CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
UNIDADE DE PÓS-GRADUAÇÃO, EXTENSÃO E PESQUISA
MESTRADO PROFISSIONAL EM GESTÃO E TECNOLOGIA EM SISTEMAS PRODUTIVOS

EMERSON MARTINS

UMA APLICAÇÃO DE *MACHINE LEARNING* NA PREVISÃO DE VENDAS NO SEGMENTO DE VAREJO.

São Paulo
Março/2023

EMERSON MARTINS

UMA APLICAÇÃO DE *MACHINE LEARNING* NA PREVISÃO DE VENDAS NO SEGMENTO DE VAREJO.

Dissertação apresentada como exigência parcial para a obtenção do título de Mestre em Gestão e Tecnologia em Sistemas Produtivos do Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos, sob a orientação do Prof. Dr. Napoleão Verardi Galegale.

Área de Concentração: Sistemas Produtivos

São Paulo
Março/2023

FICHA ELABORADA PELA BIBLIOTECA NELSON ALVES VIANA
FATEC-SP / CPS

Martins, Emerson

M386a Uma aplicação de machine learning na previsão de vendas no segmento de varejo / Emerson Martins. – São Paulo: CPS, 2023.
168 f.: il.

Orientador: Prof. Dr. Napoleão Verardi Galegale
Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). – Centro Estadual de Educação Tecnológica Paula Souza, 2023.

1. Aprendizado de máquina. 2. Análise de big data. 3. Previsão. 4. Predição. 5. Varejo. I. Galegale, Napoleão Verardi. II. Centro Estadual de Educação Tecnológica Paula Souza. III. Título.

EMERSON MARTINS

UMA APLICAÇÃO DE *MACHINE LEARNING* NA PREVISÃO DE VENDAS NO SEGMENTO DE VAREJO.

Prof. Dr. Napoleão Verardi Galegale.

Orientador – CEETEPS

Prof. Dr. José Odálio dos Santos

Examinador Externo – PUC

Prof. Dr. Marcelo Duduchi Feitosa

Examinador Interno - CEETEPS

São Paulo, 16 de Março de 2023

AGRADECIMENTOS

Agradeço primeiramente a Deus, em segundo à minha mãe, esposa e filho, pela compreensão devido a minha ausência em vários momentos durante estes dois últimos anos. Sem este apoio não teria conseguido concluir este importante trabalho no meu desenvolvimento acadêmico e profissional.

Também a todos os professores com os quais tive a oportunidade de interagir durante o mestrado e especialmente ao meu orientador, Prof. Dr. Napoleão Verardi Galegale, por todo o apoio e paciência no direcionamento desta dissertação, sempre primando pela qualidade exigida pelo programa de mestrado do CPS.

Ao Centro Paula Souza pela oportunidade e todo o apoio dos profissionais, principalmente da Débora Antunes e Vilma Capela.

Finalmente, a todos que, diretamente ou indiretamente, acreditaram, colaboraram e incentivaram durante o desenvolvimento dessa dissertação.

“A menos que modifiquemos à nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”. (Albert Einstein)

RESUMO

MARTINS, E. **Uma aplicação de *machine learning* na previsão de vendas no segmento de varejo**. 168 f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2023.

O presente trabalho tem por objetivo desenvolver um protótipo de solução computacional, baseado em aprendizado de máquina, capaz de prever a receita com a venda de produtos no segmento varejista brasileiro. A metodologia usada foi DSRM, definida como um processo de resolução de problemas que permite aos pesquisadores adquirir conhecimento e compreensão de um domínio de problema e a sua solução por meio da criação e aplicação de artefatos. Esta pesquisa propôs um novo método multidisciplinar para realização de experimentos em Inteligência Computacional, cujo qual foi implementado por meio de algumas bibliotecas disponíveis na linguagem *Python*. Seis algoritmos de inteligência computacional com base em regressão linear e cinco conjuntos de dados representando o histórico de vendas de diferentes produtos foram usados nesse sentido. Diferentes modificações críticas foram propostas e testadas em várias fases do método publicado anteriormente por Tsiliki, e os resultados obtidos foram relevantes. Pelos resultados apresentados, fica evidenciado que não existe um algoritmo melhor do que outro, ou seja, o comportamento dos dados e a correlação entre suas variáveis irá influenciar diretamente na obtenção do melhor modelo para um produto específico. Desta forma o protótipo de solução computacional resultante desta pesquisa é útil e relevante pois indica ao pesquisador qual o melhor o modelo para cada tipo de produto. Linha de Pesquisa: Sistemas de Informação e Tecnologias Digitais. Projeto de Pesquisa: Tecnologias Digitais em Sistemas Produtivos.

Palavras-chave: Aprendizado de Máquina, Análise de *Big Data*, Previsão, Predição, Preditivo, Varejo.

ABSTRACT

MARTINS, E. **A machine application in sales forecasting in the retail segment**. 168 f. Dissertation (Professional master´s degree in Management and Technology in Productive Systems. Line of Research: Information Systems and Digital Technologies. Research Project: Digital Technologies in Production Systems). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2023.

This work aims to develop a prototype computational solution, based on machine learning, capable of predicting revenue from the sale of products in the Brazilian retail segment. The methodology used was DSRM, defined as a problem-solving process that allows researchers to acquire knowledge and understanding of a problem domain and its solution through the creation and application of artifacts. This research proposed a new multidisciplinary method for carrying out experiments in Computational Intelligence, which was implemented through some libraries available in the Python language. Six computational intelligence algorithms based on linear regression and five datasets representing the sales history of different products were used in this regard. Different critical modifications were proposed and tested in several phases of the previously published method by Tsiliki, and the obtained results were relevant. From the results presented, it is evident that there is no better algorithm than another, that is, the behaviour of the data and the correlation between its variables will directly influence the obtaining of the best model for a specific product. Thus, the prototype computational solution resulting from this research is useful and relevant as it indicates to the researcher which model is best for each type of product. Line of Research: Information Systems and Digital Technologies. Research Project: Digital Technologies in Productive Systems.

Keywords: *Machine Learning*, *Big Data Analytics*, *Forecasting*, *Predictions*, *Predictive*, *Retail.*

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SIGLAS

ANN	*Artificial Neural Network*
AUROC	*Area Under the ROC Curve*
ARIMA	*Autoregressive Integrated Moving Average*
ARSA	*Autoregressive Sentiment-Aware*
ARSQA	*Autoregressive Sentiment and Quality Aware*
ATM	*Automated Teller Machine*
BDA	*Big Data Analytics*
BD	*Big Data*
CI	*Computational Intelligence*
CV	*Cross-Validation*
DBN	*Deep Belief Networks*
DNN	*Deep Neural Networks*
DOE	*Design of Experiments*
DSRM	*Design Science Research Methodology*
DSRP	*Design Science Research Process*
DT	Decision Tree
EDA	*Exploratory Data Analysis*
EM	*Expectation Maximization*
GEML	*Grid-Embedding based Multi-task Learning*
GFNN	*Gaussian-Fuzzy-Neural Network*
GBM	*Gradient Boosting Machine*
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
IoT	*Internet Of Tnings*
IPCA	Índice Nacional de Preços ao Consumidor Amplo
K-MEANS	Algoritmo de aprendizado de máquina não supervisionado para agrupamento
LASSO	*Least Absolute Shrinkage and Selection Operator*
LR	*Linear Regression*
LSTM	*Long Short Term Memory*
MAE	*Mean Absolute Error*
MAPE	*Mean Absolute Percentage ErrorM*
ML	*Machine Learning*
MRP	*Maximum Retail Price*
MSE	*Mean Square Error*

ODMP	*Origin-Destination Matrix Prediction*
OLS	*Ordinay Least Squares*
PLSA	*Probabilistic Latent Semantic Analysis*
PV	Predição de Vendas
RF	*Random Forest*
RL	*Reinforcement Learning*
RMSE	*Root Mean Square Error*
RR	*Ridge Regression*
SL	*Supervised Learning*
SI	Sistemas de Informações
SQL	*Structure Query Language*
SVR	*Support Vector Regression*
UL	*Unsupervised Learning*
URL	*Uniform Resource Locator*
XGBoost	*Extreme Gradient Boosting*

SUMÁRIO

INTRODUÇÃO

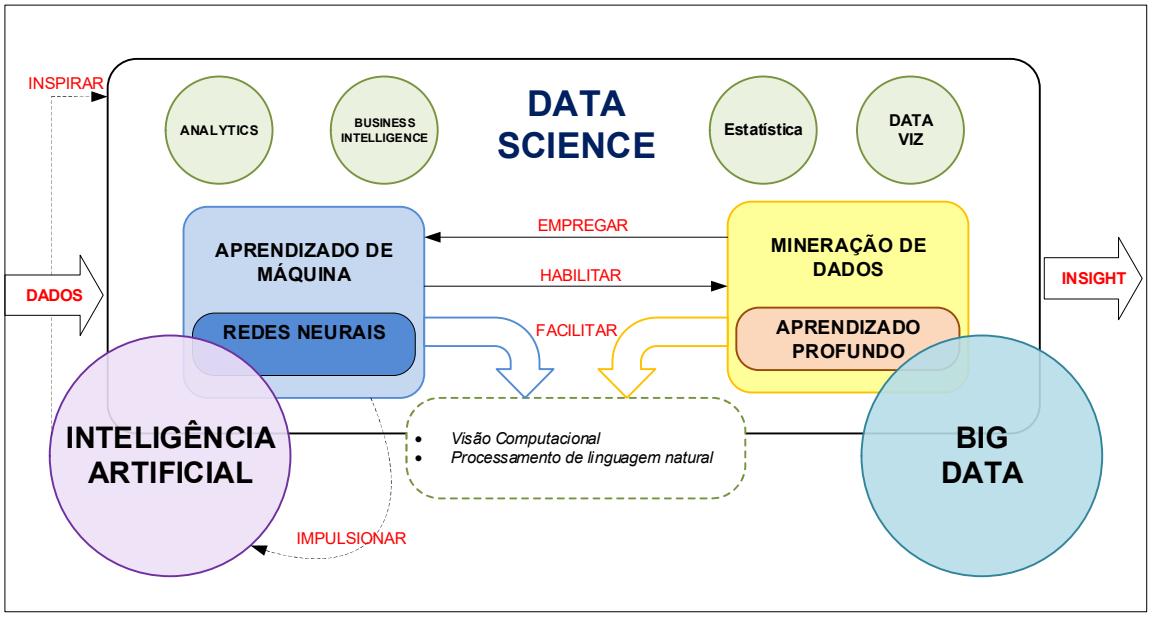
Machine Learning (ML) é o ramo da ciência onde algoritmos de computador são desenvolvidos para executar tarefas sem orientação humana, em vez de depender de regras codificadas. Em outras palavras, ML é a capacidade dos computadores de induzir novos conhecimentos, tais algoritmos têm sido usados de forma ampla e com sucesso em muitas áreas (MAXWELL *et al*., 2015).

Nas duas últimas décadas o poder da computação evolui consideravelmente, neste contexto podemos citar: grande armazenamento de dados, processadores mais robustos, conexão mais rápida com a internet, entre outros exemplos. Problemas que pareciam ser extremamente complexos ou custosos de serem resolvidos, agora estão ao nosso alcance. Novas tendências como *Big Data*, *Cybersecurity*, internet das coisas, em inglês *internet of things* (IoT) e *blockchain* surgiram, explorando conjuntamente os avanços tecnológicos mencionados acima. O IoT, que visa usar sistemas embarcados, incluindo sensores e atuadores, juntamente com a internet, para permitir o controle e o acesso imediato às informações em tempo real (Atzori, 2010; Cecchinel, 2014), representa um desses desafios, pois o relatório da "*Juniper Research*", informa que em 2024 teremos mais de 83 bilhões de dispositivos e sensores conectados (IoT). Além disso, alguns destes dispositivos terão a capacidade de gerar quantidade de dados expressiva na ordem de *Zettabytes*, informações que podem ser valiosas para estratégia de uma empresa, portanto, a previsão de vendas não pode ignorar essas novas tendências; ela deve utilizá-la como suporte para vantagem competitiva.

Atualmente o processamento dos dados por meio dos sistemas de informação para geração de conhecimento tornou-se vital para os tomadores de decisão, particularmente em algumas áreas importantes como a previsão de vendas de produtos ou de serviços no varejo, nas quais variáveis externas como o clima, índice de inflação ou economia global podem afetar a decisão de consumo das pessoas (KRAWCZYK, 2016).

Uma das técnicas recentes e populares que visa enfrentar esses novos desafios de negócios é o *Big Data Analytics* (BDA). Uma definição precisa de BDA é dada em Hofmann (2018). Em suma, é o alinhamento das técnicas de *Big Data* (BD) e ML para fornecer *insights* confiáveis para a tomada de decisões. A Figura 1, representa a arquitetura geral e elementos em um modelo BDA. Podemos observar que o ML e *Big Data* se beneficiam uma da outra, pois podem ser acopladas para criar modelos mais completos. Além disso, o principal propósito do BDA é transformar informações em conhecimento útil.

Figura 1 - Arquitetura de um modelo *Big Data Analytics*



Fonte: Adaptado de Mayo (2016)

Na Figura 1, o lado esquerdo representa a IA com a presença do aprendizado de máquina e seus diversos algoritmos como as redes neurais e do lado direto é destacado o *Big Data* com a mineração de dados e aprendizado profundo, todas estas tecnologias dentro do contexto de ciência de dados voltadas para inteligência de negócio, fornecendo *insights* sobre eventos futuros para que empresas consigam melhorar sua tomada de decisão.

Por sua vez, análise preditiva engloba métodos que utilizam informações para criar modelos e realizar simulações que fornecerão *insights* sobre eventos futuros, permitindo que os executivos mais atentos consigam prever ações estratégicas que melhorem o desempenho da sua empresa. Por definição, os resultados obtidos por meio destas técnicas não são 100% precisos, pois nenhum método pode prever o futuro, sendo assim, uma boa análise preditiva é aquela que fornece os resultados mais precisos em um tempo razoável (CASTILLO *et al*, 2017).

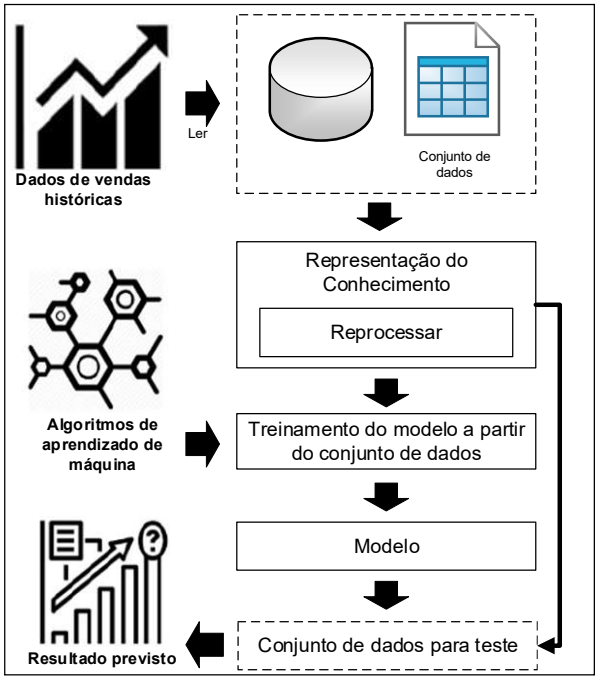
Um dos usos comuns da análise preditiva nos negócios é a previsão de vendas, mas também há várias outras aplicações em domínios como: estimativa de custo, onde (LOYER *et al*, 2016) aplicaram técnicas de ML para estimar rapidamente o custo de fabricação de componentes de motor para aviões, já que os métodos clássicos de estimativa de custos, embora sejam bastante precisos, são caros e lentos. Na avalição de desempenho, (FAN *et al*, 2013) utilizaram ML para estimar o desempenho da cadeia de suprimentos baseado no "*5 Dimensional Balanced Scorecard*" (5DBSC) e "*Levenberg-Marquadt Back Propagation*" (LMBP) com o objetivo de fornecer resultados rápidos e evitar avaliações tendenciosas de desempenho por parte dos gestores.

Finalmente, em relação a estratégia, Vahdani *et al*. (2016) empregaram ML para prever o desempenho de fornecedores de logística terceirizados "*Third-party Logistics*" (3PL) considerando elementos como o seu incremento na participação de mercado, experiência, localização, percentual de entregas sem atraso, entre outros fatores. O objetivo foi fornecer uma metodologia para apoiar a escolha de um provedor 3PL dada a importância desta decisão para a organização.

A abordagem do ML pode ser descrita como o estudo de algoritmos de computador que se aprimoram automaticamente com a experiência. É tratado com uma subárea da inteligência artificial (IA). ML constroem um modelo baseado em dados de amostra, conhecidos como "dados de treinamento", a fim de fazer previsões ou decisões sem serem explicitamente programados para isso. Os algoritmos de ML são usados em uma ampla variedade de aplicações, como filtragem de e-mail e visão computacional, onde é difícil ou inviável desenvolver algoritmos convencionais para realizar as tarefas necessárias (RASCHKA *et al*, 2017).

Os seres humanos aprendem por meio da experiência, usamos um processo de tentativa e erro para descobrir quais ações devem ser desencadeadas em determinadas circunstâncias. Isso nos permite fazer abstrações e construir o conhecimento. O ML é de alguma forma semelhante, pode ser visto como algoritmos que têm como objetivo melhorar uma medida de desempenho, derivando automaticamente suas próprias regras e criando seus próprios modelos de decisão com base em determinadas informações (Raschka *et al*., 2017) e Mitchell (1997). A Figura 2 mostra o diagrama desta arquitetura.

Figura 2 - Arquitetura ML



Fonte: Adaptado de Cheriyan *et al*. (2018)

A Figura 2 representa o processo no qual os algoritmos de ML são treinados. Primeiramente os dados são coletados a partir de uma base histórica, na fase seguinte estes dados são reprocessados com o objetivo de eliminar *outlier* e ruídos, na sequência uma parte dos dados é separada para treinamento para que o algoritmo de ML consiga interpretar seu comportamento e finalmente o melhor modelo é identificado e aplicado nos dados de teste.

Questão de pesquisa

Este estudo procura responder a seguinte questão de pesquisa: Como, por meio de um protótipo de solução computacional, baseado em ML, pode-se prever as receitas com a venda de produtos no segmento varejista brasileiro?

Objetivo Geral

O objetivo geral desta pesquisa é desenvolver um protótipo de solução computacional, baseado em ML, onde por meio de variáveis internas contidas no histórico de vendas como: categoria do produto, data da venda, quantidade, valor da venda e variáveis externas como índice de inflação IPCA, seja possível precificar as vendas futuras, contribuindo para que a empresa tenha previsibilidade de receita, viabilizando seu planejamento financeiro. Adicionalmente, evidenciar que os algoritmos de ML com base em regressão linear apresentam melhor desempenho se comparados com métodos tradicionais como média aritmética.

Objetivos Específicos

Com base no objetivo geral, foram definidos os seguintes objetivos específicos para responder à questão de pesquisa:

a) Por meio de uma revisão descritiva da literatura, identificar quais são os algoritmos aplicáveis para predição.

b) Desenvolver protótipo de solução computacional, baseado em algoritmos de ML, que possa analisar as movimentações do estoque para determinado produto. O protótipo deve apresentar um painel comparativo, destacando alguns indicadores estatísticos de performance entre estes algoritmos, utilizando como base de treinamento o próprio histórico de vendas.

c) Testar e avaliar os resultados, por meio de duas validações internas, sendo, entrevista com o usuário do departamento planejamento financeiro da empresa selecionada para este estudo e utilização DOE para aplicação do estudo de campo e uma validação externa por meio de entrevistas semiestruturadas com especialistas para avaliar a arquitetura da solução computacional proposta.

Justificativa da Pesquisa

As empresas, como sistemas produtivos, devem usar seus recursos de forma eficiente e tomar decisões estratégicas para obter receitas crescentes e estáveis, especialmente quando as condições de mercado estão ficando mais competitivas e com margens de lucro cada vez mais pressionadas. Desta forma, a previsão de vendas é crucial para manter a competitividade, porém obter previsões imprecisas pode levar à escassez de estoque, ocasionando atrasos nas entregas e gerando insatisfação dos clientes, como também, podem elevar o estoque, aumentando o custo de armazenagem, forçando a "queima" de estoque por meio de campanhas promocionais, afetando diretamente a lucratividade das empresas. (HOFMANN *et al*., 2018).

A motivação deste trabalho se fundamenta na pesquisa sobre a abordagem de ML aplicado na previsão de vendas no segmento de varejo, cujo qual apoia os gestores na tomada de decisão, contribuindo nas operações da cadeia de suprimentos, nas previsões de vendas, na gestão do estoque, no planejamento da produção e na programação da força de trabalho. Contribuindo, desta forma, no aumento de lucros e redução de custos (RANGAPURAM, 2018; SALINAS, 2020; ABURTO 2007; GUIDOTTI, 2018).

Prever a demanda de produtos e serviços e adequar a cadeia de suprimentos encontrando um equilíbrio sempre foi e continuará sendo um desafio na indústria.

Vários métodos de séries temporais podem ser encontrados em alguns estudos, como (VEIGA *et al.* 2016; PAVLYSHENKO, 2019). No entanto, existem poucos casos de predição de vendas que utilizam métodos supervisionados de ML baseados em LR como *Random Forest* (BOEHMKE *et al.*, 2019) e *Gradient Boosting Machine* (FRIEDMAN, 2001).

As pesquisas a respeito de predições utilizando ML vem apresentando resultados importantes em diversas áreas do conhecimento, porém têm sido pouco exploradas no segmento de varejo, conforme resultados da revisão da literatura apresentada na fundamentação teórica, sugerindo uma lacuna de pesquisa recente a ser investigada (FRIEDMAN, 2001; BOEHMKE et al., 2019; SAYLI et al., 2016; MAINGI, 2015; HUANG *et al*., 2017; SASTRY *et al*., 2013).

É observado que a regressão linear melhora substancialmente o desempenho do modelo base por meio de recursos extraídos e fornece um desempenho comparável a outras abordagens bem estabelecidas. A interpretação das previsões do modelo e a alta precisão preditiva da regressão linear a torna um método mais eficaz do que métodos tradicionais como ARIMA que aplicam médias móveis

(RANGAPURAM, 2018; SALINAS, 2020, ABURTO 2007; GUIDOTTI, 2018; HUANG *et al.*, 2017; SAYLI *et al.*, 2016).

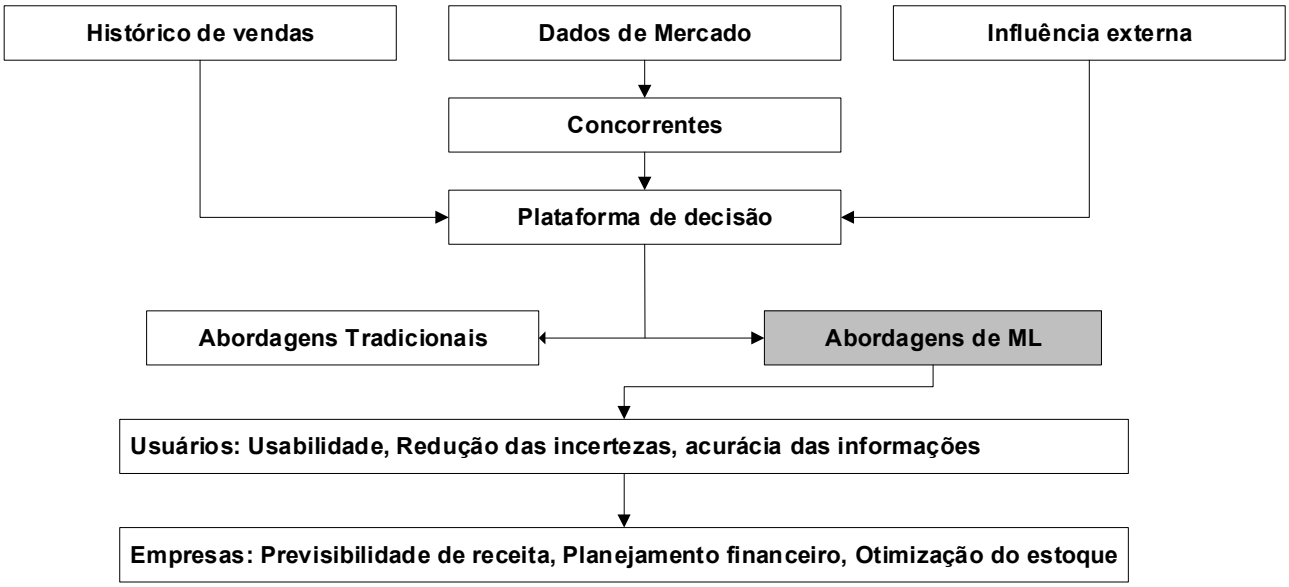
O sucesso para uma previsão eficaz está na existência, relevância e confiabilidade dos dados históricos contidos em sistemas de informações. Uma adequada previsão de vendas permiti antecipar eventuais desiquilíbrios no fluxo de caixa, evitando o impacto de erros de previsão nas finanças e consequentemente no desempenho da organização, uma vez que tais erros ocasionam aumento dos níveis de estoque, perda de vendas e frequentemente uma queda da margem bruta para o varejista, indicando o início do efeito chicote. Os sistemas de previsão que utilizam técnicas avançadas como ML são, portanto, uma forma eficiente de reduzir este fenômeno negativo para a indústria do varejo. (THOMASSEY, 2010).

Porém, em mercados com demandas constantes e poucas interferências externas, a sua utilização não se justifica pois, para estes casos, a utilização de métodos tradicionais é mais simples e menos custosa.

Contexto da Intervenção

O contexto de negócios em que se insere a pesquisa, o ponto de intervenção (bloco em fundo cinza) e suas finalidades são apresentados na Figura 3, descrevendo que a partir do histórico de vendas e de fatores externos como índice de inflação, cenário macro econômico, dentre outros é possível compilar tais informações por meio do treinamento de algoritmos ML para apoiar a tomada de decisão, aumentando a previsibilidade de receita e consequentemente o planejamento financeiro das empresas.

Figura 3 - Contexto da intervenção e finalidades da pesquisa



Fonte: Resultado da pesquisa

A tomada de decisão para predição de futuras receitas nas empresas pode ocorrer por dois caminhos, ou seja, pela abordagem tradicional, no qual o histórico de vendas é avaliado por um analista e este avalia as futuras vendas com base no comportamento passado considerando algumas variáveis externas que não inviabilizem o cálculo de forma manual, pois dependendo do número de variáveis isso pode se tornar inviável.

A outra abordagem é através da utilização da tecnologia como ferramenta de apoio, sendo este o foco desta pesquisa, desta forma o item “Abordagens de ML” é destacado no diagrama acima.

Estrutura da Dissertação

No Capítulo de INTRODUÇÃO, foram apresentados o contexto da pesquisa, sua orientação em benefício dos sistemas produtivos, o importante papel da predição de vendas no segmento varejista, o impacto da evolução tecnológica sobre o segmento do varejo. Descreveu-se também o contexto da aplicação do ML. Ao final, apresenta-se a questão de pesquisa, o objetivo geral e específicos assim como a justificativa deste trabalho.

O Capítulo 1, FUNDAMENTAÇÃO TEÓRICA, se inicia apresentando conceitos e características da previsão de vendas. É apresentado o contexto do surgimento da IA, os conceitos de ML, as formas de aprendizado, os modelos de parametrização, e as principais abordagens. As abordagens de como a regressão linear é utilizada para predição de valores são pesquisadas por meio de uma análise bibliométrica, seguida de revisão descritiva da literatura identificando os conceitos, características, frequência de ocorrência das abordagens, análise de suas aplicações comparadas às abordagens tradicionais e ao ML.

O Capítulo 2, METODOLOGIA, apresenta o método *Design Science Research Method* (DSRM) utilizado para a pesquisa da abordagem de ML e a implementação do artefato.

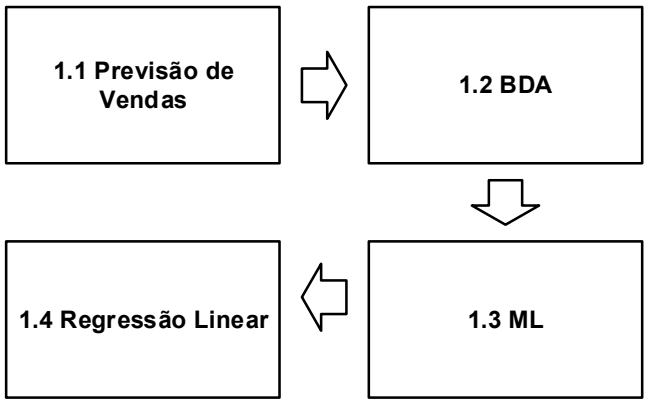
O Capítulo 3, RESULTADOS E DISCUSSÃO, em conformidade com as atividades da pesquisa em DSRM, apresenta a configuração, a execução e avaliação interna do artefato de software baseado no DOE e entrevista com usuário do departamento de planejamento financeiro da empresa selecionada para o estudo de campo, assim como uma avaliação externa por meio de entrevistas semiestruturadas com especialistas.

No Capítulo CONSIDERAÇÕES FINAIS, apresentam-se considerações quanto aos resultados obtidos e contribuições da pesquisa, e são levantadas possibilidades para continuidade desta pesquisa.

1 FUNDAMENTAÇÃO TEÓRICA

Este capítulo se estrutura em: Previsão de vendas; *Big Data Analytics,* ML e Regressão Linear aplicados na previsão de vendas, conforme Figura 4.

Figura 4 – Estrutura do referencial teórico



Fonte: Autor

A previsão de vendas sempre foi e continua sendo um grande desafio do segmento varejista, as empresas continuam à procura de uma ferramenta que consiga apoiá-las para antecipar o comportamento do mercado consumidor, como por exemplo a previsão de vendas. O BDA contribui em direção a este sentido, uma vez que um dos seus principais objetivos é consolidar este grande volume de informações e extrair conhecimento a partir do comportamento dos dados. Por sua vez as técnicas de ML conseguem otimizar a leitura do comportamento dos dados, aplicando conceitos estatísticos ao treinar os modelos, destacando os algoritmos com maior acurácia na predição. Juntamente com o ML, a regressão linear potencializa a utilização dos algoritmos ao aplicar as funções lineares que promovem clareza e fácil compreensão de como os dados históricos estão sendo interpretados na predição dos valores de vendas.

1.1 Previsão de vendas

A previsão de vendas é considerada um tópico importante no campo do BDA e ML, conforme relatam Sarwar *et al*. (2000), pois desempenha um papel essencial na inteligência empresarial moderna. A análise preditiva deve ser baseada em grandes quantidades de dados históricos, na medida que as vendas podem ser consideradas como uma série temporal. Atualmente, muitos estudiosos têm aplicado diferentes modelos de séries temporais, como ARIMA, GARCH e Holt-Winters para tentar predizer o comportamento das futuras receitas.

Vários métodos de séries temporais podem ser encontrados em alguns estudos, como (VEIGA *et al.* 2016; PAVLYSHENKO, 2019). No entanto, existem poucos casos de predição de vendas que utilizam métodos supervisionados de ML baseados em regressão linear como *Random Forest* (BOEHMKE *et al.*, 2019) e *Gradient Boosting Machine* (FRIEDMAN, 2001).

Avaliar e comparar o desempenho dos modelos construídos usando diferentes algoritmos é uma parte crucial da construção de modelos de ML. O uso de várias métricas de avalição pode evitar defeitos no modelo. Para melhorar o desempenho do modelo, também é crucial escolher as métricas de avalição correspondentes, como *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE) e *Mean Absolute Percentage Error* (MAPE), os quais têm sido amplamente utilizadas para resolver problemas de regressão, como predição no giro de estoque (CHEN *et al*., 2017) e previsão de demanda na cadeia de suprimentos (KILIMCI *et al.*,2019). Além disso a métrica de avaliação F2-score pode ser utilizada para avaliar o desempenho do modelo referente a acurácia ou precisão, já para problemas de classificação deve ser utilizado o *recall rate* que nos dá informações sobre os falsos negativos, sinalizando quanto o modelo está identificando os casos positivos corretamente (WANG *et al*., 2016).

As empresas do segmento varejista lidam com enormes repositório de dados, espera-se que este volume de dados cresça de uma forma exponencial ao passar dos anos, uma vez que o histórico de compras dos consumidores e seu comportamento de consumo são armazenados a todo momento. A indústria de *e-commerce* está sempre em busca de novas técnicas de mineração de dados e modelos inteligentes de previsão de tendencias de vendas com o maior nível possível de precisão e confiabilidade (SAYLI *et al.*, 2016).

Previsão de vendas fornece *insights* em como uma empresa deve gerenciar sua força de trabalho, fluxo de caixa e demais recursos. É um importante pré-requisito no planejamento e na tomada de decisão, fornecendo subsídios para que as empresas planejem suas estratégias de negócios de uma forma efetiva. A indústria do varejo enfrenta severos desafios para identificar uma técnica precisa e eficaz de mineração de dados (MAINGI, 2015).

Previsões precisas permitem que a organização melhore seu crescimento no mercado com um maior nível de geração de receita. As técnicas de mineração de dados são muito eficazes em ajustar grandes volumes de dados em informações úteis para previsão de custos e vendas e, portanto, a obtenção de um orçamento sólido (HUANG *et al*., 2017).

No nível organizacional, previsões de vendas são insumos essenciais para a tomada de decisão em diversas áreas funcionais, como: operações, marketing, vendas, produção e finanças. Entretanto a análise de dados de vendas enfrenta muitos desafios relacionados a alguns atributos do produto, como: fixação de preço e lançamento de novos produtos. Vários métodos de previsão, assim como estratégias de previsão de vendas e algoritmo de maximização de expectativa (EM) são discutidos em (SASTRY *et al.*, 2013).

Um estudo comparativo sobre ajuste de dados e vários algoritmos de *clustering* em dados de vendas é evidenciado em (SHRIVASTAVA *et al*., 2015).

Rajagopal (2011) demonstra que a classificação dos dados é muito importante na tomada de decisão. Técnicas de *clustering* são muito úteis em descobrir padrões de distribuição e algoritmos de agrupamento empregam medidas de similaridade baseados em métricas de distância (TSAI *et al.*, 2002).

Utilizando adequadamente técnicas de mineração de dados em um conjunto volumoso de dados é possível transformar dados dispersos em informações úteis para tomada de decisão, empregando aprendizado supervisionado ou não supervisionado (MANN *et al*., 2013).

Jain *et al*. (2015) demostram que que a previsão de venda pode ser feita usando técnicas de mineração de dados, aplicando algoritmos de regressão como LR, SVR e RF, assim como algoritmos e preditores como XGBoost e GBM, sendo possível prever as vendas em qualquer dia e em qualquer loja.

A previsão de vendas está ligada diretamente a estratégia da empresa para manter ou aumentar seu faturamento ao longo de determinado período. De qualquer forma prever o lucro operacional da organização é tão importante quanto a previsão de vendas, uma vez que as despesas administrativas, comerciais e operacionais devem ser constantemente monitoradas com o objetivo de calibrar o alvo das futuras receitas para suprir as despesas, mantendo o lucro operacional e consequentemente a saúde financeira da organização.

1.2 *Big Data Analytics*

1.2.1 Contexto

Um sistema analítico de decisão inteligente requer a integração de análise de decisão e previsões. As organizações empresariais dependem de uma base de conhecimento e exigem previsões de tendências de vendas. A precisão na previsão de vendas proporciona um grande impacto nos negócios. As técnicas de mineração de dados são ferramentas eficazes na extração de conhecimento oculto em enormes conjuntos de dados, o que proporciona aumento na precisão e na eficiência da previsão.

Os sistemas de previsão tradicionais são difíceis de lidar com grandes volumes de dados, assim como com a precisão na previsão de vendas. Esses problemas podem ser superados usando técnicas de mineração de dados.

BDA é definida como uma abordagem holística para gerenciar, processar e analisar as dimensões relacionadas aos dados em "5 Vs" (ou seja, volume, variedade, velocidade, veracidade e valor), a fim de promover *insights* para a entrega sustentável de valor, medindo o desempenho e estabelecendo vantagens competitivas (WAMBA, *et al*., 2015). Considerado por Manyika (2011) como um paradigma entre a fronteira da inovação, competição e produtividade, o BDA vem impulsionando a adoção generalizada de várias ferramentas e tecnologias, incluindo mídias sociais (Facebook, Twitter), dispositivos móveis (*laptops*, *smartphones*), sensores que viabilizam o IoT e integrações com RFID e *Bluetooth*. Todas estas tecnologias embarcadas em plataformas em nuvem que suportam os processos de negócios intraorganizacionais, assim como externos à organização para obter vantagem competitiva.

Espera-se que a difusão generalizada dessas ferramentas e tecnologias transforme a maneira como atualmente os negócios são conduzidos. Isso é particularmente verdadeiro para previsão de vendas. Estudos anteriores sobre BDA salientaram a importância de alcançar uma elevada capacidade de integração de processos e sistemas de informação integrados, a fim de alcançar um maior nível de coordenação sem descontinuidades e reduzir os esforços repetidos e as ineficiências. A combinação da análise de sentimentos e serviços inteligentes de informação habilitados para o varejo com as tecnologias de informação e de comunicação existentes deverá desempenhar um papel facilitador - tornando assim os produtos e serviços mais visíveis aos consumidores - e, em paralelo, deverá oferecer mais oportunidades para uma predição mais assertiva a respeito de futuras vendas. Neste contexto, o acesso a informações críticas para a tomada de decisões torna-se não só um pré-requisito, mas também um grande desafio.

BDA tem sido usado com sucesso para várias funções de negócios, como contabilidade, marketing, cadeia de suprimentos, operações e vendas. Atualmente, junto com o desenvolvimento recente em aprendizado de máquina o BDA na área de vendas vem ganhando cada vez mais importância.

1.2.2 Bibliometria

Para mensuração do volume de publicações na literatura cientifica sobre BDA, as bases Scopus, Web of Science, ACM e IEEE Xplore foram consultadas, conforme critérios no Quadro 1.

Quadro 1 - Critérios de busca sobre utilização de BDA na previsão de vendas

Atributo	Critério
Expressão de busca	ALL= (("*Big Data Analytics*" OR "BDA" OR "*Big Data*") AND ("*Sales Prediction*" OR "*Sales Forecast*" OR "*Retail*" OR "Sales"))
Período	1980 a 2022
Idioma	Inglês
Tipo de publicação	Artigos de periódicos e conferências
Exclusão de domínios de pesquisa	Medicina, Engenharia e Arquitetura, Educação, Psicologia, Agricultura e Biociências, Artes e Humanidades, Bioquímica, Geociências e Logística

Fonte: Autor

A Tabela 1 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 1 - Número de publicações das bases consultadas sobre utilização de BDA na previsão de vendas

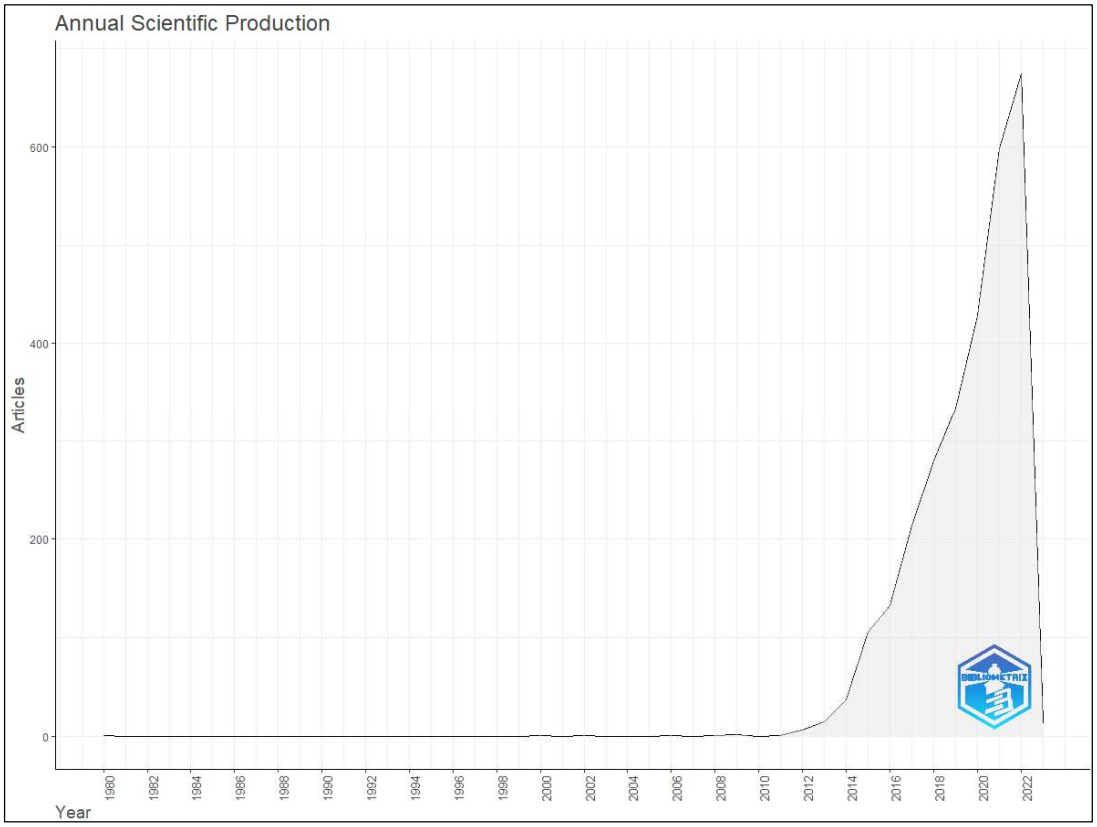
Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	1,526
Web of Science	Tópico	1,541
ACM Digital Library	Títulos, Resumos e Palavras-chave	148
IEEE Xplore	*All-Metada*	626

Fonte: Scopus (2022), Web of Science (2022), ACM Digital Library (2022), IEEE Xplore (2022)

Por meio do software *Bibliometrix,* os arquivos gerados pelas bases Scopus e Web Of Science foram consolidados, resultando na Figura 5, na qual apresenta a evolução da produção cientifica sobre BDA relacionado a predição de vendas no segmento de varejo, em número de publicações ao longo dos anos.

Neste gráfico se observa um leve crescimento no número de publicações a partir do ano de 2014, tornando-se um crescimento acentuado a partir do ano de 2016. Quanto ao ano de 2022, deve-se considerar que os dados são parciais, acumulados até o mês de Agosto, contudo observa-se que o número de publicações se mantém crescente, ou seja, não existe uma tendência de baixa no número de publicações.

Figura 5 - Evolução das publicações sobre utilização de BDA na previsão de vendas



Fonte: Scopus (2022), Web of Science (2022)

A Tabela 2 apresenta o número de publicações por ano considerando as bases Scopus e Web of Science no período de 2014 a 2022.

Tabela 2 – Evolução das publicações por ano

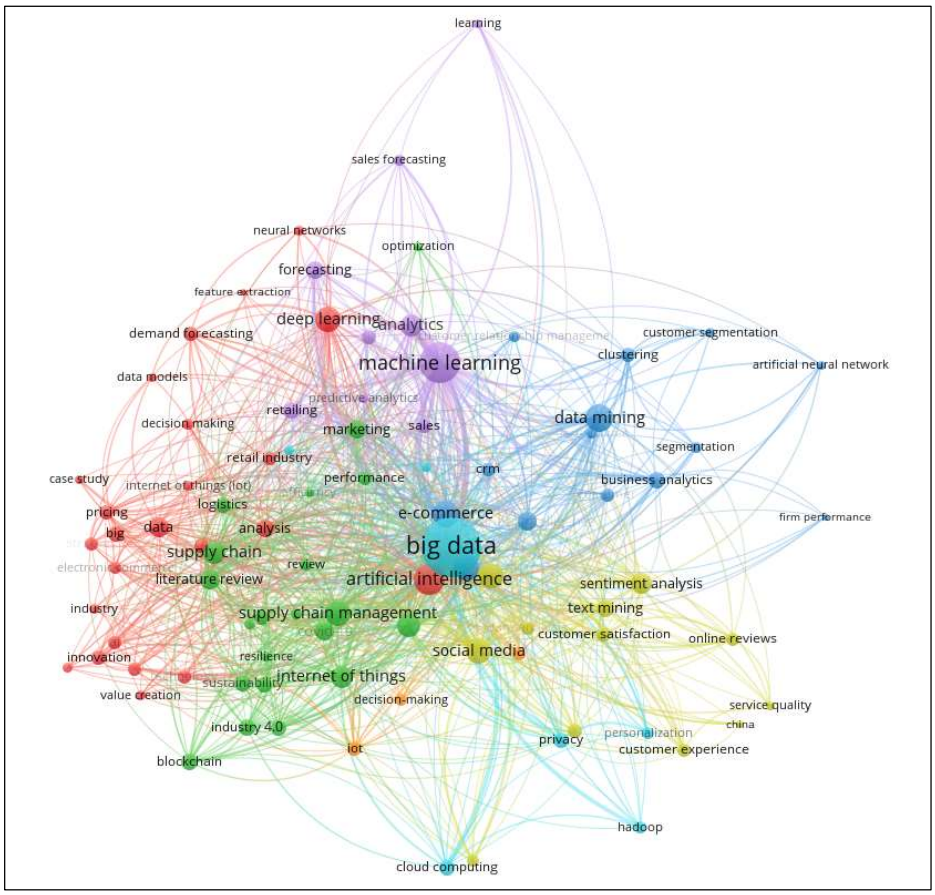
Ano	Publicações	% Crescimento
2014	37	
2015	106	286%
2016	133	7,6%
2017	213	60%
2018	279	31%
2019	333	20%
2020	428	29%
2021	598	40%
2022	687	15%

Fonte: Scopus (2022), Web of Science (2022)

É evidenciado um crescimento constante no número de publicações desde o ano de 2014, destacando o ano de 2015 que apresenta um crescimento 286% comparando o ano anterior.

Na Figura 6 é apresentado o gráfico de coocorrências de palavras-chaves por meio de integração entre o *Bibliometrix* e o *VOSviewer*, o qual também pode ser interpretado para destacar os tópicos específicos que aparecem com frequência, assim como os tópicos baseados em palavras-chaves gerais que possuem um escopo de cobertura maior. O tamanho dos círculos indica a frequência em que determinada palavra-chave ocorre. É observado que tópicos sobre *Big Data*, *Data Analytics*, *Artificial Intelligence* e *Machine Learning* são recorrentes. Ao mesmo tempo, *Data Mining*, *Social Media* e *Supply Chain Management* são tópicos menos frequentes, ou seja, com menor número de artigos, mas com alto interesse para pesquisas futuras.

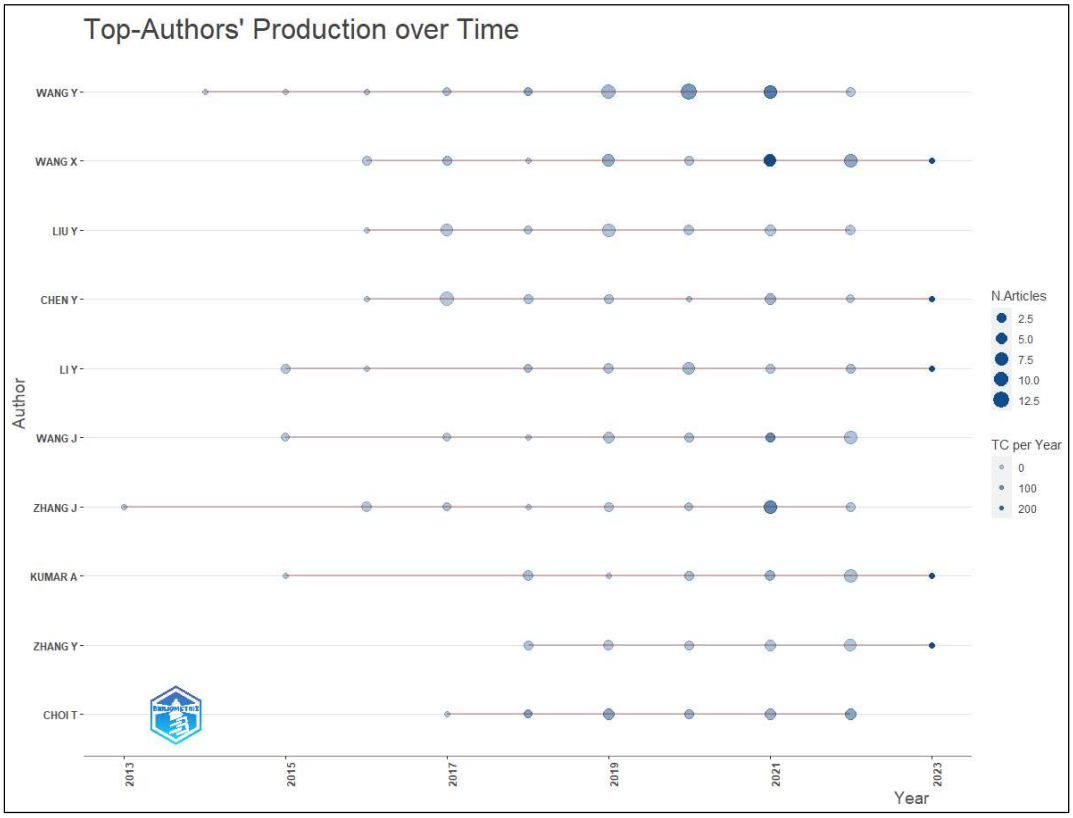
Figura 6 - Coocorrência de palavras chaves



Fonte: Scopus (2022), Web of Science (2022)

Os autores mais produtivos são mostrados na Figura 7, com destaque para o Yichuan Wang (WANG Y) com 41 artigos publicados, seguido por Xuequn Wang (WANG X) com 31 artigos e Yau Liu (LIU Y) com 30 artigos.

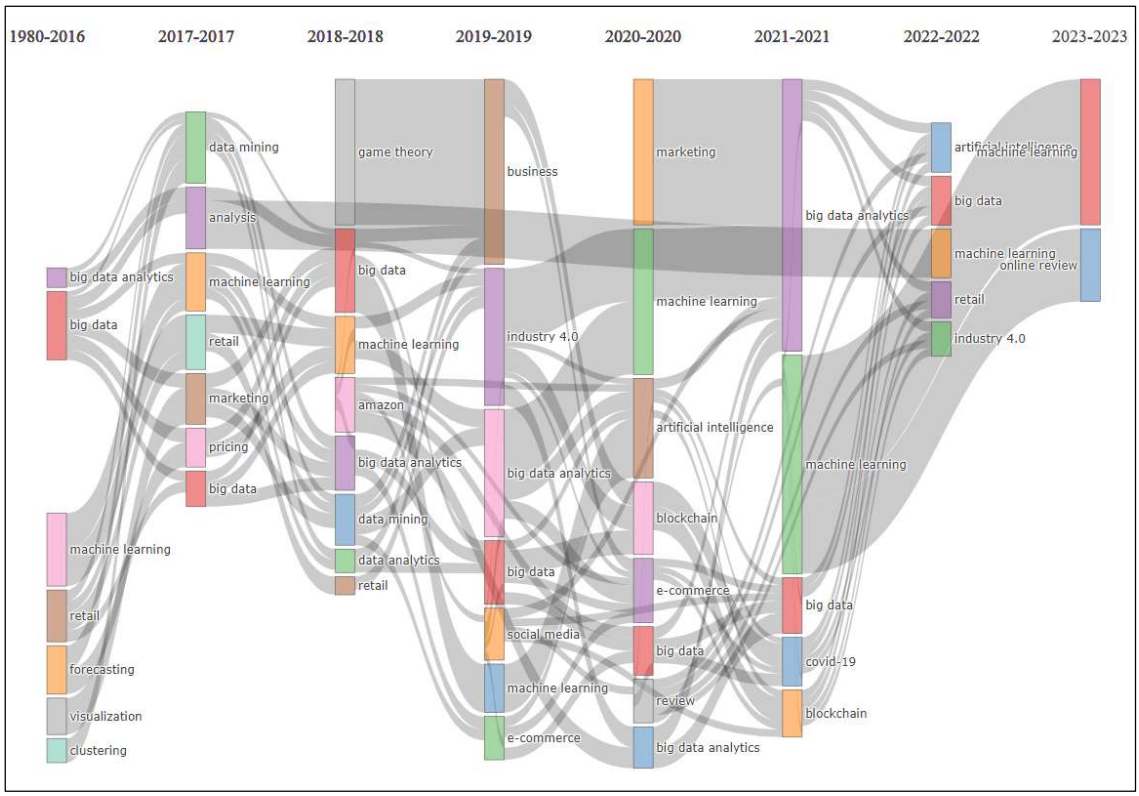
Figura 7 - Autores mais produtivos sobre BDA na previsão de vendas



Fonte: Scopus (2022), Web of Science (2022)

A Figura 8 ilustra o mapa de evolução temática para os anos de 1980-2023. O período de 2016 a 2017 gira em torno do tema *Big Data* e métodos de classificação para tratar a grande quantidade de dados. Gradualmente, outros temas evoluíram no período 2018 a 2019, por exemplo *Big Data Analytics* e *Machine Learning*, indicando a necessidade de utilizar sistemas de aprendizagem para classificar as informações. O período de 2019 a 2020 se destacam os termos *Artificial Intelligence*, *Blockchain* e *e-Commerce*, indicando a busca de mecanismos para apoiar a indústria na tomada de decisão no comércio eletrônico. Em 2020 a 2021, os termos *Machine Learning*, *Artificial Intelligence* e *Big Data Analytics*, ganham tração, indicando, assim, a demanda por utilização de métodos de aprendizado de máquina para tomada de decisão. No período atual de 2022 seguindo para 2023, fica evidenciado que os termos *Machine Learning*, *Artificial Intelligence* e *Big Data*, continuam dominando os temas dos artigos acadêmicos.

Figura 8 - Mapa de evolução temática



Fonte: Scopus (2022), Web of Science (2022)

A Tabela 3 apresenta dez publicações com conteúdo pertinente ao BDA, dentre as mais citadas, conforme as bases Scopus e Web of Sciense.

Tabela 3 - Artigos mais citados sobre abordagens de BDA aplicados na previsão de vendas

Título	Autores	Fonte	Ano	Base	TC
What can big data and text analytics tell us about hotel guest experience and satisfaction?	Xiang, Z,; Schwartz, Z.; Gerdes , J. H.; Uysal, M.	International Journal of Hospitality Management	2015	WoS	450
The Future of Retailing	Grewal, D; Roggeveen, AL; Nordfalt, J	Journal of Retailing	2017	WoS	445
Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation	Tirunillai, S; Tellis, GJ	Journal of Marketing Research	2014	WoS	365
Marketing Analytics for Data-Rich Environments	Wedel, M; Kannan, PK	Journal of Marketing	2016	WoS	320
Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics	Salehan, M; Kim, DJ	Decision Support Systems	2016	WoS	314
Big Data Analytics in Operations Management	Choi T.-M., Wallace S.W., Wang Y.	Production and Operations Management	2018	Scopus	306
Harvesting big data to enhance supply chain innovation capabilities: An analytic infrastructure based on deduction graph	Tan K.H.; Zhan Y.; Ji G.; Ye F.; Chang C.	International Journal 0f Production Economics	2015	Scopus	267

Título	Autores	Fonte	Ano	Base	TC
DeepAR: Probabilistic forecasting with autoregressive recurrent networks	Salinas, D; Flunkert, V; Gasthaus, J; Januschowski, T	International Journal of Forecasting	2020	WoS	256
Big data analytics in supply chain management: A state-of-the-art literature review	Nguyen T., ZHOU L., Spiegler V., Ieromonachou P., Lin Y.	Computers and Operations Research	2018	Scopus	245
Analysis of Dimensionality Reduction Techniques on Big Data	Reddy, GT; Reddy, MPK; Lakshmanna, K; Kaluri, R; Rajput, DS; Srivastava, G; Baker, T	IEEE ACCESS	2020	WoS	242

Fonte: Scopus (2022), Web of Science (2022)

Embora a análise de *big data* tenha sido apontada como um novo paradigma de pesquisa em muitas disciplinas, Xiang *et al*. (2015) relatam que poucas aplicações no campo de hotelaria exploram plenamente suas capacidades, desta forma o estudo aplica análise textual para classificar uma grande quantidade de avaliações de dados on-line, avaliando também a qualidade desses dados e identificando as relações entre dois domínios de variáveis (higiene e experiência) na gestão hoteleira. O estudo demonstrou a utilidade da análise de *big data* na identificação de novos padrões de comportamento dos hóspedes, refletindo a maneira como os consumidores "falam" sobre suas experiências em avaliações on-line. O conhecimento do domínio mostrou-se crítico na orientação do processamento de dados e do processo analítico antes de chegar ao ponto em que as relações significativas emergem.

Uma compra fornece ao varejista uma infinidade de informações díspares, incluindo dados transacionais (preço pago, quantidade comprada ou composição do carrinho de compras), dados do consumidor (sexo, idade ou composição familiar) e dados ambientais (temperatura, época do ano ou datas comemorativas). Os varejistas que podem extrair *insights* eficazes do big data podem fazer melhores previsões sobre o comportamento do consumidor, projetar ofertas mais atraentes, segmentar melhor seus clientes e desenvolver ferramentas que incentivem os consumidores a tomar decisões de compra que favoreçam seus produtos. Assim, o *big data* pode iniciar processos benéficos e cíclicos de consumo e engajamento do consumidor que, por sua vez, levam a uma maior lucratividade. Os mundos online e offline estão convergindo. Saber o que é diferente e o que é semelhante nesses dois mundos, bem como as novas tecnologias vão impactar ambos, é fundamental para o futuro do varejo. (GREWAL *et al*. 2017).

Wedel *et al*. (2017) relatam que os dados têm sido chamados de "o petróleo" da economia digital. A captura rotineira de informações digitais por meio de aplicativos on-line e móveis produz vastos fluxos de dados sobre como os consumidores se sentem, se comportam e interagem em torno de produtos e

serviços, e como eles respondem aos esforços de marketing. Os dados estão assumindo um papel cada vez mais central nas organizações, à medida que os profissionais de marketing buscam aproveitá-los para construir e manter relacionamentos com os clientes, personalizar produtos e serviços. O crescimento explosivo de mídia, canais, dispositivos digitais e aplicativos de software proporcionou às empresas oportunidades sem precedentes de alavancar dados para fornecer mais valor aos clientes, aprimorar suas experiências, aumentar sua satisfação e lealdade e extrair valor. Embora inicialmente o potencial do *big data* possa ter sido exagerado e as empresas possam ter investido demais em captura e armazenamento de dados e não o suficiente em análise, está ficando claro que a disponibilidade de *big data* está gerando culturas de decisão baseadas em dados nas empresas, está fornecendo vantagens competitivas e está tendo um impacto significativo em seu desempenho financeiro.

Segundo Wong (2012), o fator chave para obter vantagem competitiva no ambiente de negócios em rápida evolução de hoje é a capacidade de extrair informações úteis a partir do *big data* para gerar *insights* para condução dos negócios. Ser capaz de usar o *big data*, permite que as empresas alcancem excelentes desempenhos contra seus concorrentes. De acordo com Werdigier (2009), os varejistas podem potencialmente aumentar suas margens operacionais em até 60%, aproveitando os valores ocultos no *big data*, no entanto, há uma falta de ferramentas e técnicas analíticas para ajudar as empresas a gerar *insights* úteis a partir de dados para impulsionar a estratégia ou melhorar o desempenho da indústria. (TAN *et al.,* 2015).

1.3 *Machine Learning*

1.3.1 Contexto – Tipos de algoritmos

O problema proposto nesta dissertação é a predição de receita, onde é utilizada a base histórica de vendas de produtos de informática para treinar algoritmos de regressão linear para estimar a receita de venda em períodos futuros.

O problema é definido com as três palavras-chaves: ML, predição e regressão linear, e elas são sobrepostas, ou seja, são termos intercambiáveis, é como chamar o ML como modelagem preditiva e vice-versa.

Em ternos gerais, as tarefas de aprendizagem de máquina são classificadas em três categorias principais: aprendizado de máquina supervisionado, não supervisionado e aprendizado por reforço.

Aprendizado supervisionado (SL)

Esse tipo de aprendizado é caracterizado pelo uso de estruturas de dados que possuem uma coleção de recursos apontando para um resultado (saída). Como a saída desejada já é conhecida, os modelos supervisionados aprendem progressivamente a imitar a saída necessária. Nesse propósito, o sistema de aprendizagem cria sua própria lógica, que pode produzir saídas para consultas sobre novos recursos (JORDAN *et al*., 2015). A Aprendizagem Supervisionada é frequentemente usada para classificação e regressão.

Na classificação o objetivo é prever o rótulo de classe categórico de novas instâncias utilizando regras aprendidas a partir de observações passadas (RASCHKA, *et al*., 2017). A classificação pode ser binária ou pode ser uma série múltipla de classes. Na classificação binária, o modelo aprende uma série de regras que distinguirão entre duas possibilidades (WANG *et al*., 2013), ou a previsão de tendência das ações na bolsa de valores, classificando se o preço aumentará ou diminuirá usando DT e SVM; enquanto a classificação multiclasse tenta mapear uma nova instância em uma das várias possibilidades. Por exemplo, diferenciar amostras de vinho provenientes de três produtores diferentes conhecendo suas características intrínsecas, como ácido málico, alcalinidade de cinzas, nível de magnésio etc. (LICHMAN, 2013).

Por outro lado, a regressão consiste em encontrar uma relação entre as variáveis preditoras e uma variável de resposta contínua para prever um desfecho (RASCHKA *et al*., 2017). O foco desta pesquisa se concentra principalmente neste tipo de problema, pois vários autores têm usado métodos ML muito

poderosos para enfrentar problemas para previsão de vendas. Além disso, mesmo que as abordagens de ML sejam computacionalmente mais caras, elas provaram oferecer melhor desempenho preditivo do que os modelos tradicionais de séries temporais (KE *et al*., 2017), o que prova a necessidade de pesquisa neste campo.

Aprendizado Não-Supervisionado (UL)

Em contraste com o tipo de aprendizagem mencionado acima, um sistema de aprendizagem não supervisionado não é alimentado com uma saída esperada ou feedback explícito para criar suas regras; em vez disso, o sistema deve descobrir padrões explorando uma estrutura de dados e extraindo informações significativas (Zhou *et al*., 2017). Esse tipo de aprendizado é, por exemplo, usado para agrupamento de dados, ou seja, uma maneira de organizar um conjunto de dados em subgrupos, a fim de reconhecer padrões ou estruturas ocultas. Também é utilizado na redução da dimensionalidade, que consiste na escolha das variáveis mais significativas para o modelo. Esta última aplicação é conveniente quando os modelos são complexos e computacionalmente caros, pois reduz o número de recursos necessários.

O aprendizado não supervisionado é uma ferramenta útil para predição de eventos, pois consegue identificar tendências ocultas em estruturas de dados com informações valiosas, onde seja necessário encontrar padrões. Além disso, os conjuntos de dados muitas vezes têm características irrelevantes que só introduzem mais complexidade aos modelos, de modo que aplicar a redução de dimensionalidade é obrigatório. Por exemplo, (Chang *et al*., 2009) usaram *K-Means* para agrupamento de dados para reduzir o ruido nos dados de entrada e efetivamente prever as vendas na indústria de placas de circuito impresso.

Aprendizado por Reforço (RL)

O aprendizado por reforço está relacionado à Aprendizagem Supervisionada, mas em vez de exemplos de treinamento que indicam a saída correta para uma determinada entrada, os dados de treinamento na aprendizagem por reforço são assumidos para fornecer apenas uma indicação sobre se uma ação está correta ou não (JORDAN *et al*., 2015). Em suma, o objetivo é maximizar a recompensa esperada ao longo do tempo, aprendendo uma série de ações que evitarão punições ou penalidades. Uma das aplicações mais impressionantes e recentes de aprendizado de reforço é o *Alpha Go*, o programa de Inteligência Artificial que venceu o campeão europeu Fan Hui por 5-0 no jogo de tabuleiro chinês Go em outubro de 2015 (SILVER *et al.*, 2016). Uma nova versão de Alpha Go, *Alpha Go Zero*, foi lançada em 2017. Ao contrário de suas versões anteriores, *Alpha Go Zero* só aprendeu pelo aprendizado de reforço e os resultados foram surpreendentes: *Alpha Go Zero* venceu *Alpha Go* por 100-0 (SILVER, *et al.*, 2017).

1.3.2 Contexto – Aprendizado do modelo

Existem diferentes métricas para medir o desempenho de algoritmos em ML, sendo que as mais conhecidas para modelos de classificação são: *Accuracy*, *F1-Score*, *Precision*, *Recall* e *AUROC.* Já para modelos de regressão as mais usadas são: *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE) e *Mean Absolute Percentage Error* (MAPE) e *R-squared* (R^2).

Mean Squared Error (MSE)

A métrica MSE calcula a média dos erros do modelo ao quadrado, ou seja, diferenças menores têm menos importância, enquanto diferenças maiores recebem maior peso. O RMSE é a raiz quadrada do MSE, neste caso o erro volta a ter as unidades de medida original da variável dependente. O cálculo é realizado por meio da diferença entre o valor previsto calculado e o valor real. Em seguida, o valor médio é obtido de todas as instâncias. Por fim o RMSE é calculado tomando a raiz quadrada do valor médio resultante. A métrica RMSE é uma das medidas dependentes de escala preferidas devido à sua adequação para avaliar vários modelos construídos usando o mesmo conjunto de dados. A semelhança teórica RMSE com modelos estatísticos o tornaram bem utilizado na avaliação de desempenho de modelos de previsão (HYNDMAN e KOEHLER, 2016).

Mean Absolute Error (MAE)

Parecido com MSE, a métrica MAE ao invés de elevar a diferença entre a previsão do modelo e o valor real ao quadrado, está métrica utiliza o valor absoluto, ou seja, em vez de atribuir um peso de acordo com a magnitude da diferença, ela atribui o mesmo peso a todas as diferenças, de maneira linear.

Mean Absolute Percentage Error (MAPE)

O MAPE calcula a média percentual do desvio absoluto entre as previsões e a realidade. É utilizado para avaliar sistemas de previsões de vendas e outros sistemas no qual a diferença percentual seja mais interpretável, ou mais importante, do que os valores absolutos.

R-Squared (R^2)

Em estatística, o coeficiente de determinação, denotado R^2 e pronunciado "*R-squared",* é a proporção da variação na variável dependente que é previsível a partir das variáveis independentes. É uma métrica utilizada no contexto de modelos estatísticos cujo principal objetivo é a previsão de resultados futuros ou o teste de hipótese, com base em outras informações relacionadas. Ele fornece uma medida de quão bem os resultados observados são replicados pelo modelo, com na base proporção da variação total dos resultados explicados pelo modelo (WANG *et al*., 2021).

Quanto maior o resultado de *R-squared* melhor, ou seja, significa que o modelo está conseguindo predizer a variável independente com assertividade, porém deve-se tomar cuidado para não chegar a um resultado de 100%, caso contrário ocorrerá um super ajustamento a base de dados de treinamento e não conseguira tratar uma base real.

Em ML geralmente dividimos a massa de dados em dois subconjuntos: dados de treinamento e dados de teste. Desta forma é realizado um ajuste no modelo aos dados de treinamento, a fim de fazer previsões sobre os dados de teste. Quando este procedimento é realizado, um de dois cenários podem ocorrer: super ajustamos (*Overfitting*) ou sub (Underfitting) ao modelo. Não é interessante que estes cenários aconteçam, pois podem afetar a previsibilidade do modelo, provocando um modelo com menor precisão e/ou não generalizado, ou seja, não será possível generalizar previsões utilizando outras massas de dados.

O *Overfitting* é um ponto de atenção, pois significa que o modelo está super ajustado ao conjunto de dados de treinamento. Isso geralmente acontece quando o modelo é complexo (ou seja, muitos recursos/variáveis em comparação com o número de observações). Este modelo será muito preciso nos dados de treinamento, mas provavelmente não será muito preciso em dados não treinados ou novos. Desta forma esse modelo não é generalizado (*not AS generalized*), o que significa que você pode generalizar os resultados e não pode fazer nenhuma inferência em outros dados, que é, em última análise, o objetivo do modelo. Basicamente, quando isso acontece, o modelo aprende ou descreve o "*noise*" nos dados de treinamento em vez dos relacionamentos reais entre as variáveis nos dados. Esse ruído, obviamente, não faz parte de nenhum novo conjunto de dados e não pode ser aplicado a ele.

O *Underfitting* ocorre quando um modelo está sub ajustado, isso significa que o modelo não se ajusta aos dados de treinamento e, portanto, perde as tendências nos dados. Isso também significa que o modelo não pode ser generalizado para novos dados. Isso geralmente é o resultado de um modelo muito simples (não há preditores/variáveis independentes suficientes). Isso também pode acontecer quando, por exemplo, ajustamos um modelo linear (como regressão linear) a dados que não são lineares. Desta forma este modelo terá baixa capacidade preditiva (em dados de treinamento e não pode ser generalizado para outros dados).

No entanto, às vezes é necessário avaliar o desempenho do modelo usando uma medida *ad-hoc*, pois conforme visto anteriormente, o excesso de treinamento (super ajustamento) e a ausência de adequação

do modelo (sub ajustamento) devem ser evitados para garantir que o modelo obtido forneça bons resultados utilizando dados desconhecidos.

Cross-Validation (CV)

Técnicas como validação cruzada (CV), podem ser úteis para evitar o *Overfitting* e *Underfitting*. A CV é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados (KOHAVI, 1995).

Esta técnica é amplamente empregada em problemas onde o objetivo da modelagem é a predição. Busca-se então estimar o quão preciso é este modelo na prática, ou seja, o seu desempenho para um novo conjunto de dados (MCLACHLAN *et al.*, 2005).

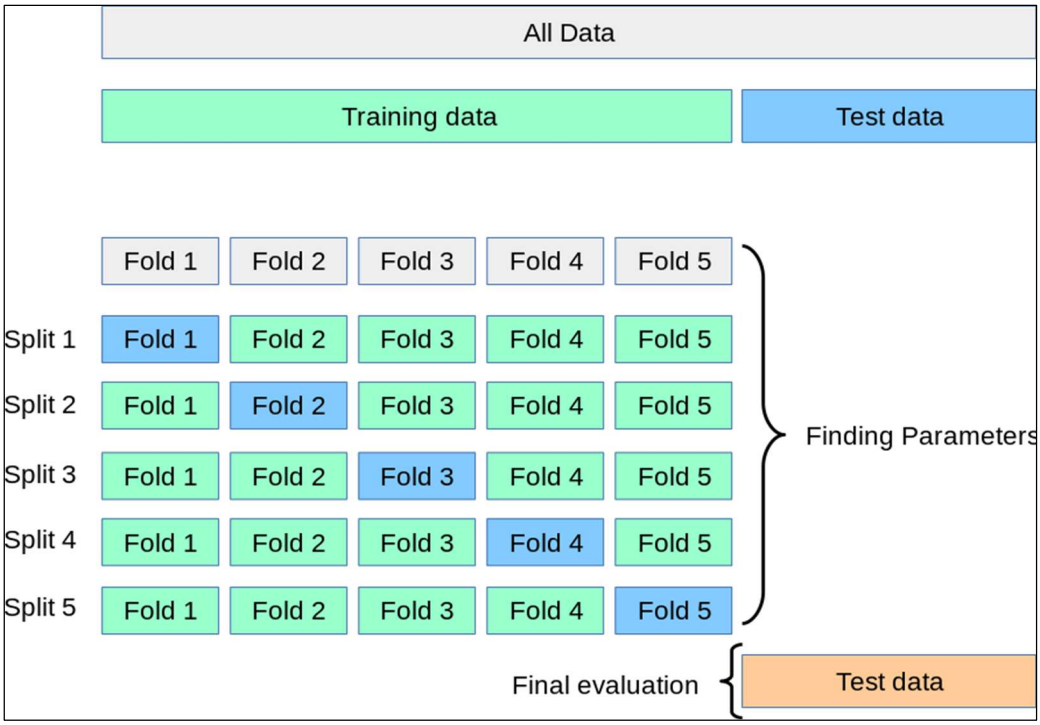
O conceito central das técnicas de validação cruzada é o particionamento do conjunto de dados em subconjuntos mutuamente exclusivos, e posteriormente, o uso de alguns destes subconjuntos para a estimação dos parâmetros do modelo (dados de treinamento), sendo os subconjuntos restantes (dados de validação ou teste) empregados na validação do modelo.

Conforme relatado por Kohavi (1995), existem algumas formas para realizar o particionamento dos dados, sendo as três mais utilizadas: o método *holdout*, o *k-fold* e o *leave-one-out*.

A medida de desempenho relatada pela validação cruzada *k-fold* é então a média dos valores calculados no loop. Essa abordagem pode ser computacionalmente cara, mas utiliza plenamente os dados disponibilizados evitando desperdício, como é o caso ao se fixar um conjunto de validação arbitrário ao utilizar o método *train_test_split,* no qual obrigatoriamente é definido um percentual para dados de treino e um percentual para dados de teste, normalmente utilizado 70% e 30% respectivamente. Com isso o CV *k-fold* possui uma grande vantagem em problemas como inferência inversa, onde o número de amostras é muito pequeno.

A Figura 9 Figura 9, extraída do site *scikit-learn.org* mostra a estrutura do *k-fold*.

Figura 9 - Estrutura do *Cross-Validation k-fold*



Fonte: https://scikit-learn.org/stable/modules/cross_validation.html

Na validação cruzada de *k-fold* o conjunto de dados se divide em um número K. Na Figura 9 é mostrado o conceito da validação cruzada de 5 vezes ou K+5, mesmo método utilizado nesta pesquisa. Neste cenário, o método dividirá o conjunto de dados em cinco dobras. O algoritmo utiliza a primeira dobra na primeira iteração (em azul) para testar o modelo, as demais dobras (em verde) é o restante do conjunto de dados utilizados para treinar o modelo. O mesmo processo repete-se até que o conjunto de testes utilize cada uma das cinco dobras.

1.3.3 Enquadramento do Problema

O problema apresentado nesta dissertação é uma tarefa supervisionada como a maioria das tarefas práticas de aprendizagem de máquina, pois o algoritmo tem algum conhecimento prévio sobre qual deve ser a saída. Uma tarefa supervisionada de aprendizagem de máquina é definida da seguinte forma: o processo de aprendizado é obtido a partir da alimentação do modelo com alguns dados para ser capaz de aprender uma função de mapeamento que não sabemos, mas que os algoritmos tentarão descobrir.

Basicamente para problemas de aprendizado de máquina existem as variáveis de entrada X (também chamadas de variáveis independentes) e as variáveis de saída Y (também chamadas de variáveis dependentes), e usando diferentes algoritmos de aprendizado de máquina uma função de mapeamento de X para Y é aprendida como na seguinte equação: {Y = f(X)}. Além disso, há o fator de erro que é independente dos preditores: {Y = f(X) + e} que é chamado de erro irredutível porque não pode ser reduzido não importa o quão boa a função de mapeamento seja estimada. O poder desta ciência é que o modelo treinado será então capaz de prever valores de novos dados "invisíveis" além dos pontos de dados de treinamento que foram inicialmente alimentados ao modelo.

Existem dois tipos de problemas de aprendizagem supervisionados: classificação e regressão. O problema aqui proposto é uma regressão porque os valores da variável alvo ou variável dependente “Previsão de Vendas” é representada pelo eixo Y na equação e necessariamente deve ser um tipo numérico. Assim, quando um algoritmo é aplicado para mapear Y a partir de X, trata-se de um processo de desenvolvimento de um modelo de predição cujo objetivo final de desenvolvimento é fornecer as previsões mais precisas que sejam mais próximas dos valores reais.

Otimização de hiper parâmetros

Otimização ou ajuste de hiper parâmetros é um processo pelo qual o melhor conjunto de hiper parâmetros são escolhidos para que resultem no melhor desempenho do modelo. O modelo basicamente fará trilhas até que os melhores hiper parâmetros sejam encontrados. Cada trilha é, em si, um processo completo de treinamento do modelo. Portanto, é um processo por meio do qual o processo de treinamento pode ser controlado até certo ponto e seu objetivo é otimizar o desempenho do modelo dependendo dos possíveis valores que esses parâmetros podem assumir. Os hiper parâmetros diferem dos parâmetros normais, pois são definidos antes do processo de aprendizagem começar, e a maioria dos algoritmos de aprendizagem de máquina disponibilizam esses hiper parâmetros, mas são específicos para cada tipo de algoritmo. O efeito dos hiper parâmetros se estende não apenas à duração do processo de treinamento, mas também à precisão das previsões produzidas pelo algoritmo. Existem duas abordagens comuns para ajuste de hiper parâmetros: pesquisa de grade e pesquisa aleatória. Na abordagem de pesquisa de grade, haverá um espaço de grade no qual são definidos valores possíveis dos hiper parâmetros e o modelo testará cada combinação deles. Enquanto na pesquisa aleatória, nenhum conjunto discreto de valores é fornecido como uma grade para exploração, mas sim uma combinação aleatória de faixas de valores é escolhida e testada para o melhor desempenho.

1.3.4 Bibliometria

Para mensuração do volume de publicações na literatura cientifica sobre ML, as bases Scopus, Web of Science, ACM e IEEE Xplore foram consultadas, conforme critérios no Quadro 2.

Quadro 2 - Critérios de busca sobre utilização de ML na previsão de vendas

Atributo	Critério
Expressão de busca	ALL= ("*Machine Learning*" AND ("*Sales Prediction*" OR "*Sales Forecast*") AND ("*Retail*" OR "Sales"))
Período	1994 a 2022
Idioma	Inglês
Tipo de publicação	Artigos de periódicos e conferências
Exclusão de domínios de pesquisa	Medicina, Engenharia e Arquitetura, Educação, Psicologia, Agricultura e Biociências, Artes e Humanidades, Bioquímica, Geociências e Logística

Fonte: Autor

A Tabela 4 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 4 - Número de publicações das bases consultadas sobre utilização de ML na previsão de vendas

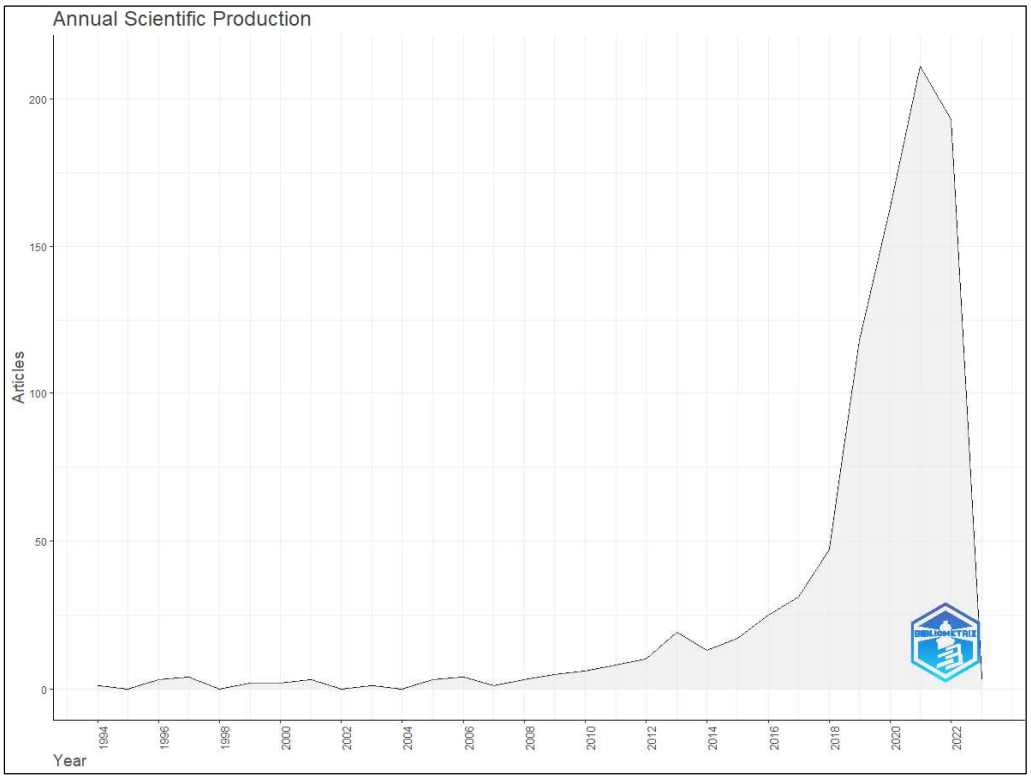
Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	897
Web of Science	Tópico	55
ACM Digital Library	Títulos, Resumos e Palavras-chave	64
IEEE Xplore	*All-Metada*	57

Fonte: Scopus (2022), Web of Science (2022), ACM Digital Library (2022), IEEE Xplore (2022)

Com os arquivos de dados consolidados gerados pelas bases Scopus e Web Of Science, foi elaborado um gráfico, conforme Figura 10, que apresenta a evolução da produção cientifica sobre ML relacionado a predição de vendas no segmento de varejo, em número de publicações, ao longo dos anos.

Neste gráfico se observa um discreto crescimento no número de publicações a partir do ano de 2008, tornando-se pronunciado na década de 2010, e mais acentuado no ano de 2021. Quanto ao ano de 2022, deve-se considerar que os dados são parciais, acumulados até o mês de Agosto, porém observa-se uma estagnação no número de publicações, ou seja, não existe uma tendência no aumentou das publicações como apresentado entre os anos de 2017 e 2022. Mesmo assim se verifica que o patamar do número de publicações se mantém. Constata-se assim um interesse relevante e consistente em volume de publicações no início desta década.

Figura 10 - Evolução das publicações sobre utilização de ML na previsão de vendas



Fonte: Scopus (2022), Web of Science (2022)

Na Tabela 5 são apresentados os crescimentos percentuais das publicações nas bases Scopus e Web of Science, nos períodos de 10 anos, 5 anos e 1 ano.

Tabela 5 - Crescimento das publicações sobre utilização de ML na previsão de vendas

Períodos	**10 anos**	**5 anos**	**1 ano**
Bases	**2012 a 2022**	**2017 a 2022**	**2021 a 2022**
Scopus	1.820 %	562 %	-8,5 %
Web of Science	800 %	60 %	-33 %

Fonte: Scopus (2022), Web of Science (2022)

A consulta à base Scopus e Web of Science possibilitou a identificação dos países com maior número de publicações sobre ML relacionado a predição de vendas no segmento de varejo. A Tabela 6 mostra os 10 países que mais publicaram. Verifica-se a liderança da China, com a Índia em segundo, seguidos dos Estados Unidos, Turquia e Reino Unido.

Tabela 6 - Dez países que mais publicaram sobre utilização de ML na previsão de vendas

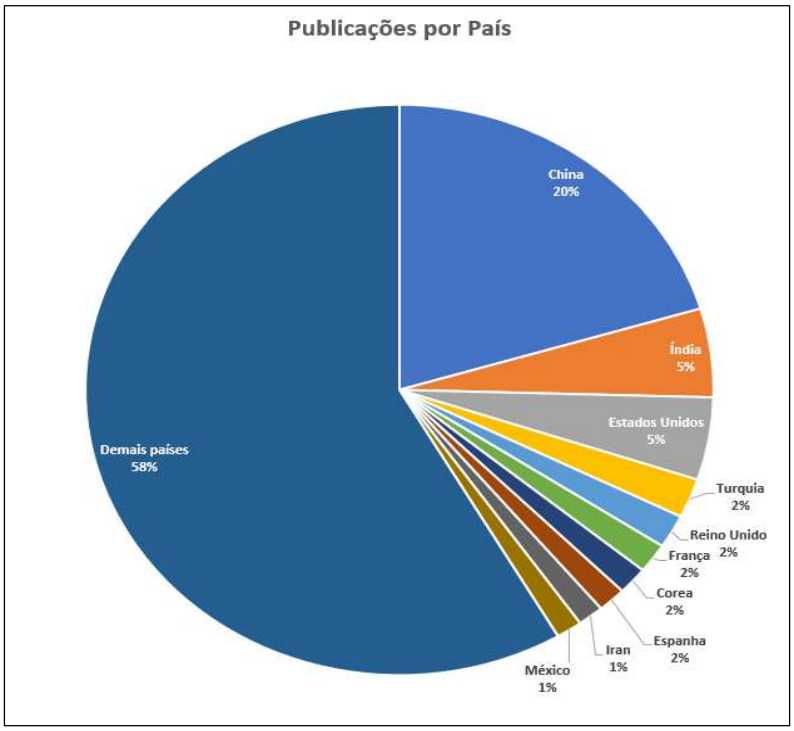
Países	**Publicações**
China	183
Índia	45

Países	Publicações
Estados Unidos	42
Turquia	20
Reino Unido	17
França	15
Corea	14
Espanha	13
Iran	12
México	12
Demais países	524

Fonte: Scopus (2022), Web of Science (2022)

Na Figura 11, visualiza-se a participação percentual dos dez países que mais publicaram, também conforme dados das bases Scopus e Web of Sciense.

Figura 11 - Dez países que mais publicam sobre utilização de ML aplicados na previsão de vendas



Fonte: Scopus (2022), Web of Science (2022)

A Tabela 7 apresenta dez publicações com conteúdo pertinente ao ML, dentre as mais citadas, conforme as bases Scopus e Web of Sciense.

Tabela 7 - Artigos mais citados sobre abordagens de ML aplicados na previsão de vendas

Título	Autores	Fonte	Ano	Base	TC
Deriving the pricing power of product features by mining consumer reviews	Archak N.	Management Science	2007	Scopus	164

Título	Autores	Fonte	Ano	Base	TC
Sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm	R. J. Kuo	European Journal of Operational Research	2001	Scopus	129
Sales forecasts in clothing industry: The key success factor of the supply chain management	Thomassey S.	International Journal of Production Economics	2010	Scopus	127
A hybrid sales forecasting system based on clustering and decision trees	Thomassey S.	Decision Support Systems	2006	Scopus	118
Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling	Yuandong Wang e Hongzhi Yin	Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining	2019	Scopus	91
Exploring the use of deep neural networks for sales forecasting in fashion retail	A.L.D. Loureiro,	Decision Support Systems	2018	Scopus	79
Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management	H.D. Nguyen	International Journal of Information Management	2021	Scopus	70
Analysis of book sales prediction at Amazon marketplace in India: a machine learning approach	S. K. Sharma	Information Systems and e-Business Management	2019	WoS	56
Mining online reviews for predicting sales performance: A case study in the movie domain	Xiaohui Y.	IEEE Transactions on Knowledge and Data Engineering	2012	Scopus	24
A Deep Learning Approach for the Prediction of Retail Store Sales	Y. Kaneko	IEEE 16th International Conference on Data Mining Workshops (ICDMW).	2016	WoS	13

Fonte: Scopus (2022), Web of Science (2022)

Com ênfase na análise textual, Archak *et al.* (2007) indicam que a análise de vendas dos produtos deve ser multifacetada, ou seja, além dos histórico quantitativo das vendas com base em variáveis numéricas, deve ser considerado a análise qualitativa com base na experiência dos consumidores. Desta forma o conceito de regressão hedônica foi utilizado para estimar o peso no qual os clientes atribuem a cada característica individual do produto e a pontuação de avalição implícita que os clientes atribuem a cada recurso. Para esse objetivo, foi desenvolvido uma nova técnica hibrida combinando mineração de texto e econometria.

R. J. Kuo (2001), integra os resultados obtidos pelo modelo GFNN com pesos iniciais a um modelo ANN usando dados da série temporal de uma loja de conveniência. Os resultados indicam que o sistema proposto fornece previsões mais confiáveis do que o método estatístico convencional utilizando uma única ANN.

Dentre os documentos listados, vale destacar o artigo "*Sales forecasts in clothing industry: The key success factor of the supply chain management"* de Thomassey com mais de cem citações, cujo qual se aprofunda no segmento de varejo da indústria têxtil, destacando as características de previsões de longo

e curto prazo, ambas com diferentes restrições e técnicas. Indicando que a chave do sucesso para uma previsão eficaz está na existência, relevância e confiabilidade dos dados históricos contidos em sistemas de informações dos distribuidores. Uma simulação de previsão no processo de fabricação permitiu quantificar o impacto dos erros de previsão no financeiro e consequentemente no desempenho da organização, ratificando que tais erros ocasionam aumento dos níveis de estoque, perda de vendas e frequentemente uma queda da margem bruta para o varejista, indicando o início do efeito chicote. Os sistemas de previsão que utilizam técnicas avançadas como ML são, portanto, uma forma eficiente de reduzir este fenômeno negativo para a indústria de vestuário.

Thomassey e Fiordaliso (2006), abordam a previsão de vendas utilizando técnicas de cluster com algoritmo *k-means* e classificadores de arvore de decisão, úteis para estimar perfis de vendas de novos itens para os quais não existe dados históricos de vendas.

Por meio de um modelo PLSA, Xiaohui *et al.* (2012) abordam como a análise de sentimentos pode apoiar a predição de vendas. Inicialmente é aplicado o modelo ARSA para tratar os sentimentos complexos deixados por usuários ao comentar filmes e na sequência é aplicado o modelo ARSQA. O uso generalizado de avaliações online como forma de transmitir opiniões e comentários tem proporcionado uma oportunidade única para compreender os sentimentos do público em geral e derivar inteligência de negócios. A precisão e eficácia dos modelos propostos foram confirmados pelos experimentos em dois conjuntos de dados de filme.

Tomar decisões adequadas é de fato um fator chave para ajudar as empresas a enfrentar os desafios das cadeias de suprimentos nos dias de hoje. Nguyen *et al.* (2021), propõe duas abordagens orientadas a dados que permitem tomar melhores decisões na gestão da cadeia de suprimentos. Em particular, é sugerido um método baseado em rede de LSTM para prever dados de séries temporais multivariadas e um método baseado em rede LSTM *Autoencoder* combinado com um algoritmo de máquina de vetor de suporte de uma classe para detectar anomalias nas vendas. Os resultados obtidos mostram que o método baseado em LSTM *Autoencoder* leva a um melhor desempenho para detecção de anomalias em comparação com o método baseado em LSTM sugerido em um estudo anterior. O método de previsão proposto para dados de séries temporais multivariadas também tem um desempenho melhor do que alguns outros métodos baseados em um conjunto de dados fornecido pela NASA.

Os aplicativos de carona estão se tornando cada vez mais populares para fornecer aos motoristas e passageiros serviços de transporte convenientes. Para obter os padrões de mobilidade dos passageiros, as plataformas online de serviços de carona precisam prever antecipadamente o número de demandas de

passageiros de uma região para outra. Por meio de uma matriz de previsão de origem e destino ODMP, Wang e Yin (2019), relatam que o problema ODMP é mais desafiador do que a previsão de demanda comum, pois além do número de demandas de uma região, o modelo também precisa prever o destino delas. Além disso, a escassez de dados é um problema grave. Para resolver o problema de forma eficaz, é proposto um modelo unificado, aprendizado multitarefa baseado em *Grid-Embedding* (GEML), que consiste em dois componentes com foco em informações espaciais e temporais, respectivamente. A parte *Grid-Embedding* é projetada para modelar os padrões de mobilidade espacial dos passageiros e relações vizinhas de diferentes áreas, cujo agregador preponderado visa detectar a dispersão e a variedade de dados. O framework *Multi-task Learning* se concentra em modelar atributos temporais e capturar vários objetivos do problema ODMP. Os resultados experimentais demonstram a superioridade do GEML em relação a outras abordagens de última geração.

No setor de varejo de moda cada vez mais competitivo, as empresas estão constantemente adotando estratégias focadas em ajustar as características dos produtos para atender de perto as necessidades e preferências dos clientes. Embora os ciclos de vida dos produtos de moda sejam muito curtos, a definição de estratégias de estoque e compras pode ser suportada pela grande quantidade de dados históricos que são coletados e armazenados nos bancos de dados das empresas. Loureiro *et al.* (2018), exploram o uso de uma abordagem de aprendizado profundo para prever vendas na indústria da moda, prevendo as vendas de novos produtos individuais em estações futuras. Os modelos foram desenvolvidos considerando um conjunto amplo e diversificado de variáveis, nomeadamente as características físicas dos produtos e a opinião de especialistas do domínio. O estudo compara as previsões de vendas obtidas com a abordagem de aprendizado profundo com um conjunto de técnicas rasas, ou seja, Árvores de Decisão, Floresta Aleatória, Regressão Vetorial de Suporte, Redes Neurais Artificiais e Regressão Linear. Verificou-se que o modelo que emprega aprendizado profundo tem um bom desempenho para prever vendas no mercado de varejo de moda, no entanto, para parte das métricas de avaliação consideradas, ele não apresenta desempenho significativamente melhor do que algumas das técnicas rasas, como Random Forest.

Os resultados demonstram que o uso de DNN e outras técnicas de mineração de dados para realizar previsão de vendas no setor de varejo de moda quando não há dados históricos de vendas é muito promissora. Os modelos propostos podem constituir uma importante ferramenta para auxiliar os gestores na aquisição de seus produtos. Em particular, a DNN superou as técnicas restantes em parte do desempenho das técnicas consideradas. Portanto, embora geralmente sugerido como apropriado para a análise de grandes bancos de dados, as técnicas de DNN também podem ter um bom desempenho quando aplicadas a conjuntos de dados menores. No entanto, é importante ressaltar que na prática essa

técnica pode não ser a mais indicada, pois seu processo de treinamento é mais complexo quando comparada com outras técnicas mais simples que ainda apresentam a mesma capacidade de predição, como RF.

A previsão na demanda dos clientes é uma parte importante do gerenciamento da cadeia de suprimentos, pois ajuda a evitar excesso ou falta de produção e reduz o tempo de entrega. No contexto do comércio eletrônico, a previsão precisa na demanda do cliente, normalmente é capturada pelo volume de vendas, cuja qual requer uma análise cuidadosa de vários fatores, como o tipo de produto, país de compra, preço, taxa de desconto, opção de entrega gratuita, sentimento de avaliação online, entre outras. Para grandes varejistas eletrônicos, esse tipo de capacidade de previsão também é extremamente importante para gerenciar a cadeia de suprimentos com eficiência e garantir a satisfação do cliente. Sharma *et al*. (2019), investigam a eficácia de várias técnicas de modelagem, como análise de regressão, análise de árvore de decisão e rede neural artificial, para prever as vendas de livros de uma grande varejista, usando vários fatores relevantes e suas interações como variáveis preditoras. A análise de sentimento é realizada para medir a polaridade das revisões online, que são incluídas como preditores nesses modelos. A importância de cada variável preditora independente, como taxa de desconto, sentimento de revisão etc., é analisada com base no resultado de cada modelo para determinar os principais preditores significativos que podem ser controlados pelo profissional de marketing para influenciar as vendas. Em termos de precisão de previsão, o modelo de rede neural artificial tem um desempenho melhor do que o modelo baseado em árvore de decisão. Além disso, a análise de regressão, com e sem fatores de sentimento e interação, geram resultados comparáveis. Todos os três modelos confirmam que o volume de revisão, ou seja, livros com edições revisadas é o preditor mais importante e significativo das vendas de livros. Em segundo lugar, a taxa de desconto, o valor do desconto e as classificações médias têm um efeito mínimo ou insignificante na previsão de vendas. Em terceiro lugar, tanto o sentimento negativo quanto o sentimento positivo das revisões são preditores individualmente significativos de acordo com o modelo de regressão e árvore de decisão, mas não são significativos de acordo com o modelo de rede neural.

Kaneko *et al.* (2016), construíram um modelo de previsão de vendas para lojas de varejo usando a abordagem de ML. Por meio da utilização de dados históricos de três anos do ponto de venda de uma loja do varejo é construído um modelo de previsão de vendas, cujo qual prevê as mudanças nas vendas do dia seguinte. Os resultados indicaram que o ML é altamente adequado para a construção de modelos que incluam múltiplas variáveis.

1.4 Regressão linear

1.4.1 Contexto

A previsão de séries temporais envolve coletar e analisar observações passadas para desenvolver um modelo para prever essas observações para o futuro. A previsão de eventos futuros é importante em muitos campos para apoiar a tomada de decisões, pois contribui para reduzir a incerteza futura. Esta pesquisa aplica um método multidisciplinar para selecionar o melhor algoritmo de ML baseado em regressão linear para predição de valor de venda, cujo método iterativo começa com um modelo base e explica os erros do modelo por meio do RMSE. A cada iteração, o caminho que leva ao erro mais baixo é adicionado como uma nova variável ao modelo base. Nesse sentido, essa abordagem pode ser considerada uma melhoria em relação aos métodos encontrados na literatura para comparar modelos de regressão linear, pois permite incorporar características não lineares por explicação residual. Por meio de um estudo numérico detalhado os algoritmos são comparados e avaliados. RMSE é uma medida de desvio da diferença entre o valor real e o valor previsto, tendo a mesma unidade de medida do atributo alvo, tornando mais fácil mensurar a exatidão da predição.

É observado que a regressão linear melhora substancialmente o desempenho do modelo base por meio de recursos extraídos e fornece um desempenho comparável a outras abordagens bem estabelecidas. A interpretação das previsões do modelo e o alto desempenho preditivo da regressão linear a torna um método mais eficaz do que métodos tradicionais como ARIMA que aplicam médias móveis (ILIC *et al*., 2021).

A previsão de séries temporais tem aplicações importantes em vários domínios, incluindo energia (Deb *et al*., 2017) , finanças (Krollner *et al*., 2010) e clima (Baboo *et al.*, 2010) . Previsões precisas fornecem *insights* sobre as tendências no domínio e servem como insumos para decisões envolvendo eventos futuros. Nas operações da cadeia de suprimentos, as previsões de vendas e demanda dos produtos são essenciais para o controle de estoque, planejamento da produção e programação da força de trabalho. Assim, ferramentas de previsão eficazes estão diretamente ligadas ao aumento de lucros e redução de custos. Os métodos de previsão quantitativa são geralmente divididos em duas categorias: modelos gerais de séries temporais e modelos baseados em regressão. Modelos gerais de séries temporais, como suavização exponencial e média móvel integrada autorregressiva ARIMA, são normalmente derivados das informações estatísticas nos dados históricos. Por outro lado, os modelos de regressão dependem da construção de uma relação entre variáveis independentes (características como observações anteriores) e variáveis dependentes (resultados alvo). Há uma ampla gama de abordagens de regressão usadas para previsão de séries temporais, incluindo regressão linear, métodos de conjunto e redes neurais.

Embora a maioria dos estudos sobre previsão de séries temporais se concentre em previsões pontuais, muitas aplicações se beneficiam de ter previsões probabilísticas que possam fornecer informações sobre incerteza futura. Por exemplo, em negócios de varejo, as previsões probabilísticas permitem gerar diferentes estratégias para uma série de possíveis resultados fornecidos pelos intervalos de previsão. Uma previsão probabilística normalmente consiste em limites superior e inferior, e o intervalo correspondente pode ser tomado como um intervalo de confiança em torno das previsões pontuais. Métodos padrões como suavização exponencial e ARIMA geram previsões probabilísticas por meio de simulações ou expressões de forma fechada para a distribuição preditiva alvo (BOX *et al*., 2015). Estudos recentes propõem modelos de aprendizado profundo para previsão probabilística que visam prever os parâmetros da distribuição de probabilidade subjacente (ou seja, média e variância) para a próxima etapa de tempo e mostram melhorias de desempenho sobre abordagens padrão para grandes conjuntos de dados que consistem em muitas séries temporais (RANGAPURAM, 2018; SALINAS, 2020).

Na modelagem preditiva, muitas vezes os modelos são avaliados medindo-se seu desempenho de previsão obtido por meio de um conjunto de testes baseado em métricas como erro médio absoluto e erro quadrático médio, desconsiderando a interpretação das previsões do modelo. No entanto, os modelos interpretáveis têm certos benefícios, como criar uma confiança em relação ao modelo pela caracterização explícita da contribuição dos fatores para as previsões e fornecer uma melhor compreensão científica do modelo. O valor da interpretação do modelo foi reconhecido em estudos recentes e levou a novos caminhos para esta pesquisa (ABURTO 2007; GUIDOTTI, 2018).

Vários estudos recentes sobre previsão de séries temporais recorrem a arquiteturas complexas de aprendizado profundo, que normalmente produzem resultados relativamente precisos quando os dados disponíveis são abundantes. As desvantagens de tais abordagens incluem sua carga computacional e a natureza de caixa preta (RANGAPURAM, 2018; SALINAS, 2020). A este respeito, os modelos lineares podem fornecer um bom compromisso entre precisão e simplicidade. Especificamente, modelos lineares com formas matemáticas simples são geralmente preferidos por sua interpretação e explicação dos resultados do modelo.

Este estudo propõe um método de previsão de séries temporais adequado para previsões determinísticas e probabilísticas que melhora iterativamente suas previsões por meio da geração de características baseadas na exploração residual. O modelo proposto possui duas etapas. Na primeira etapa, um modelo de previsão genérico (ou seja, um aprendiz de base) é treinado para obter as previsões de base. No segundo estágio, é aplicado a validação cruzada, na qual o *dataset* é divido em várias partes

para treinamento e teste e não somente em duas partes como 70% e 30% comumente utilizado. Uma representação visual do método proposto é fornecida na seção 3.5.1.3.

Especificamente, os algoritmos de regressão linear ajustam os resíduos obtidos do modelo de predição com base na validação cruzada, ou seja, a cada iteração, um novo ciclo de treinamento e teste é executado, com isso todo o *dataset* passa pelo processo de treinamento e teste, descartando a utilização de uma taxa de aprendizagem fixa.

Os modelos básicos de séries temporais e outros modelos lineares dependem de tendências e sazonalidade e deixam componentes inexplicáveis como ruído. Devido ao número exponencial de interações potenciais, métodos baseados em kernel (MÜLLER *et al.*, 1997) são introduzidos para capturar tais interações, bem como a não linearidade sem introduzir explicitamente os recursos no modelo. No entanto, esses kernels fazem com que o modelo perca sua interpretação. Nesta pesquisa as interações entre as feições por meio da validação cruzada e os testes paramétricos adicionam explicitamente essas informações ao modelo. Para a previsão de vendas no varejo, se houver uma interação implícita entre as variáveis de feriado e dia de promoção que crie um efeito maior do que seus efeitos individuais, uma nova variável (não linear) é incluída no modelo ajustado. Esta variável implica que a promoção tem maior efeito nas vendas quando aplicada em feriado. De maneira semelhante, esta pesquisa é capaz de capturar muitos termos de interação polinomial e incorporá-los ao modelo ajustado dependendo do algoritmo que apresentar o melhor desempenho.

Conforme relatado por Al-Gunaid *et al.* (2018), para conjunto de dados com pequenas amostras não é possível usar efetivamente algoritmos com base em redes neurais, uma vez que tais redes exigem um grande volume de dados históricos, além disso uma desvantagem das redes neurais é a presença de muitos parâmetros livres, os quais permitem que o sistema neural ajuste os dados de treinamento de forma arbitrária, resultando em uma rede sobre ajustada, ou seja, *overfitting* (DESAI, 1998).

Mediante aos achados acima, esta pesquisa compara seis algoritmos em ML com base em regressão linear:

ElasticNet Regression - ElasticNET

Trata-se de um método de regressão regularizado que combina linearmente as penalidades dos métodos LASSO e RIDGE. *ElasticNET* é particularmente útil quando o número de preditores é muito maior que o número de observações (ZOU *et al*., 2005).

Lasso Regression - LASSO

Em estatística e ML, *Least Absolute Shrinkage and Selection Operator* (LASSO) é um método de análise de regressão que realiza seleção e regularização de variáveis para melhorar a precisão da previsão e a interpretação do modelo estatístico resultante. Foi originalmente introduzido na geofísica por Santosa e Symes (1986) e posteriormente por Tibshirani (1996).

Linear Regression - LR

A regressão linear é uma técnica estatística que usa a existência de uma relação de associação entre uma variável dependente (variável de resultado/alvo) e uma variável independente (variável preditora). Na LR, os relacionamentos são modelados usando funções de previsão linear cujos parâmetros de modelo desconhecidos são estimados a partir dos dados. Tais modelos são chamados de modelos lineares. Como todas as formas de análise de regressão, a LR se concentra na distribuição de probabilidade condicional da resposta conforme os valores dos dados preditores, em vez da distribuição de probabilidade conjunta de todas essas variáveis, que é domínio da análise multivariada (FREEDMAN, 2009).

A forma funcional da LR é a seguinte:

$$Yi = \beta 0 + \beta l\ Xi + \varepsilon i$$

Onde:

Yi = Valor da enésima observação da variável dependente

Xi = Variável independente da observação

εi = Erro aleatório (resíduos) na previsão do valor Yi

β0 e βl = são parâmetros da regressão

Os parâmetros β0 e βl são estimados minimizando a soma dos erros quadrados.

Random Forest - RF

Esta técnica deriva das árvores de decisão, em inglês *Decision Tree* (DT) e foi desenvolvida para superar as fragilidades reveladas pelo DT, permitindo melhorar a precisão e contornando problemas como a alta sensibilidade a pequenas variações nos dados. É baseado em conjunto de árvores seguido pelo cálculo do valor médio das previsões obtidas no nó final de cada árvore e evita a ausência de robustez demonstrada por uma única árvore de decisão. Nesta abordagem, cada árvore é cultivada

usando um subconjunto de variáveis independentes selecionadas aleatoriamente. Parecido com o DT, RF é considerado um método simples (BREIMAN, 2001).

Cutler *et al.* (2007), relatam tanto para classificação quanto para regressão, RF é um dos melhores métodos disponíveis e superior à maioria dos métodos de uso comum. Como o nome sugere, a RF combina muitas árvores de classificação para produzir classificações mais precisas. Os subprodutos dos cálculos de RF incluem medidas de importância variável e medidas de similaridade de pontos de dados que podem ser usados para agrupamento, dimensionamento multidimensional e representação gráfica.

Ridge Regression - RR

RR é um método para estimar os coeficientes de modelos de regressão múltipla em cenários onde as variáveis independentes são altamente correlacionadas. Este método tem sido usado em muitos campos, incluindo econometria, química e engenharia (HILT e SEEGRIST, 1977).

Support Vector Regression - SVR

A regressão de vetores de suporte (SVR) é uma categoria de *Support Vector Machine* que visa mapear os dados originais em um espaço de recursos de alta dimensão. Nesse espaço, o SVR encontra os melhores hiperplanos de regressão, permitindo estimar o valor da variável (WITTEN *et al*. 2011).

1.4.2 Bibliometria

Para mensuração do volume de publicações na literatura cientifica sobre regressão linear, as bases Scopus, Web of Science, ACM e IEEE Xplore foram consultadas, conforme critérios no Quadro 3.

Quadro 3 - Critérios de busca sobre utilização de LR na previsão de vendas

Atributo	Critério
Expressão de busca	ALL = ("*Linear Regression*" AND ("*Sales Prediction*" OR "*Sales Forecast*" OR "*Sales*" OR "*Retail*"))
Período	1970 a 2022
Idioma	Inglês
Tipo de publicação	Artigos de periódicos e conferências
Exclusão de domínios de pesquisa	Medicina, Engenharia e Arquitetura, Educação, Psicologia, Agricultura e Biociências, Artes e Humanidades, Bioquímica, Geociências e Logística

Fonte: Autor

A Tabela 8 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 8 – Número de publicações das bases consultadas sobre utilização de LR na previsão de vendas

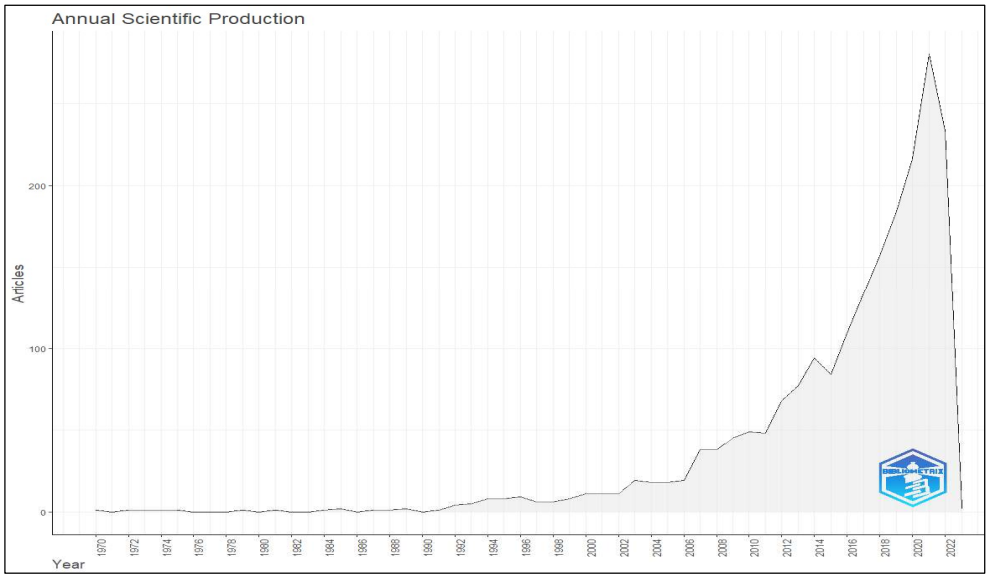
Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	1.450
Web of Science	Tópico	789
ACM Digital Library	Títulos, Resumos e Palavras-chave	961
IEEE Xplore	*All-Metada*	2.635

Fonte: Scopus (2022), Web of Science (2022), ACM Digital Library (2022), IEEE Xplore (2022)

Um estudo bibliométrico é realizado para analisar as contribuições dos principais autores e países em termos de produtividade e citações bibliográficas. Uma amostra de 1.893 artigos entre o período de 1970 a 2022 são extraídos das bases de dados mencionadas na Tabela 8. Para analisar estas informações, são utilizadas as ferramentas *Bibliometrix* e *VOSViewer*, os quais demonstram por meio de diagramas de redes temáticas as tendências da regressão linear aplicadas no segmento varejista.

O volume da produção científica sobre regressão linear relacionada a previsão de vendas ao longo dos anos é apresentado na Figura 12, na qual consolida as publicações das bases Scopus e Web of Science. Observa-se um crescimento no número de publicações, mais notadamente a partir de 2014, seguindo de forma crescente e ganhando novo impulso a partir de 2018, ano que antecedeu a crise sanitária mundial do COVID-19, momento em que ocorre alta demanda no mercado varejista, especificamente no *e-commerce*. Embora o ano de 2022 os dados sejam parciais até o mês de Agosto, observa-se que o número de publicações está diminuindo.

Figura 12 – Evolução das publicações sobre utilização de LR na previsão de vendas



Fonte: Scopus (2022), Web of Science (2022)

Na Tabela 9 são apresentados os crescimentos percentuais das publicações sobre regressão linear nas bases Scopus e Web of Sciense, no período de 10 anos, 5 anos e 1 ano.

Tabela 9 – Crescimento das publicações sobre utilização de LR na previsão de vendas

Períodos / Bases	10 anos 2012 a 2022	5 anos 2017 a 2022	1 ano 2021 a 2022
Scopus	353 %	121 %	-3 %
Web of Science	321 %	101 %	-82 %

Fonte: Scopus (2022), Web of Science (2022)

Foram identificados também os países com maior número de publicações na base Web of Science. A Tabela 10 mostra os dez países que mais publicaram sobre o tema, em que se observa a predominância dos Estados Unidos, país que sedia as maiores empresas varejistas do mundo: Walmart e Amazon, assim como a China, país que sedia as empresas Alibaba Group e JD.com, ambas empresas com destaque internacional no *e-commerce*. Na terceira posição está o Brasil, no qual sedia as empresas Magazine Luiza e Via Varejo.

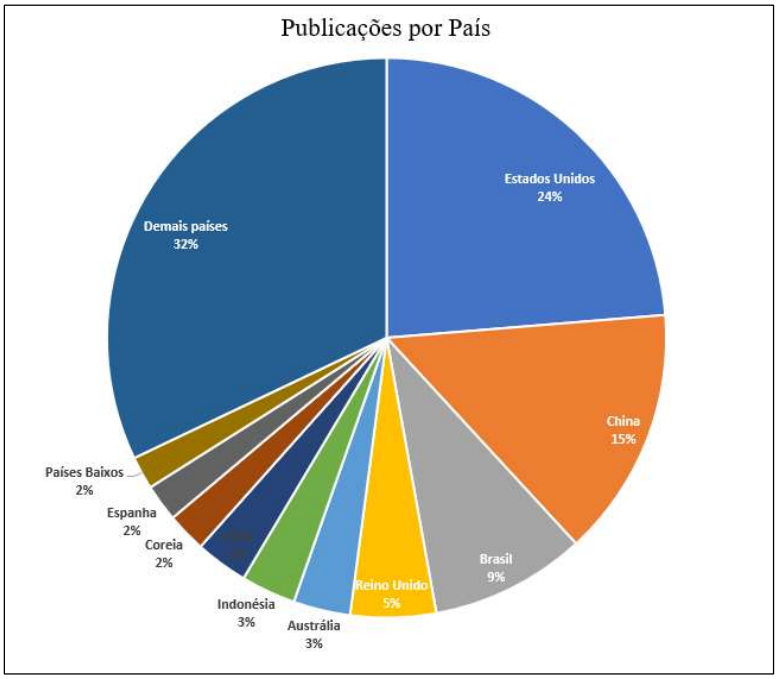
Tabela 10 - Dez países que mais publicaram sobre utilização de LR na previsão de vendas

Países	Publicações
Estados Unidos	187
China	114
Brasil	71
Reino Unido	39
Austrália	26
Indonésia	25
Índia	24
Coreia	18
Espanha	17
Países Baixos	15
Demais países	253

Fonte: Web of Science (2022)

A Figura 13 mostra a participação percentual dos dez países com maior número de publicações, conforme a base Web of Science. Os Estados Unidos respondem por 24% das publicações, enquanto a China púbica 15% da produção cientifica e o Brasil aparece na terceira posição com 9% das publicações.

Figura 13 – Dez países que mais publicam sobre utilização de LR aplicados na previsão de vendas



Fonte: Web of Science (2022)

A Tabela 11 apresenta os vinte artigos mais citados dentre os artigos resultantes da busca sobre a base Scopus e Web of Science, sobre abordagens de LR aplicadas ao varejo.

Tabela 11 – Artigos sobre abordagens de LR aplicados na previsão de vendas mais citados

Título	Autores	Fonte	Ano	TC
Exploring the use of deep neural networks for sales forecasting in fashion retail	Loureiro A., *et al.*	Decision Support Systems	2018	67
A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting	Arunraj N., Ahrens D.	International Journal of Production Economics	2015	56
Sales forecasting for computer wholesalers a comparison of multivariate adaptive regression splines and artificial neural networks	Lu C., Lee T., Lian C.	Decision Support Systems	2012	56
Supply chain collaboration for improved forecast accuracy of promotional sales	Ramanathan U.	International Journal of Operations & Production Management	2012	43
Tactical sales forecasting using a very large set of macroeconomic indicators	Sagaert Y., Aghezzaf E., Kourentzes N., Desmet B.	European Journal of Operational Research	2018	40
A two-stage dynamic sales forecasting model for the fashion retail	Ni Y., Fan F.	Expert Systems with Applications	2011	38
A big data approach to black friday sales	Awan M., Rahim M., Nobanee A., Khalaf O., Ishfaq U.	Intelligent Automation and Soft Computing	2021	36
Improving sustainable office building operation by using historical data and linear models to predict energy usage	Safa M., Allen J., Shahi C.	Sustainable Cities and Society	2017	24
Prediction analysis for business-to-business B2B sales of telecommunication services using machine learning techniques	Wisesa O., Adriansyah A., Khalaf O	Majlesi Journal of Electrical Engineering	2020	23
Social media and stock market prediction a big data approach	Awan M., Rahim M., Nobanee A., Yasin A., Zain A.	CMC-Computers Materials & Continua	2021	20

Título	Autores	Fonte	Ano	TC
Algorithms for generalized clusterwise linear regression	Park Y., Jiang Y., Klabjan D., Williams L.	Informs Journal on Computing	2017	17
The impact of store flyers on store performance a format and customer related approach	Luceri B., Latusi S., Vergura D., Lugli G.	International Journal of Retail and Distribution Management	2014	12
Sales forecasting of retail stores using machine learning techniques	Krishna A., Akhilesh V., Aich A., Hegde C.	Proceedings 2018 3rd international conference on computational systems and information technology for sustainable solutions, csitss 2018	2018	10
Machine learning model for sales forecasting by using XGboost	Dairu X., Shilong Z.	2021 ieee international conference on consumer electronics and computer engineering, iccece 2021	2021	5
Predictive analysis for big mart sales using machine learning algorithms	Ranjitha P., Spandana M.	Proceedings - 5th international conference on intelligent computing and control systems, iciccs 2021	2021	3
Multiple linear regression with Kalman filter for predicting end prices of online auctions	Li X., Dong H., Han S.	Proceedings - IEEE 18th International Conference on Dependable	2020	2
Smart shopping prediction on smart shopping with linear regression method	Nastiti M., Abdurohman M.; Putrada A.	2019 7th International Conference on Information and Communication Technology, ICOICT	2019	2
Sales prediction using online sentiment with regression model	Punjabi S., Shetty V., Pranav S., Yadav A.	Proceedings of the International Conference on Intelligent Computing and Control Systems ICICCS	2020	1
Evaluating the short-term effect of cross market discounts in purchases using neural networks a case in retail sector	Migueis V., Camanho A., Falcao E. C. J.	Expert Systems	2019	1
Predicting sneaker resale prices using machine learning	Raditya D., Erlin N., Amanda N.	5th International Conference on Computer Science and Computational Intelligence	2021	1

Fonte: Scopus (2022), Web of Science (2022)

Os autores mais citados dentre os artigos selecionados são apresentados na Tabela 12, contendo também o número de publicações (NP), a contagem de citações (TC), a razão do número de citações pelo número de publicações, os índices (h, g, m) e o ano da primeira publicação (PY_Start), conforme informações extraídas por meio do pacote Bibliometrix nas bases Scopus e Web of Science.

Tabela 12 – Autores sobre abordagens de LR aplicados na previsão de vendas mais citados

Autor	Instituição Afiliação	País	NP	TC	TC/NP	h_index	g_index	m_index	PY_Start
Sales, J.	University of South Bohemia České Budějovice	República Tcheca	18	354	19,67	10	18	0.4545455	2001
Liu, Y.	China Agricultural University	China	15	162	10,80	7	12	0.5384615	2010
Wang, J.	Xuchang University	China	14	225	16,07	6	14	0.5000000	2011
Wang, Y.	Fuzhou University	China	13	365	28,08	6	13	0.3750000	2007
Lee, S	Sejong University	Coréia do Sul	13	191	14,69	6	13	0.2608696	2000
Li, Y.	The University of Kitakyushu Kitagata Campus	Japão	12	88	7,33	5	9	10.000.000	2018
Park, S.	Kumoh National Institute of Technology	Coréia do Sul	8	80	10,00	5	8	0.2500000	2003
Chen, Y.	University Hong Kong	China	8	404	50,50	6	8	0.4285714	2009

Autor	Instituição Afiliação	País	NP	TC	TC/NP	h_index	g_index	m_index	PY_Start
Singh, S.	International Sales and Marketing, Emmbi Industries Limited	Índia	7	100	14,29	5	7	0.4545455	2012
Wang, Z.	University of Electronic Science and Technology of China	China	7	116	16,57	4	7	0.5714286	2016
Kim, J.	Hong Kong University of Science and Technology	China	7	218	31,14	6	7	0.4615385	2010
Sales, C.	Universidade de São Paulo	Brasil	6	45	7,50	5	6	0.7142857	2016
Li, X.	Harbin Engineering University	China	6	18	3,00	3	4	0.2307692	2010
Zhang, J.	Beijing Jiaotong University	China	5	65	13,00	4	5	0.3333333	2011
Chen, X.	Shanghai University	China	5	44	8,80	3	5	0.5000000	2017
Bunea, O.	The Bucharest University of Economic Studies	Romênia	5	10	2,00	2	2	0.3333333	2017
Yang, Y.	University of Memphis	Estados Unidos	5	103	20,60	4	5	0.3333333	2011
Li, J.	Renmin University of China	China	5	246	49,20	3	5	0.1578947	2004
Zhang, X.	Academy of Mathematics and Systems Science	China	4	18	4,50	2	4	0.4000000	2018
Zhang, Y.	East China University of Science and Technology	China	3	184	61,33	2	3	0.4000000	2018

Fonte: Scopus (2022), Web of Science (2022)

1.4.3 Revisão Descritiva da Literatura

Para o estudo de regressão linear aplicada na previsão de vendas no varejo, foram analisados os títulos, palavras-chave e resumo dos 50 artigos mais citados resultantes da busca da análise bibliométrica sobre as quatro bases: Scopus, Web of Sciense, ACM e IEEE Xplore. O critério de ordenação de artigos considerou a soma de citações nestas bases e o número de bases em que foram listados, buscando-se ponderar a seleção não apenas pelo número de citações, mas também pela presença nas bases consultadas.

A seleção dos documentos foi realizada mediante leitura dos resumos, obedecendo aos critérios do Quadro 4.

Quadro 4 – Critérios de seleção sobre abordagens de LR aplicadas na previsão de vendas

Tipo de Critério	Critério
Inclusão	Conteúdo se concentra em abordagens de regressão linear aplicadas na previsão de vendas
	Conteúdo apresenta aspectos relevantes sobre regressão linear aplicada na previsão de vendas
Exclusão	Conteúdo fora da área de interesse ou marginalmente relacionado ao tema de busca
	Documento não disponível para consulta online
	Documento não disponível na língua portuguesa ou inglesa
	Documento duplicado

Fonte: Autor

Os artigos resultantes foram analisados quanto a características como: Previsão estatística linear, Abordagens híbridas, Previsões probabilísticas e exploração dos resíduos.

Previsão estatística linear

A previsão de séries temporais tem sido bem estudada na literatura. Enquanto os estudos anteriores se concentravam em métodos de previsão estatística linear, como suavização exponencial e ARIMA, modelos altamente não lineares, incluindo redes neurais profundas, demonstraram ter um desempenho melhor para várias tarefas de previsão (ALON, 2001; CHU, 2003). Além disso, árvores de decisão e seus conjuntos, como florestas aleatórias e árvores impulsionadas por gradiente, foram frequentemente usados para previsão de séries temporais devido ao seu poder preditivo e estrutura interpretável (GALICI, 2019; TAIEB, 2014). O alto desempenho de conjuntos baseados em árvores e a força dos modelos simples foram destacados na competição de previsão M5 *forecasting challenge* disponível no site Kaggle, onde os métodos baseados em *Light Gradient Boosting Machines* estavam consistentemente entre os modelos de melhor desempenho (MAKRIDAKIS *et al.*, 2020).

Abordagens híbridas

Estudos recentes focaram particularmente em abordagens de aprendizagem profunda e meta-aprendizagem, relatando melhorias substanciais de desempenho para a tarefa de previsão de séries temporais sobre os modelos lineares, bem como abordagens de aprendizagem supervisionada padrão (MA, 2020; RANGAPURAM, 2018; SALINAS, 2020). Embora esses modelos mostrem uma promessa significativa na geração de resultados altamente precisos, especialmente quando os dados são abundantes, eles são normalmente considerados como modelos de caixa preta, que não são considerados interpretáveis/explicáveis, conforme relatado por Parmezan *et al.* (2019), o qual fornece uma visão detalhada dos modelos estatísticos e de ML para previsão de séries temporais.

Várias abordagens híbridas têm sido consideradas para incorporar relações não lineares entre variáveis de entrada e saída. Zhang (2003) assumiu que cada série temporal pode ser representada como uma combinação de componentes lineares e não lineares, e desenvolveu um modelo híbrido ARIMA e rede neural artificial (RNA) para previsão. As previsões no modelo foram obtidas como uma combinação da previsão do ARIMA para o componente linear e a previsão da RNA para o componente não linear. Khashei e Bijari (2012), investigaram o desempenho de modelos híbridos de previsão comparando o modelo híbrido generalizado ARIMA/ANN, o modelo híbrido ARIMA/ANN de Zhang (2003) e os modelos ANN (p, d, q). Sua análise com três conjuntos de dados diferentes mostrou que o modelo híbrido generalizado ARIMA/ANN, que visa encontrar relações lineares usando ARIMA no primeiro estágio e relações não lineares usando RNA no segundo estágio, teve o melhor desempenho entre as três abordagens. Aladag *et al.* (2009) substituíram a rede *neural feed forward* no modelo de Zhang (2003) por uma rede neural recorrente (RNN), que levou a melhorias na precisão da previsão.

Taskaya-Temizel e Casey (2005), realizaram análises comparativas em ARIMA e ANN usando nove conjuntos de dados diferentes, mostrando que os componentes dos modelos híbridos superaram seus equivalentes em cinco das nove instâncias. Concluíram que a seleção cuidadosa dos modelos a serem combinados é importante para o desempenho dos modelos híbridos. Arunraj e Ahrens (2015), propuseram um modelo híbrido com ARIMA sazonal (SARIMA) e regressão quantílica onde o último foi usado para prever os quantis em vez de pontos de dados individuais. Vários outros estudos consideraram modelos híbridos construídos a partir de variantes de modelos ARIMA, como SARIMA e SARIMAX (COOLS, 2009; CORNELSEN, 2012).

O sucesso de abordagens híbridas e o poder de modelos simples na captura de características de séries temporais foram enfatizados como os principais resultados da competição M4 (MAKRIDAKIS; SMYL, 2020), o vencedor da competição M4, propôs um método de previsão híbrido que combina suavização exponencial com redes neurais LSTM onde a suavização exponencial e LSTM respectivamente visam capturar padrões simples (por exemplo, sazonalidade) e tendências complexas não lineares.

Previsões probabilísticas

Embora a abordagem comum para previsão seja a previsão do valor esperado de um valor alvo, entender a incerteza nas previsões de um modelo pode ser importante em diferentes áreas, como macroeconomia ou no planejamento de vendas. Assim, muitos estudos sobre previsão focam em incertezas de modelagem que levam a previsões probabilísticas. Uma abordagem comum é usar regressão quantílica (LIU *et al.*, 2015), enquanto alguns outros estudos consideram um conjunto de modelos aprendidos para gerar previsões probabilísticas (AHMED MOHAMMED *et al.*, 2016). Estudos recentes empregaram redes neurais profundas para gerar parâmetros de média e variância das distribuições preditivas. Especificamente, Salinas *et al.* (2020) propuseram o modelo DeepAR, que é um modelo global baseado em rede recorrente autorregressivo que considera observações de diferentes séries temporais de treinamento para construir um único modelo de previsão probabilística. Rangapuram *et al.* (2018) combinaram modelos de espaço de estados com aprendizado profundo parametrizando um modelo de espaço de estados linear usando uma rede neural recorrente (RNN). Wang *et al.* (2019) integraram uma rede neural profunda global com modelos gráficos probabilísticos locais onde a estrutura global extrai padrões não lineares complexos e a estrutura local captura efeitos aleatórios individuais. Wen *et al.* (2021) sugeriram que assumir uma distribuição de erro para fazer previsões probabilísticas pode fornecer previsões imprecisas para algumas aplicações, e propuseram combinar RNN e rede neural convolucional (CNN) com regressão quantílica para previsão probabilística. Oreshkin *et al.* (2019) desenvolveram uma arquitetura neural profunda para previsão de séries temporais univariadas com base

em camadas profundas totalmente conectadas e links residuais. Chen *et al.* (2020), propuseram uma estrutura de previsão probabilística baseada em CNN com o objetivo de estimar a densidade de probabilidade dada várias séries temporais relacionadas. Combinando diferentes métodos de previsão probabilística, Alexandrov *et al.* (2020) forneceram uma extensa biblioteca Python de modelos probabilísticos de séries temporais.

Exploração dos resíduos

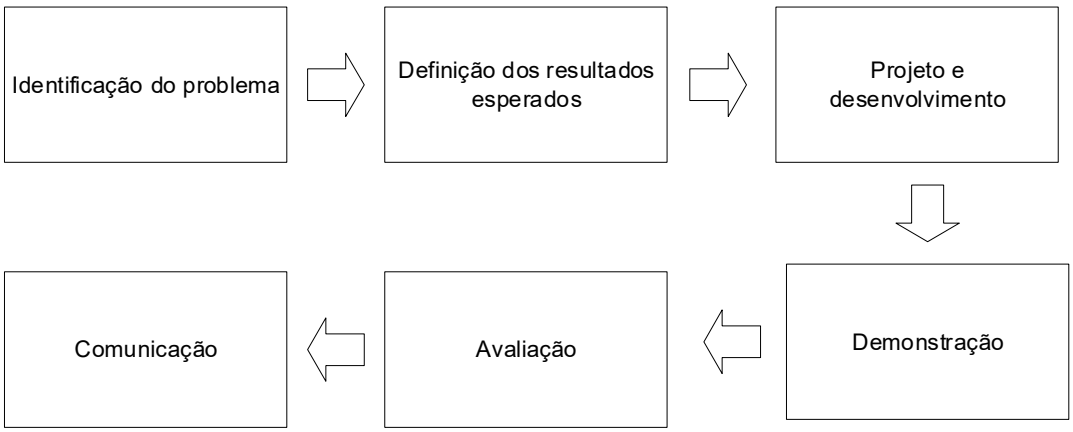
Poucos estudos na literatura se concentraram na construção de um modelo de previsão por meio da exploração de resíduos. Aburto e Weber (2007) consideraram um ARIMA e um modelo de rede neural combinados onde o modelo ARIMA foi usado para modelar a série temporal original e a rede neural foi usada para prever possíveis erros de previsão. A previsão resultante foi tomada como a soma dos valores previstos por esses dois modelos. Gur Ali e Pinar (2016) propuseram um modelo de compartilhamento de informações em dois estágios para o problema de previsão de vendas no varejo considerando vários períodos. Em seu modelo, o primeiro estágio estimou diversas variáveis como calendário, sazonalidade e promoções por meio de uma análise de regressão, e o segundo estágio extrapolou a série temporal residual. A previsão resultante foi obtida pela combinação das previsões da primeira etapa com os resíduos extrapolados da segunda etapa.

2 METODOLOGIA

Este trabalho utiliza o método *Design Science Research Methodology* (DSRM) em Sistemas de Informações para a condução da pesquisa aplicada, desta forma, este estudo seguiu as diretrizes da DSRM com base em Hevner *et al.* (2004) e Peffers *et al*. (2006, 2007). O método DSRM é um processo de resolução de problemas que permite aos pesquisadores adquirir conhecimento e compreensão de um domínio de problema e a sua solução por meio da criação e aplicação de artefatos de TI (HEVNER *et al*., 2004).

Peffers *et al.* (2006, 2007) apresentam um processo conceitual para DSRM em sistemas de informações baseado em frameworks em estudos de pesquisa em *design*. O *Design Science Research Process* (DSRP), ou seja, o processo de pesquisa em ciência do *design* consiste em seis atividades distintas, as quais podem ser executadas em ordem sequencial ou não, pois o método de pesquisa poderá ser utilizado de maneira diferente, sendo seu ponto de início modificado de acordo com os objetivos do pesquisador. A saída esperada de cada etapa é apresentada na Figura 14. O DSRP está intimamente relacionado com as diretrizes DSRM de Hevner *et al.* (2004), pois também ressalta a importância de uma declaração adequada do problema, a construção de uma solução viável e a avaliação dos resultados quanto à sua utilidade, bem como à sua comunicação.

Figura 14 - Método de pesquisa proposto por Peffers *et al*. (2007)



Fonte: Adaptado de Peffers *et al.* (2007)

Neste capítulo serão apresentadas apenas as etapas e respectivos objetivos do DSRM, sendo que os resultados da sua aplicação serão apresentados no próximo capítulo.

2.1 Identificação do Problema e Motivação

Esta etapa requer uma compreensão fundamental do problema de pesquisa e a capacidade para encontrar soluções potencialmente significativas para resolver tais problemas, de sorte que as soluções não sejam apenas uma resposta pontual a certo problema em determinado contexto, mas que o

conhecimento gerado em um contexto específico, quando generalizado, possa ser enquadrado em determinada classe de problemas para ser acessado por outros pesquisadores ou organizações que apresentem problemas similares. Isso vem ao encontro da afirmação de que a *design science* não se preocupa com a ação em si mesma, mas com o conhecimento que pode ser utilizado para projetar as soluções (VAN AKEN, 2004).

O pesquisador deve justificar a importância da pesquisa, considerando sua relevância e a importância do problema que está sendo investigado, além da aplicabilidade da solução que será proposta (PEFFERS *et al.* 2007).

Classe de Problemas

O conhecimento gerado a partir do DSRM deve ser passível de generalização e, consequentemente, pode ser enquadrado em uma determinada classe de problema. As classes permitem que os artefatos e, por consequência, suas soluções não sejam apenas uma resposta pontual a certo problema em determinado contexto, mas que o conhecimento gerado em um contexto específico, quando generalizado, possa ser enquadrado em determinada classe de problemas para ser acessado por outras organizações que apresentem problemas similares. Isso vem ao encontro da afirmação de que a *design science* não se preocupa com a ação em si mesma, mas com o conhecimento que pode ser utilizado para projetar as soluções (VAN AKEN, 2004).

2.2 Definição dos resultados esperados

Nesta etapa os resultados podem ser quantitativos, com indicadores que quantifiquem uma melhora de desempenho, ou qualitativos, que se concentrem em atributos que possam evidenciar uma melhor solução para o problema (PEFFERS *et al.* 2007).

2.3 Projeto e desenvolvimento

Esta etapa envolve projetar e desenvolver os artefatos que auxiliarão na solução do problema. É fundamental que nesse momento sejam definidas as funcionalidades desejadas, sua arquitetura e seu desenvolvimento em si. Para isso, o pesquisador deverá fazer uso do conhecimento teórico existente, a fim de propor artefatos que suportem a solução do problema (PEFFERS *et al.* 2007).

O artefato pode ser um constructo, um modelo, um método ou uma instanciação. O artefato deve ser eficaz quanto ao uso dos meios disponíveis para alcançar seus propósitos, obedecendo às diretrizes no ambiente do problema (HEVNER *et al.* 2004). O Quadro 5 resume as descrições dos tipos de artefatos documentados na DSRM.

Quadro 5 - Tipos de Artefatos da DSRM

Tipo	Descrição
Constructos	Constructos ou conceitos formam o vocabulário de um domínio. Eles constituem uma conceituação utilizada para descrever os problemas dentro do domínio e para especificar as respectivas soluções. (MARCH; SMITH, 1995).
Modelos	De acordo com March e Smith (1995), os modelos podem ser entendidos como um conjunto de proposições ou declarações que expressam as relações entre os constructos. São considerados representações da realidade que apresentam tanto as variáveis de determinado sistema como suas relações, não obstante, embora um modelo possa ser impreciso sobre os detalhes da realidade, ele precisa ter condições de capturar a estrutura geral da realidade, buscando assegurar sua utilidade.
Métodos	Os métodos podem estar ligados aos modelos, e as etapas do método podem utilizar partes do modelo como uma entrada que o compõe. Os métodos favorecem sobremaneira tanto a construção quanto a representação das necessidades de melhoria de um determinado sistema. Além disso, favorecem a transformação dos sistemas em busca de sua melhoria. Os métodos são criações típicas das pesquisas fundamentadas em *design science* (MARCH; SMITH, 1995).
Instanciações	As instanciações são artefatos que operacionalizam outros artefatos (constructos, modelos e métodos), cuja qual, visa demonstrar a viabilidade e a eficácia dos artefatos construídos (MARCH; SMITH, 1995).

Fonte: Adaptado de (LACERDA *et al*., 2013; MARCH; SMITH, 1995)

Conforme o método DSRM, na etapa de projeto deve ser informado o desempenho esperado, que vai garantir uma solução satisfatória para o problema em estudo, adicionalmente o método DSRM propõe a utilização do conhecimento da teoria disponível e aplicável para a determinação da funcionalidade do artefato e sua arquitetura (PEFFERS *et al.,* 2007).

Na etapa de desenvolvimento podem ser utilizadas diferentes abordagens, como algoritmos computacionais, representações gráficas, protótipos, maquetes etc., ou seja, a construção do ambiente interno do artefato (SIMON, 1996).

O DSRM não foca exclusivamente no desenvolvimento de um produto, embora possa ser utilizado para este fim, mas tem um objetivo mais amplo: gerar conhecimento que seja aplicável e útil para a solução de problemas, melhoria de sistemas existentes e criação de novas soluções e/ou artefatos (VENABLE, 2006).

Ao fim desta etapa, é possível obtermos o artefato em seu estado funcional, ou somente a heurística de construção, que pode ser formalizada a partir do desenvolvimento do artefato, sendo que a heurística de construção, proveniente do desenvolvimento de artefatos, é uma das contribuições do *design science* para o avanço do conhecimento.

2.4 Demonstração

Esta etapa é desenvolvida por meio de experimentação, simulação. Trata-se da demonstração do artefato para solucionar o problema em questão (PEFFERS *et al.* 2007).

2.5 Avaliação

Esta etapa visa comparar os resultados obtidos com os requisitos definidos na segunda etapa do método. Caso o resultado encontrado não seja o esperado, poderá retornar à etapa de projeto e desenvolvimento a fim de desenvolver um novo artefato. A avaliação deve comparar os objetivos do artefato com os resultados observados na demonstração, podendo incluir qualquer evidência empírica apropriada ou prova lógica. Pode assumir várias formas, dependendo da natureza do local do problema e do artefato, como a comparação da funcionalidade do artefato com os objetivos da solução, utilizando medidas quantitativas de desempenho, como orçamento, produção, pesquisas de satisfação, feedback do cliente e simulações; como também medidas de desempenho do sistema como o tempo de resposta ou a disponibilidade (PEFFERS *et al.* 2007).

Design do artefato: Utiliza-se o método analítico para conduzir a análise estática e da arquitetura para avaliação da funcionalidade e consistência do artefato, considerando as variáveis endógenas e exógenas, assim como atributos dependentes e independentes, caracterizados como atributos qualitativos relevantes de um artefato de TI (HEVNER *et al.*, 2004).

Conforme Tremblay, Hevner e Berndt (2010), a pesquisa sustentada pela DSRM não pode estar preocupada somente com o desenvolvimento do artefato em si. Devem-se expor evidências de que o artefato, efetivamente, pode ser utilizado para resolver problemas reais (TREMBLAY; HEVNER; BERNDT, 2010).

2.6 Comunicação

A última etapa apresenta o problema que foi estudado e sua importância. Ademais, é nessa fase que deverá ser demonstrado o rigor com o qual a pesquisa foi conduzida, bem como o quão eficaz foi a solução encontrada para o problema (PEFFERS *et al.* 2007).

3 RESULTADOS E DISCUSSÃO

A discussão dos resultados será apresentada de acordo com as seis atividades do método DSRM, em conformidade com as proposições de Peffers *et al.* (2007).

3.1 Identificação do Problema e Motivação

A identificação do problema e justificativas já foram apresentadas no capítulo de introdução.

Nesta pesquisa a previsão de vendas esta enquadra na classe de problemas de apoio a tomada de decisão conforme classificado por Andrade *et al.* (2006) e Kepner (1980).

A previsão de vendas não é uma tarefa trivial, uma vez que os dados necessários para tal análise, costumam possuir grandes volumes, apresentam ruídos, excesso de categorias e diversos outros problemas, além da dificuldade de seleção do modelo mais adequado para o problema que será analisado.

Porém, tais análises podem fortalecer a comercialização de produtos específicos, permitir o melhor gerenciamento de estoques e otimizar o processo de negociação com fornecedores. Para alcançar os benefícios citados, é necessário a exploração de um conjunto de técnicas de pré-processamento dos dados coletados, além da modelagem e avaliação das ferramentas de predição.

Esta pesquisa apresenta uma abordagem baseada em modelos de ML, para inferir a precificação de vendas e analisar as relações de causalidade entre as variáveis que influenciam as vendas futuras.

Nesta pesquisa, os dados foram obtidos a partir de uma empresa que atua no mercado varejista brasileiro, sendo que a seleção da empresa para este estudo de campo não ocorreu de forma aleatória. Ao apresentar um método estruturado de previsão de vendas utilizando média aritmética simples, a empresa foi convidada a participar do estudo mediante o termo de autorização para coleta de dados, cujo qual está arquivado no Centro Paula Souza e faz parte integrante desta pesquisa. Trata-se de uma empresa multinacional japonesa atuante no mercado varejista brasileiro desde 1972, na qual concordou em participar da pesquisa desde que sua identidade não fosse revelada assim como informações sigilosas ou dados estratégicos.

Para garantir que a amostra dos produtos selecionados represente efetivamente a população objeto deste estudo, foram selecionados produtos de três diferentes linhas de negócio as quais representam 90%

do faturamento da empresa. Dentro de cada linha de negócio, foram selecionados produtos com histórico de movimentação acima de cinco anos.

Os produtos selecionados estão classificados em três grupos conforme mostrado na Tabela 13.

Tabela 13- Número de transações no estoque por produto

Produto	Linha de Negócio	Transações	Período
13M1S1	Digitalizador biométrico	42.887	2006 a 06/2018
0-0001	Consumível para digitalizador de imagem	2.998	2008 a 05/2022
0-B051	Digitalizador de imagem	1.973	2014 a 05/2022
8-K011	Consumível para digitalizador de imagem	2.642	2004 a 05/2022
1-B301	Digitalizador de imagem	568	2016 a 05/2022

Fonte: Resultados da Pesquisa

Digitalizador biométrico

Sistema utilizado para autenticação utilizando a mais recente tecnologia biométrica de segurança. Está presente em caixas eletrônicos - ATM, assim como no controle de acesso a estabelecimentos por meio de catracas eletrônicas conectadas a um sistema para liberação dos usuários.

Digitalizador de imagem

Digitalizador, em inglês *scanner* é um periférico de entrada responsável por digitalizar imagens, fotos e textos impressos para o computador, ou seja, constitui na passagem dos dados físicos para o meio digital, eliminando pilhas de papéis ao passar estes documentos para o formato digital.

Consumível para digitalizador de imagem

Pequenas peças para realizar manutenção em aparelhos de *scanner*, com o objetivo de manutenção preventiva indispensável que ajuda a melhorar o desempenho, reduzir os custos de serviço e manter o scanner na sua capacidade máxima de desempenho.

3.2 Definição dos resultados esperados

Propõe-se implementar uma abordagem de ML aplicada na previsão de vendas no segmento de varejo, na forma de um artefato de software que utilize modelos estatísticos adequados para identificar o algoritmo com melhor desempenho ao aplicar as métricas de coeficiente de determinação *R-squared* (R^2) e de erro RMSE.

3.3 Projeto e desenvolvimento

Embora a literatura sobre predição de vendas no segmento de varejo aplicando ML sejam escassas, na revisão da literatura foram identificados trabalhos anteriores que enfatizam abordagens para examinar os determinantes de preços com base em ML para outros campos de aplicação, como a indústria de energia (Deb *et al.*, 2017), finanças (Krollner *et al*., 2010) e clima (Baboo *et al.*, 2010). Todos estes trabalhos possuem um ponto em comum, são publicações que focam na aplicação de algoritmos de aprendizado supervisionado para identificar determinantes de preços.

Os artigos sobre abordagens de ML aplicado na previsão de vendas, obtidos mediante Revisão Descritiva da Literatura (PARÉ et al., 2014), foram selecionados por critérios como contagem de citações, fatores de impacto dos periódicos, índice-h dos autores e pertinência ao tema, limitados até o dia 15 de agosto de 2022. Foram identificados que os métodos de previsão quantitativa são geralmente divididos em duas categorias: modelos gerais de séries temporais e modelos baseados em regressão. Modelos gerais de séries temporais, como suavização exponencial e média móvel integrada autorregressiva ARIMA, são normalmente derivados das informações estatísticas nos dados históricos. Por outro lado, os modelos de regressão dependem da construção de uma relação entre variáveis independentes e variáveis dependentes.

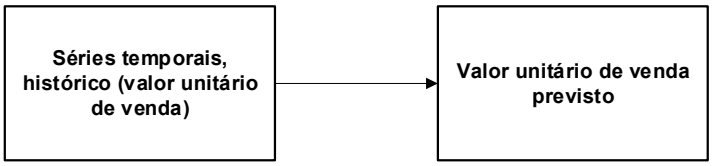
É observado que a regressão linear melhora substancialmente o desempenho do modelo base por meio de recursos extraídos e fornece um desempenho comparável a outras abordagens bem estabelecidas. A interpretação das previsões do modelo e a alta precisão preditiva da regressão linear a torna um método mais eficaz do que métodos tradicionais como ARIMA que aplicam médias móveis (RANGAPURAM, 2018; SALINAS, 2020, ABURTO 2007; GUIDOTTI, 2018; HUANG *et al.*, 2017; SAYLI *et al.*, 2016). Informações mais detalhadas constam da Revisão Descritiva da Literatura, no capítulo de Fundamentação Teórica, com uma análise dos modelos de regressão linear.

Como resultado desta pesquisa, foram gerados os seguintes artefatos: constructos, modelo, métodos e instanciações.

3.3.1 Constructos

Neste contexto a Figura 15 representa o diagrama conceitual desta pesquisa, cujo objetivo é obter o modelo com melhor desempenho ao aplicar as métricas de coeficiente de determinação *R-squared* (R^2) e de erro RMSE para precificação de vendas futuras a partir do desenvolvimento de um protótipo de solução computacional, comparando seis algoritmos de ML com base em regressão linear.

Figura 15 - Diagrama conceitual - Constructo



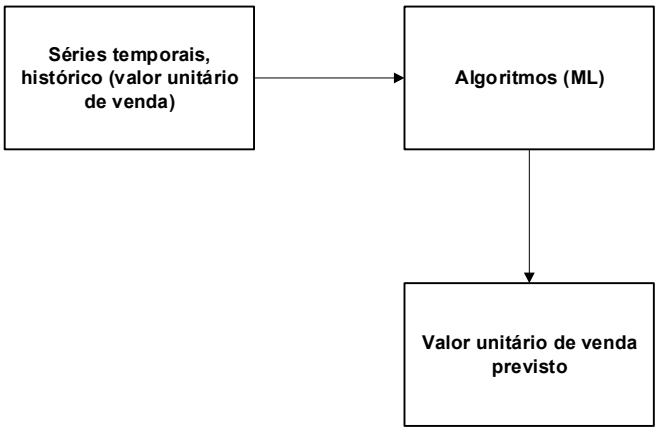
Fonte: Autor

3.3.2 Modelo

Os modelos podem ser entendidos, segundo March e Smith (1995), como um conjunto de proposições ou declarações que expressam a relação entre os constructos, desta forma, os modelos de regressão dependem da construção de uma relação entre variáveis independentes, sejam elas internas (endógenas) ao contexto do estudo como o histórico de vendas do produto ou externas (exógenas) como índice de inflação ou qualquer outro indicador que esteja fora do contexto da organização, desta forma o conjunto destas variáveis são considerados como séries temporais pois refletem o comportamento dos dados em um espaço temporal. Por meio da aplicação do ML, os modelos de regressão são treinados para obtenção do da variável dependente ou variável alvo, representada nesta pesquisa pela previsão do valor unitário de venda.

A Figura 16 representa a relação entre as variáveis no contexto desta pesquisa.

Figura 16 - Diagrama conceitual - Modelos



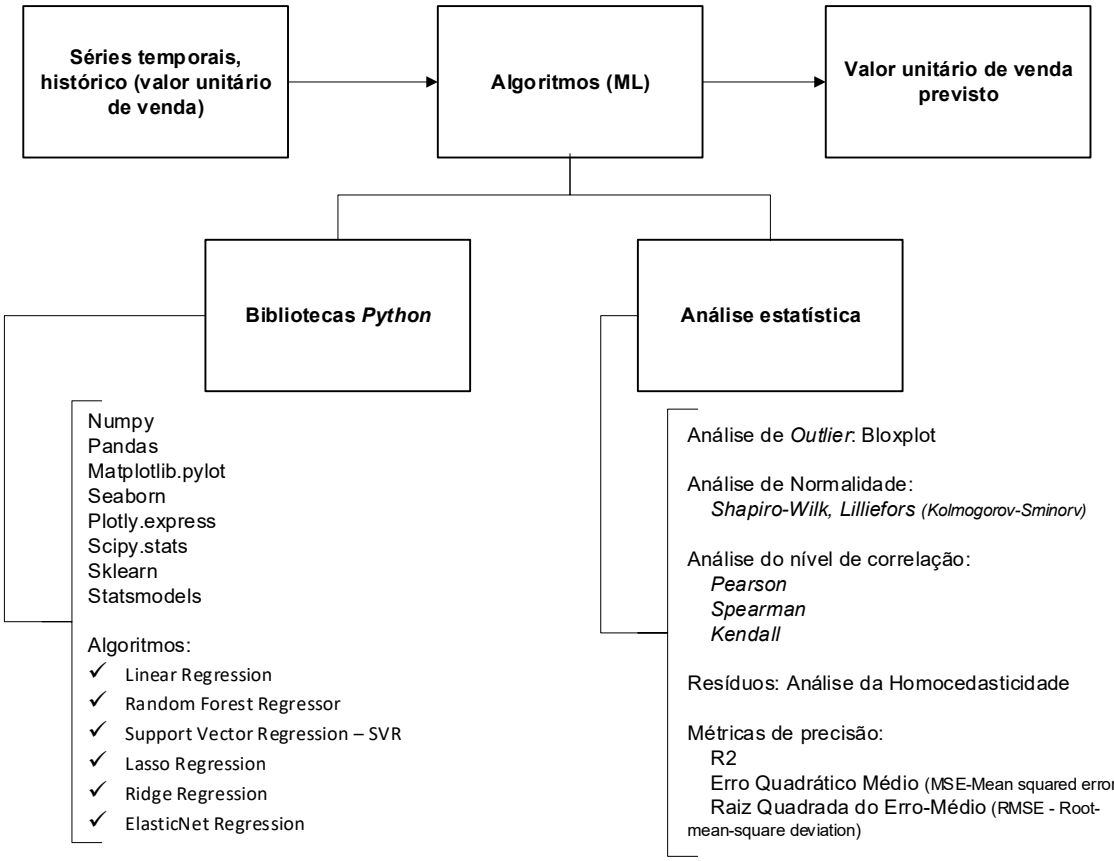
Fonte: Autor

3.3.3 Métodos

Os métodos estão ligados aos modelos quando utilizam partes do modelo como uma entrada, descrevendo os passos necessários para desempenhar determinada tarefa e atingir o resultado, com isso, para alcançar os resultados esperados, esta pesquisa utiliza seis tipos de regressão linear por meio de algumas bibliotecas disponíveis na linguagem *Python*, assim como elementos estatísticos para tratamento dos dados e métricas de precisão para comparação dos modelos.

A Figura 17, demonstra um resumo dos elementos envolvidos no método desta pesquisa e detalhados no delineamento de pesquisa na seção 3.5.1.

Figura 17 – Diagrama conceitual - Métodos



Fonte: Autor

3.3.4 Instanciações (Artefato de Software)

As instanciações são artefatos que operacionalizam os constructos, modelos e métodos vistos anteriormente, demonstrando a viabilidade e a eficácia destes artefatos em um determinado ambiente. O artefato de software resultante desta pesquisa, tem como objetivo comparar seis diferentes tipos de

algoritmos com base em regressão linear, evidenciando o algoritmo que apresente o melhor desempenho ao aplicar as métricas de coeficiente de determinação *R-squared* (R^2) e de erro RMSE. Para atingir este objetivo a solução computacional realiza a carga dos dados em seu estado bruto com os respectivos parâmetros de filtro. Na fase de pré-processamento dos dados os valores nulos são removidos e o diagrama de caixa *boxplot* é aplicado para identificar e remover valores *outliers*, na sequência as escalas das variáveis são padronizadas para tornar os dados comparáveis como pré-requisito para o treinamento dos modelos, na próxima etapa é aplicado a redução da dimensionalidade para escolha das variáveis mais significativas para o modelo, na etapa seguinte as variáveis independentes correlacionadas são removidas para evitar multicolinearidade, na sequência os dados são particionados utilizando a técnica *k-fold* para aplicação da validação cruzada para diminuir a possibilidade de *Overfitting* e *Underfitting*, após o treinamento, os resíduos de cada modelo são avaliados quanto a sua homocedasticidade e caso apresentem indícios de heterocedasticidade, são aplicados os testes paramétrios: *Breusch-Pagan*, *Bartlett* e Teste *t*. Finalmente é aplicado a validação cruzada externa, no qual o modelo com a melhor desempenho é obtido por meio do coeficiente de determinação R^2 mais alto e o menor índice de erro RMSE.

O projeto consiste na instanciação de um artefato viável e de propósito definido, entendido como a implementação de um modelo que demonstra a sua viabilidade, possibilitando a avaliação da sua adequação aos propósitos pretendidos e a compreensão do seu funcionamento (HEVNER *et al*., 2004). Nesta pesquisa o artefato se apresenta na forma de um programa de computador composto de modelo lógico e código fonte que implementa um conjunto de seis algoritmos baseados em regressão linear aplicados na previsão de vendas no segmento de varejo. O código fonte, está publicamente disponível na plataforma GitHub no endereço: **https://github.com/emersonemtech/bashboard-previsao-vendas**, no arquivo "jupyter-001.ipynb".

O programa utiliza a linguagem de código aberto Python v. 3.9.2, o ambiente integrado de desenvolvimento Jupyter Notebook 6.4.8 e as bibliotecas: Numpy v. 1.22.1 para processamento matemático, Pandas v. 1.3.5 para operações de análise de dados, MatPlotLib v. 3.5.1 para apresentação gráfica, cx_Oracle v. 8.3.0 para conexão com banco de dados Oracle, Seaborn v0.11.2 para apresentação gráfica, Scikit-learn v. 1.0.2 para pré-processamento, padronização dos dados e aplicação dos modelos de IA aos seis algoritmos com base em regressão linear selecionados, Scipy v 1.7.3 para apresentações de gráficos QQ-Plot, Statsmodels v. 0.13.2 para teste de normalidade, Openpyxl v. 3.0.10 para exportar *datasets* para planilha eletrônica e Researchpy v. 0.3.2 para aplicar teste paramétrico.

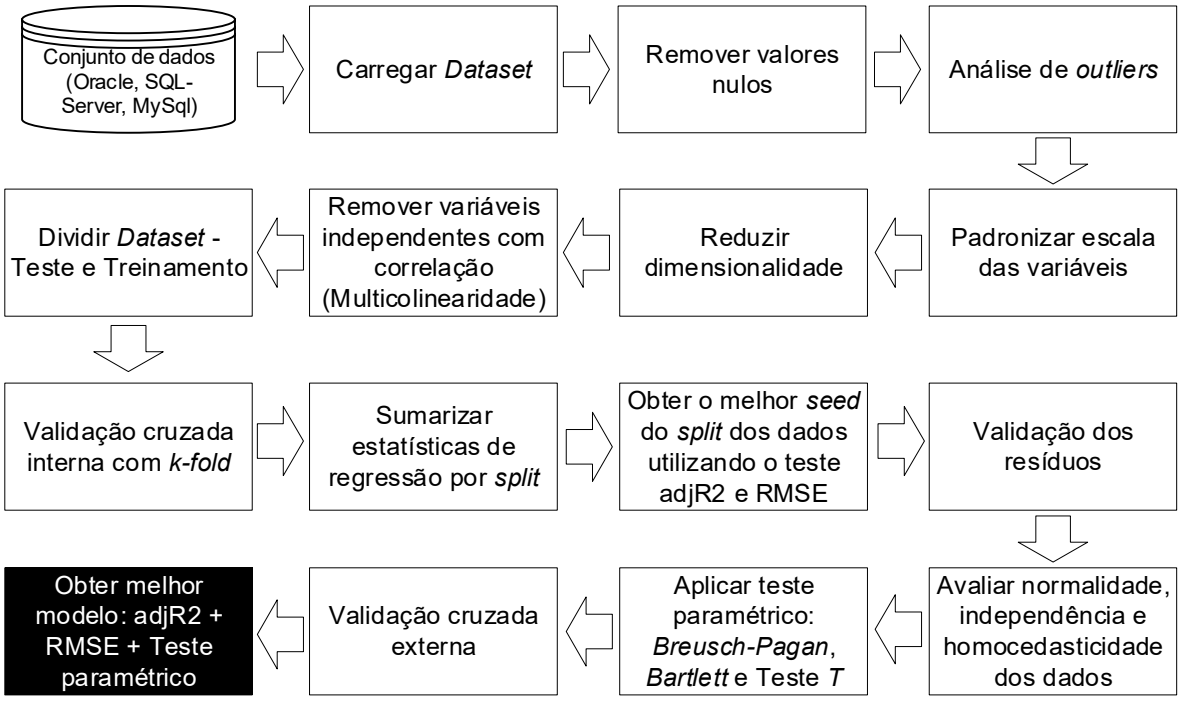
A configuração do ambiente de desenvolvimento segue as orientações dos sites Python.org e Anaconda.com.

As parametrizações e o uso de bibliotecas de funções da abordagem LR utilizam configurações e recomendações de Zou *et al*. (2005), Tibshirani (2006), Freedman (2009), Breiman (2001), Hilt e Seegrist (1977) e Witten *et al.* (2011) e disponibilizadas em regime código aberto (*open source*) no repositório GitHub, respeitando os direitos de propriedade intelectual.

O modelo foi executado em notebook Fujitsu, modelo LIFEBOOK U748 com 16Gb de memória RAM, processador Intel64 Family 6 Model 142 Stepping 9 GenuineIntel ~2701 Mhz e sistema operacional Microsoft Windows 10 Professional.

O método proposto nesta pesquisa é definido em um fluxograma integrado e desenvolvido com o objetivo de criar e comparar múltiplos modelos de regressão linear, inicialmente introduzido por (Tsiliki *et al*., 2015a). Tal modelo foi adaptado nesta pesquisa conforme Figura 18. Este diagrama é implementado e testado por meio de uma solução computacional desenvolvida em *Python*.

Figura 18 - Modelo lógico - diagrama de processos



Fonte: Autor

Com foco na última etapa do fluxograma, o melhor modelo de regressão linear dessa metodologia foi selecionado com base nos seguintes critérios: I) A performance de todos os modelos foi avaliada (inclusive com o conjunto de dados de teste) de acordo com o *R-squared* (R^2), estabelecendo um ranking. II) Em seguida, foi levado em consideração o resultado *R-squared* (R^2) de todos os modelos que estivessem com uma variação de (0,05), com isso um novo ranking foi reordenado, no qual foram selecionados os modelos que apresentaram o menor (RMSE) obtido nos testes.

Assim, é obtido o melhor modelo de todas as execuções, combinando o uso de duas medidas de desempenho com um critério específico.

Resultado comparativo dos modelos

Uma vez atendidos os pressupostos de normalidade, independência e heterocedasticidade, os conjuntos de dados são submetidos aos algoritmos de ML utilizando a *seed* que apresentou a melhor precisão no processo de *Cross-Validation* apresentadas na seção 3.5.1.3.

O *dataset* contendo as movimentações históricas dos produtos foi separado em 70% para treinamento e 30% para teste, além disso foi utilizado o *seed* que apresentou a melhor precisão no processo de validação cruzada interna. A Figura 19 apresenta o método *train_test_split* presente na biblioteca *sklearn.model_selection* utilizado para separação inicial dos dados.

Figura 19 – Declaração do *train_test_split*

```
In [342]: # Criandos dataframes para o eixo X e Y
          # no eixo X deve-se eliminar a variavel dependente VL_VENDA_TOT, VL_VENDA_UNIT, variavel do tipo string MES_EXT
          y = df_mes[["VL_VENDA_TOT"]]
          x = df_mes.drop(["MES_EXT", "VL_VENDA_TOT", "VL_VENDA_UNIT"], axis=1)
          print(x)


          # O dataset sera separado em 70% para treinamento e 30% para teste, para avaliar o modelo test_size=0.3
          x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=746)
```

Fonte: Resultados da Pesquisa

A Tabela 14, evidência os resultados obtidos por meio de uma matriz comparativa entre os algoritmos. Os resultados apresentados em cinza escuro destacam os modelos com o melhor desempenho. O processo para identificar os algoritmos com o melhor desempenho está detalhado no Apêndice C.

Tabela 14 – Matriz comparativa dos modelos

		13M1S1	0-0001	0-B051	8-K011	1-B301
Linear Regression	R^2	0.9859703	0.9655112	0.9884350	0.9698459	0.9961951
	RMSE	30.634	204	15.366	556	1.091
Random Forest Regressor	R^2	0.9676439	0.9558832	0.9700913	0.9626484	0.9699347
	RMSE	46.522	231	24.711	619	3.069
Support Vector Regression	R^2	**0.9913870**	**0.9716644**	0.9687610	0.9639821	0.9460529
	RMSE	**24.002**	**185**	25.254	608	4.111
Lasso Regression	R^2	0.9859438	0.9643137	0.9884473	0.9702801	0.9962689
	RMSE	30.663	208	15.358	552	1.081
Ridge Regression	R^2	0.9859974	0.9659043	0.9886798	0.9702801	**0.9980811**
	RMSE	30.604	203	15.202	552	**775**
ElasticNet Regression	R^2	0.9862212	0.9658657	**0.9896923**	**0.9717601**	0.9971229
	RMSE	30.359	203	**14.507**	**538**	949
Média Aritmética Simples	R^2	0.9403611	0.6257375	0.8977438	0.7912402	0.9976336
	RMSE	63.161	673	45.692	1.464	861
Distribuição Normal (Shapiro-Wilk)	*p-value* > 0.05	Não	Não	Sim	Não	Sim
Correlação		-	-	Pearson	-	Pearson
		Spearman	Spearman	-	Spearman	-
Homocedasticidade	Teste de *Breusch-Pagan*	Sim	Não	Não	Não	Não
	Teste de *Bartlett*	-	Sim	Sim	Sim	Não
	Teste *T*	-	-	-	-	Sim

Fonte: Resultados da Pesquisa

Fica evidenciado que o algoritmo SVR apresentou a melhor predição para os produtos 13M121 e 0-0001 com *R-squared* (R^2) de 0,9914 e 0,9717, além do RMSE de 24,002 e 185 respectivamente. Para os produtos 0-B051 e 8-K011 o melhor algoritmo foi *ElasticNet* com *R-squared* (R^2) de 0,9897 e 0,9718, além do RMSE de 14,507 e 538 respectivamente. O algoritmo *Ridge* apresentou a melhor predição para o produto 1-B301 com *R-squared* (R^2) de 0,9981 e RMSE de 775.

A análise de regressão gera uma equação para descrever a relação entre uma ou mais variáveis independentes e a variável resposta. Após ajustar o modelo de regressão e verificar o resultado por meio dos gráficos de resíduos, é necessário interpretar os valores-p e os coeficientes que aparecem na saída da análise da regressão linear.

O valor-p "P>|t|" para cada termo testa a hipótese nula de que o coeficiente é igual a zero (sem efeito). Um valor-p inferior a 0,05 indica que a hipótese nula pode ser rejeitada, ou seja, uma variável que tenha um valor-p baixo provavelmente será significativa ao modelo, porque as alterações no valor dela estão relacionadas às alterações na variável resposta.

Por outro lado, um valor-p superior a 0,05 sugere que as mudanças na variável não estão associadas às mudanças na variável resposta.

A Tabela 15, mostra os valores-p e coeficientes obtidos em cada algoritmo.

Tabela 15 – Valores-p e coeficientes

			13M1S1	0-0001	0-B051	8-K011	1-B301
	Algoritmo		**Support Vector Regression**	**Support Vector Regression**	**ElasticNet Regression**	**ElasticNet Regression**	**Ridge Regression**
Intercept	**Valor**		-3.297,08075388	-2.135,86022129	-198.533,97412236	-7.915,36327135	-5.933,50995303
	P>|t|		0,008	0,000	0,000	0,000	0,053
Variáveis Independentes (Coeficientes)	**IPCA**	**Valor**	0,37089298	4,56836718	1,49856996e+03	62,31798964	-8,01685912e+02
		P>|t|	**0,786**	0,003	0,005	0,007	**0,226**
	QUANTIDADE	**Valor**	-284,16947437	-3,36728982	3,83090439e+02	39,9499042	1,77854211e+03
		P>|t|	0,000	**0,507**	0,000	0,000	0,000
	CMV_TOTAL	**Valor**	2,96582083	2,80555821	1,64752460e+00	2,28711455	2,30295935e-01
		P>|t|	0,000	0,000	0,000	0,000	**0,122**
	PERC_MARGEM	**Valor**	54,53815384	34,74717362	3,89795307e+03	129,01527573	1,51315050e+02
		P>|t|	0,003	0,000	0,000	0,000	0,050

Fonte: Resultados da Pesquisa

Na Tabela 15, fica evidenciado que a variável PERC_MAGEM é estatisticamente significativa em todos os algoritmos, apresentado p-valor inferior a 0,05. A variável CMV_TOTAL apresentou significância para todos os modelos, com exceção do produto "1-B031" cujo valor de 0,122 foi superior a 0,05 indicando não ser estatisticamente significativa. A variável QUANTIDADE também apresentou significância para todos os modelos, com exceção do produto "0-0001" cujo valor apresentado foi de 0,507. A variável IPCA apresentou ser estatisticamente significativa para os produtos 0-0001, 0-B051 e 8-K011 e pouca significativa para os produtos 13M1S1 e 1-B301.

Relevância da solução, Utilidade e Ineditismo do Artefato: O artefato contribui para a pesquisa sobre ML em predição de vendas no segmento do varejo brasileiro, lacuna de pesquisa identificada em revisões da literatura, implementando uma abordagem no estado da arte, por meio de artefato de software viável aplicando algoritmos com base em regressão linear múltipla.

Eficácia: Nos gráficos de dispersão contendo o valor realizado e o valor previsto, observa-se a aderência dos valores da predição aos valores efetivos nos períodos. Em todos os produtos avaliados, o desempenho dos modelos preditivos foi superior à média aritmética simples.

Embora a inspeção visual possa parecer indicar boa capacidade de predição, deve-se levar em consideração uma investigação cuidadosa quanto à possibilidade de *Overfitting* ou *Underfitting*.

O desenho experimental proposto neste trabalho já foi utilizado anteriormente em outras pesquisas, provando ter bom desempenho em tarefas de predição ou classificação. Por meio de uma revisão narrativa da literatura o autor desta pesquisa compara a acurácia de diferentes algoritmos de ML na detecção de fraudes no segmento de crédito financeiro (Martins *et al*., 2022a). O estudo supracitado utilizou as quatro fases do desenho experimental normalizado, aplicando diferentes abordagens de seleção de características para redução de dimensionalidade. Nesta pesquisa, os resultados mostraram que para todos os conjuntos de dados gerados, a metodologia proposta apresentou resultados reprodutíveis, comparáveis e alcançados em igualdade de condições. Além disso, um trabalho recente realizado pelo autor do presente estudo empregou essa abordagem para comparar o desempenho entre métodos tradicionais e o ML na previsão de vendas no segmento de varejo (MARTINS *et al.*, 2022b).

Nesta pesquisa, a mesma metodologia para a formalização de desenhos experimentais utilizada por Tsiliki *et al.* (2015a) foi adaptada com validação cruzada e testes estatísticos e comprovada sobre um conjunto de dados no segmento varejista relacionados a tarefa de regressão. Os resultados mostraram um excelente desempenho dos modelos finais selecionados e demonstraram a importância de levar em consideração a variabilidade dos modelos para escolher o melhor. Essa seleção não deve se basear apenas no melhor resultado alcançado em uma única métrica. Nesta pesquisa é avaliado o desempenho de todos os modelos (com o conjunto de teste), utilizando-se o *R2-Squared* para estabelecer um ranking dos modelos. Na etapa seguinte, leva-se em consideração a normalidade, independência e homocedasticidade dos dados, selecionando o modelo com menor RMSE obtido no teste.

Os resultados também permitiram entender que selecionar o modelo apenas de acordo com o melhor *R2-Squared* ou a maior precisão e levar em consideração apenas uma execução não é estatisticamente válido. Um valor discrepante ou incomum no *dataset* pode ser um *outlier* e pode surgir devido a uma partição específica dos dados durante o aprendizado do modelo e deve ser descartado em favor de métodos mais estáveis. Aspectos como estabilidade nos resultados devem ser levados em consideração para fazer essa seleção.

Esta pesquisa propõe a combinação de duas profundas modificações necessárias na metodologia introduzida por Tsiliki *et al.* (2015a), ou seja, processo de validação cruzada externa na fase de aprendizado e análise estatística na fase de seleção do melhor modelo, além de uma simples consideração sobre a diversidade dos dados, a fim de lidar com dados de contagem durante o pré-processamento da fase de dados.

Os achados desta pesquisa são relevantes e pode-se supor que outros modelos de ML, em geral, devem se comportar de maneira semelhante aos apresentados nesta pesquisa, bem como os mesmos algoritmos com outros conjuntos de dados. De um modo geral, se o processo de treinamento ou os algoritmos são estocásticos de alguma forma, deve-se repetir várias execuções experimentais para encontrar os melhores resultados, e é crucial usar uma comparação estatística para avaliar se as diferenças na pontuação de desempenho são relevantes antes de eleger finalmente o melhor modelo. Caso contrário, as conclusões finais podem estar erradas e o modelo final deve ser descartado.

Por fim, ficou evidenciado que a metodologia utilizada nesta pesquisa é aplicável em outros campos, pois é bastante flexível para adicionar novas fases, podendo ser aplicada em outros conjuntos de dados de escopos diferentes. Ao mesmo tempo, conforme os resultados apresentados, pode-se afirmar que o mesmo comportamento é esperado com outros algoritmos de ML e espera-se que estudos estatísticos posteriores sejam relevantes nesse sentido.

O principal objetivo deste tipo de metodologia na modelagem preditiva é ajudar na exploração inicial de extensos bancos de dados, a fim de projetar métodos abrangentes de triagem no mercado varejista, reduzindo ao máximo os custos econômicos e apoiar na previsibilidade financeira da empresa e ao mesmo tempo, garantindo a qualidade e confiabilidade dos métodos de ML usados no processo.

3.4 Demonstração

Nesta pesquisa, os dados foram obtidos a partir de uma empresa que atua no mercado varejista brasileiro, sendo que a seleção da empresa para este estudo de campo não ocorreu de forma aleatória.

Ao apresentar um método estruturado de previsão de vendas utilizando média aritmética simples, a empresa foi convidada a participar do estudo mediante o termo de autorização para coleta de dados, cujo qual está arquivado no Centro Paula Souza e faz parte integrante desta pesquisa. Trata-se de uma empresa multinacional japonesa atuante no mercado varejista brasileiro desde 1972, na qual concordou em participar da pesquisa desde que sua identidade não fosse revelada assim como informações sigilosas ou dados estratégicos.

O protótipo permite que o usuário selecione o período e o produto que deseja analisar. Com os parâmetros definidos, a solução computacional exibirá graficamente qual o melhor modelo com base no resultado do coeficiente de determinação *R-squared* (R^2), juntamente com a menor variação do índice de erro RMSE, como mostra a Figura 20.

Figura 20 – Resultados do *OLS Regression Results*

OLS Regression Results

Dep. Variable:	VL_VENDA_TOT	R-squared:	0.984
Model:	OLS	Adj. R-squared:	0.983
Method:	Least Squares	F-statistic:	1232.
Date:	Sat, 12 Nov 2022	Prob (F-statistic):	2.68e-70
Time:	08:34:23	Log-Likelihood:	-928.29
No. Observations:	84	AIC:	1867.
Df Residuals:	79	BIC:	1879.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.904e+05	1.59e+04	-11.958	0.000	-2.22e+05	-1.59e+05
IPCA	3003.4686	4892.873	0.614	0.541	-6735.551	1.27e+04
QUANTIDADE	373.8518	98.020	3.814	0.000	178.748	568.956
CMV_TOTAL	1.6459	0.092	17.975	0.000	1.464	1.828
PERC_MARGEM	3730.4294	271.578	13.736	0.000	3189.867	4270.992

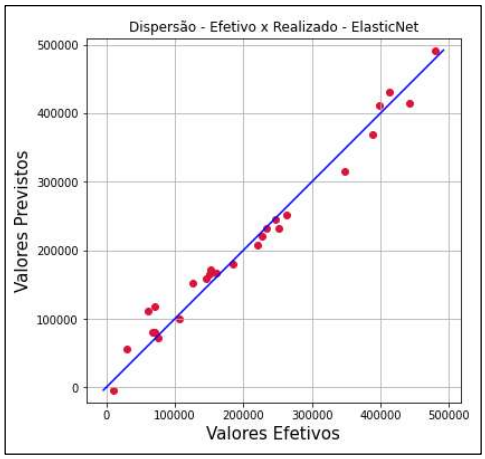
Fonte: Resultados da Pesquisa

Na Figura 20 são apresentados os resultados estatísticos por meio da aplicação da biblioteca *statsmodels.stats.api*, utilizando o método *summary()*, para o produto 0-B051. Dentre as informações é destacada a variável dependente "VL_VENDA_TOT", assim como o coeficiente de determinação *R-squared* (R^2) com 0.984, evidenciando que o modelo se ajustou em 98,4% aos dados. Além disso são apresentados o atributo "INTERCEPT" e os coeficientes "IPCA", "QUANTIDADE", "CMV_TOTAL" e "PERC_MARGEM".

Embora as variáveis independentes "IPCA" e "CMV_TOTAL" tenham apresentado sinais contrário ao esperado, decidiu-se por mantê-las no modelo preditivo considerando o histórico de ajuste imediato de preço da empresa, uma vez que se trata de produtos importados com histórico de vendas constante, ou seja, não cíclico ao longo do período analisado.

Após a separação dos dados em 70% para treinamento e 30% para teste conforme mostrado na Figura 19, o protótipo apresenta o gráfico de dispersão, destacando o modelo que apresentou o melhor desempenho, neste caso o algoritmo ElasticNet.

Figura 21 - Gráfico de dispersão



Fonte: Resultados da Pesquisa

A reta azul na Figura 21 representa a função linear múltipla definida pelo algoritmo ElasticNet por meio do Intercepto e dos coeficientes das variáveis independentes. Os pontos vermelhos representam os valores reais contidos nos dados de treinamento. O objetivo do modelo é obter uma função linear, cuja reta traçada fique mais próxima possível das observações reais.

Por meio da biblioteca *sklearn* o método *metrics* é utilizado para obter os parâmetros “Intercept_” e “coef_” conforme mostrado na Figura 22.

Figura 22 - Biblioteca sklearn com o método metriccs

```
In [213]: # ELASTIC NET REGRESSION ----------------------------------------------------------------------------------
          from sklearn import metrics

          #import warnings
          #warnings.filterwarnings('ignore')
          #warnings.filterwarnings('always')

          from sklearn.linear_model import ElasticNet

          modelo_el = ElasticNet()

          # Treinar
          modelo_el.fit(x_train, y_train)

          # Predicao
          previsao_el = modelo_el.predict(x_test)

          print(f'ElasticNet: {metrics.r2_score(y_test, previsao_el)}')

          print('********** Parametros funcao linear ********************************')
          print('intercept_ -> {} '.format(modelo_el.intercept_))
          print('coef_      -> {} '.format(modelo_el.coef_))

          ElasticNet: 0.9759422544722696
          ********** Parametros funcao linear ********************************
          intercept_ -> [-202957.00573622]
          coef_      -> [9.27529753e+02 4.09319865e+02 1.59927236e+00 4.04448617e+03]
```

Fonte: Resultados da Pesquisa

O valor de -202957.00 representa o intercepto e os valores de 9.275e02, 4.093e02, 1.599 e 4.044e03 representam os coeficientes para as variáveis independentes "IPCA", "QUANTIDADE", "CMV_TOTAL" e "PERC_MARGEM" respectivamente.

As métricas *R-squared (R^2)* e o RMSE comparativos entre os seis algoritmos testados pela aplicação, além da média aritmética simples são mostrados na Figura 23.

Figura 23 - Métricas R-squared (R2) e RMSE

```
R2 Score ------------------------------------------------------------------------------------------
LinearRegression               : 0.9755563
RandomForestRegressor          : 0.9669644
Support Vector Regression - SVR: 0.9358383
Lasso Regression               : 0.9755637
Ridge Regression               : 0.9755812
ElasticNet Regression          : 0.9759423
Simple Mean                    : 0.8286912
The best model r2 is: ElasticNet Regression n with value: 0.9759422544722696
```

```
Raiz quadrada do erro-médio (RMSE - Root-mean-square deviation) Score ----------------------------------
LinearRegression               : 20872.3005023
RandomForestRegressor          : 24834.9039078
Support Vector Regression - SVR : 33816.2276964
Lasso Regression               : 20869.1533202
Ridge Regression               : 20861.6784588
ElasticNet Regression          : 20706.8743892
Simple Mean                    : 55255.6870855
The best model RMSE is: ElasticNet Regression n with value: 20706.87438922723
```

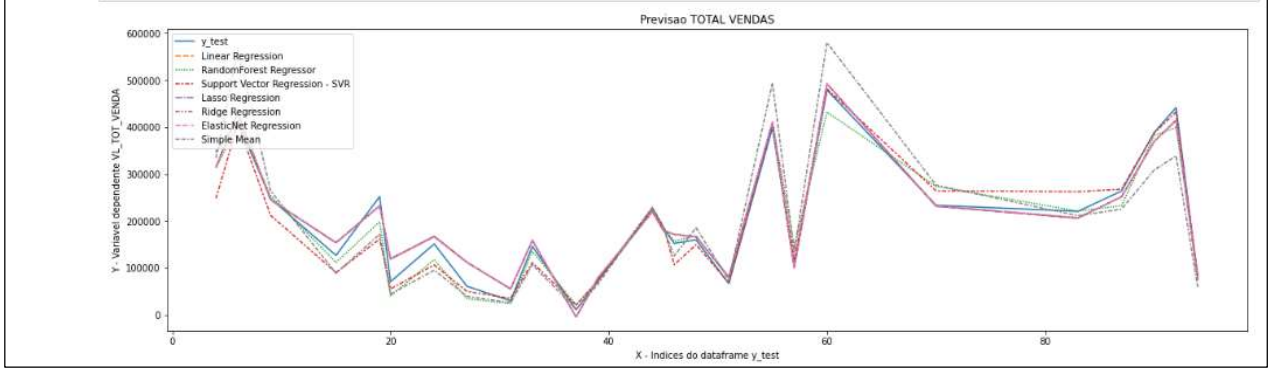
Fonte: Resultados da Pesquisa

A Figura 23 apresenta o maior índice entre os algoritmos comparados pela aplicação para o coeficiente de determinação *R-squared (R^2),* destacando o índice que apresentou melhor desempenho, neste caso foi o *ElasticNet Regression*. Da mesma forma também é destacado o menor índice entre os algoritmos comparados pela aplicação para a variação de erro RMSE.

Adicionalmente, o protótipo apresenta o gráfico mostrado na Figura 24.

Figura 24 - Gráfico comparativo entre os modelos

```
In [221]: plt.figure(figsize=(21, 6))
          sns.lineplot(data=tab_aux)
          plt.title("Previsao TOTAL VENDAS")
          plt.ylabel('Y - Variavel dependente VL_TOT_VENDA')
          plt.xlabel('X - Indices do dataframe y_test')
          plt.show()
```



Fonte: Resultados da Pesquisa

Por meio da biblioteca *matplotlib.pyplot* a Figura 24 apresenta o gráfico comparativo entre as observações efetivas dos dados (y_test), da média aritmética simples e das predições calculadas pelos seis algoritmos treinados.

3.5 Avaliação

Com o objetivo de evidenciar que o artefato de solução computacional desenvolvido a partir desta pesquisa contribui de forma relevante para predição de vendas, a fase de avaliação foi dividida entre os resultados comparativos dos modelos (DOE), entrevista com usuário da empresa (avalição interna) e entrevista com especialistas (avaliação externa).

3.5.1 Delineamento de Pesquisa (DOE)

A maioria dos algoritmos de aprendizagem de máquina possuem hiper parâmetros. Em uma rede neural artificial é necessário determinar o número de camadas ocultas, nódulos e muitos outros parâmetros relacionados ao processo de montagem do modelo. Apesar disso, não há um consenso claro sobre como ajustá-los. A metodologia mais popular é uma busca exaustiva na pesquisa de grade, que pode ser altamente ineficiente e às vezes inviável. Outra solução comum é mudar um hiper parâmetro de cada vez e medir seu efeito no desempenho do modelo. No entanto, isso também pode ser ineficiente e não garante resultados ideais, pois ignora as interações entre os hiper parâmetros (SNOEK *et al*., 2012). Nesta pesquisa, será utilizado a metodologia *Design of Experiments* (DOE) para ajustar os hiper parâmetros dos algoritmos de aprendizagem de máquina utilizados. Os benefícios incluem menos ciclos de treinamento, melhor seleção de parâmetros e uma abordagem disciplinada baseada em teoria estatística.

O ajuste de hiper parâmetro é essencial para otimizar a performance de qualquer algoritmo de ML. Apesar dessa importância, entender como os hiper parâmetros interagem com o desempenho do modelo, continua sendo um desafio. Vários esforços foram feitos para ajustar algoritmos de ML específicos. Por exemplo, Lalor *et al*. (2017) demonstram uma abordagem para afinação de redes neurais profundas utilizando subconjuntos de dados para treinar o modelo. Maclaurin *et al.* (2015), propuseram uma abordagem baseada em gradiente para a sintonia do modelo de rede neural que calcula os derivados da validação cruzada em relação aos hiper parâmetros por meio de um modelo *stochastic algorithm descent*. Nickson *et al.* (2014), utilizaram um método de ajuste de algoritmos estocásticos para b*ig data* uma vez que o treinamento com um conjunto de dados completo não foi viável. No entanto, este método só é aplicável a dados de séries temporais de processos gaussianos. Bardenet *et al.*, (2013) apresentaram um

algoritmo de seleção de hiper parâmetros que incorpora o conhecimento aprendido em experimentos anteriores usando técnicas de classificação e otimização baseadas em substituição. Um método de otimização bayesiana baseado em processo gaussiano para selecionar hiper parâmetros numéricos também foi proposto em (SNOEK *et al*., 2012).

A pesquisa aleatória é um método que recentemente se tornou uma alternativa popular à pesquisa em grade. Os autores Bergstra e Bengio (2012) afirmam que a busca aleatória é mais eficiente do que a busca em grade porque normalmente apenas um subconjunto dos hiper parâmetros ajustáveis de um modelo é importante para otimizar o desempenho. No entanto, a busca aleatória mostrou-se pouco confiável para ajustar os hiper parâmetros de *Deep Belief Networks* (DBNs) (BERGSTRA *et al.*, 2011); os autores desse trabalho introduziram duas estratégias de seleção sequencial que superaram tanto a busca aleatória quanto as buscas guiadas DBNs. No entanto, essas abordagens não medem o efeito que cada um dos hiper parâmetros tem no desempenho do modelo e ignoram possíveis interações entre hiper parâmetros. O efeito que o número de nós em uma rede neural tem no desempenho do modelo pode depender do número de camadas ocultas. Consequentemente, as atuais estratégias de ajuste podem ser vistas como otimizando uma função de caixa preta desconhecida (SNOEK *et al*., 2012), ignorando amplamente a questão de entender o comportamento interno do sistema.

Conforme restrições mencionadas anteriormente para identificar adequadamente os hiper parâmetros, é proposto nesta pesquisa a utilização da metodologia de *design* de experimentos (DOE) como o primeiro passo para rastrear os hiper parâmetros (fatores) mais significativos de um algoritmo de ML.

Reduzir o número de fatores para um subconjunto que tenha o maior efeito no desempenho do modelo, reduz consideravelmente o número de execuções de ajuste de modelo na próxima rodada de experimentos de ajuste de hiper parâmetros. A fase de triagem é feita usando delineamentos fatoriais fracionários, que são bem adequados para cenários em que não temos recursos computacionais suficientes para executar muitos experimentos.

O DOE em Inteligência Computacional (CI) é um dos aspectos mais importantes em cada processo de pesquisa, por isso é crucial definir corretamente todas as etapas que devem ser tomadas para garantir bons resultados.

Um DOE incorreto ou uma definição incorreta de uma de suas etapas pode direcionar a uma escolha errada do melhor método para resolver um determinado problema.

Hoje vivemos uma era de informações disponíveis publicamente, ou seja, temos vários bancos de dados abertos, especialmente quando consideramos que a disponibilidade de conjuntos de dados em domínio público disparou nos últimos anos, contudo parece não haver nenhuma orientação ou conjunto de procedimentos comumente aceitos para a preparação na análise destes dados (FOURCHES *et al*., 2010).

Esta pesquisa propõe uma normalização do padrão DOE para aplicar os requisitos de simulação dos algoritmos e balanceamento dos hiper parâmetros em quatro fases: I) Conjunto de dados, II) Pré-processamento de dados, III) Treinamento do modelo e IV) Seleção do melhor modelo. Estas fases incluem as operações ou etapas, as quais devem ser seguidas para obter resultados reprodutíveis e comparáveis.

Todas as fases propostas neste desenho experimental são importantes, mas a fase final de seleção do melhor modelo é onde erros ou variações referente ao nosso modelo podem ocorrer, ou simplesmente nossas recomendações podem não ser levadas em consideração. Por esta razão, a metodologia proposta possui especial atenção neste ponto, fornecendo diretrizes estatísticas robustas para garantir a reprodutibilidade dos resultados, e propõe algumas melhorias ou modificações na metodologia padrão do DOE, a fim de alcançar o objetivo final, ou seja, confiabilidade nos modelos de previsão.

Este método multidisciplinar não é um fluxo de trabalho engessado nas diferentes fases, pois deve ser adaptável aos diferentes campos, cada um deles com diferentes etapas internas. Assim, trata-se de uma proposta mais ampla que pode ser tomada com uma boa prática de trabalho, válido para qualquer tipo de experimentação onde algoritmos de ML estejam envolvidos.

Esta pesquisa apresenta uma estrutura conceitual de DOE, cujo foco principal está na fase de identificação do melhor modelo por meio de uma abordagem essencialmente estatística. Essa abordagem fornece uma maneira robusta, estável e reprodutível de como realizar um DOE aplicado na análise de algoritmos de ML com base em regressão linear. Segundo Baker (2016b), a não reprodutibilidade cientifica é uma preocupação importante e crescente, pois foi constatado que, com mais de 1.500 pesquisadores, 87% indicaram ter identificado um desenho experimental ruim devido à ausência de reprodutibilidade e, também, um número expressivo de 89% detectaram falhas na análise estatística (BAKER, 2016a).

Para esta pesquisa foram selecionados cinco diferentes produtos que compõem o portifólio de vendas da empresa onde esta pesquisa foi aplicada, tais produtos foram nomeados como 13M1S1, 0-0001, 0-B051, 8-K011 e 1-B301. Como critério de escolha dos produtos, foram identificados aqueles que possuem movimentações de vendas registradas entre os anos de 2006 e 2022 e que estejam em três categorias diferentes: Digitalizador biométrico, Digitalizador de imagens e consumíveis.

Na primeira fase, foram aplicados seis algoritmos de ML utilizando bibliotecas Python para identificar o algoritmo que apresentasse o maior coeficiente de determinação R^2 conforme o produto selecionado e seu respectivo histórico de vendas. Na segunda e última fase, os resultados obtidos serão comparados com os resultados obtidos usando a validação cruzada externa.

O objetivo deste estudo é apresentar um conjunto de diretrizes para realizar análises multivariadas a fim de obter modelos de ML estatisticamente sólidos para uma comparação precisa dos diferentes resultados obtidos por esses métodos. Assim como, apresentar uma metodologia abrangente que suporte a modelagem preditiva.

Este experimento está organizado da seguinte forma: I) descrever a metodologia e os ajustes necessários em suas diferentes etapas: conjunto de dados, pré-processamento de dados, aprendizado de modelos e seleção do melhor modelo; II) apresentar os resultados por meio de uma comparação com os algoritmos de ML e uma análise experimental do desempenho da metodologia proposta; III) apresentar as discussões e considerações finais.

Metodologia proposta

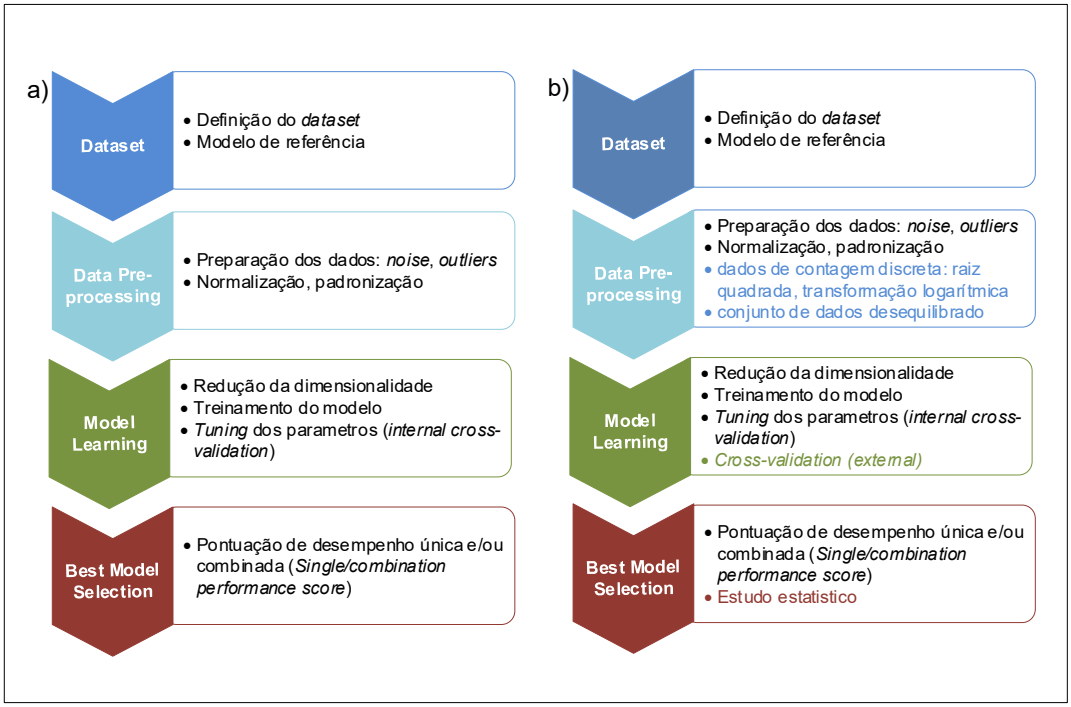
Por meio de uma normalização de projetos experimentais para problemas de inteligência computacional, assim como as disciplinas relacionadas onde é necessário selecionar o melhor modelo de ML, foi aplicada uma metodologia implementada em *Python* para automatizar a modelagem preditiva em problemas de regressão.

A linguagem *Python* é escolhida com o objetivo de fornecer ao usuário final uma aplicação intuitiva, possibilitando ao usuário realizar simulações selecionando as variáveis independentes disponíveis no modelo. Além disso foi implementado a conexão direta ao banco de dados, evitando que os dados tenham que ser exportados a partir do banco de dados, para somente depois serem importados na aplicação, ou seja, com o *Python* é fornecido um ambiente focado na usabilidade do usuário final.

Em Tsiliki *et al.* (2015a) e Tsiliki *et al*. (2015b), os autores observaram que havia uma necessidade de padronização de metodologias em diferentes partes da análise dos algoritmos: análise dos dados, métodos de validação cruzada, parâmetros específicos de regressão e critérios na seleção do melhor modelo.

Desta forma, a normalização foi implementada a partir da definição de um fluxo de trabalho que contém as seguintes fases para evidenciar onde estão localizadas as diferenças dentro da proposta desta pesquisa e o modelo proposto por Tsiliki: Conjunto de dados, Pré-processamento de dados, Aprendizado do modelo e Seleção do melhor modelo, que são representadas graficamente na Figura 25. Os parágrafos a seguir descrevem mais detalhadamente cada uma destas fases.

Figura 25 – Fluxo de trabalho para seleção dos algoritmos com melhor desempenho



Fonte: Adaptado de Tsiliki et al. (2015a)

Na Figura 25 (a), é mostrado o fluxo padrão utilizado na maioria dos projetos experimentais em inteligência computacional publicados na literatura. A Figura 25 (b), detalha as fases em que são propostas mudanças metodológicas para garantir que os modelos de ML não sejam enviesados, assim como os resultados obtidos contemplem o melhor desempenho possível.

É importante ressaltar que esta metodologia utiliza algoritmos de ML para resolver problemas de regressão e, consequentemente, é uma metodologia universal. Apesar da capacidade dessas técnicas de resolver problemas do mundo real, elas também apresentam inconvenientes e, obviamente, limitações particulares que devem ser levadas em consideração quando usadas. Mais precisamente, a metodologia

proposta por Tsiliki *et al*. (2015a) não leva em consideração que o desempenho das técnicas de ML está diretamente relacionado às observações utilizadas para treinar os modelos.

Assim, uma análise estatística da variabilidade e estabilidade das técnicas é essencial dentro de diferentes execuções e diferentes cargas iniciais para separar os dados. Além disso, a validação cruzada é necessária não apenas para selecionar os melhores parâmetros (fase de ajuste interno - *internal tuning phase*) para cada técnica proposta, mas também externamente, para garantir que o treinamento do modelo não seja enviesado ou falho, conforme mostrado na Figura 25 (b).

Há também duas considerações pontuais sobre o pré-processamento dos dados que surgem quando os modelos de ML são aplicados: I) como lidar com os dados de contagem e II) como lidar com conjuntos de dados desequilibrados.

3.5.1.1 Conjunto de dados (Dataset)

Primeiramente, o conjunto de dados deve ser gerado, definindo suas características particulares. A definição deve conter as variáveis envolvidas no estudo e uma breve descrição de cada uma delas para garantir a reprodutibilidade dos testes por pesquisadores externos. Em algumas situações, para garantir que os dados sejam suficientemente representativos para o problema específico em estudo, é necessária a ajuda de especialistas para definir quais variáveis independentes são necessárias para suporte na predição das variáveis dependentes.

Nesta pesquisa, os dados foram obtidos a partir de uma empresa que atua no mercado varejista brasileiro, sendo que a seleção da empresa para este estudo de campo não ocorreu de forma aleatória. Ao apresentar um método estruturado de previsão de vendas utilizando média aritmética simples, a empresa foi convidada a participar do estudo mediante o termo de autorização para coleta de dados, cujo qual está arquivado no Centro Paula Souza e faz parte integrante desta pesquisa. Trata-se de uma empresa multinacional japonesa atuante no mercado varejista brasileiro desde 1972, na qual concordou em participar da pesquisa desde que sua identidade não fosse revelada assim como informações sigilosas ou dados estratégicos.

Desta forma, foram coletadas 48.977 transações no estoque (instâncias ou linhas) e 16 atributos (variáveis ou colunas), distribuídos em 5 diferentes produtos conforme Tabela 16.

No Apêndice A é apresentado o código fonte para a conexão com o banco de dados e a coleta das informações.

Tabela 16- Número de transações no estoque por produto

Produto	Tipo	Transações	Período
13M1S1	Digitalizador biométrico	42.887	2006 a 06/2018
0-0001	Consumível para digitalizador de imagem	2.998	2008 a 05/2022
0-B051	Digitalizador de imagem	1.973	2014 a 05/2022
8-K011	Consumível para digitalizador de imagem	2.642	2004 a 05/2022
1-B301	Digitalizador de imagem	568	2016 a 05/2022

Fonte: Resultados da Pesquisa

Cada transação possui 17 atributos, os quais são detalhados na Tabela 17.

Tabela 17 - Atributos da transação do estoque

#Id	Atributo	Finalidade	Variável	Classificação	Tipo
01	COD_ESTABEL	Identificação do estabelecimento	Categórica	Endógena	Independente
02	IT_CODIGO	Código do item	Categórica Nominal	Endógena	Independente
03	TP_TRANS	Tipo da transação (1-Entrada / 2-Saída)	Categórica Intervalar	Endógena	Independente
04	DT_TRANS	Data da transação (DD/MM/AAAA)	Contínua	Endógena	Independente
05	NRO_DOCTO	Número do documento	Contínua	Endógena	Independente
06	COD_EMITENTE	Identificação do emitente (Cliente/Fornecedor)	Categórica Intervalar	Endógena	Independente
07	NOME_EMIT	Nome do emitente	Categórica Nominal	Endógena	Independente
08	COD_LOCALIZ	Localização no estoque	Categórica Intervalar	Endógena	Independente
09	ESP_DOCTO	Espécie da transação (NFE-NF de entrada; NFS-NF de saída)	Categórica Intervalar	Endógena	Independente
10	COD_DEPOS	Identificação do depósito logístico	Categórica Intervalar	Endógena	Independente
11	QUANTIDADE	Quantidade do item para movimentação do estoque	Discreta	Endógena	Independente
12	VL_VENDA_UNIT	Valor de venda unitário	Contínua	Endógena	Dependente
13	VL_VENDA_TOT	QUANTIDADE * VL_VENDA_UNIT	Contínua	Endógena	Dependente
14	VL_MATERIAL_UNIT	Valor de compra unitário	Contínua	Endógena	Independente
15	CMV_UNITARIO	Custo médio do estoque no mês da transação	Contínua	Endógena	Independente
16	PERC_MARGEM	Percentual de margem bruta (*Gross Margin*) (((Quantidade * Vl_venda_unit) – (Quantidade * Vl_material_unit)) / (Quantidade * Vl_venda_unit)) * 100	Contínua	Endógena	Independente
17	IPCA	Percentual de inflação medida no mês (MM/AAAA)	Contínua	Exógena	Independente

Fonte: Resultados da Pesquisa

Os atributos do conjunto de dados estão classificados em dois grupos, aqueles relacionados ao histórico de movimentação do item (16 atributos) classificados como variáveis endógenas, ou seja, são variáveis obtidas por fatores internos do próprio ambiente e um atributo externo IPCA classificado como

variável exógena, ou seja, são variáveis obtidas por fatores externos e são fixadas no momento que são introduzidas no modelo.

Os atributos também são classificados entre variáveis independentes e variáveis dependentes. O protótipo foi desenvolvido para predizer o valor de receita para os próximos períodos, desta forma os atributos VL_VENDA_UNIT e VL_VENDA_TOT foram categorizadas como variáveis dependentes, ou seja, são variáveis alvo.

A análise exploratória de dados, em inglês *exploratory data analysis* (EDA) é o primeiro passo para todas as tarefas de ML, pois permite um melhor e profundo entendimento dos dados e recursos fornecidos. Explorar os dados visualmente é uma das maneiras mais eficazes de entender as distribuições das variáveis, encontrar valores ausentes e pensar na melhor forma de lidar com eles e investigar as relações entre as variáveis.

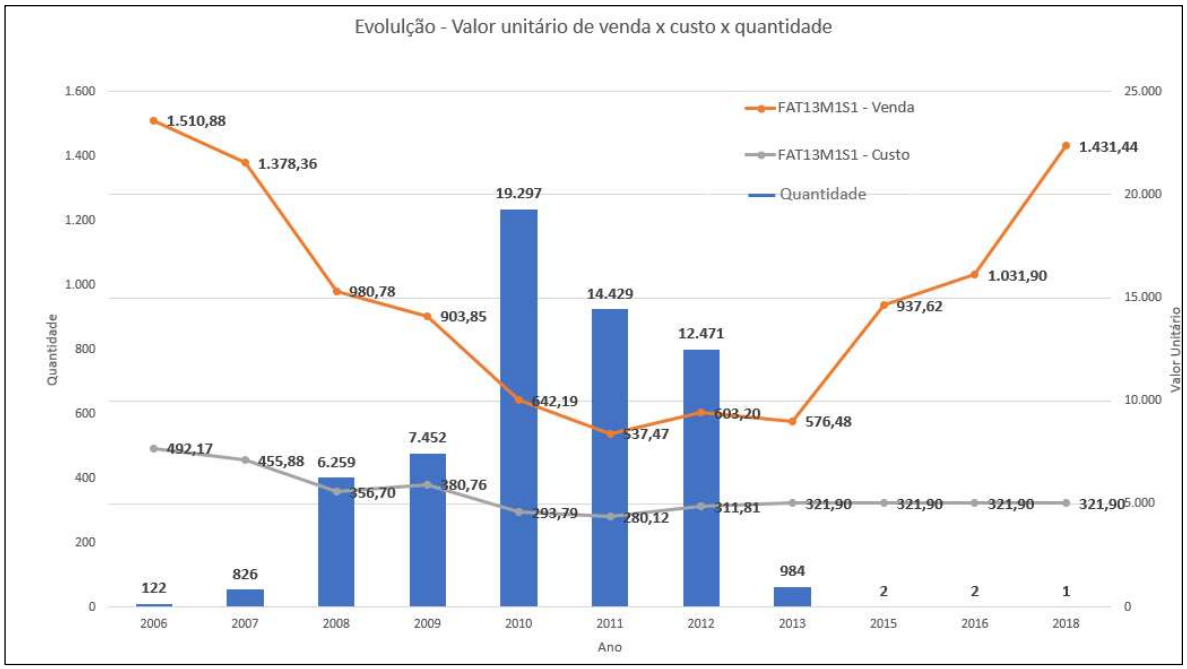
Segundo Rumsey (2017), em estatística a variável pode ser definida como a característica que é medida ou avaliada em cada elemento da amostra ou população.

Para fins organizacionais, a primeira parte da EDA envolve a separação das seguintes variáveis:

- Variável categórica: Contêm um número finito de categorias ou grupos distintos. Os dados categóricos podem não ter uma ordem lógica. Por exemplo, os preditores categóricos incluem gênero, tipo de material e método de pagamento (RUMSEY, 2017).
- Variável discreta: São variáveis numéricas que têm um número contável de valores entre quaisquer dois valores. Uma variável discreta é sempre numérica. Por exemplo, o número de reclamações de clientes ou o número de falhas ou defeitos (RUMSEY, 2017).
- Variável continua: São variáveis numéricas que têm um número infinito de valores entre dois valores quaisquer. Uma variável contínua pode ser numérica ou data/hora. Por exemplo, o cumprimento de uma peça ou a data e hora em que um pagamento é recebido (RUMSEY, 2017).

Para as variáveis categóricas os dados serão apresentados em forma de gráficos de barras e para as variáveis contínuas serão utilizados histogramas. Nos próximos parágrafos são levantadas algumas considerações a respeito desta fase exploratória.

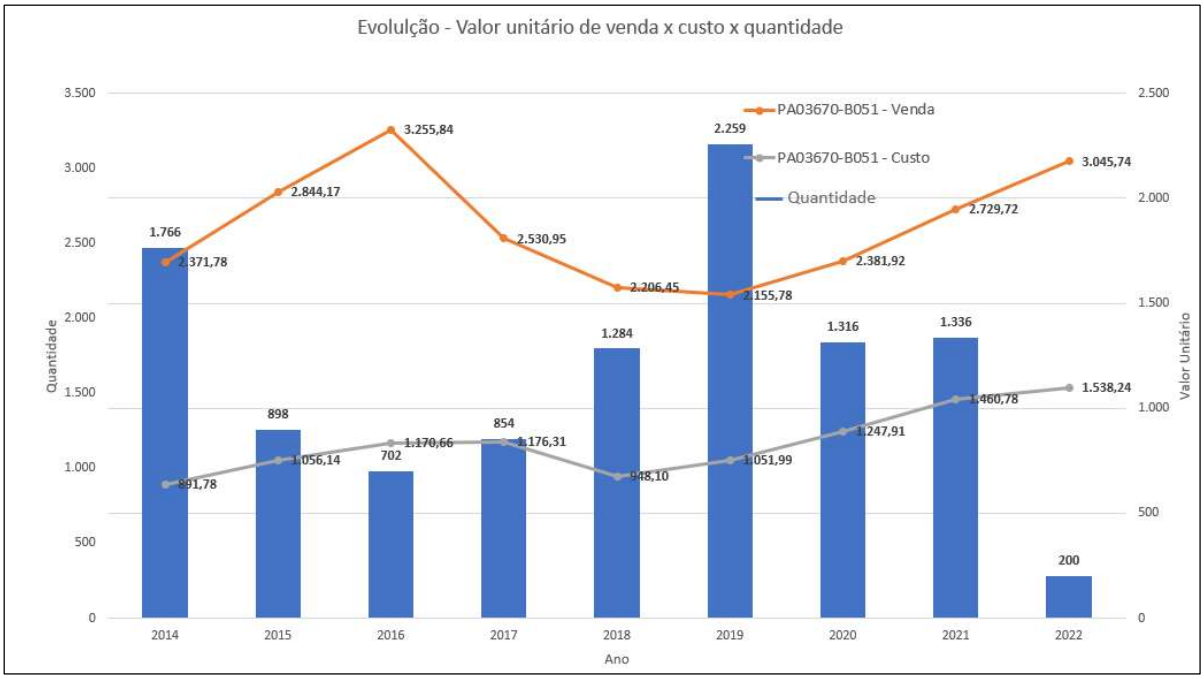
Figura 26 - Histórico valor unitário venda x custo x quantidade (produto 13M1S1)



Fonte: Resultados da Pesquisa

Na Figura 26, a barra azul representa a quantidade de venda ao longo dos anos, enquanto a linha laranja representa o valor unitário médio de venda e a linha cinza o custo médio do produto 13M1S1, evidenciando que entre os anos de 2009 e 2012, com o aumento na quantidade de itens vendidos houve um decréscimo no valor unitário de venda, entretanto, à medida que a quantidade vendida diminui o valor unitário de venda tende a aumentar, evidenciando uma correlação inversa entre quantidade e valor unitário de venda.

Figura 27 - Histórico valor unitário venda x custo x quantidade (produto 0-B051)



Fonte: Resultados da pesquisa

O comportamento da evolução do valor unitário de venda para o produto 0-B051 apresentado na Figura 27 é diferente do produto 13M1S1 apresentado na Figura 26, ou seja, não existe uma correlação direta ou inversa entre os atributos quantidade e valor unitário de venda.

Os períodos mostrados nos gráficos são diferentes pois a comercialização do produto 13M1S1 iniciou em 2006 e sendo mais recente o produto 0-B051 teve o início de sua comercialização em 2014. O produto 13M1S1 foi selecionado devido ao elevado número de transações, mesmo sendo descontinuado em 2018 o seu produto substituto não foi selecionado devido ao baixo volume de transações.

A mesma análise foi realizada para todos os cinco produtos, ficando evidenciado que para estes produtos não existe vendas sazonais, ou seja, as vendas são distribuídas de forma aleatória durante todos os meses dos anos, não existindo períodos específicos de menor ou maior demanda.

Após a EDA, uma análise bivariada ocorre onde as relações ocultas entre a variável alvo (dependente) e as variáveis independentes (preditoras) são exploradas principalmente usando gráficos de dispersão e, em seguida, é aplicado o pré-processamento de dados e engenharia de recursos.

Nesta pesquisa, foram utilizados os cinco conjunto de dados mencionados na Tabela 16, sendo que as colunas não numéricas foram eliminadas. Para criação do *dataset* foi utilizado a biblioteca "*Pandas*". Em programação de computadores, "*Pandas*" é uma biblioteca de software criada para a linguagem *Python* para manipulação e análise de dados. Em particular, oferece estruturas e operações para manipular tabelas numéricas e séries temporais. Na Figura 28Figura 28, é apresentado o comando utilizado para filtrar somente os atributos numéricos e a utilização do método *loc* para filtrar somente as movimentações de vendas, com base nos atributos "ESP_DOCTO" e "TP_TRANS".

Figura 28 - Utilizando biblioteca "*Pandas*" para eliminar variáveis não numéricas

```
In [25]: # filtrar somente os movimentos de saida (faturamento)
         df = df[['ANO', 'MES', 'IPCA', 'QUANTIDADE', 'VL_VENDA_UNIT', 'VL_VENDA_TOT', 'CMV_TOTAL', 'PERC_MARGEM']].loc[(df['ESP_DOCTO']==
                                                                                                                        (df['TP_TRANS']=='SAI')]
```

Fonte: Resultados da Pesquisa

3.5.1.2 Pré-processamento de dados (Data pre-processing)

Após a geração do conjunto de dados, os dados estão em estado bruto ou puro. Os dados brutos são muitas vezes difíceis de analisar, por isso geralmente requerem um estudo preliminar ou uma etapa de pré-processamento. Nesta fase é verificado se há registros com informações incompletas, *outliers* ou

noise. Caso algumas das situações mencionadas estejam presentes no conjunto de dados, diferentes abordagens devem ser aplicadas para evitá-las. Somente uma vez finalizado esse processo, considera-se que os dados estão prontos para análise. Para entender a importância dessa etapa, Dasu *et al*. (2013) relata que 80% do esforço de uma análise de dados é utilizado compilando dados corretamente para análise.

Além disso, normalmente as variáveis apresentam escalas ou tamanhos diferentes, dificultando a comparação em igualdade de condições. Assim, técnicas de normalização ou padronização são necessárias para tornar os dados comparáveis, ou seja, compatíveis. Ambas as técnicas têm certamente desvantagens e nenhuma é melhor que a outra. Além disso, o conjunto de dados deve ser estudado para cada problema específico antes de aplicá-lo. Por exemplo, se houver uma tentativa de realizar uma etapa de normalização e houver *outliers* nos dados (não removidos anteriormente), essa etapa dimensionará indevidamente os dados úteis podendo ocasionar algum tipo de viés no modelo, sendo este um comportamento não desejável.

Análise de outliers

Antes de iniciar a análise de *outlier* é necessário validar se o *dataset* possui algum valor nulo. Utilizando a combinação dos métodos *isnull()* e *sum()* contidos na biblioteca *Pandas*, é possível validar se todos os valores nulos foram removidos, pois a quantidade de registros retornados para cada atributo deve ser zero, indicando a ausência de valores nulos, conforme evidenciado na Figura 29.

Figura 29 - Validar se o dataset possui valor nulos

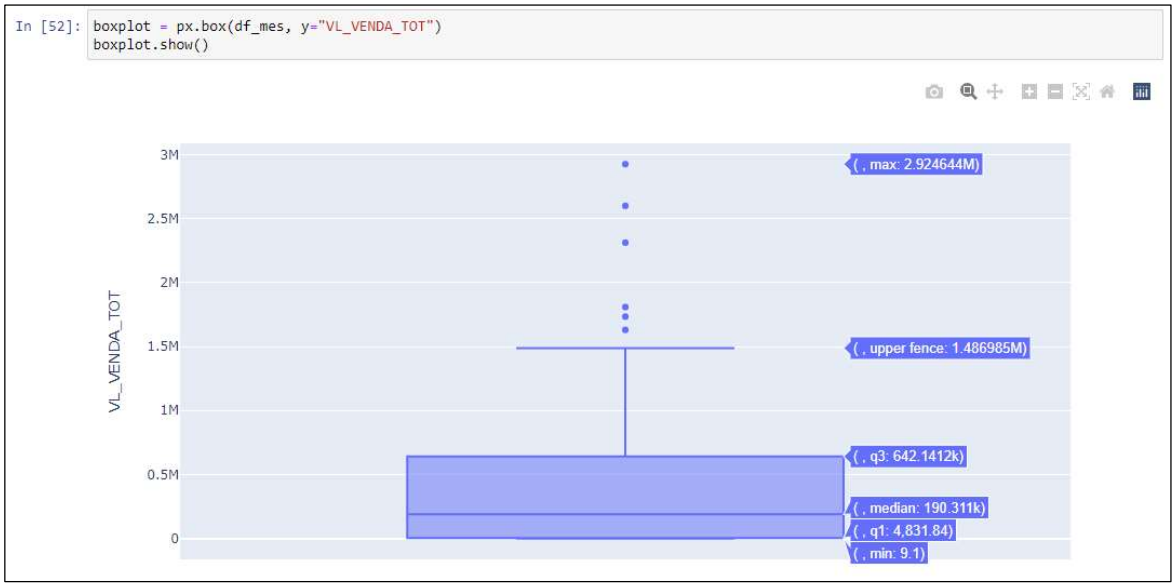
```
In [17]: print('Tratamento inicial do dataset para verificar valores nulos ########
         print(df_mes.isnull().sum())

         Tratamento inicial do dataset para verificar valores nulos ###############
         ANO             0
         MES             0
         IPCA            0
         QUANTIDADE      0
         VL_VENDA_UNIT   0
         VL_VENDA_TOT    0
         CMV_TOTAL       0
         MES_EXT         0
         PERC_MARGEM     0
         dtype: int64
```

Fonte: Resultados da Pesquisa

O *boxplot* ou diagrama de caixa é uma ferramenta gráfica que permite visualizar a distribuição e valores discrepantes (*outliers*) dos dados, fornecendo assim um meio complementar para desenvolver uma perspectiva sobre o caráter dos dados. Além disso, o *boxplot* também é uma disposição gráfica comparativa. As medidas de estatísticas como o mínimo, máximo, primeiro quartil, segundo quartil ou mediana e o terceiro quartil formam o *boxplot*.

Faz parte da fase de pré-processamento dos dados, a eliminação de possíveis *outliers,* com isso o *boxplot* é uma poderosa ferramenta visual para realizarmos esta identificação. Esta fase é aplicada para o atributo que contém o valor total de vendas "VL_VENDA_TOT", assim como para os demais atributos do *dataset*.

Figura 30 - Boxplot para o atributo total de vendas



Fonte: Resultados da Pesquisa

Na Figura 30 são evidenciados seis valores *outliers* superiores ao valor de 1.5M de vendas, desta forma, tais registros são eliminados para que o modelo não fique desbalanceado, sendo que o mesmo procedimento é realizado para os demais atributos numéricos do *dataset*.

Análise da Normalidade

Analisar a normalidade dos dados é importante, pois o resultado desta análise indicará qual será o método aplicado para tratar o tipo de correlação entre as variáveis. O protótipo desenvolvido a partir desta pesquisa está preparado para aplicar três tipos de métodos para análise de correlação: *Pearson*, *Spearman* ou *Kendall*.

Passando no teste da normalidade, será aplicado o coeficiente de *Pearson* caso exista uma correlação linear, caso contrário será aplicado o coeficiente de *Spearman* para correlação não linear e *Kendall* será aplicado para os conjuntos de dados que apresentem um número elevado de valores iguais e poucos registros, ou seja, normalmente até 30 registros. (PUTH *et al*., 2015).

Conforme Wiedermann *et al.* (2015), a correlação de *Pearson* avalia a relação linear entre duas variáveis contínuas e quantitativas. Uma relação é linear quando a mudança em uma variável é associada

a uma mudança proporcional na outra variável, ou seja, as variáveis se movem na mesma direção ou em direção inversa, porém a uma taxa constante.

A correlação de *Spearman* avalia a relação monotônica entre duas variáveis contínuas ou ordinais, ou seja, em uma relação monotônica, as variáveis tendem a mover-se na mesma direção relativa, mas não necessariamente a uma taxa constante. O coeficiente de correlação de *Spearman* baseia-se nos valores classificados de cada variável, em vez dos dados brutos, além disso é muito usada para avaliar relações envolvendo variáveis ordinais. Por exemplo, você poderia usar a correlação de *Spearman* para avaliar se a ordem na qual os funcionários executam determinada atividade está relacionada ao número de meses de emprego. (XU *et al*., 2012).

O coeficiente de correlação por postos de *Kendall*, é uma medida de associação entre duas variáveis. É semelhante à correlação de *Spearman*, por também ser calculado através dos postos das variáveis. Assim como na correlação de *Spearman*, o *Kendall* descreve a relação entre as variáveis por meio de uma função monotônica, isso quer dizer que ele analisa se o valor de uma variável aumenta ou diminui conforme o valor da outra variável aumenta ou diminui. Quanto mais próximos dos extremos (-1 ou 1), maior é a força da correlação, enquanto valores próximos de zero implicam em correlação mais fracas ou inexistentes. (XU *et al*., 2012).

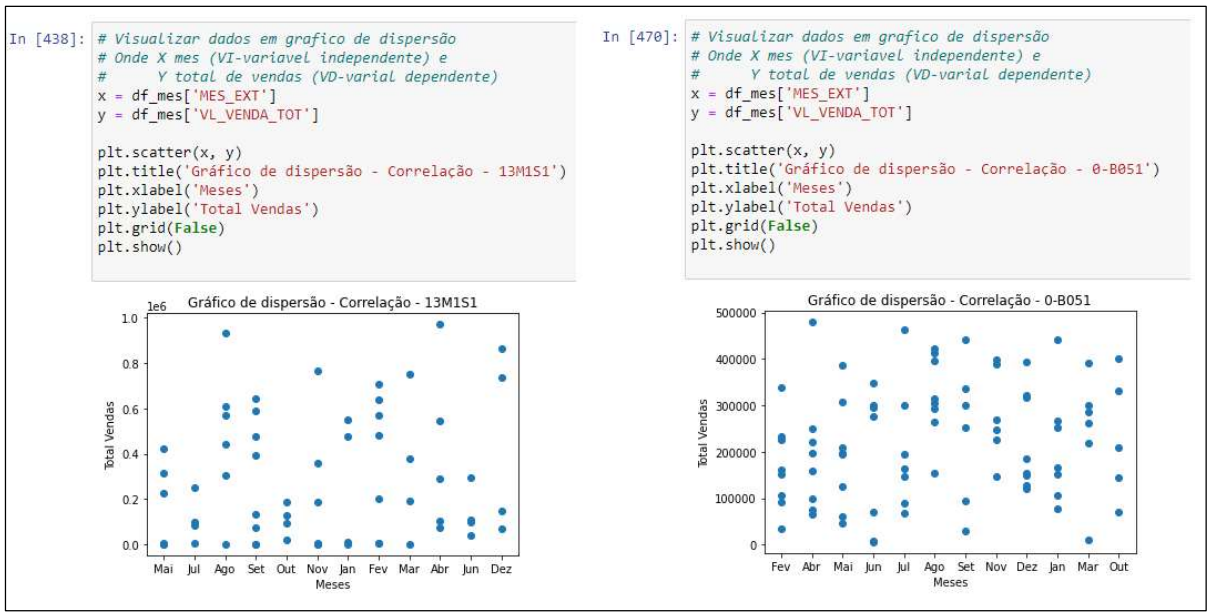
Quando o resultado do coeficiente é positivo, indica que o aumento em uma variável implica no aumento na outra variável, quando negativo, indica que o aumento em uma variável implica no decréscimo da outra. (HEUVEL *et al*., 2022).

O coeficiente de *Kendall* e *Spearman*, servem propósitos muito semelhantes. De maneira geral, o coeficiente de *Kendall* tem níveis de significância mais confiáveis em amostrar pequenas do que o coeficiente de *Spearman*. (PUTH *et al*., 2015).

Para examinar se a distribuição entre as varáveis é Normal, este protótipo utiliza o gráfico de dispersão, uma vez que os coeficientes de correlação medem apenas relacionamentos lineares (*Pearson*) ou monotônicos (*Kendall* ou *Spearman*), sendo que outras relações também são possíveis.

Na Figura 31Figura 31 é destacado o gráfico de dispersão, comparando os meses (eixo X) com o total de vendas (eixo Y) para os produtos 13M1S1 e 0-B051.

Figura 31 - Gráfico de dispersão para os produtos 13M1S1 e 0-B051



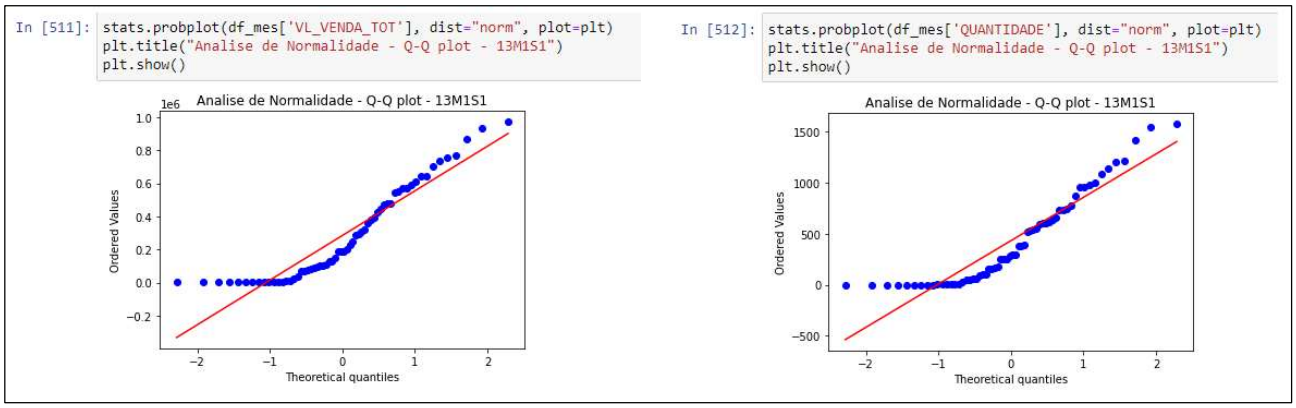
Fonte: Resultados da Pesquisa

Na Figura 31Figura 31 fica evidenciado que não existe uma correlação entre o total de vendas com relação ao mês, ou seja, neste caso não se trata de produtos com vendas sazonais uma vez que as vendas estão bem distribuídas durante todos os meses dos anos conforme períodos mencionados na Tabela 16.

O gráfico histograma é adequado para avaliarmos a normalidade quando a distribuição é normal, porém quando a distribuição não é normal, este gráfico pode gerar confusão na interpretação, desta forma este protótipo também utiliza o gráfico *quantile plot* (qq-plot), o qual irá dispor uma comparação, dois a dois dos quantis teóricos de uma Normal e os quantis de seus dados. Se os pontos se concentrarem em torno de uma reta, então temos indícios de que a distribuição é Normal. (MARDEN, 2004).

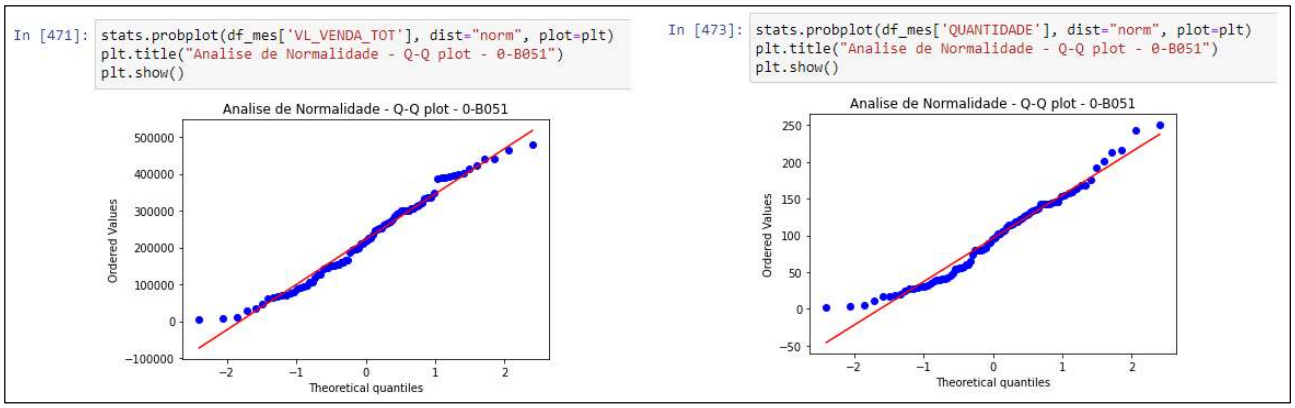
Na Figura 32 e na Figura 33, fica evidenciado pelo resultado de ambos os gráficos, tanto para o valor total de vendas, quanto para quantidade, que se trata de uma distribuição normal, uma vez que os pontos azuis permanecem próximos a linha vermelha ao longo do eixo X referente aos quantis teóricos. Desta forma em uma distribuição Normal pode ser utilizado o coeficiente de *Pearson*. (WIEDERMANN, *et al*., 2015).

Figura 32 - Análise de Normalidade utilizando qq-plot – (produto 13M1S1)



Fonte: Resultados da Pesquisa

Figura 33- Análise de Normalidade utilizando qq-plot – (produto 0-B051)



Fonte: Resultados da Pesquisa

No entanto, para algumas situações a visualização gráfica pode se tornar subjetiva, com isso se torna necessário aplicar outros métodos para validarmos se a distribuição é normal. Por meio da biblioteca *statsmodels* é possível aplicar o teste *Shapiro-Wilk*, cujo qual tem como objetivo avaliar se a distribuição é semelhante a uma distribuição normal. A distribuição normal também pode ser chamada de gaussiana. (JURECKOVÁ *et al*., 2007).

Utilizando o método *shapiro* da biblioteca *statsmodels*, esta pesquisa aplica as hipóteses contidas na Tabela 18.

Tabela 18 - Hipótese nula ou alternativa para *Shapiro-Wilk*

Hipótese	Tipo da Distribuição	*p*-valor
H_o	Normal	> 0.05
H_a	Diferente de Normal	<= 0.05

Fonte: Resultados da Pesquisa

Na Figura 34Figura 34, é apresentado o resultado do teste *Shapiro-Wilk* para o atributo total de vendas para os produtos 13M1S1 e 0-B051.

Figura 34 - Resultado do teste *Shapiro-Wilk*

```
In [91]: print('Teste Shapiro-Wilk  para analise de Normalidade - 13M1S1')
         estatistica, p = stats.shapiro(df_mes['VL_VENDA_TOT'])
         print('Estatística do teste SHAPIRO: {}'.format(estatistica))
         print('p-valor SHAPIRO: {}'.format(p))

         Teste Shapiro-Wilk  para analise de Normalidade - 13M1S1
         Estatística do teste SHAPIRO: 0.8806207776069641
         p-valor SHAPIRO: 2.1068375644972548e-05
```

```
In [127]: print('Teste Shapiro-Wilk  para analise de Normalidade - 0-B051')
          estatistica, p = stats.shapiro(df_mes['VL_VENDA_TOT'])
          print('Estatística do teste SHAPIRO: {}'.format(estatistica))
          print('p-valor SHAPIRO: {}'.format(p))

          Teste Shapiro-Wilk  para analise de Normalidade - 0-B051
          Estatística do teste SHAPIRO: 0.9738746881484985
          p-valor SHAPIRO: 0.0851423442363739
```

Fonte: Resultados da Pesquisa

Conforme demonstrado na Figura 34, o parâmetro *p*-valor retorna o resultado do método *shapiro*, ou seja, para o produto 13M1S1 o valor retornado foi 2.106^{e-05}, evidenciando que a hipótese alternativa (H_a) foi aceita, uma vez que o valor é inferior a 0.05, desta forma, o conjunto de dados não possui distribuição normal. Para o produto 0-B051 o valor retornado foi de 0.085, evidenciando que a hipótese nula (H_0) foi aceita, uma vez que o valor é superior a 0.05, indicando que o conjunto de dados possui distribuição normal.

Adicionalmente o protótipo aplica o teste de *Lilliefors*, cujo qual é uma adaptação do teste de *Kolmogorof-Smirnoff*, usado para verificação de normalidade de um conjunto de dados. A diferença básica entre os testes é que o teste de *Lilliefors*, testa a normalidade a partir de média e desvio padrão fornecidos, enquanto o *Kolmogorof-Smirnoff* utiliza a média e o desvio padrão calculado no próprio conjunto de dados. (BLAIN, 2014).

Os valores de *p*-valor para teste de hipótese, são os mesmos utilizados pelo *Shapiro-Wilk*, contidos na Tabela 18.

Figura 35 - Resultado do teste *Lilliefors*

```
In [170]: import statsmodels
          from statsmodels.stats.diagnostic import lilliefors

          print('Teste Lilliefors para analise de Normalidade - 13M1S1')
          estatistica, p = statsmodels.stats.diagnostic.lilliefors(df_mes['VL_VENDA_TOT'], dist='norm')
          print('Estatística de teste lilliefors : {}'.format(estatistica))
          print('p-valor: {}'.format(p))

          Teste Lilliefors para analise de Normalidade - 13M1S1
          Estatística de teste lilliefors : 0.16059397595991004
          p-valor: 0.0009999999999998899
```

```
In [129]: import statsmodels
          from statsmodels.stats.diagnostic import lilliefors

          print('Teste Lilliefors para analise de Normalidade - 0-B051')
          estatistica, p = statsmodels.stats.diagnostic.lilliefors(df_mes['VL_VENDA_TOT'], dist='norm')
          print('Estatística de teste lilliefors : {}'.format(estatistica))
          print('p-valor: {}'.format(p))

          Teste Lilliefors para analise de Normalidade - 0-B051
          Estatística de teste lilliefors : 0.08296746427532803
          p-valor: 0.21055191135796522
```

Fonte: Resultados da Pesquisa

Os resultados apresentados no teste de *Lilliefors* Figura 35, corrobora que que o produto 13M1S1 não possui distribuição normal conforme retorno do parâmetro *p*-valor igual a 0.000999, evidenciando que ($H_{o)}$ foi rejeitada. Referente ao produto 0-B051 a hipótese nula (H_o) foi aceita, uma vez que o valor retornado foi de 0.2105, confirmando que este conjunto de dados possui distribuição normal.

Após a etapa de normalização, os dados são dimensionados no intervalo [0,1] no caso de valores numéricos. Caso seja realizado um processo de padronização, os dados apresentam valor médio zero e desvio padrão igual a um, portanto, são independentes da unidade de medida. Dependendo dos tipos de dados, existem outras abordagens para minimizar a influência dos valores, como elevar uma característica a uma potência.

No caso de dados de contagem discreta, o uso da transformação de raiz quadrada é recomendado para normalizar os dados de contagem e depois transformá-los em logaritmo para análise posterior (Cuesta *et al*., 2008). No entanto, trabalhos sugerem que mais estudos são necessários para escolher a melhor técnica para lidar com os dados originais sem transformações (O'Hara *et al.*, 2010), ao invés do pré-processamento desses dados.

Outro ponto relevante ao usar métodos de ML é como lidar com conjuntos de dados desequilibrados. Existem principalmente duas abordagens diferentes para lidar com esses conjuntos de dados, sobreamostragem (*oversampling*), - criando exemplos sintéticos da classe minoritária, como proposto por Chawla *et al*. (2002) e subamostragem (*undersampling*), remoção de amostras da classe majoritária dos dados, conforme proposto por Seiffert *et al.* (2010) para fins de balanceamento.

Com base no resultado dos testes *Shapiro-Wilk* e *Lilliefors*, o protótipo aplica o método de correlação linear correspondente. Os métodos de correlação linear: *Pearson*, *Spearman* e *Kendall*, utilizam os parâmetros contidos na Tabela 19 para avaliar se existe correlação linear entre as variáveis.

Tabela 19 – Hipóteses para os métodos *Pearson*, *Spearman* e *Kendall*

Hipótese	Tipo da Distribuição	*p*-valor
H_o	Não há correlação linear	> 0.05
H_a	Existe correlação linear	<= 0.05

Fonte: Resultados da Pesquisa

Na Figura 36, é aplicado o método de *Spearman* para o produto 13M1S1, cujo conjunto de dados não possui distribuição normal.

Figura 36 - Correlação Linear de *Spearman* – (produto 13M1S1)

```
In [47]: print('Utilizando o coeficiente de SPEARMAN para validar se existe Correlação Linear - 13M1S1')
         coef, p = stats.spearmanr(df_mes['MES'], df_mes['VL_VENDA_TOT'])
         print('Coeficiente de correlação SPEARMAN com MES : {}'.format(coef))
         print('p-valor: {}'.format(p))

         Utilizando o coeficiente de SPEARMAN para validar se existe Correlação Linear - 13M1S1
         Coeficiente de correlação SPEARMAN com MES : -0.04923110701285823
         p-valor: 0.7039545479910251

In [49]: print('Utilizando o coeficiente de SPEARMAN para validar se existe Correlação Linear - 13M1S1')
         coef, p = stats.spearmanr(df_mes['QUANTIDADE'], df_mes['VL_VENDA_TOT'])
         print('Coeficiente de correlação SPEARMAN com QUANTIDADE : {}'.format(coef))
         print('p-valor: {}'.format(p))

         Utilizando o coeficiente de SPEARMAN para validar se existe Correlação Linear - 13M1S1
         Coeficiente de correlação SPEARMAN com QUANTIDADE : 0.9696981361886025
         p-valor: 1.9954706640894314e-38
```

Fonte: Resultados da Pesquisa

Fica evidenciado que para os atributos "QUANTIDADE" e "VL_VENDA_TOT" existe uma forte correlação, cujo parâmetro *p*-valor retornado foi de 1.9954^{e-38}, confirmando que a hipótese alternativa (H_a) foi aceita, com um coeficiente de correlação linear diretamente proporcional de 0.9696.

O coeficiente de correlação pode variar de -1 a +1, quanto mais próximo de 0 (zero) significa que não possui correlação, por outro lado, quanto mais próximo dos extremos -1 ou +1, indica a intensidade da correlação e sua direção, podendo ser inversamente proporcional caso negativo ou diretamente proporcional caso positivo. Baba *et al*. (2014) relatam que os índices de correlação podem ser interpretados conforme a Tabela 20.

Tabela 20 - Interpretação dos índices de correlação

Valor de σ (+ ou -)	Interpretação
0.00 a 0.19	Correlação muito fraca
0.20 a 0.39	Correlação fraca
0.40 a 0.69	Correlação moderada
0.70 a 0.89	Correlação forte
0.90 a 1.00	Correlação muito forte

Fonte: Baba *et al*. (2014)

Na Figura 37Figura 37, foi aplicado o método de *Pearson* para o produto 0-B051, cujo conjunto de dados possui distribuição normal.

Figura 37 - Correlação Linear de *Pearson* – (produto 0-B051)

```
In [91]: print('Utilizando o coeficiente de PEARSON para validar se existe Correlação Linear - 0-B051')
         coef, p = stats.pearsonr(df_mes['MES'], df_mes['VL_VENDA_TOT'])
         print('Coeficiente de correlação PEARSON com MES : {}'.format(coef))
         print('p-valor: {}'.format(p))

         Utilizando o coeficiente de PEARSON para validar se existe Correlação Linear - 0-B051
         Coeficiente de correlação PEARSON com MES : 0.17476855098891356
         p-valor: 0.11182509357022125

In [92]: coef, p = stats.pearsonr(df_mes['QUANTIDADE'], df_mes['VL_VENDA_TOT'])
         print('Coeficiente de correlação PEARSON com QUANTIDADE : {}'.format(coef))
         print('p-valor: {}'.format(p))

         Coeficiente de correlação PEARSON com QUANTIDADE : 0.948565970244441
         p-valor: 1.0131174737390465e-42
```

Fonte: Resultados da Pesquisa

Ficou evidenciado que não existe correlação linear entre o atributo "MES" e o atributo "VL_VENDA_TOT", uma vez que o parâmetro *p*-valor retornado foi de 0.1118, confirmando que a hipótese nula (H_o) foi aceita, entretanto para os atributos "QUANTIDADE" e "VL_VENDA_TOT" existe uma forte correlação, cujo parâmetro *p*-valor retornado foi de 1.01311^{e-42}, confirmando que a hipótese alternativa (H_a) foi aceita, com um coeficiente de correlação linear diretamente proporcional de 0.9485.

Na Figura 38Figura 38, é apresentado a matriz de correlação dos atributos do conjunto de dados utilizando o método *Pearson*.

Figura 38 - Matriz de correlação com método Pearson (produto 0-B051)

```
In [94]: print('MATRIZ Correlação entre os campos do dataframe - metodo PEARSON')
         correlacoes = df_mes.corr(method='pearson')
         correlacoes

         MATRIZ Correlação entre os campos do dataframe - metodo PEARSON
```

Out[94]:

	MES	IPCA	QUANTIDADE	VL_VENDA_TOT	CMV_TOTAL	PERC_MARGEM	VL_VENDA_UNIT
MES	1.000000	0.019019	0.189545	0.174769	0.184361	-0.061386	-0.030349
IPCA	0.019019	1.000000	0.055522	0.172369	0.147235	0.060285	0.291850
QUANTIDADE	0.189545	0.055522	1.000000	0.948566	0.943766	-0.449159	-0.525719
VL_VENDA_TOT	0.174769	0.172369	0.948566	1.000000	0.963591	-0.349724	-0.304480
CMV_TOTAL	0.184361	0.147235	0.943766	0.963591	1.000000	-0.560303	-0.387880
PERC_MARGEM	-0.061386	0.060285	-0.449159	-0.349724	-0.560303	1.000000	0.624458
VL_VENDA_UNIT	-0.030349	0.291850	-0.525719	-0.304480	-0.387880	0.624458	1.000000

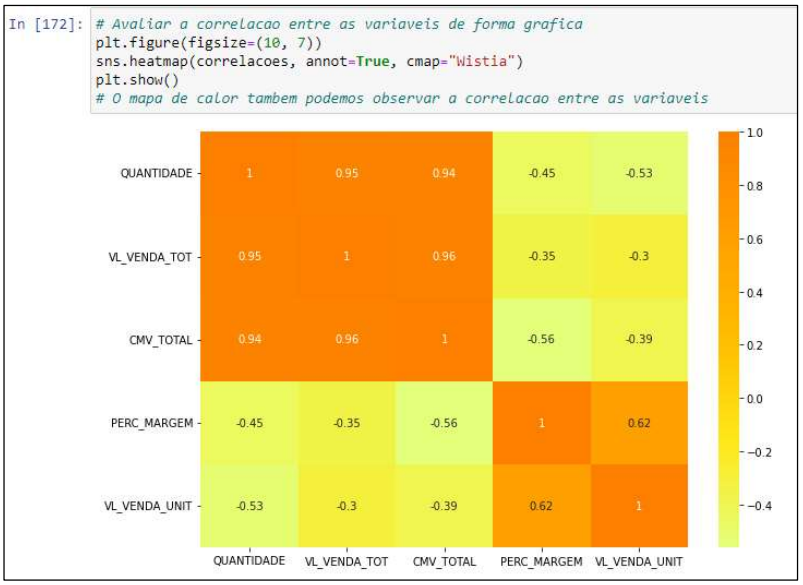
Fonte: Resultados da Pesquisa

Pelos resultados obtidos na Figura 38, ficou evidenciado uma forte correlação linear entre os atributos "CMV_TOTAL" e "VL_VENDA_TOT" com coeficiente de correlação de 0.9635, além de "QUANTIDADE" e "VL_VENDA_TOT" com coeficiente de correlação de 0.9485. Existe uma correlação

moderada entre os atributos "PERC_MARGEM" e "VL_VENDA_UNIT" com um coeficiente de correlação de 0.6244. Os demais atributos apresentam uma fraca correlação. (BABA *et al*., 2014)

Este protótipo também apresenta o mapa de calor (*Heatmap*), para facilitar a visualização dos coeficientes de correlação conforme apresentado na Figura 39.

Figura 39 - Mapa de Calor (*Heatmap*)

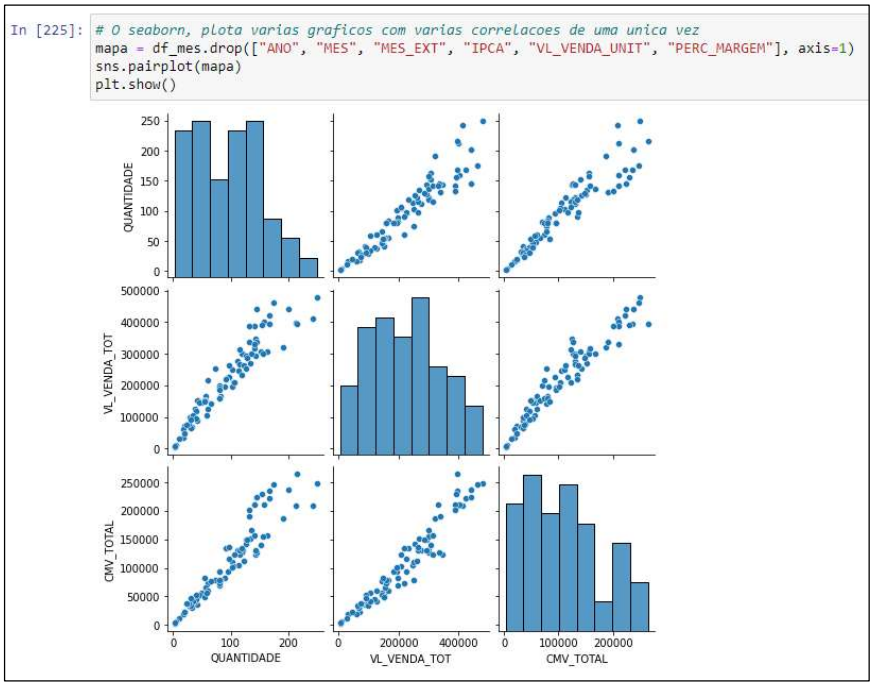


Fonte: Resultados da Pesquisa

No mapa de calor, quanto maior a intensidade da cor, maior é o coeficiente de correlação entre as variáveis, destacando-se os atributos com uma forte correlação linear: "QUANTIDADE", "VL_VENDA_TOT" e "CMV_TOTAL", assim como os atributos "PERC_MARGEM" e "VL_VENDA_UNIT" com correlação moderada.

Por meio da biblioteca *Seaborn*, o protótipo apresenta outro tipo de mapa de correlação, utilizando gráficos de dispersão para representar a matriz de correlação entre as variáveis – Figura 40.

Figura 40 - Matriz de Correlação utilizando *Seaborn*



Fonte: Resultados da Pesquisa

Após executadas todas as etapas acima para os cinco produtos, foram obtidos os coeficientes de correlação por produto apresentados na Tabela 21.

Tabela 21 - Coeficientes de correlação por produto

Produto	Normalidade (p-valor) (Sim; > 0.05) (Não; <= 0.05)		Teste *Shapiro-Wilk*	Teste *Lilliefors*	*PEARSON* (VL_VENDA_TOT)					*SPEARMAN* (VL_VENDA_TOT)				
					MES	QUANT.	IPCA	% MARGEM	CMV TOTAL	MES	QUANT.	IPCA	% MARGEM	CMV TOTAL
13M1S1		Não	2.10683^{e-05}	0.000999						-0.049231	0.969698	0.276440	-0.350306	0.988567
0-0001		Não	0.015922	0.0094575						0.121409	0.818277	0.158418	0.240220	0.927266
0-B051	Sim		0.085142	0.210551	0.174769	0.948566	0.172369	-0.349724	0.963591					
8-K011		Não	3.281^{e-07}	0.000999						0.107835	0.935193	0.039959	0.170447	0.960103
1-B301	Sim		0.293697	0.633170	0.216120	0.998709	-0.193716	0.140028	0.970311					

Fonte: Resultados da Pesquisa

Para os conjuntos de dados analisados, foi evidenciado que os produtos 0-B051 e 1-B301, possuem distribuição normal, uma vez que os resultados do teste de *Shapiro-Wilk* retornaram 0.085142 e 0.293697 respectivamente, ou seja, maior que 0.05. Para os demais produtos, os resultados do teste de *Shapiro-Will* foram inferiores a 0.05, evidenciando que não possuem distribuição normal.

Conforme discutido anteriormente, quando existe normalidade é aplicado o método de *Pearson* para avaliar a correlação entre as variáveis, caso contrário é aplicado o método de *Spearman*.

Ambos os métodos apresentaram uma forte correlação entre os atributos "VL_VENDA_TOT", "QUANTIDADE" e "CMV_TOTAL" conforme retorno demonstrado na Tabela 21, pois os valores retornando são maiores que 0.90 conforme classificação de Baba *et al*. (2014), demonstrados na Tabela 20.

3.5.1.3 Aprendizado do Modelo (Model Learning)

Considerada por Raschka *et al* (2017), uma das fases mais importantes na inteligência computacional, a qual preconiza a necessidade de aplicar um modelo de referência para verificar os resultados alcançados por meio de uma técnica proposta. Esse modelo de referência pode ser extraído de um estudo bibliográfico de determinado segmento com base no estado da arte, ou construído a partir de um conjunto de dados padrão. Em ambos os casos, os resultados obtidos a partir deste modelo de referência estão sendo tratados neste *design* de experimento.

Estabelecido o modelo de referência, faz-se necessário a construção e teste deste modelo com o objetivo de fornecer as melhores soluções para predição. Existem vários métodos disponíveis na linguagem *Python* para resolvermos problemas de classificação e regressão.

Alguns pontos-chaves surgem neste momento, como a necessidade de medir o desempenho que deve indicar claramente o desempenho das técnicas durante a fase de treinamento. Existem diferentes métricas de desempenho conforme relatado na seção 1.3.2, sendo que nesta pesquisa serão aplicadas as métricas *R-Squared (R2)* e RMSE.

Esta pesquisa também utiliza o método de validação cruzada denominado *k-fold*, cujo qual consiste em dividir o conjunto total de dados em *k* subconjuntos mutuamente exclusivos do mesmo tamanho, ou seja, ora o mesmo subconjunto é utilizado para treinamento e ora para teste. Com isso um subconjunto é utilizado para teste e os *k-1* restantes são utilizados para estimação dos parâmetros, fazendo-se o cálculo do desempenho do modelo. Este processo é realizado *k* vezes alternando de forma circular o subconjunto de teste.

Existem dois objetivos principais ao usar CV durante o processo de aprendizagem do modelo. Em primeiro lugar, o CV é utilizado para medir o grau de generalização do modelo durante a fase de treinamento, avaliando o desempenho particular do modelo e estimando o desempenho com dados desconhecidos (MCLACHLAN *et al.*, 2005).

Em segundo lugar, uma comparação é realizada com seis algoritmos que foram treinados nas mesmas condições e com o mesmo conjunto de dados, porém ao invés de executar o experimento apenas uma vez, são realizadas cinco separações "*splits*" e processadas dez execuções "*seeds*" para separar os dados de cada conjunto de dados independentemente.

Nesta pesquisa, para cada uma das dez execuções, cada conjunto de dados original foi separado em cinco divisões aleatórias (70% de treinamento e 30% de teste) e o processo de CV foi realizado apenas com os dados de treinamento, conforme Figura 41.

Figura 41 – Aplicando *k-fold* para separação dos dados

```
# Definicação de funcao para comparar os resultados para 6 algoritimos diferentes.

# create an Empty DataFrame object
df_result_cv = pd.DataFrame()


def ApplyesKfold(x_axis, y_axis, p_seed):
    #Linear Models.
    #print('x_axis -> {}  {}'.format(x_axis, y_axis))

    print('seed {}'.format(p_seed))


    # KFold settings.
    seed = p_seed
    kfold = KFold(n_splits=5, random_state=seed, shuffle=True)

    # Axis
    x = x_axis
    y = y_axis

    # Models instances.
    linearRegression = LinearRegression()
    elasticNet       = ElasticNet()
    ridge            = Ridge()
    lasso            = Lasso()
    svr              = SVR(kernel='linear')
    rf               = RandomForestRegressor()


    # Applyes KFold to models.
    linearRegression_result    = cross_val_score(linearRegression, x, y, cv=kfold, scoring='r2')
    elasticNet_result          = cross_val_score(elasticNet, x, y, cv=kfold, scoring='r2')
    ridge_result               = cross_val_score(ridge, x, y, cv=kfold, scoring='r2')
    lasso_result               = cross_val_score(lasso, x, y, cv=kfold, scoring='r2')
    svr_result                 = cross_val_score(svr, x, y, cv=kfold, scoring='r2')
    rf_result                  = cross_val_score(rf, x, y, cv=kfold, scoring='r2')

    print(linearRegression_result)
```

Fonte: Autor

O resultado do coeficiente de determinação *R-squared* (R^2) do processo de CV é apresentado na Tabela 22, juntamente com a métrica de erro RMSE para o produto 0-0001, evidenciando que o melhor resultado encontrado foi na execução da semente "*seed*" 747, cujo algoritmo LR apresentou o melhor resultado, ou seja, teve um percentual de predição de 96,24% e RMSE 156, embora os algoritmos *Ridge* e *Lasso* tenham apresentado o mesma variação no índice de erro RMSE.

Tabela 22 - Resultado R-Squared (R2) e (RMSE) ao utilizar CV para o produto 0-0001

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE
745	0,95895991	167	0,95666	165	0,959188	166	0,95891	167	0,954245	170	0,933534	214
746	0,95811821	167	0,956089	167	0,958297	166	0,95806	167	0,954599	168	0,931003	203
747	**0,96238317**	**156**	0,958321	161	0,962279	**156**	0,962355	**156**	0,959986	159	0,948782	178

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE
748	0,95923916	160	0,958663	158	0,959759	159	0,959364	160	0,957696	162	0,945595	191
749	0,96086412	159	0,961162	158	0,962051	157	0,961191	158	0,960199	158	0,937479	187
750	0,96084187	162	0,957169	164	0,960806	161	0,960803	162	0,956084	166	0,93307	212
751	0,95988286	158	0,955148	162	0,959627	157	0,959752	157	0,953708	166	0,942767	199
752	0,96151157	162	0,959846	162	0,96178	161	0,961593	162	0,959114	165	0,940192	187
753	0,95949416	157	0,956018	161	0,959569	157	0,959525	157	0,955303	163	0,909688	228
754	0,95460207	167	0,952101	166	0,954681	166	0,954548	167	0,95318	166	0,915489	223

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared (R^2)* do processo de CV é apresentado na Tabela 23, juntamente com a métrica de erro RMSE para o produto 13M1S1, evidenciando que o melhor resultado encontrado foi na execução da *seed* 745, cujo algoritmo *ElasticNet* apresentou o melhor resultado, ou seja, teve um percentual de predição de 97,62% e a menor variação no índice de erro RMSE ocorreu no processo de execução 751 no algoritmo SVR.

Tabela 23 - Resultado R-Squared (R2) e (RMSE) ao utilizar CV para o produto 13M1S1

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE
745	0,97483414	29.462	**0,97618851**	28.336	0,97546852	28.965	0,97483625	29.460	0,97280307	26.136	0,93494024	47.138
746	0,96572020	31.953	0,96867935	29.908	0,96717472	30.819	0,96572314	31.951	0,97148790	26.151	0,93016346	44.396
747	0,96636469	30.022	0,97087334	28.178	0,96850819	29.102	0,96636990	30.020	0,97315880	27.207	0,92106779	48.359
748	0,96829008	31.001	0,96884107	30.475	0,96856818	30.750	0,96829161	31.000	0,97047406	26.299	0,90803553	53.241
749	0,96654648	31.251	0,96727947	30.751	0,96684406	31.062	0,96654689	31.251	0,96924481	26.272	0,89561563	53.779
750	0,97039761	30.028	0,97357840	28.965	0,97190940	29.504	0,97040094	30.026	0,97417736	27.110	0,93344956	46.548
751	0,97174025	28.876	0,97367897	27.238	0,97276141	27.806	0,97174299	28.873	0,97297250	**25.825**	0,93672128	39.860
752	0,96992777	28.511	0,97224119	27.408	0,97103232	28.017	0,96993062	28.510	0,97172332	26.266	0,90562946	43.352
753	0,97108922	27.946	0,97165038	27.906	0,97133234	27.907	0,97109054	27.945	0,97193301	25.939	0,92221801	49.061
754	0,96219268	31.696	0,96543168	30.261	0,96373723	30.993	0,96219642	31.695	0,96851711	26.173	0,92081753	42.835

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared (R^2)* do processo de CV é apresentado na Tabela 24, juntamente com a métrica de erro RMSE para o produto 0-B051, evidenciando que o melhor resultado encontrado foi na execução da *seed* 753, cujo algoritmo *ElasticNet* apresentou o melhor resultado, ou seja, teve um percentual de predição de 97,74% e a menor variação no índice de erro RMSE ocorreu no mesmo algoritmo, porém na *seed* 754.

Tabela 24 - Resultado R-Squared (R2) e (RMSE) ao utilizar CV para o produto 0-B051

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE
745	0,975360356	12.666	0,976125131	12.276	0,975529119	12.588	0,975363128	12.664	0,909638126	24.707	0,935963028	22.130
746	0,973727566	12.828	0,973993921	12.758	0,973766088	12.819	0,973728477	12.828	0,897363585	26.416	0,939717305	22.967
747	0,975118857	13.133	0,975475354	13.027	0,97518797	13.114	0,97512002	13.133	0,917379219	24.910	0,931919933	22.678
748	0,964763392	13.523	0,965814609	13.367	0,964943822	13.491	0,9647662	13.523	0,918237808	24.632	0,934885553	23.963
749	0,97342435	13.208	0,973680042	13.031	0,973481269	13.176	0,973425791	13.207	0,902403851	26.505	0,937810881	23.112
750	0,973187927	13.203	0,974274472	12.829	0,97346436	13.116	0,973191024	13.202	0,905038809	25.028	0,937055021	22.964
751	0,969899546	12.584	0,9707074	12.387	0,970089878	12.548	0,969902174	12.583	0,886273184	24.694	0,913455661	23.626
752	0,971853286	13.263	0,973450351	12.734	0,972241676	13.126	0,971857724	13.262	0,904252011	25.103	0,931706448	23.175
753	0,976286158	12.777	**0,977398194**	12.430	0,976526707	12.702	0,97628893	12.777	0,913718462	25.080	0,940198932	22.875
754	0,975498375	12.183	0,975650722	**12.090**	0,975540536	12.164	0,975499582	12.182	0,909758096	25.011	0,941789337	22.786

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared (R^2)* do processo de CV é apresentado na Tabela 25, juntamente com a métrica de erro RMSE para o produto 8-K011, evidenciando que o melhor resultado encontrado foi na execução da *seed* 747, cujo algoritmo *ElasticNet* apresentou o melhor resultado, ou seja, teve um percentual de predição de 96,22% e a menor variação no índice de erro RMSE ocorreu no processo de execução 750 no algoritmo SVR.

Tabela 25 - Resultado R-Squared (R2) e (RMSE) ao utilizar CV para o produto 8-K011

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE	*R-squared (R^2)*	RMSE
745	0,95566069	468	0,95655498	459	0,95583868	467	0,95563712	468	0,949256023	428	0,94633459	499
746	0,95831784	453	0,95841310	453	0,95834968	453	0,95828961	453	0,952149989	422	0,94099450	493
747	0,96182522	463	**0,96221924**	458	0,96191648	462	0,96182254	463	0,955231644	421	0,93899353	517
748	0,9588000	468	0,95842707	467	0,95877271	468	0,95877392	468	0,954445383	426	0,93931143	529
749	0,9587247	472	0,95917792	470	0,95883248	471	0,95873122	472	0,953205966	426	0,93521342	536
750	0,95943187	464	0,96130127	454	0,95975898	462	0,95948875	464	0,955304256	**418**	0,94063521	495
751	0,95936121	457	0,95944522	457	0,95941194	456	0,95936251	457	0,954264574	422	0,93354479	573
752	0,95970849	467	0,96012494	463	0,95980923	466	0,95971807	467	0,953521848	424	0,94379817	508
753	0,96165541	460	0,96214367	456	0,96174696	459	0,96166457	460	0,957833541	423	0,94258984	504
754	0,95963671	470	0,95988930	466	0,95969606	469	0,95964113	470	0,954540482	425	0,94283252	525

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared (R^2)* do processo de CV é apresentado na Tabela 26, juntamente com a métrica de erro RMSE para o produto 1-B301, evidenciando que o melhor resultado encontrado foi na execução da *seed* 748, cujo algoritmo *ElasticNet* apresentou o melhor

resultado, ou seja, teve um percentual de predição de 99,04% e a menor variação no índice de erro RMSE ocorreu no processo de execução 747 do algoritmo *Ridge* com 1.054.

Tabela 26- Resultado R-Squared (R2) e (RMSE) ao utilizar CV para o produto 1-B301

Seed	LR		ElasticNet		Ridge		Lasso		SVR		RF	
	R-squared (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE	*R-squared* (R^2)	RMSE
745	0,98498893	1.312	0,98264322	1.360	0,98454172	1.289	0,98495866	1.311	0,78993953	4.514	0,86420163	4.175
746	0,98494008	1.482	0,98304069	1.311	0,98575709	1.355	0,98497641	1.478	0,83020012	4.229	0,88641602	3.436
747	0,98532830	1.136	0,98348094	1.178	0,98580874	**1.054**	0,98531115	1.135	0,26014219	5.402	0,78178009	3.699
748	0,98640285	1.314	**0,99043256**	1.154	0,98957381	1.195	0,98644870	1.313	0,74462962	5.161	0,89789651	3.675
749	0,97568618	1.275	0,98211885	1.196	0,97687093	1.237	0,97567419	1.275	0,61829933	4.503	0,88001669	3.472
750	0,98093858	1.372	0,98156249	1.332	0,98136595	1.313	0,98092410	1.372	0,74734417	4.542	0,86382571	3.644
751	0,98378083	1.314	0,98455923	1.210	0,98402924	1.231	0,98376470	1.313	0,09723940	4.904	0,86413130	3.236
752	0,98960606	1.326	0,98730555	1.316	0,98962750	1.275	0,98960364	1.325	0,87565766	4.375	0,92713402	3.275
753	0,98884315	1.408	0,98949123	1.310	0,98944660	1.334	0,98883128	1.408	0,86420023	4.180	0,90141079	3.578
754	0,98911951	1.177	0,98678967	1.161	0,98928153	1.108	0,98912956	1.176	0,86459133	3.961	0,88481627	4.174

Fonte: Resultados da Pesquisa

Uma vez que o modelo foi finalmente treinado, 30% dos dados originais que permaneceram absolutamente desconhecidos foram usados para fins de validação. No entanto, o ponto chave neste caso é que a validação cruzada até o momento foi realizada somente com dados conhecidos (*Internal Cross-Validation*), utilizado somente para selecionar a melhor combinação de parâmetros para cada técnica.

O referido processo de CV foi aprimorado, com a formalização de um processo de CV externo (*External Cross-Validation*), para garantir que os modelos não fossem super treinados evitando a ocorrência de *Overfitting* ou sub treinados ocorrendo *Underfitting*, evidenciando que a melhor pontuação de desempenho não fosse encontrada para uma combinação específica de parâmetros.

Ao fazê-lo, evitou-se um *design* de experimento falho, devido à distribuição particular das observações no conjunto de dados. Assim, dentro de cada execução *seed*, foi mantido o processo interno de CV para a seleção dos parâmetros propostos, mas foi assegurado que um conjunto externo de dados fosse desconhecido para o modelo, evitando um possível viés na seleção do modelo.

Na Tabela 27, são evidenciados os resultados dos modelos com o maior coeficiente de determinação *R-squared* (R^2) e menor RMSE por produto utilizando CV interna e externa.

Tabela 27 – Resultados *R-squared* (R^2) e RMSE utilizando CV Interna e Externa

	Internal Cross-Validation							**External Cross-Validation**					
Produto		**LR**	**Elastic Net**	**Ridge**	**Lasso**	**SVR**	**RF**	**LR**	**Elastic Net**	**Ridge**	**Lasso**	**SVR**	**RF**
13M1S1	*Seed*		**745**			**751**			**748**			**748**	
	R^2	0,9748341	**0,9761885**	0,9754685	0,97483	0,9741773	0,9367212	0,96048	**0,9709992**	0,9702803	0,96048899	0,9653728	0,8743046
	RMSE	27.946	27.238	27.806	27.945	**25.825**	39.860	35.200	26.736	28.287	35.196	**25.435**	67.169
0-0001	*Seed*	**747**							**753**				
	R^2	**0,96238**	0,96116	0,96227	0,96235	0,96019	0,948782	0,95432	**0,95768**	0,95539	0,95423	0,95122	0,87607
	RMSE	**156**	158	156	156	158	178	152	**149**	152	153	159	238
0-B051	*Seed*		**753/754**					**751**	**751**				
	R^2	0,9762	**0,9773**	0,9765	0,9762	0,9182	0,9417	**0,9844**	0,9841	0,9843	0,9844	0,9486	0,9158
	RMSE	12.183	**12.090**	12.164	12.182	24.632	22.130	10.398	**10.284**	10.366	10.398	17.786	24.025
8-K011	*Seed*		**747**			**750**			**750**	**750**			
	R^2	0,9618	**0,9622**	0,9619	0,9618	0,9578	0,9463	0,9568	0,9563	**0,9569**	0,9568	0,9421	0,8984
	RMSE	453	453	453	453	**418**	493	414	**408**	412	414	432	641
1-B301	*Seed*		**748**	**747**							**753/751**		
	R^2	0,9896	**0,9904**	0,9896	0,9896	0,8756	0,9271	0,9738	0,9871	0,9923	**0,9949**	0,9596	0,7151
	RMSE	1.136	1.154	**1.054**	1.135	3.961	3.236	1.310	1.605	1.176	**873**	3.062	8.085

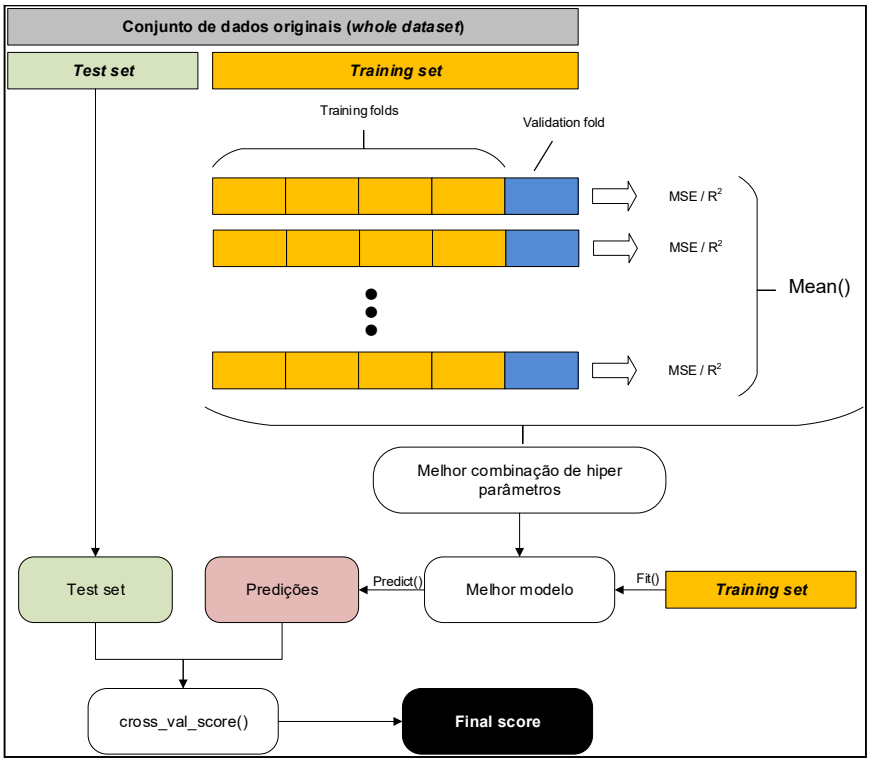
Fonte: Resultados da Pesquisa

A medida final de desempenho relatada é, neste caso, a média dos valores computados para o as execuções "*seeds*", para o coeficiente de determinação *R-squared* (R^2) e a métrica de erro RMSE utilizado a CV interna, todos computados e demonstrados nas Tabelas 22, 23, 24, 25 e 26.

Porém ficou evidenciado que o algoritmo ElasticNet apresentou melhor desempenho quando aplicado com a CV externa, ou seja, quando aplicado aos 30% dos dados desconhecidos. Conforme relatado por Zou *et al.* (2005) na seção 1.2.1, este modelo se beneficia quando o número de preditores é maior que o número de observações.

O processo mostrado na Figura 42, é repetido dez vezes para cada tipo de algoritmo em cada produto.

Figura 42 - Utilizando k-fold com CV interna e externa



Fonte: Autor

Nesta pesquisa, foi modificada a proposta inicial feita por Tsiliki *et al.* (2015a), adicionando a etapa de validação cruzada externa (*External Cross-Validation*), a fim de garantir que, em cada uma das dobras (*k-fold*), fosse realizada uma validação cruzada externa para selecionar o melhor conjunto de parâmetros, diminuindo a ocorrência de viés *Overfitting* ou *Underfitting*, uma vez que dados desconhecidos foram utilizados para validação externa em cada *k-fold*.

Os materiais suplementares disponíveis no Apêndice B, incluem o código fonte em *Python* para aplicação da validação cruzada interna e externa utilizando *k-fold*, assim como o log de execução.

Redução da dimensionalidade

A dimensionalidade dos dados deve ser levada em consideração. Quanto maior a dimensionalidade dos dados de entrada, maior é o número de exemplos necessários para o aprendizado. Além disso, segundo Saeys *et al*. (2007), as técnicas de redução da dimensionalidade são geralmente significativas para fornecer o melhor modelo possível.

Assim, essas técnicas permitem uma redução da complexidade do modelo gerado, além disso, implicam também na redução do tempo necessário para treinamento e melhora a capacidade geral do sistema (DONOHO, 2000).

Seis modelos de regressão linear disponíveis na linguagem *Python*, foram empregados neste delineamento de pesquisa. Linear Regression (FREEDMAN, 2009), Lasso Regression (SANTOSA e SYMES, 1986; TIBSHIRANI, 1996), ElasticNet Regression (ZOU et al., 2005), Random Forest (BREIMAN, 2001), Support Vector Regression - SVR (WITTEN et al. 2011) e Ridge Regression (HILT e SEEGRIST, 1977).

3.5.1.4 Melhor seleção do modelo (Best Model Selection)

Na seção anterior foi destacado que existem algumas medidas conhecidas para medir o desempenho de um modelo de regressão. Isso não significa que seja possível comparar diferentes algoritmos utilizados nas mesmas condições, usando o mesmo conjunto de dados com apenas uma corrida "execução" e uma taxa de erro. Neste ponto, cada técnica deve ser executada várias vezes para garantir que os resultados não sejam tendenciosos devido à distribuição das observações dentro dos dados ou que o número de corridas internas de CV sejam realizadas a fim de encontrar a melhor combinação de parâmetros e não direcione para um viés nos resultados (há uma alta probabilidade de que uma boa combinação de parâmetros/escore de desempenho seja encontrada para uma distribuição particular de observações quando o número de corridas experimentais de CV aumenta).

Com os resultados obtidos por essas técnicas e para determinar se o desempenho de uma determinada técnica é estatisticamente melhor do que as demais, é necessário um teste de hipótese nula. Além disso, para que um teste paramétrico ou não paramétrico pudesse ser utilizado, algumas condições necessárias devem ser verificadas: normalidade, independência e heteroscedasticidade (GARCÍA *et al*., 2010).

Note-se que essas suposições não se referem ao conjunto de dados utilizados como entrada para treinamento dos modelos, mas à distribuição do desempenho dos modelos. Na estatística, um evento é independente de outro se o fato de um ocorrer não modificar a probabilidade dos outros. Assim, na inteligência computacional, diferentes conjuntos de algoritmos, onde as sementes iniciais, em inglês *seeds*, são usadas aleatoriamente para a separação de dados em treinamento e teste, as quais cumprem a condição de independência.

A normalidade é considerada o comportamento de uma observação que segue uma distribuição normal ou gaussiana; para verificar esta condição, existem diferentes testes, como Kolmogorov-Smirnov ou Shapiro-Wilk (SHAPIRO e WILK, 1965). Finalmente, a violação da hipótese de igualdade de variâncias, ou heteroscedasticidade deve ser verificada usando, por exemplo, os testes de Levene ou Bartlett (BARTLETT, 1937).

Para uma comparação técnica independente, como parte deste *design* de experimento, deve ser aplicado um teste estatístico adequado, com base na distribuição estatística do desempenho observado. A maioria das comparações de inteligência computacional na literatura apenas aplicam um teste *T* usando as pontuações de desempenho para verificar se uma técnica é significativamente melhor que as outras (GARCÍA *et al*., 2010).

Em alguns casos, a distribuição dos dados não atende aos requisitos de teste paramétrico, consequentemente é necessário o emprego do teste não paramétrico. Apesar de um teste paramétrico ser mais objetivo conforme relatam García *et al.* (2010), ele não deve ser usado quando as condições de independência, normalidade e homocedasticidade não são totalmente atendidas. Em vez disso, é melhor empregar um teste não paramétrico, pois foi projetado especificamente para estes casos e o resultado será mais preciso e ajustado às características intrínsecas dos dados. Dependendo do número específico de técnicas utilizadas na comparação, deve ser aplicado um teste estatístico diferente, alguns dos testes utilizados em regressão linear são listados na Tabela 28.

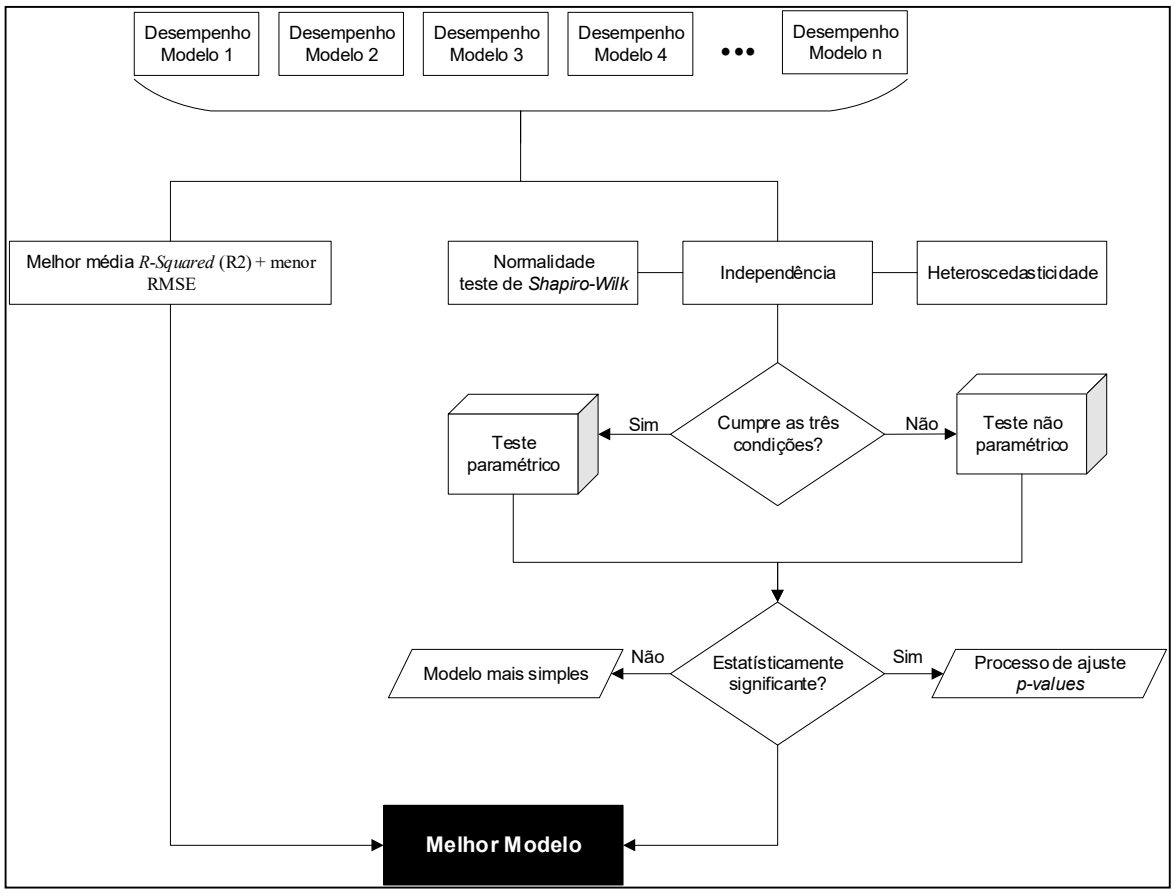
Tabela 28 - Testes estatísticos paramétricos e não paramétricos

Grupo de amostras	Paramétrico	Não Paramétrico
n = 2	Teste T	Wilcoxon
n > 2	Teste de Breusch-Pagan Teste de Bartlett	Friedman, Quade

Fonte: Resultados da Pesquisa

A Figura 43, apresenta a proposta utilizada neste estudo para selecionar o melhor modelo de acordo com a média *R-Squared* (R2), métrica RMSE e adicionalmente os critérios estatísticos.

Figura 43 - Critérios para seleção do melhor modelo



Fonte: Autor

Nesta última fase, caso os resultados não sejam estatisticamente significativos, o modelo mais simples será escolhido dentre os modelos vencedores do teste de hipóteses nulas, seja em termos de complexidade ou tempo de execução.

Em um sentido mais rigoroso, também pode-se realizar um novo teste de hipótese nula usando apenas os resultados dos modelos vencedores, mas neste caso outra medida de desempenho deve ser empregada. Pode ser utilizado o número de recursos *"features*", caso um processo de seleção de recursos seja utilizado. Portanto, pode-se concluir que entre os modelos vencedores inicialmente selecionados, há uma diferença estatisticamente significativa de acordo com este novo critério de desempenho, segundo Fernandez-Lozano *et al.* (2016).

3.5.1.5 Análise dos modelos

Seis modelos de regressão linear disponíveis na linguagem *Python*, foram aplicados no conjunto de dados mencionados na Tabela 16, como ponto de partida para comparar o método proposto em Tsiliki *et al*. (2015a).

A Tabela 29 mostra os resultados dos coeficientes de determinação *R-squared* (R^2), para cada conjunto de dados (*Datasets*), utilizando os seis modelos de regressão linear mencionados na seção 1.4.1. Para cada produto, é evidenciado em negrito o coeficiente com o melhor coeficiente de determinação *R-squared* (R^2). Todos os experimentos foram executados com *Python* 3.9

O resultado do coeficiente de determinação *R-squared* (R^2) segue as mesmas condições de Tsiliki *et al*. (2015a), ou seja, com 5 divisões (*splits*) de conjunto de dados diferentes, usando validação cruzada (CV) e aplicando 10 execuções com o melhor modelo e randomizando a variável de resposta para garantir que nenhum viés seja incluído na análise (TROPSHA, 2010).

Tabela 29 - Resultado R-Squared (R2) para os *Datasets* com 10 *splits*

	13M1S1	**0-0001**	**0-B051**	**8-K011**	**1-B301**
LR	0,97483414	**0,962383173**	**0,984413083**	0,961825224	0,973859010
ElasticNET	**0,97618851**	0,961161807	0,984122121	**0,962219246**	0,987147349
Ridge	0,97546852	0,962279379	0,984391965	0,961916488	0,992325200
Lasso	0,97483625	0,962354569	0,984411107	0,961822542	**0,994970684**
SVR	0,97417736	0,960199150	0,948610677	0,957833541	0,959664823
RF	0,93672128	0,948782046	0,915828492	0,946334591	0,715105657

Fonte: Resultados da Pesquisa

Normalidade

A Tabela 21 na seção 3.5.1.2 evidencia que os produtos 0-B051 e 1-B301 possuem distribuição normal e, portanto, é aplicado o coeficiente de correlação *Pearson*, já para os demais produtos o teste de *Shapiro-Wilk* corroborado pelo teste *Lilliefors* retornaram *p-value* inferior 0.05 indicando que não possuem distribuição normal, aplicando-se neste caso o coeficiente de correlação *Spearman*, sendo que para todos os produtos constatou-se uma forte correlação das variáveis independentes "QUANTIDADE" e "CMV_TOTAL" com a variável dependente "VL_VENDA_UNIT", além de moderada correlação com a variável exógena "IPCA".

Independência

Algumas premissas para regressão linear são necessárias na construção de um modelo que consiga fornecer bons resultados com os dados, desta forma é de suma importância validar estas premissas no conjunto de dados fornecido (YALÇIN *et al.*, 2021).

- O relacionamento entre a variável dependente e independente devem ser lineares
- Não deve haver correlação entre as variáveis independentes
- Os resíduos devem ter variância constante (homocedasticidade)
- Os resíduos não devem apresentar autocorrelação

Quando as variáveis independentes são correlacionadas entre si, ocorre a multicolinearidade. Isso leva ao desenvolvimento de um modelo com coeficientes que possuem valores que dependem da presença de outras variáveis, ou seja, cria-se um modelo que mudará drasticamente se uma variável independente for removida, portanto um modelo impreciso (YALÇIN *et al.*, 2021).

Desta forma, ao realizar o processo de *feature engineer*, no qual as variáveis dependentes e independentes são selecionadas, deve-se selecionar as variáveis independentes que possuem uma forte correlação com a variável dependente (alvo) e que não tenham correlação com as outras variáveis independentes ou que tenham uma correlação fraca, conforme visto na seção 3.5.1.2

Heteroscedasticidade

A heterocedasticidade, em estatística, ocorre quando os erros não são constantes ao longo de toda a amostra. O termo é contrário a homocedasticidade, ou seja, em modelos de regressão linear diz que há heterocedasticidade quando a variância dos erros não é a mesma em todas as observações feitas.

A homogeneidade de variâncias caracteriza os grupos de dados de forma a direcioná-los aos tratamentos estatísticos mais convenientes e, para muitos casos, é ponto de partida para se obter resultados estatisticamente válidos e significativos. Existem diversos testes de variâncias destinados à verificação da homogeneidade de dados. Para compreender seus princípios, consideram-se grupos de dados obtidos de um determinado estudo e formulam-se as seguintes hipóteses:

H_o: $\sigma_1^2 = \sigma_2^2 = \sigma_n^2$

H_a: pelo menos um dos $\sigma^{2,}$ é diferente

Onde σ_i^2 = represente a variância de cada um dos grupos de dados. Os testes de variância verificam, por meio de cálculos pertinentes, qual das hipóteses acima é verdadeira. Comparam grupos grandes e pequenos, de tamanhos iguais ou diferentes, e que obedeçam à distribuição normal ou não. Três testes são utilizados nesta pesquisa: teste de Breusch-Pagan, teste de Bartlett e o teste *T*.

Para análise dos resíduos, este protótipo utiliza a biblioteca *statsmodels*, conforme declaração destacada na Figura 44.

Figura 44 – Utilizando *statsmodels* para análise dos resíduos

```
In [844]: import statsmodels.formula.api as smf
          import statsmodels.stats.api as sms

          # Criação do modelo
          # ols = funcao do modelo de regressao linear simples
          # ' v1 ~ v2 + vn = v1 = variavel dependente, v2, vn = variavel independente
          regressao = smf.ols('VL_VENDA_TOT ~ QUANTIDADE', data=df_mes).fit()
          #regressao = smf.ols('VL_VENDA_TOT ~ QUANTIDADE + IPCA + PERC_MARGEM', data=df_mes).fit()
```

Fonte: Autor

No modelo de regressão linear, cuja forma $Y = X\beta + e$, os elementos e_i, do vetor e são as diferenças entre os valores observados y_i e aqueles esperados pelo modelo. Essa diferença é chamada de resíduo e considera-se como pressuposto do modelo que os resíduos sejam independentes e que tenham distribuição normal, sendo que a distribuição esperada pode mudar conforme o modelo, além disso a normalidade deve ser verificada nos resíduos e não na variável resposta Y.

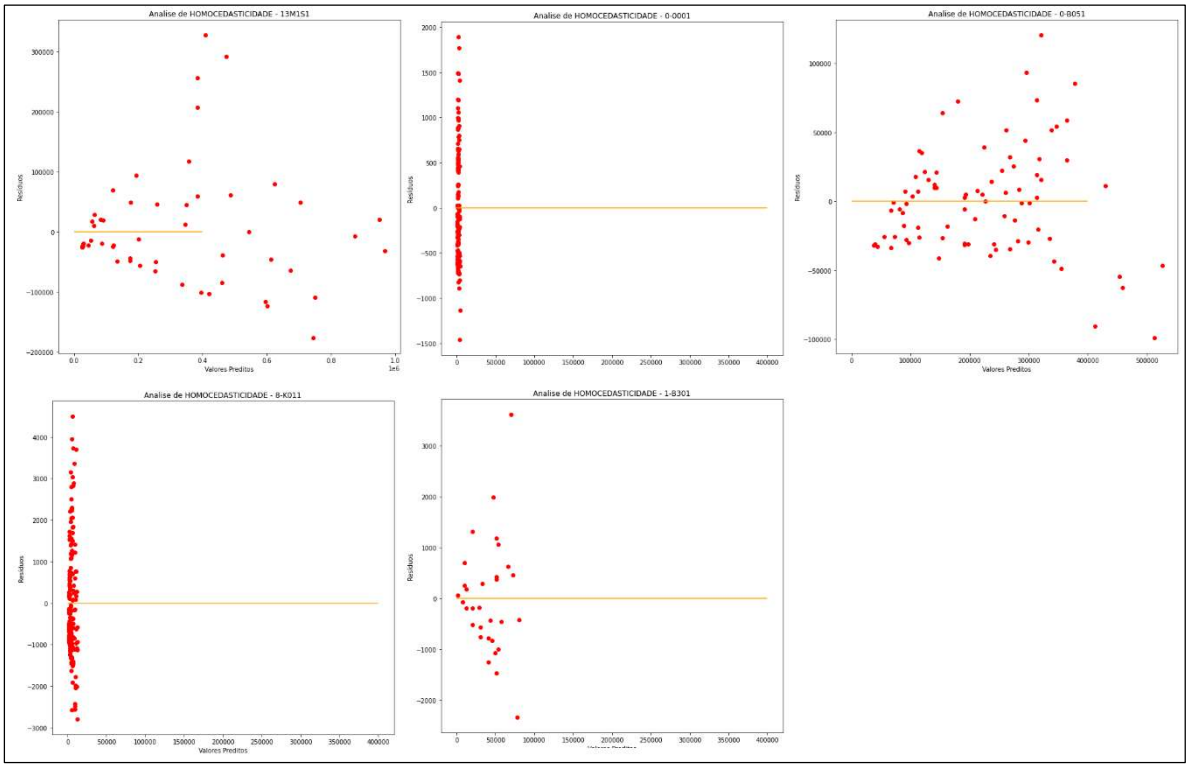
Os resíduos indicam a variação natural dos dados, um fator aleatório (ou não) que o modelo não capturou. Se as pressuposições do modelo são violadas, a análise será levada a resultados duvidosos e não confiáveis para inferências. Essas falhas do modelo nos pressupostos podem ser oriundas de diversos fatores como não linearidade, não-normalidade, heterocedasticidade, não-independência e isso pode ser causado por pontos atípicos (observações discrepantes), que podem influenciar, ou não, no ajuste do modelo.

Considerando as hipóteses demonstradas na Tabela 18, é utilizado o teste de *Shapiro-Wilk* para avaliar se a distribuição dos resíduos é normal ou não. Além disso, o gráfico *quantile plot* (qq-plot), utilizado na seção 3.5.1.2 para avaliar a normalidade do conjunto de dados, também é aplicado na análise dos resíduos.

Uma vez evidenciado que os resíduos possuem distribuição normal, é aplicado o teste de homocedasticidade por meio da biblioteca *matplotlib.pylot*, na qual avalia se os resíduos apresentam distribuição de erro constante, ou seja, uma distribuição homogênea. Caso os resíduos tenham uma

distribuição constante, significa que os resíduos passaram no teste de homocedasticidade, caso contrário o resultado indica a presença de heterocedasticidade. Na Figura 45, é mostrado o resultado da análise para cada um dos produtos analisados.

Figura 45 – Análise de homocedasticidade utilizando *matplotlib.pylot*



Fonte: Resultados da Pesquisa

Por meio da linha de referência laranja y=0, é possível avaliar como os pontos estão distribuídos, sendo possível avaliar visualmente se a distribuição de erros é constante ou não, atendendo desta forma o teste de homocedasticidade, porém somente se os pontos vermelhos formarem um retângulo perfeito com a linha laranja no centro do gráfico é possível ter uma indicação visual que os resíduos possuem homocedasticidade, contudo, como esta análise se torna subjetiva, é aplicado o teste *Breusch-Pagan*, cujo qual utiliza as hipóteses declaradas na Tabela 30.

Tabela 30 – Hipóteses para o teste de *Breusch-Pagan*

Hipótese	**Tipo da Distribuição**	***p*-valor**
H_o	Existe homocedasticidade	> 0.05
H_a	Não existe homocedasticidade	<= 0.05

Fonte: Resultados da Pesquisa

Teste de Breusch-Pagan

O teste *Breusch-Pagan* permite testar a hipótese de homocedasticidade do termo de erro de um modelo de regressão linear. Foi proposto por Trevor Breusch e Adrian Pagan em um artigo publicado em 1979 na revista Econométrica, o teste procura determinar a natureza da variância do termo do erro: se a variância é constante, então o resultado dos resíduos possui homocedasticidade, por outro lado, se variar, considera-se que a variância dos resíduos apresenta heterocedasticidade. (BREUSCH, 1979).

Na Figura 46, é demonstrado a utilização da biblioteca *statsmodels.stats.api*, e obtenção do resultado *p-valor* para compará-lo com as hipóteses descritas na Tabela 30 e avaliar se os resíduos apresentam erros constantes ou não.

Figura 46 - Teste de *Breusch-Pagan*

```
In [852]: from statsmodels.compat import lzip
          import statsmodels.stats.api as sms

In [859]: estatistica, p, f, fp = sms.het_breuschpagan(regressao.resid, regressao.model.exog)
          print('Estatística de teste Breusch-Pagan : {}'.format(estatistica))
          print('p-valor: {}'.format(p))
          print('f-valor: {}'.format(f))
          print('f_p-valor: {}'.format(fp))

          Estatística de teste Breusch-Pagan : 3.6420049290232814
          p-valor: 0.05633905065613974
          f-valor: 3.7444791493543614
          f_p-valor: 0.057699975412221556
```

Fonte: Autor

Na Tabela 31, são apresentados os resultados de *p-valor*, obtidos por meio do teste de *Breusch-Pagan* para cada item demonstrado no teste de homocedasticidade contido na Figura 46.

Tabela 31 – Resultado do teste de *Breusch-Pagan* por produto

Produto	Resultado *p-valor*	Hipótese
13M1S1	0.05633	H_o
0-0001	0.00597	H_a
0-B051	4.5575^{e-05}	H_a
8-K011	3.7762^{e-05}	H_a
1-B301	0.0223	H_a

Fonte: Resultados da Pesquisa

Observa-se, dos cinco produtos testados, somente o produto 13M1S1 aceitou a hipótese nula, ou seja, os demais produtos rejeitaram a hipótese nula, desta forma apresentaram indícios de heterocedasticidade, sendo necessário a aplicação do teste de Bartlett para corroborar esta suspeita.

Teste de Bartlett

O teste de igualdade de variâncias de Bartlett é um teste paramétrico, usado para avaliar se *k* amostras independentes vêm de populações com a mesma variância, ou seja, possui homocedasticidade. Este teste

entra em colapso completamente assim que alguma ocorrência se desvia, mesmo que ligeiramente, da distribuição gaussiana (PHONN *et al*., 2013).

De acordo com Brown *et al.* (1974), um dos testes de variância mais utilizados é o teste de Bartlett.

Phoon *et al.* (2013) afirmam que o teste de Bartlett para homogeneidade de variâncias é uma ferramenta eficiente somente se as variáveis possuem distribuição normal. Quando a suposição de normalidade é violada, o tamanho do teste pode ser muito maior do que o nível de significância fixado.

Considerando o nível de significância *alpha* de 0.05, o resultado de *p-value* segue o mesmo critério para avaliação das hipóteses nula e alternativa apresentadas na Tabela 30. Na Figura 47, é demonstrado a utilização da biblioteca *scipy.stats* para utilização do teste Bartlett, cujo qual, foi aplicado para todos os produtos.

Figura 47 - Teste de igualdade de variâncias de Bartlett – produto 13M1S1

```
In [969]: import scipy.stats as stats

          # data separado em 12 grupos representando as vendas por mes
          data = [df_mes[df_mes['MES'] == 1]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 2]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 3]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 4]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 5]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 6]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 7]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 8]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 9]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 10]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 11]['VL_VENDA_TOT'],
                  df_mes[df_mes['MES'] == 12]['VL_VENDA_TOT']]

          # performing Bartlett's test
          test_statistic, p_value = stats.bartlett(data[0],data[1],data[2],data[3],data[4],data[5],data[6],data[7],data[8],data[9],data[10]

          print(test_statistic, p_value)

          14.217442847956354 0.22119416728812694
```

Fonte: Autor

Pelo teste de Bartlett fica evidenciado na Tabela 32 que a hipótese nula foi aceita para os primeiros quatro produtos, ou seja, possui homocedasticidade quando comparado sua variância em grupos, neste teste foram considerados doze grupos independentes, representando os meses de faturamento dos respectivos anos contidos no *dataset*.

Tabela 32 – Resultados do teste de Bartlett

Produto	Resultado *p-valor*	Hipótese
13M1S1	0.22119	H_o
0-0001	0.88654	H_o
0-B051	0.97231	H_o
8-K011	0.73867	H_o
1-B301	nan	H_a

Fonte: Resultados da Pesquisa

Porém, a hipótese nula foi rejeitada para o produto 1-B301, desta forma é aplicado o teste *T*, trata-se de um teste paramétrico usado para testar uma diferença estatisticamente significativa nas médias entre dois grupos. Tal como acontece com todos os testes paramétricos, existem certas condições que precisam ser atendidas para que os resultados do teste sejam considerados confiáveis: I) As distribuições da população são normais; II) As amostras têm variações iguais; III) As duas amostras são independentes (RASCH *et al*., 2017).

Teste t

O teste *T* de *Student* ou somente teste *T* é um teste de hipótese que usa conceitos estatísticos para rejeitar ou não uma hipótese nula quando a estatística de teste (t) segue uma distribuição t de *Student*. Essa premissa é normalmente usada quando a estatística de teste, na verdade, segue uma distribuição normal, mas a variância da população σ^2 é desconhecida. Nesse caso, é usada a variância amostral s^2 e, com este ajuste, a estatística de teste passa a seguir uma distribuição *T* de *Student*.

Se forem feitas inúmeras amostras de tamanho *n* a partir da mesma população e se fossem tiradas as médias de uma variável dessa população que possui uma distribuição normal, a distribuição dessas inúmeras médias seguiria uma distribuição *T* de *Student* (RASCH *et al*., 2017).

O teste *T* consiste em formular uma hipótese nula e consequentemente uma hipótese alternativa, calcular o valor de *T* e aplicá-lo à função densidade de probabilidade da distribuição *T* de *Student* medindo o tamanho da área abaixo dessa função para valores maiores ou iguais a *T*. Essa área representa a probabilidade de a média dessas amostras em questão terem apresentado os valores observados. Se a probabilidade desse resultado ter ocorrido for muito pequena, podemos concluir que o resultado observado é estatisticamente relevante. Essa probabilidade também é chamada de *p*-valor ou valor p. Consequentemente, o nível de confiança α é igual a (1 - *p*-valor).

Neste estudo é usado um "ponto de corte" de 5% para o nível de confiança para definir se a hipótese nula deve ser rejeitada ou não. Se o *p*-valor for menor que esse "ponto de corte", a hipótese nula é rejeitada. Caso contrário, a hipótese nula é aceita. Desta forma, caso a área abaixo da função densidade de probabilidade da distribuição *T* de *Student* seja menor que 5%, pode-se afirmar que a hipótese nula é rejeitada com nível de confiança de 95%, conforme Tabela 33.

Tabela 33 - Hipóteses para o teste *T*

Hipótese	**Tipo de variância das amostras**	***p*-valor**
H_o	Idêntica	> 0.05

Hipótese	Tipo de variância das amostras	*p*-valor
H_a	Diferente	<= 0.05

Fonte: Resultados da Pesquisa

Para avaliar se os resíduos do produto 1-B301 possuem variância, o *dataset* foi divido em dois semestres, sendo o primeiro semestre correspondente aos valores de vendas dos meses de janeiro a junho e o segundo semestre referente aos valores de vendas dos meses de julho a dezembro. Para aplicar o teste *T* foram utilizadas as bibliotecas *researchpy* e *scipy.stats*, conforme Figura 48 e Figura 49.

Figura 48 – Aplicando teste *T* por meio da biblioteca *researchpy*

```
In [1207]: #!pip install researchpy
           import researchpy as rp

           #print(df_mes.info())

           #INDEPDENT T-TEST USING RESEARCHPY

           # The method returns 2 data frames, one that contains the summary statistical information and the other that contains the
           # statistical test information. If the returned data frames are not stored as a Python object then the output will be less
           # clean than it can be since it will be displayed as a tuple - see below

In [1185]: summary, results = rp.ttest(group1= df_mes['VL_VENDA_TOT'][df_mes['MES'] <= 6], group1_name= "Semestre-1",
                                       group2= df_mes['VL_VENDA_TOT'][df_mes['MES'] >= 7], group2_name= "Semestre-2")
           print(summary)

                Variable     N          Mean            SD          SE     95% Conf.      Interval
           0  Semestre-1  14.0  39876.278571  25695.432510  6867.393204  25040.177541  54712.379601
           1  Semestre-2  17.0  40447.796471  19134.653742  4640.835205  30609.665327  50285.927615
           2    combined  31.0  40189.691613  21942.337730  3940.960195  32141.177155  48238.206071

In [1186]: print(results)

                                 Independent t-test   results
           0  Difference (Semestre-1 - Semestre-2) =  -571.5179
           1                Degrees of freedom =    29.0000
           2                                 t =    -0.0710
           3             Two side test p value =     0.9439
           4            Difference < 0 p value =     0.4720
           5            Difference > 0 p value =     0.5280
           6                        Cohen's d =    -0.0256
           7                        Hedge's g =    -0.0249
           8                    Glass's delta =    -0.0222
           9                    Pearson's r =     0.0132
```

Fonte: Autor

O método *ttest_ind* da biblioteca *scipy.stats* é um teste bilateral para a hipótese nula de que duas amostras independentes têm valores médios (esperados) idênticos. Este teste assume que as populações têm variâncias idênticas por padrão.

Figura 49 – Aplicando teste *T* por meio da biblioteca *scipy.stats*

```
In [ ]: import scipy.stats as stats

In [1191]: stats.ttest_ind(df_mes['VL_VENDA_TOT'][df_mes['MES'] <= 6],
                           df_mes['VL_VENDA_TOT'][df_mes['MES'] >= 7])

Out[1191]: Ttest_indResult(statistic=-0.070962784386682 94, pvalue=0.9439144973485499)
```

Fonte: Autor

O retorno de *p*-valor em ambas as bibliotecas foi de 0.9439, desta forma, a hipótese nula foi aceita, ou seja, as duas amostras (semestres 1 e 2), não possuem variância relevante nos resíduos gerados, com isso, o *dataset* pode ser utilizado no treinamento dos algoritmos.

Resíduos

O Método dos Mínimos Quadrados (MMQ) ou *Ordinay Least Squares* (OLS), é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados, tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados, tais diferenças são chamadas resíduos conforme visto nos parágrafos anteriores.

É a forma de estimação amplamente utilizada na econometria. Consiste em um estimador que minimiza a soma dos quadrados dos resíduos da regressão, de forma a maximizar o grau de ajuste do modelo aos dados observados.

Um requisito para o método dos mínimos quadrados é que o fator imprevisível (erro) seja distribuído aleatoriamente e essa distribuição seja normal. O Teorema Gauss-Markov garante que o estimador de mínimos quadrados é um estimador não-enviesado de mínima variância linear na variável resposta.

Outro requisito é que o modelo apresente parâmetros lineares, ou seja, as variáveis apresentam uma relação linear entre si. Caso contrário, deveria ser usado um modelo de regressão não-linear.

Por meio do método *summary()* disponível na biblioteca *statsmodels* são obtidas diversas informações do modelo relacionadas ao resultado de regressão utilizando OLS. A Figura 50 mostra o resultado do treinamento para o produto 1-B301.

Figura 50 – Resultado do modelo de regressão linear simples
statsmodels.formula.api.ols

```
In [735]: print(regressao.summary())

                            OLS Regression Results
==============================================================================
Dep. Variable:           VL_VENDA_TOT   R-squared:                       0.998
Model:                            OLS   Adj. R-squared:                  0.997
Method:                 Least Squares   F-statistic:                     2978.
Date:                Sat, 19 Nov 2022   Prob (F-statistic):          3.46e-34
Time:                        21:30:44   Log-Likelihood:                -258.35
No. Observations:                  31   AIC:                             526.7
Df Residuals:                      26   BIC:                             533.9
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -5983.1550   2944.918     -2.032      0.053    -1.2e+04      70.210
IPCA       -1143.1165    921.665     -1.240      0.226   -3037.626     751.393
QUANTIDADE  1747.6055    197.392      8.853      0.000    1341.860    2153.351
CMV_TOTAL      0.2573      0.161      1.597      0.122      -0.074       0.589
PERC_MARGEM  153.9498     74.791      2.058      0.050       0.215     307.685
==============================================================================
```

Fonte: Resultados da Pesquisa

O atributo "*Adj. R-squared*", representa o ajuste do modelo, ou seja, indica qual o percentual em que o modelo se ajustou aos dados de treinamento para realizar a predição, quanto maior o percentual, maior é a precisão na predição de valores. Além disso os coeficientes *Intercept,* IPCA, QUANTIDADE, CMV_VOTAL e PERC_MARGEM são utilizados na fórmula para calcular o valor previsto e obtenção do resíduo, ou seja, a diferença entre o valor total de venda realizado e o previsto.

Com base nas informações contidas na Figura 50, o modelo aplica a seguinte formula:

Valor previsto = Coeficiente *Intercept* + (Coeficiente IPCA * IPCA *Dataset*) + (Coeficiente QUANTIDADE * Quantidade *Dataset*) + (Coeficiente CMV_TOTAL * CMV_TOTAL *Dataset*) + (Coeficiente PERC_MARGEM * PERC_MARGEM *Dataset*)

Para exemplificar a utilização da fórmula acima, as cinco primeiras observações do conjunto de dados do item 1-B301 são apresentados na Figura 51, por meio do método *head()* contido na biblioteca *Pandas*.

Figura 51 – Cinco primeiras observações do *dataset* para o produto 1-B301

```
In [672]: print(df_mes.head())

     ANO  MES  IPCA  QUANTIDADE  VL_VENDA_TOT  CMV_TOTAL MES_EXT  PERC_MARGEM  VL_VENDA_UNIT
3   2016    6  0.35          15      30293.16   16873.08     Jun    44.300694    2019.544000
4   2016    7  0.52           6      12425.63    6749.23     Jul    45.682996    2070.938333
5   2016    8  0.44          16      33210.01   17390.55     Ago    47.634614    2075.625625
6   2016    9  0.08          23      49380.42   24375.24     Set    50.637844    2146.974783
7   2016   11  0.18          34      73750.83   36032.96     Nov    51.142299    2169.142059
```

Fonte: Resultados da Pesquisa

Por meio do método *resid* contido na biblioteca *statsmodels.stats.api,* são listados os resíduos obtidos com a aplicação da fórmula mencionada acima. A lista dos cinco primeiros resíduos é apresentada na Figura 52.

Figura 52 - Método *resid* para obtenção dos resíduos

```
In [738]: residuos = regressao.resid
          print(residuos)

          3     -699.568619
          4     -252.039481
          5      -73.856287
          6     1192.135099
          7     3375.752635
```

Fonte: Resultados da Pesquisa

Na Tabela 34 é simulado o cálculo do valor previsto pelo modelo por meio dos coeficientes resultantes do método *summary()* mostrados na Figura 51, para as cinco primeiras observações do *dataset.*

Tabela 34 - Simulação do cálculo dos resíduos utilizando o coeficiente *Intercept*

Idx	IPCA (*Dataset*)	QUANTIDADE (*Dataset*)	CMV (*Dataset*)	MARGEM (*Dataset*)	VL_VENDA_TOT (*Dataset*)	VALOR PREVISTO	RESÍDUO
3	0,35	15	16.873,08	44,30069	30.293,16	30.992,36	-699,20
4	0,52	6	6.749,23	45,68300	12.425,63	12.677,52	-251,89
5	0,44	16	17.390,55	47,63461	33.210,01	33.283,49	-73,48
6	0,08	23	24.375,24	50,63784	49.380,42	48.187,76	1.192,66
7	0,18	34	36.032,96	51,14230	73.750,83	70.374,30	3.376,53

Fonte: Resultados da Pesquisa

3.5.2 Validação interna com usuário do departamento de planejamento financeiro

O protótipo de solução computacional foi disponibilizado para o usuário do departamento de planejamento financeiro realizar simulações das predições de vendas no ambiente produtivo.

Conforme relatado anteriormente, o protótipo foi desenvolvido utilizando a base de dados replicada do ambiente produtivo com a data de corte do dia 31/05/2022, desta forma toda massa de dados utilizada para o treinamento dos algoritmos se limitou a esta data.

O protótipo foi conectado a base produtiva com dados de vendas mais recente, desta forma o usuário conseguiu avaliar a performance e o resultado das predições considerando as movimentações desconhecidas ao modelo, ou seja, transações compreendidas entre 01/06/2022 até 30/11/2022.

Na Tabela 35, fica evidenciado que os valores das predições apresentadas pelos algoritmos, tiveram uma maior precisão do que o cálculo de média simples utilizado pela área de planejamento financeiro.

Tabela 35 - Comparativo das predições com dados desconhecidos ao modelo

Produto	Ano	Mês	Qtd	Total de Vendas	Valor Unitário	% Margem Bruta	IPCA	Resultado do Algoritmo com maior precisão			Média Simples	
								Modelo	Valor Unitário	Resíduo	Valor Unitário	Resíduo
0-0001	2022	08	8	1.136,89	142,11	62,38 %	-0,36 %	Lasso	141,50	-4,89	73,02	-552,67
0-0001	2022	09	3	392,34	130,78	59,13 %	-0,29 %	Lasso	113,67	-51,34	73,02	-173,27
8-K011	2022	07	4	2.569,39	642,35	70,00 %	-0,68 %	LR	664,25	87.61	308,37	-1.335,88
8-K011	2022	10	3	2.047,36	682,45	71,70 %	0,59 %	LR	939,00	769,64	308,37	-1.122,27
1-B301	2022	8	22	70.878,72	3.221,76	39,90 %	-0,36 %	ElasticNet	3.141,59	-1.763,72	2.229,50	-21.829,77
1-B301	2022	11	15	46.800,00	3.120,00	39,20 %	0,41 %	ElasticNet	3.053,60	-996,00	2.229,50	-13.357,53

Fonte: Resultados da Pesquisa

Os produtos 13M1S1 e 0-B051 não foram avaliados neste testes, pois não houve vendas para estes produtos a partir de 01/06/2022.

Para os demais produtos, os meses com início em junho foram selecionados de forma aleatória pelo próprio algoritmo, destacando-se a coluna “Resíduo”, presente no resultado calculado pelo algoritmo assim como a média simples.

Ficou evidenciado que o resíduo presente na média simples ficou muito superior ao resíduo do algoritmo, confirmando que a precisão dos algoritmos é superior à média aritmética simples e que este

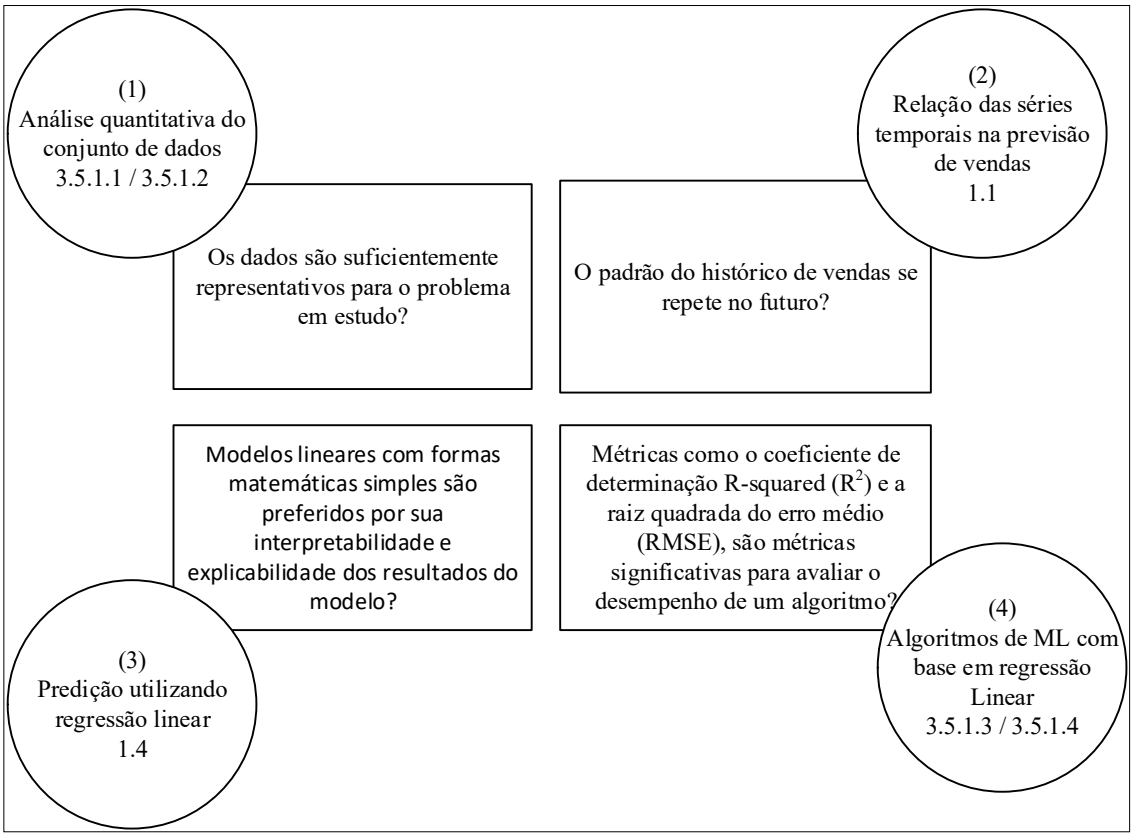
tipo de abordagem pode ser utilizado na predição de vendas, podendo gerar resultados positivos na empresa no que tange ao planejamento financeiro, aumentando a previsibilidade do fluxo de caixa.

3.5.3 Entrevista com especialistas

As entrevistas semiestruturadas foram conduzidas individualmente com os cinco especialistas, os quais possuem mais de 10 anos de experiência no segmento de varejo ou com IA. O roteiro da entrevista seguiu com 11 perguntas como parte de um questionário submetido a todos os especialistas, os quais tiveram que responder a pesquisa antes de ser realizada a entrevista.

As 11 perguntas contidas na pesquisa foram classificadas em 4 grupos conforme destacado na Figura 53.

Figura 53 - Classificação das perguntas do questionário



Fonte: Resultados da pesquisa

O grupo (1) está relacionado com a análise quantitativa do conjunto de dados, cujo objetivo é avaliar a importância da qualidade dos dados, evitando a existência de viés que possa influenciar os modelos gerados pelos algoritmos. O grupo (2) aborda a análise das séries temporais, esclarecendo se eventos passados podem se repetir no futuro e qual impacto no modelo. No grupo (3) é questionado se os modelos

lineares oferecem uma forma simples de predição se comparados a outros técnicas como ARIMA que lidam com séries temporais. Por fim no grupo (4), são abordadas as métricas para avaliar o desempenho dos algoritmos com base em regressão linear.

As 11 perguntas e sua separação nos quatro grupos mencionados anteriormente são apresentadas no Quadro 6.

Quadro 6 - Questionário utilizado na entrevista semiestruturada

#	Descrição	Grupos
P01	Os métodos de previsão podem ser separados em técnicas de Julgamento (Abordagens Qualitativas ou Técnicas Subjetivas) e técnicas Estatísticas ou Quantitativas (ARMSTRONG, 2008). Os métodos quantitativos podem ser classificados em dois tipos, Análises de Séries Temporais e Métodos Causais (PASSARI, 2003; LEMOS, 2006) ou, de acordo com a proposição de Armstrong (2008), com apenas uma variável, quando se realiza a extrapolação a partir da própria série temporal e múltipla quando se analisa mais de uma variável na previsão. Seja qual for o tipo de método quantitativo utilizado pressupõe-se o uso de dados históricos quantitativos. Dessa forma, para o desenvolvimento de modelos estatísticos de previsão se faz necessário uma base de dados. A sua maior vantagem é sua imparcialidade diante da previsão, tanto previsões ruins como boas podem ser feitas, porém o seu desempenho depende da qualidade dos dados de entrada (ARMSTRONG, 2008).	(1) - Análise quantitativa do conjunto de dados 3.5.1.1 / 3.5.1.2
P02	A abordagem quantitativa é mais eficiente na geração de previsões, quando existe uma demanda histórica do produto.	
P03	As análises de dados de séries temporais servem para obter propriedades e informações de relevância estatística dos dados.	
P04	As séries temporais podem apresentar tendência, ciclos e sazonalidade.	(2) - Relação das séries temporais na previsão de vendas 1.1
P05	Presume-se que o padrão histórico de vendas se repetirá no futuro.	
P06	A tendência de uma série indica o seu comportamento “de longo prazo”, isto é, se ela cresce, decresce ou permanece estável, e qual a velocidade destas mudanças	
P07	A série temporal possui uma sequência de dados equidistantes no tempo. A análise de séries temporais é realizada com o intuito de explorar o comportamento passado e de prever o comportamento futuro em um determinado problema	(3) - Predição utilizando regressão linear 1.4
P08	A análise de regressão consiste na realização de uma análise estatística com o objetivo de verificar a existência de uma relação funcional entre uma variável dependente com uma ou mais variáveis independentes	
P09	Ao analisar o histórico de vendas a partir de um conjunto de dados de séries temporais utilizando regressão linear, podemos considerar que se trata de um mecanismo útil para prever as receitas futuras.	
P10	O R-squared (R2) cuja medida estatística revela o quão próximo os dados estão da linha de regressão ajustada, assim como o Erro Quadrático Médio (MSE) cujo qual indica a diferença entre o valor predito e o valor real, são indicadores estatísticos relevantes para comparar modelos de regressão linear.	(4) - Algoritmos de ML com base em regressão Linear 3.5.1.3 / 3.5.1.4
P11	A utilização de uma ferramenta que compare o resultado de diferentes algoritmos em ML aplicando R-quadrado e MSE é relevante para que seja identificado o modelo com maior desempenho na predição.	

Fonte: Autor

O questionário foi desenvolvido com perguntas fechadas, porém em cada questão foi aberto a opção de comentários, no qual o respondente pôde expressar comentários a respeito da sua resposta, contribuindo para uma análise qualitativa das respostas.

Uma pergunta aberta e opcional foi aplicada no questionário para explorar questões ou temas vivenciados pelos especialistas e que não foram inicialmente abordadas na pesquisa, contribuindo para testar a validade e legibilidade do conteúdo antes da entrevista individual.

Para avaliar a efetividade da arquitetura da solução computacional, as perguntas fechadas são avaliadas utilizando 3 níveis: I) discordo; II) concordo parcialmente; III) concordo totalmente.

Na Tabela 36, foi adotada a seguinte simbologia para avaliação da opinião apresentada em cada resposta: se o respondente não concordou com a definição informada, a célula é mantida vazia, se o respondente concordou parcialmente, a célula é apresenta com meia bolinha "◗"; se o respondente concordou totalmente, a célula é apresentada com bolinha cheia " ●".

Para uma análise quantitativa, a meia bolinha representa 0,5 ponto e a bolinha inteira 1 ponto.

Tabela 36 - Resultados das entrevistas

	Entrevistados						
	A	**B**	**C**	**D**	**E**	**Total**	**Média por grupo**
P01	●	◗	◗	◗	◗	3,0	(1) 3,50
P02	◗	●	●	◗	◗	3,5	
P03	◗	◗	●	●	●	4,0	
P04	●	◗	●	●	●	4,5	(2) 2,83
P05	●	-	-	◗	-	1,5	
P06	◗	-	◗	●	◗	2,5	
P07	●	●	◗	●	◗	4,0	(3) 4,00
P08	●	●	●	●	●	5,0	
P09	◗	◗	◗	●	◗	3,0	
P10	◗	●	●	●	◗	4,0	(4) 4,00
P11	●	◗	●	●	◗	4,0	

Fonte: Resultados da Pesquisa

Entrevistado "A"

Com relação a importância na qualidade dos dados utilizados para treinar o modelo, o entrevistado "A" respondeu: "Sem uma base representativa pode-se construir um castelo de cristal sobre areia movediça", entretanto, ao utilizar uma série histórica longa não significa que o modelo será preciso o suficiente, uma vez que o futuro apresenta surpresas, ou seja, situações não captadas no passado, confirmando que nem sempre o futuro é a reprodução do passado.

O coeficiente de determinação *R-squared* (R^2) é uma medida de ajuste da regressão, porém a regressão pressupõe que você irá adicionar no modelo as variáveis explicativas para identificar as variáveis dependentes, sendo que a qualidade das variáveis independentes é medida pelos testes de significância, seja pelo teste T ou teste F. Quando se trabalha com séries temporais você não tem preocupação com a explicação, uma vez que o passado, em tese deve explicar o futuro, já os modelos de regressão servem para explicar fenômenos, o qual explica a variação de determinada variável, não sendo o melhor modelo para previsões de curto prazo, os modelos de séries temporais são mais indicados nestas situações. Porém em ambientes com pouca interferência externa os modelos de regressão, juntamente com uma base histórica longa e representativa conseguem superar os modelos tradicionais de séries temporais como o modelo ARIMA.

Entrevistado “B”

A qualidade dos dados é primordial para uma análise quantitativa, porém é insuficiente para uma predição assertiva, uma vez que a análise qualitativa em cima dos fatores externos ao meio deve ser considerada.

Quando existe um fluxo constante de informação, como no processo produtivo automotivo, a abordagem quantitativa assume um papel importante uma vez que nestes processos os fatores externos são conhecidos e gerenciáveis.

Em ambientes com grandes variações externas, não é possível considerar que o histórico de venda se repetirá no futuro, portanto a predição de valor nestes casos deve ser considerada apenas como um parâmetro de apoio, mas não o único fator decisório.

A regressão linear funciona bem para antecipar tendências, porém deve ser utilizada em conjunto com outras análises, ou seja, não deve ser usada de forma isolada para tomada de decisões.

As métricas *R-squared* (R^2) e RMSE são importantes para que seja possível identificar a margem de erro da predição e avaliar o impacto das variáveis independentes no resultado preditivo. Caso o índice de erro seja baixo, significa que o resultado pode se aproximar muito da realidade, indicando que análise está no caminho certo.

Entrevistado “C”

Uma base de dados representativa é de suma importância para treinar os modelos e obter um resultado mais próximo da realidade, porém em base de dados de qualquer natureza sempre existirá

algum viés de tendência que pode distorcer a previsão, principalmente quando um novo cenário está se desenhando ou quando ocorre alguma ruptura no comportamento do consumidor, como o que está ocorrendo nos últimos dois anos devido ao Covid-19, cujo comércio eletrônico teve um crescimento exponencial devido ao período de quarentena.

Podemos considerar que as séries temporais sejam uma importante fonte de consulta dos dados históricos para prever o comportamento futuro, desde que o espaço temporal seja reduzido, ou seja, não é possível considerar que o comportamento passado se perpetuará ao longo de muito tempo.

Não é possível presumir que o padrão histórico de vendas se repetirá no futuro, haja vista a recente crise sanitária que vivemos, a qual impactou profundamente o comportamento do consumidor.

Analisar a correlação existente entre as variáveis dependentes e independentes é de suma importância para que se encontre uma relação funcional consistente e independente de qualquer viés.

As predições para o médio e longo prazo é complicado, uma vez que as métricas *R-squared* (R^2) e RMSE podem distorcer a relação funcional entre as variáveis, pois quanto maior o período, maiores são as chances de surgirem fatorem externos que possam impactar na relação entre as variáveis e este fatores podem não estar contemplados nos parâmetros do modelo.

Resultados das entrevistas

As entrevistas realizadas com os especialistas foram realizadas com base no questionário enviado previamente aos mesmos, com a finalidade de coletar informações, para que fosse possível analisar o nível de aderência do protótipo de solução computacional na predição do valor unitário de venda. Assim os respondentes tiveram que responder as onze questões , utilizando-se apenas uma das opções disponíveis, tais como: “Discordo”, “Concordo Parcialmente” ou “Concordo Totalmente”. Além de existir um campo de comentários em cada pergunta, no final do questionário foi disponibilizada a seguinte pergunta: “Qual questão você acha importante e não foi abordada nesta pesquisa?”, oferecendo espaço ao especialista para descrever suas sugestões.

As informações adquiridas foram organizadas, considerando a opinião de cada especialista em relação a cada questão. Dessa forma, com base nos resultados coletados, fica evidenciado que as onze questões foram classificadas em sua maioria entre os critérios de “Concorda Parcialmente” e “Concorda Totalmente”. Portanto estes critérios são considerados como respostas positivas e apenas duas questões em particular receberam resposta classificada como “Discordo”.

Na Tabela 37, é apresentado o resultado da composição dos pontos por grupo de perguntas, no qual obedeceu ao seguinte critério: de 0 a 2 pontos para “Discordo”; de 2,1 a 3,9 pontos para “Concordo Parcialmente” e de 4,0 a 5,0 pontos para “Concordo Totalmente”.

Tabela 37 - Resultados das entrevistas por grupo de perguntas

Grupos de Perguntas	**Média de pontos**	**Resultado**
(1) - Análise quantitativa do conjunto de dados	3,50	Concordo Parcialmente
(2) - Relação das séries temporais na previsão de vendas	2,83	Concordo Parcialmente
(3) - Predição utilizando regressão linear	4,00	Concordo Totalmente
(4) - Algoritmos de ML com base em regressão Linear	4,00	Concordo Totalmente

Fonte: Resultados da Pesquisa

Observando os resultados apresentados na Tabela 37, nota-se que o protótipo de solução computacional conseguiu alcançar um alto nível de aderência entre os especialistas, com dois grupos de perguntas apresentando “Concordo Totalmente” e dois grupos de perguntas apresentando “Concordo Parcialmente”.

3.6 Comunicação

Este trabalho foi elaborado visando públicos da comunidade acadêmica, das áreas de engenharia de produção, tecnologia da informação e planejamento financeiro. Para o cumprimento da atividade, a pesquisa deve ser apresentada e aprovada perante banca de avaliação de dissertação de mestrado em gestão e tecnologia em sistemas produtivos. A aprovação e disponibilização na forma impressa e digital representa a comunicação formal da pesquisa. O artefato apresentado será objeto de artigo, cujo qual deverá ser apresentado em congresso da área de engenharia de produção e posteriormente ser encaminhado a um periódico qualificado e com linha de pesquisa adequada ao tema.

CONCLUSÃO

Esta pesquisa propôs um novo método multidisciplinar para realização de experimentos em Inteligência Computacional, cujo qual foi implementado por meio de algumas bibliotecas disponíveis na linguagem *Python*. Seis algoritmos de inteligência computacional com base em regressão linear e cinco conjuntos de dados representando o histórico de vendas de diferentes produtos foram usados nesse sentido. Diferentes modificações críticas foram propostas e testadas em várias fases do método publicado anteriormente por Tsiliki, e os resultados obtidos foram relevantes. Para uma melhor compreensão da pesquisa, todas as etapas foram aplicadas e demonstradas ao exemplo de linha de base de um estudo experimental em conjuntos de dados.

A fase final do estudo estatístico foi detalhadamente descrita, por ser a modificação mais crítica, assim como a validação cruzada externa. A partir dos cinco conjuntos de dados, cujos resultados foram comparados com métodos simples de previsão como média aritmética simples, pode-se concluir que, seguindo a metodologia desta pesquisa, os resultados podem ser verificados quanto à significância estatística e, portanto, confiáveis em modelos preditivos e deve ser proposto à comunidade científica. Além disso, com o método demonstrado neste estudo é possível resolver problemas de regressão linear em outros escopos de pesquisa, obtendo resultados estáveis, reprodutíveis e relevantes.

Pelos resultados apresentados, fica evidenciado que não existe um algoritmo melhor do que outro, ou seja, o comportamento dos dados e a correlação entre suas variáveis irá influenciar diretamente na obtenção do melhor modelo para um produto específico. Desta forma o protótipo de solução computacional resultante desta pesquisa é útil e relevante pois indica ao pesquisador qual o melhor o modelo para cada tipo de produto.

Nenhum método consegue prever o futuro, porém com o apoio de novas tecnologias é possível obter uma predição eficaz, sendo esta, essencial para vários segmentos, principalmente no setor de varejo que historicamente trabalha com baixas margens de lucro em um mercado altamente competitivo.

Embora se trate de um estudo de caso específico de uma empresa de revenda de produtos importados no segmento de digitalização de imagens e biometria, este estudo pode ser generalizado a outras empresas uma vez que as variáveis preditoras como “Quantidade de venda”, “Valor do custo unitário”, “Percentual da margem bruta” e “Percentual de inflação do mês - IPCA”, podem fazer parte do contexto de negócio de outras empresas.

Reconhece-se, entretanto, as limitações desta pesquisa quanto à capacidade de processamento disponível, limitando o conjunto de dados em um conjunto de cinco produtos. Para pesquisas futuras, sugere-se a utilização de outros produtos em segmentos diferentes do varejo.

Para pesquisas futuras, é sugerido explorar a relação entre margem de lucro e receita de vendas, comparando o histórico do lucro operacional relacionado com a receita das vendas, obtendo um modelo que consiga prever qual seria a receita alvo para que a empresa consiga manter sua sustentabilidade financeira.

REFERÊNCIAS

ABURTO, L.; WEBER, R. **Improved supply chain management based on hybrid demand forecasts.** Appl. Soft Comput. 7 (1) 136-144, 2007.

AHMED MOHAMMED, A.; AUNG, Z. **Ensemble learning approach for probabilistic forecasting of solar power generation**. Energies 9 (12) 1017, 2016.

ALADAG, C. H.; EGRIOGLU, E.; KADILAR, C. **Forecasting nonlinear time series with a hybrid methodology**. Appl. Math. Lett. 22 (9) 1467-1470, 2009.

Al-GUNAID, M. A.; SHCHERBAKOV, M. V.; KRAVETS, A. G.; LOSHMANOV, V. L.; SHUMKIN, A. M.; TRUBITSIN, V. V.; VAKULENKO, D. V. **Analysis of Drug Sales Data based on Machine Learning Methods**. Proceedings of the SMART-2018, IEEE Conference, 2018.

ALEXANDROV, A.; BENIDIS, K.; BOHLKE-SCHNEIDER, M.; FLUNKERT, V.; GASTHAUS, J.; JANUSCHOWSKI, T.; MADDIX, D. C.; RANGAPURAM, S.; SALINAS, D.; SCHULZ, J. *et al*. **GluonTS: probabilistic and neural time series modeling in Python.** J. Mach. Learn. Res. 21 (116) 1-6, 2020.

ALON, I.; QI, M.; SADOWSKI, R. J. **Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods.** J. Retailing Consum. Serv. 8 (3) 147–156, 2001.

ANDRADE, L. A. et al. **Pensamento sistêmico: caderno de campo**. Porto Alegre: Bookman, 2006.

ARUNRAJ, N. S.; AHRENS, D. **A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting**. Int. J. Prod. Econ. 170 321-335, 2015.

ATZORI, L.; IERA, A.; MORABITO, G. **The internet of things: A survey**. Computer Networks. 2010.

BABA, R. K.; VAZ, M. S. M. G.; COSTA, J. D. **Correção de dados agro meteorológicos utilizando métodos estatísticos**. Revista Brasileira de Meteorologia, v.29, n.4, 515 - 526, 2014

BABOO, S. S.; SHEREEF, I. K. **An efficient weather forecasting system using artificial neural network**, Int. J. Environ. Sci. Dev. 1 (4) 321, 2010.

BAKER, M. **1,500 scientists lift the lid on reproducibility**. Nature 533:452-454, 2016a.

BAKER, M. **Reproducibility: seek out stronger science**. Nature 537:703-704, 2016b.

BARDENET, R.; BRENDEL, M.; KÉGL, B.; SEBAG, M. **Collaborative hyperparameter tuning**. In International conference on machine learning (pp. 199–207), 2013.

BARTLETT, M. S. **Properties of Sufficiency and Statistical Tests**. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 160(901), 262-282, DOI: 10.1098/rspa.1937.0109, 1937.

BERGSTRA, J.; BENGIO, Y. **Random search for hyper-parameter optimization**. Journal of Machine Learning Research, 13 (Feb), 281–305, 2012.

BERGSTRA, J. S.; BARDENET, R.; BENGIO, Y.; KÉGL, B. **Algorithms for hyperparameter optimization.** In Advances in neural information processing systems (pp. 2546–2554), 2011.

BLACKBURN, R.; LURZ, K.; PRIESE, B.; GÖB, R.; DARKOW, I.-L. **A predictive analytics approach for demand forecasting in the process industry** - International Transactions in Operational Research, Vol. 22 No. 3, pp. 407-428. 2015.

BLAIN, G. C. **Revisiting the critical values of the Lilliefors test: towards the correct agrometeorological use of the Kolmogorov-Smirnov framework**. Agrometeorology - <https://doi.org/10.1590/brag.2014.015>, 2014.

BOEHMKE, B.; GREENWELL, B. M. **in Hands-onMachine Learning with R**, 1st ed., New York: CRC Press, pp. 488, 2019

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C., LJUNG, G. M. **Time Series Analysis: forecasting and Control.** John Wiley & Sons, 2015.

BREIMAN, L. **Random forests**. Machine Learning, 45(1), 5-32, 2001.

BREUSCH, T. S.; PAGAN, A. R. **A Simple Test for Heteroscedasticity and Random Coefficient Variation.** Econometrica, Vol. 47 No. 5, pp. 1287-1294, 1979.

BROWN, M. B; FORSYTHE, A. B. **Robust Tests for the Equality of Variances**. Journal of the American Statistical Association, 69:346, 364-367, 1974.

CASTILLO, P. A.; MORA, A. M.; FARIS, H.; MERELO, J. J.; GARCÍA-SÁNCHEZ, P.; FERNÁNDEZ-ARES, A. J.; CUEVAS, P. D.; GARCÍA-ARENAS, M. I. **Applying computational intelligence methods for predicting the sales of newly published books in a real editorial business management environment** - Knowledge-Based Systems, 115:133–151, 2017.

CECCHINEL, C.; JIMENEZ, M.; RIVEILL, M.; MOSSER, S. **An architecture to support the collection of big data in the internet of things**. IEEE World Congress on Services. 2014.

CHANG, P.C.; LIU, C.H.; FAN, C.Y. **Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry**. Knowledge-Based Systems, 22:344-355, 2009.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. **SMOTE: synthetic minority over-sampling technique**. Journal of Artificial Intelligence, 2002.

CHEN, Y.; HAO, Y. **A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction**, *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.

CHEN, Y.; KANG, Y.; CHEN, Y.; WANG, Z. **Probabilistic forecasting with temporal convolutional neural network.** Neurocomputing, 2020.

CHERIYAN, S.; IBRAHIM, S.; MOHANAN, S.; TREESA, S. **Intelligent Sales Prediction Using Machine Learning Techniques** – IEEE 978-1-5386-4904-6/18. 2018.

CHU, C.-W.; ZHANG, G. P. A comparative study of linear and nonlinear models for aggregate retail sales forecasting, Int. J. Prod. Econ. 86 (3) 217–231, 2003.

CUTLER, D. R., EDWARDS, T. C., BEARD, K. H., CUTLER, A., HESS, K. T., GIBSON, J., LAWLER, J. J. **RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY**. Ecology, 88(11),

2783–2792. Doi:10.1890/07-0539.1, 2007.

COOLS, M.; MOONS, E.; WETS, G. **Investigating the variability in daily traffic counts through use of ARIMAX and SARIMAX models: assessing the effect of holidays on two site locations.** Transp. Res. Rec. 2136 (1) 57-66, 2009.

CORNELSEN, L.; NORMAND, C. **Impact of the smoking ban on the volume of bar sales in Ireland: evidence from time series analysis.** Health Econ. 21 (5) 551-561, 2012.

CUESTA, D.; TABOADA, A.; CALVO, L.; SALGADO, J. **Short-and-medium-term effects of experimental nitrogen fertilization on arthropods associated with Calluna vulgaris heathlands in north-west Spain**. Environmental Pollution 152(2):394-402, 2008.

DASU, T; JOHNSON, T. **Exploratory data mining and data cleaning**. Hoboken: John Wiley & Sons. Vol. 479, 2003.

DEB, C.; ZHANG, F.; YANG, J.; LEE, S. E.; SHAH, K. W. **A review on time series forecasting techniques for building energy consumption, Renew.** Sustain. Energy Rev. 74, 902–924, 2017.

DESAI, V. S.; BHARATI, R. **A comparison of linear regression and neural network methods for predicting excess returns on large stocks.** Annals of Operations Research, 1998.

DONOHO, D. L. **High-dimensional data analysis: the curses and blessings of dimensionality**. In: Mathematical challenges of the 21st century conference of the American Mathematical Society. Los Angeles. American Mathematical Society, 2000.

FAN, X.; ZHANG, S.; WANG, L.; YANG, Y.; HAPESHI, K. **An evaluation model of supply chain performances using 5DBSC and LMBP neural network algorithm** - Journal of Bionic Engineering. 2013.

FERNANDEZ-LOZANO, C.; SEOANE, J. A.; GESTAL, M.; GAUNT, T. R.; DORADO, J.; PAZOS, A.; CAMPBELL, C. **Texture analysis in gel electrophoresis images using an integrative kernel-based approach**. Scientific Reports. 6:19256, 2016.

FOURCHES, D.; MURATOV, E.; TROPSHA, A. **Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research**. Journal of Chemical Information and Modeling 50(7):1189-1204, 2010.

FREEDMAN, D. A. **Statistical Models: Theory and Practice**. Cambridge University Press, p. 26, 2009.

FRIEDMAN, J. H. **Greedy function approximation: A gradient boosting machine**, *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

GALICIA, A.; TALAVERA-LLAMES, R.; TRONCOSO, A.; KOPRINSKA, I.; MARTÍNEZ-ÁLVAREZ, F. Multi-step forecasting for big data time series based on ensemble learning, Knowl. Based Syst. 163 830-841, 2019.

GARCÍA, S.; FERNÁNDEZ, A.; LUENGO, J.; HERRERA, F. **Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power**. Information Sciences, 180(10):2044-2064, 2010.

GUIDOTTI, R.; MONREALE, A.; RUGGIERI, S.; TURINI, F.; GIANNOTTI, F.; PEDRESCHI, D. **A survey of methods for explaining black box models**. ACM Comput. Surv. (CSUR) 51 (5) 1–42, 2018.

GUR ALI, O.; PINAR, E. **Multi-period-ahead forecasting with residual extrapolation and information sharing-utilizing a multitude of retail series.** Int. J. Forecast. 32 (2) 502-517, 2016.

HEUVEL, E.V. D.; ZHAN, Z. **Myths About Linear and Monotonic Associations: Pearson's, Spearman's, and Kendall's.** The American Statistician. 2022.

HEVNER, A.R.; MARCH, S.T.; PARK, J.; RAM, S. **Design Science in Information Systems Research**. MIS Q., 28, 75–105. 2004.

HILT, D. E.; SEEGRIST, D. W. **Ridge, a computer program for calculating ridge regression estimates**. DOI:10.5962/bhl.title.68934, 1977.

HOFMANN, E.; RUTSCHMANN, E. **Big data analytics and demand forecasting in supply chains: a conceptual analysis** - The International Journal of Logistics Management - Vol. 29 No. 2, 2018 - pp. 739-766 - © Emerald Publishing Limited - 0957-4093 - DOI 10.1108/IJLM-04-2017-0088, 2018.

HUANG, Q.; ZHOU, F. **Research on retailer data clustering algorithm based on spark**. In AIP Conference Proceedings (Vol. 1820, No. 1, p. 080022). AIP Publishing, 2017.

HYNDMAN, R. J.; KOEHLER, A. B. **Another look at measures of forecast accuracy. Int. J. Forecasting**, vol. 22, no. 4, pp. 679 688, 2006.

IBGE. Disponível em: <https://www.ibge.gov.br/estatisticas/economicas/precos-e-custos/9256-indice-nacional-de-precos-ao-consumidor-amplo.html?=&t=downloads>. Acesso em 20 Junho 2022.

ILIC, I.; GÖRGÜLÜ, B.; CEVIK, M.; GÖKÇE, M.; BAYDOGAN, G. **Explainable boosted linear regression for time series forecasting**. Pattern Recognition, 2021.

JORDAN, M.I.; MITCHELL, T.M. **Machine learning: Trends perspectives and prospects. Science**, 349:255–260, 2015.

JUNIPER RESEARCH. 2020. Disponível em: <https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024?ch=IOT%20CONNECTIONS%20TO%20GROW>. Acesso em 29 Maio 2022.

JURECKOVÁ, J.; PICEK, J. **Shapiro–Wilk-type test of normality under nuisance regression and scale**. Computational Statistics & Data Analysis. Volume 51, Issue 10, 15 June 2007, Pages 5184-5191, 2007.

KANEKO, Y.; YADA, K. **A Deep Learning Approach for the Prediction of Retail Store Sales**. IEEE 16th International Conference on Data Mining Workshops (ICDMW). doi:10.1109/icdmw.2016.0082, 2016.

KE, J.; ZHENG, H.; YANG, H.; CHEN, V. M. **Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach** - Transportation Research Part C, 85:591–608, 2017.

KEPNER, C. H.; TREGOE, B. B. **O administrador racional: uma abordagem sistemática à solução**

de problemas e tomada de decisão. 2. ed. São Paulo: Atlas, 1980.

KHASHEI, M.; BIJARI, M. **Which methodology is better for combining linear and nonlinear models for time series forecasting?** J. Ind. Syst. Eng., 2012.

KILIMCI, Z. H.; AKYUZ, A. O.; UYSAL, M.; AKYOKUS, S.; UYSAL, M. O. *et al.* **An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain**, *Complexity*, vol. 2019, pp. 15, 2019.

KOHAVI, R. **A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection**. International Joint Conference on Artificial Intelligence (IJCAI), v. 14, p. 1137–1145, 1995.

KRAWCZYK, B. **Learning from imbalanced data open challenges and future directions** - Progress in Artificial Intelligence – Springer. 2016.

KROLLNER, B.; VANSTONE, B. J.; FINNIE, G. R. **Financial time series forecasting with machine learning techniques: a survey.** ESANN, 2010.

JAIN, A.; MENON, M. N.; CHANDRA, S. **Sales Forecasting for Retail Chains**. Computer Science, 2015.

LACERDA, D. P. *et al*. **Design Research: método de pesquisa para a engenharia de produção Design Science Research: a research method to production engineering**. Gestão & Produção, v. 20, n. 4, p. 741-761, 2013.

LALOR, J. P.; WU, H.; YU, H. **Improving machine learning ability with finetuning**. arXiv preprint arXiv: 1702.08563, 2017.

LICHMAN, M. 2013 - **UCI Machine Learning Repository**. Disponível em <https://archive.ics.uci.edu/ml/datasets/wine>. Acesso em 06 Junho 2022.

LIU, B.; NOWOTARSKI, J.; HONG, T.; WERON R. **Probabilistic load forecasting via quantile regression averaging on sister forecasts.** IEEE Trans. Smart Grid 8 (2) 730-737, 2015.

LOUREIRO, A. L. D.; MIGUÉIS, V. L.; DA SILVA, L. F. M. **Exploring the use of deep neural networks for sales forecasting in fashion retail**. Decision Support Systems. doi:10.1016/j.dss.2018.08.010, 2018.

LOYER, J.L.; HENRIQUES, E.; FONTUL, M.; WISEALL, S. **Comparison of machine learning methods applied to the estimation of manufacturing cost of jet engine components**. -International Journal of Production Economics. 2016.

MA, S.; FILDES, R. **Retail sales forecasting with meta-learning**. Eur. J. Oper. Res., 2020.

MACLAURIN, D.; DUVENAUD, D.; ADAMS, R. **Gradient-based hyperparameter optimization through reversible learning**. In International conference on machine learning (pp. 2113–2122), 2015.

MAINGI, M. N. **A Survey on the Clustering Algorithms in Sales Data Mining**. International Journal of Computer Applications Technology and Research Volume 4– Issue 2, 126 - 128, ISSN: 2319–8656, 2015.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The M5 accuracy competition: results,

findings and conclusions. Int. J. Forecast, 2020.

MANN, A. K.; KAUR, N. **Review paper on clustering techniques**. Global Journal of Computer Science and Technology, 2013.

MANYIKA, J. **Big data: the next frontier for innovation, competition and productivity.** McKinsey Global Institute, 2011.

MARCH, S. T.; SMITH, G. F. **Design and natural science research on information technology**. Decision Support Systems, v. 15, p. 251-266, 1995.

MARDEN, J. I. **Positions and QQ Plots**. Statistical Science, Vol. 19, No. 4, 606–614, 2004.

MARTINS, E.; GALEGALE, N. V. **Detecção de fraudes no segmento de crédito financeiro utilizando aprendizado de máquina: Uma revisão da literatura**. E-TECH: Tecnologias para Competitividade Industrial, 2022a.

MARTINS, E.; GALEGALE, N. V. **Retail sales forecasting information systems: Comparison between traditional methods and machine learning algorithms**. 15th IADIS International Conference - Information Systems - PORTO, Portugal, 2022b.

MAXWELL, A. E.; WARNER, T. A.; STRAGER M. P.; CONLEY, J.F.; SHARP, A.L. **Assessing machine-learning algorithms and image-and lidar-derived variables for GEOBIA classification of mining and mine reclamation**. International Journal of Remote Sensing Vol 36, 2015.

MAYO, M. **The data science puzzle, explained**. 2016. Disponível em: <https://www.kdnuggets.com/2016/03/data-science-puzzle-explained.html/2>. Acesso em 31 Maio 2022.

MCLACHLAN, G; DO, K-A; AMBROISE, C. **Analyzing microarray gene expression data**. Hoboken: John Wiley & Sons. Vol. 422, 2005.

MÜLLER, K. R.; SMOLA, A. J.; RÄTSCH, G.; SCHÖLKOPF, B.; KOHLMORGEN, J.; VAPNIK, V. **Predicting time series with support vector machines**, in: ICANN, Springer, pp. 999-1004, 1997.

MITCHELL, T.M. **Machine Learning** - McGraw Hill. 1997.

NGUYEN, H. D.; TRAN, K. P.; THOMASSEY, S.; HAMAD, M. **Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management**. International Journal of Information Management, 2021.

NICKSON, T.; OSBORNE, M. A.; REECE, S.; ROBERTS, S. J. **Automated machine learning on big data using stochastic algorithm tuning**. arXiv preprint arXiv: 1407. 7969, 2014.

O'HARA, R. B.; KOTZE, D. J. **Do not log-transform count data**. Methods in Ecology and Evolution 1(2):118-122, 2010.

ORESHKIN, B. N.; CARPOV, D.; CHAPADOS, N.; BENGIO, Y. **N-BEATS: neural basis expansion analysis for interpretable time series forecasting.** arXiv preprint arXiv:1905.10437, 2019.

PARMEZAN, A. R. S.; SOUZA V. M.; BATISTA, G. E. **Evaluation of statistical and machine**

learning models for time series prediction: identifying the state-of-the-art and the best conditions for the use of each model. Inf. Sci. 484 302-337, 2019.

PAVLYSHENKO, B. M. **Machine-learning models for sales time series forecasting**. Data, vol. 4, no. 1, pp. 15, 2019.

PEFFERS, K.; TUUNANEN, T.; GENGLER, C.E.; ROSSI, M.; HUI, W.; VIRTANEN, V.; BRAGGE, J. **The design science research process: A model for producing and presenting information systems research**. In Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA, USA, 24–25, 2006.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M.A.; CHATTERJEE, S. **A Design Science Research Methodology for Information Systems Research**. J. Manag. Inf. Syst., 24, 45–77, 2007.

PHOON, K-K; ASCE, M; QUEK, S-T; AN, .P **Identification of Statistically Homogeneous Soil Layers Using Modified Bartlett Statistics**. Journal Of Geotechnical and Geoenvironmental Engineering. DOI: 10.1061/(ASCE)1090-0241(2003)129:7(649), 2003.

PUTH, M-T.; NEUHAUSER, M.; RUXTON, G. D. **Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits.** Animal Behaviour. 2015.

QUADE, D. **Using weighted rankings in the analysis of complete blocks with additive block effects**. Journal of the American Statistical Association, 74:367, 680-683, 1979.

RAJAGOPAL, D. **Customer data clustering using data mining technique**. arXiv:1112.2663, 2011.

RANGAPURAM, S. S.; SEEGER, M. W.; GASTHAUS, J.; STELLA, L.; WANG, Y.; JANUSCHOWSKI, T. **Deep state space models for time series forecasting**, in: Advances in Neural Information Processing Systems, pp. 7785–7794, 2018.

RASCH, D.; TEUSCHER, F.; GUIARD, V. **How robust are tests for two independent samples?** Journal of Statistical Planning and Inference, 137(8), 2706-2720, DOI: 10.1016/j.jspi.2006.04.011, 2007.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning, 2nd Ed**.- Packt Publishing, Birmingham. 2017.

RUMSEY, D. J. **Statistical Literacy as a Goal for Introductory Statistics Courses**. Journal of Statistics Education. 2017.

SAEYS, Y.; INZA, I.; LARRAÑAGA, P. **A review of feature selection techniques in bioinformatics**. Bioinformatics, 23(19):2507-2517, 2007.

SALINAS, D.; FLUNKERT, V.; GASTHAUS, J.; JANUSCHOWSKI, T. **DeepAR: probabilistic forecasting with autoregressive recurrent networks**, Int. J. Forecast. 36 (3), 1181–1191, 2020.

SANTOSA, F.; SYMES, W. W. **Linear inversion of band-limited reflection seismograms**. SIAM Journal on Scientific and Statistical Computing. SIAM. 7 (4): 1307-1330. DOI: 10.1137/0907087, 1986.

SASTRY, S. H.; BABU, P.; PRASADA, M. S. **Analysis & Prediction of Sales Data in SAP-ERP System using Clustering Algorithms**. arXiv preprint arXiv:1312.2678, 2013.

SARWAR, B.; KARYPIS, G.; KONSTAN, J.; RIEDL, J. **Analysis of recommendation algorithms for e-commerce** - in Presented at the Proc. of the 2nd ACM Conf. on Electronic Commerce, Minneapolis, Minnesota, USA, 2000

SAYLI, A.; OZTURK, I.; USTUNEL, M. **Brand loyalty analysis system using K-Means algorithm**. Journal of Engineering Technology and Applied Sciences, 1(3): 107-126, 2016.

SEIFFERT, C.; KHOSHGOFTAAR, T. M.; HULSE, J. V.; NAPOLITANO, A. **RUSBoost: a hybrid approach to alleviating class imbalance**. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 40(1):185-197, 2010.

SHAPIRO, S. S.; WILK, M. B. **An analysis of variance test for normality (complete samples)**. Biometrika, 52(3-4), 591-611. DOI: 10.1093/biomet/52.3-4.591, 1965.

SHARMA, S. K.; CHAKRABORTI, S.; JHA, T. **Analysis of book sales prediction at Amazon marketplace in India: a machine learning approach**. Information Systems and e-Business Management, 17(2-4), 261–284. doi:10.1007/s10257-019-00438-3, 2019.

SHRIVASTAVA, V.; ARYA, N. **A study of various clustering algorithms on retail sales data**. International Journal of Computing, Communications and Networking, ISSN 2319-2720, 2012.

SILVER, D. *et al*. **Mastering the game of go with deep neural networks and tree search**. Nature, 529:484-489, 2016.

SILVER, D. *et al*. **Mastering the game of go without human knowledge**. Nature, 550:354-359, 2017.

SIMONM, H.A. **The sciences of the artificial**. 3a ed. Cambridge: MIT Press, 1996.

SMYL, S. **A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting**. Int. J. Forecast. 36 (1) 75-85, 2020.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. **Practical bayesian optimization of machine learning algorithms**. In Advances in neural information processing systems (pp. 2951–2959), 2012.

TAIEB, S. B.; HYNDMAN, R. J. **A gradient boosting approach to the Kaggle load forecasting competition**. Int. J. Forecast. 30 (2) 382-394, 2014.

TASKAYA-TEMIZEL, T.; CASEY, M. C. **A comparative study of autoregressive neural network hybrids.** Neural Netw. 18 (5-6) 781-789, 2005.

TIBSHIRANI, R. **Regression Shrinkage and Selection via the lasso**. Journal of the Royal Statistical Society. Series B (Methodological). Wiley. 58 (1): 267-88. JSTOR 2346178, 1996.

THOMASSEY, S. **Sales forecasts in clothing industry: The key success factor of the supply chain management**. International Journal of Production Economics, 128(2), 470–483. doi:10.1016/ j.ijpe.2010.07.018, 2010.

TREMBLAY, M. C.; HERVNER, A. R.; BERNDT, D. J, **Focus Groups for Artifact Refinament and Evaluation in Design Research**. Communications of the Association for Information Systems, v. 26, n. 27, p. 599-618, 2010.

TROPSHA, A. **Best Practices for QSAR Model Development, Validation, and Exploitation**.

Molecular Informatics. DOI: 10.1002/minf.201000061 - 29: 476-488, 2010.

TSAI, C. F.; WU, H. C.; TSAI, C. W. **A new data clustering approach for data mining in large databases**. In Parallel Architectures, Algorithms and Networks, 2002. I-SPAN'02. Proceedings. International Symposium on (pp. 315-320). IEEE, 2002.

TSILIKI, G.; MUNTEANU, C.R.; SEOANE, J.A.; FERNANDEZ-LOZANO, C.; SARIMVEIS, H.; WILLIGHAGEN, E. L. **RRegrs: an R package for computer-aided model selection with multiple regression models**. Journal of Cheminformatics 7:1-16, (2015a).

TSILIKI, G.; MUNTEANU, C. R.; SEOANE, J. A.; FERNANDEZ-LOZANO, C.; SARIMVEIS, H.; WILLIGHAGEN, E. L. **Using the RRegrs R package for automating predictive modelling**. In: MOL2NET, international conference on multidisciplinary sciences, (2015b).

VAHDANI, B.; RAZAVI, F.; MOUSAVI, S.M. **A high performing meta-heuristic for training support vector regression in performance forecasting of supply chain** - Neural Computing and Applications, 2016.

VAN AKEN, J.E. **Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules**. Journal of Management Studies, v. 41, n.2, p. 219-246, 2004.

VEIGA, C. P.; VEIGA, C. R. P.; PUCHALSKI, W.; COELHO, L. S.; TORTATO, U. **Demand forecasting based on natural computing approaches applied to the foodstuff retail segment**. Journal of Retailing and Consumer Services, vol. 31, pp. 174–181, 2016.

VENABLE, J.R. **The role of theory and theorizing in design science research**. In: International Conference on Design Science Research in Information Systems and Technology. Claremont, 2006.

WAMBA, S. F.; AKTER, S.; EDWARDS, A.; CHOPIN, G.; GNANZOU, D. **How "big data" can make big impact: Findings from a systematic review and a longitudinal case study**. International Journal of Production Economics, 165, 234–246. DOI:10.1016/j.ijpe.2014.12.031, 2015.

WANG, D.; LIU, X.; WANG, M. **A dt-svm strategy for stock futures prediction with big data** - IEEE 16th International Conference on Computational Science and Engineering, 2013.

WANG, W.; CHAKRABORTY, G.; CHAKRABORTY, B. **Predicting the Risk of Chronic Kidney Disease (CKD) using Machine Learning Algorithm**. Appl. Sci. 2021, 11, 202. 2021.

WANG, Y.; FENG, D.; LI, D.; CHEN, X.; ZHAO, Y. *et al.* **A mobile recommendation system based on logistic regression and gradient boosting decision trees**, in *Presented at the the Int. Joint Conf. on Neural Networks*, Vancouver, BC, Canada, 2016.

WANG, Y.; SMOLA, A.; MADDIX, D.; GASTHAUS, J.; FOSTER, D.; JANUSCHOWSKI, T. **Deep factors for forecasting.** in: ICML, PMLR, pp. 6607-6617, 2019.

WANG, Y.; YIN, H.; CHEN, H.; WO, T.; XU, J.; ZHENG, K. **Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling**. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. doi:10.1145/3292500.3330877, 2019.

WEN, R.; TORKKOLA, K.; NARAYANASWAMY, B.; MADEKA, D. A multi-horizon quantile

recurrent forecaster. arXiv preprint arXiv:1711.11053, 2021.

WERDIGIER, J. **Tesco, British grocer, uses weather to predict sales**, New York Times p. 1, 2009.

WIEDERMANN, W.; HAGMANN, M. **Asymmetric properties of the Pearson correlation coefficient: Correlation as the negative association between linear regression residuals.** Communications in Statistics - Theory and Methods. 2015.

WITTEN, I. H.; FRANK, E.; HALL, M. **A. Data Mining: Practical Machine Learning Tools and Techniques**. 3rd edition. Morgan Kaufmann Series, 2011.

WONG, D. **Data is the Next Frontier, Analytics the New Tool: Five Trends in Big Data and Analytics, and Their Implications for Innovation and Organisations**. Big Innovation Centre, London, 2012.

XU, W.; HOU, Y.; HUNG, Y. S.; ZOU, Y. **A comparative analysis of Spearman's rho and Kendall's tau in normal and contaminated normal models.** Signal Processing. 2012.

YALÇIN, M. O.; DINCER, N. G.; DEMIR, S. **Fuzzy panel data analysis**. Kuwait J. Sci., Vol. 48(3), pp(1-13), 2021.

ZHANG, G. P. **Time series forecasting using a hybrid ARIMA and neural network model**. Neurocomputing 50, 159-175, 2003.

ZHOU, L.; PAN, S.; WANG, J.; VASILAKOS, A.V. **Machine learning on big data: Opportunities and challenges**, Neurocomputing, 237:350–361, 2017.

ZOU, H.; HASTIE, T. **Regularization and Variable Selection via the Elastic Net**. Journal of the Royal Statistical Society, Series B. 67 (2): 301-320. CiteSeerX 10.1.1.124.4696. DOI:10.1111/j.1467-9868.2005.00503, 2005.

APÊNDICES

APÊNDICE A – Propriedades e instrução SQL para coletar o histórico de vendas dos produtos

No código fonte abaixo são apresentadas as importações das bibliotecas *Python*, a conexão com o banco de dados *Oracle,* instrução SQL para coletar as movimentações do estoque e a instanciação do *Dataset.*

```
import cx_Oracle as ora
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

import scipy.stats as stats

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics

conn = ora.connect("system", "xxxxx", "localhost/XE")
cursor = conn.cursor()
desired_width = 320
pd.set_option('display.width', desired_width)
pd.set_option('display.max_columns', 20)
pd.set_option('display.max_rows', 300)

sql = """
            SELECT ME.U##COD_ESTABEL,
                   REPLACE(IT.IT_CODIGO,CHR(34),'') IT_CODIGO,
                   REPLACE(REPLACE(IT.DESC_ITEM,CHR(10),''),CHR(13),'') DESC_ITEM,
                   DECODE(ME.TIPO_TRANS,1,'ENT','SAI') TP_TRANS,
                   ME.DT_TRANS,
                   TO_CHAR(ME.DT_TRANS,'YYYY') ANO,
                   TO_CHAR(ME.DT_TRANS,'MM')   MES,
                   ME.U##NRO_DOCTO,
                   ME.U##SERIE_DOCTO,
                   ME.COD_EMITENTE,
                   EMIT.NOME_EMIT,
                   ME.U##NAT_OPERACAO,
                   REPLACE(REPLACE(NATOP.DENOMINACAO,CHR(10),''),CHR(13),'') DENOMINACAO,
                   ME.SEQUEN_NF,
            --       ME.DESCRICAO_DB,
                   ME.U##COD_LOCALIZ,
                   DECODE(NVL(ME.ESP_DOCTO,0),
                            1,'ACA',
                            2,'ACT',
                            5,'REQ',
                            6,'DIV',
                            8,'EAC',
                            16,'IPL',
                            18,'NC',
                            20,'NFD',
                            21,'NFE',
                            22,'NFS',
                            23,'NFT',
                            25,'REF',
                            28,'DEV',
                            32,'STR',
                            33,'TRA',
                            35,'SOB',
                            ME.ESP_DOCTO) ESP_DOCTO,
                   ME.COD_DEPOS,
                   ME.QUANTIDADE,
```

```
                  ROUND(ITNF.VL_PREUNI,2) VL_VENDA_UNIT,
                  ROUND((NVL(ME.QUANTIDADE,0) * ITNF.VL_PREUNI),2) VL_VENDA_TOT,
                  ROUND(ME.VALOR_MAT_M##1 /
(DECODE(NVL(ME.QUANTIDADE,0),0,1,ME.QUANTIDADE)),2) VL_MATERIAL_UNIT,
                  ME.VALOR_MAT_M##1 VL_MATERIAL_TOT,
                  CMV.CMV_UNITARIO,
                  ROUND((NVL(ME.QUANTIDADE,0) * NVL(CMV.CMV_UNITARIO,0)),2) CMV_TOTAL,
                  CASE
                      WHEN ME.ESP_DOCTO=22 AND ITNF.VL_PREUNI >=
(ROUND(ME.VALOR_MAT_M##1 / (DECODE(NVL(ME.QUANTIDADE,0),0,1,ME.QUANTIDADE)),2)) THEN
                        ROUND((((NVL(ME.QUANTIDADE,0) * ITNF.VL_PREUNI) -
ME.VALOR_MAT_M##1) / (NVL(ME.QUANTIDADE,0) * ITNF.VL_PREUNI)) * 100,1)
                      WHEN ME.ESP_DOCTO=22 AND ITNF.VL_PREUNI < (ROUND(ME.VALOR_MAT_M##1
/ (DECODE(NVL(ME.QUANTIDADE,0),0,1,ME.QUANTIDADE)),2)) THEN
                        ROUND((((NVL(ME.QUANTIDADE,0) * ITNF.VL_PREUNI) -
ME.VALOR_MAT_M##1) / (NVL(ME.QUANTIDADE,0) * ITNF.VL_PREUNI)) * 100,1)
                      ELSE
                        0
                  END PERC_MARGEM,
                  CMV.SALDO,
                  CMV.CMV_TOT_SALDO,
                  INFL.IPCA
           FROM EMS2_1.ITEM        IT,
                EMS2_1.MOVTO_ESTOQ ME,
                EMS2_1.NATUR_OPER  NATOP,
                (SELECT TO_CHAR(PERIODO,'YYYY/MM') ANO_MES,
                        U##IT_CODIGO,
                        SUM(NVL(QUANTIDADE,0)) SALDO,ROUND(SUM(DISTINCT
NVL(VAL_UNIT_MAT_M##1,0)),4) CMV_UNITARIO,
                        ROUND(SUM(NVL(QUANTIDADE,0) * NVL(VAL_UNIT_MAT_M##1,0)),4)
CMV_TOT_SALDO
                 FROM EMS2_1.SL_IT_PER
                 GROUP BY TO_CHAR(PERIODO,'YYYY/MM'),U##IT_CODIGO
                 ORDER BY 1) CMV,
                (SELECT
U##COD_ESTABEL,U##SERIE,U##NR_NOTA_FIS,NR_SEQ_FAT,U##IT_CODIGO,VL_PREUNI
                 FROM EMS2_1.IT_NOTA_FISC) ITNF,
                EMS2MULT.EMITENTE EMIT,
                EMS2_1.INFLACAO INFL
           WHERE IT.U##IT_CODIGO = ME.U##IT_CODIGO(+)
             AND ME.DT_TRANS BETWEEN :DT_INICIAL AND :DT_FINAL
             AND ME.U##IT_CODIGO                = CMV.U##IT_CODIGO(+)
             AND TO_CHAR(ME.DT_TRANS,'YYYY/MM') = CMV.ANO_MES(+)
             AND ME.U##NAT_OPERACAO             = NATOP.NAT_OPERACAO(+)
             AND ME.U##COD_ESTABEL              = ITNF.U##COD_ESTABEL(+)
             AND ME.U##SERIE_DOCTO              = ITNF.U##SERIE(+)
             AND ME.U##NRO_DOCTO                = ITNF.U##NR_NOTA_FIS(+)
             AND ME.SEQUEN_NF                   = ITNF.NR_SEQ_FAT(+)
             AND ME.U##IT_CODIGO                = ITNF.U##IT_CODIGO(+)
             AND ME.COD_EMITENTE                = EMIT.COD_EMITENTE(+)
             AND TO_CHAR(ME.DT_TRANS,'MM')      = INFL.MES(+)
             AND TO_CHAR(ME.DT_TRANS,'YYYY')    = INFL.ANO(+)
             AND IT.U##IT_CODIGO                = :IT_CODIGO
             AND NVL(NATOP.EMITE_DUPLIC,0)      = 1
          ORDER BY IT.U##IT_CODIGO,
                   TO_CHAR(ME.DT_TRANS,'YYYY'),
                   TO_CHAR(ME.DT_TRANS,'MM'),
                   ME.U##COD_ESTABEL
         """

sIT_CODIGO = 'xxxxxxx'
df = pd.read_sql(sql=sql, con=conn, params={'IT_CODIGO': sIT_CODIGO,
                                            'DT_INICIAL': '01-JAN-2006',
                                            'DT_FINAL': '31-MAY-2022'})
```

APÊNDICE B – Código fonte para aplicação do K-fold com o respectivo log de execução.

```
import warnings
warnings.filterwarnings('ignore')

y = df_mes[["VL_VENDA_TOT"]]
x = df_mes.drop(["MES_EXT", "VL_VENDA_TOT", "VL_VENDA_UNIT"], axis=1)

test_size=0.3
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)

x = x_train.copy()
y = y_train.copy()

seed = 7459

model  = LinearRegression()
seed = 1
kfold = KFold(n_splits=5, random_state=seed, shuffle=True)

result = cross_val_score(model, x, y, cv = kfold)

print("K-Fold (R^2) Scores: {0}".format(result))
print("Mean R^2 for Cross-Validation K-Fold: {0}".format(result.mean()))

df_result_cv = pd.DataFrame()

def ApplyesKfold(x_axis, y_axis, p_seed):

    print('seed {}'.format(p_seed))

    # KFold settings.
    seed = p_seed
    kfold = KFold(n_splits=5, random_state=seed, shuffle=True)

    # Axis
    x = x_axis
    y = y_axis

    # Models instances.
    linearRegression = LinearRegression()
    elasticNet       = ElasticNet()
    ridge            = Ridge()
    lasso            = Lasso()
    svr              = SVR(kernel='linear')
    rf               = RandomForestRegressor()


    linearRegression_result    = cross_val_score(linearRegression, x, y, cv=kfold,
scoring='r2')
    elasticNet_result          = cross_val_score(elasticNet, x, y, cv=kfold,
scoring='r2')
    ridge_result               = cross_val_score(ridge, x, y, cv=kfold, scoring='r2')
    lasso_result               = cross_val_score(lasso, x, y, cv=kfold, scoring='r2')
    svr_result                 = cross_val_score(svr, x, y, cv=kfold, scoring='r2')
    rf_result                  = cross_val_score(rf, x, y, cv=kfold, scoring='r2')

    print(linearRegression_result)


    # Creates a dictionary to store Linear Models.
    dic_models_r2 = {
      "LinearRegression": linearRegression_result.mean(),
      "ElasticNet": elasticNet_result.mean(),
      "Ridge": ridge_result.mean(),
      "Lasso": lasso_result.mean(),
      "SVR": svr_result.mean(),
      "RandomForestRegressor": rf_result.mean()
    }
```

```
    df_result_cv=pd.DataFrame.from_dict([dic_models_r2])
    df_result_cv['Seed']=p_seed
    df_result_cv['Scoring']='R2'
    df_result_cv['Item']=sIT_CODIGO

    # Select the best model
    bestModel = max(dic_models_r2, key=dic_models_r2.get)

    print("""          Linear Regression Mean (R^2): {0}
          Elastic Net Mean (R^2): {1}
          Ridge Mean (R^2): {2}
          Lasso Mean (R^2): {3}
          SVR Mean (R^2): {4}
          Random Forest Mean (R^2): {5}""".format(linearRegression_result.mean(),
                                          elasticNet_result.mean(),
                                          ridge_result.mean(),
                                          lasso_result.mean(),
                                          svr_result.mean(),
                                          rf_result.mean()))

    print("The best model r2 is: {0} with value: {1}".format(bestModel,
dic_models_r2[bestModel]))

    linearRegression_result_er = cross_val_score(linearRegression, x, y, cv=kfold,
scoring='neg_mean_absolute_error')
    elasticNet_result_er       = cross_val_score(elasticNet, x, y, cv=kfold,
scoring='neg_mean_absolute_error')
    ridge_result_er            = cross_val_score(ridge, x, y, cv=kfold,
scoring='neg_mean_absolute_error')
    lasso_result_er            = cross_val_score(lasso, x, y, cv=kfold,
scoring='neg_mean_absolute_error')
    svr_result_er              = cross_val_score(svr, x, y, cv=kfold,
scoring='neg_mean_absolute_error')
    rf_result_er               = cross_val_score(rf, x, y, cv=kfold,
scoring='neg_mean_absolute_error')


    dic_models_mse = {
           "LinearRegression": -linearRegression_result_er.mean(),
           "ElasticNet": -elasticNet_result_er.mean(),
           "Ridge": -ridge_result_er.mean(),
           "Lasso": -lasso_result_er.mean(),
           "SVR": -svr_result_er.mean(),
           "RandomForestRegressor": -rf_result_er.mean()
    }

    df_result_cv2=pd.DataFrame.from_dict([dic_models_mse])
    df_result_cv2['Seed']=p_seed
    df_result_cv2['Scoring']='MSE'
    df_result_cv2['Item']=sIT_CODIGO

    df_result_cv = df_result_cv.append(df_result_cv2, ignore_index=True)

    # Select the best model.
    bestModel = min(dic_models_mse, key=dic_models_mse.get)


    print("""          Linear Regression Mean (MSE): {0}
          Elastic Net Mean (MSE): {1}
          Ridge Mean (MSE): {2}
          Lasso Mean (MSE): {3}
          SVR Mean (MSE): {4}
          Random Forest Mean (MSE):""".format(-linearRegression_result_er.mean(),
                                          -elasticNet_result_er.mean(),
                                          -ridge_result_er.mean(),
                                          -lasso_result_er.mean(),
                                          -svr_result_er.mean(),
                                          -rf_result_er.mean()))

    print("The best model MSE is: {0} with value: {1}".format(bestModel,
dic_models_mse[bestModel]))
```

```
    return(df_result_cv)

df_cv = pd.DataFrame()

for num in range(745, 755):
    z = ApplyesKfold(x, y, num)
    df_cv = df_cv.append(z, ignore_index=True)
```

```
K-Fold (R^2) Scores: [0.97560035 0.95710211 0.96202386 0.98027632 0.93558951]
Mean R^2 for Cross-Validation K-Fold: 0.962118426645819
seed 745
[0.95920562 0.94679118 0.96978672 0.98586732 0.99188243]
          Linear Regression Mean (R^2): 0.9707066533672467
          Elastic Net Mean (R^2): 0.973237305340889
          Ridge Mean (R^2): 0.9719889792890882
          Lasso Mean (R^2): 0.9707107689614272
          SVR Mean (R^2): 0.9721715852308559
          Random Forest Mean (R^2): 0.9243212514380141
The best model r2 is: ElasticNet with value: 0.973237305340889
          Linear Regression Mean (MSE): 32128.046404281202
          Elastic Net Mean (MSE): 30519.783651824273
          Ridge Mean (MSE): 31421.133193557274
          Lasso Mean (MSE): 32125.721914978214
          SVR Mean (MSE): 26119.254953343763
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 26119.254953343763
seed 746
[0.95214704 0.96470401 0.95596226 0.93814505 0.98891759]
          Linear Regression Mean (R^2): 0.9599751916687751
          Elastic Net Mean (R^2): 0.9631386201006957
          Ridge Mean (R^2): 0.9615636393810265
          Lasso Mean (R^2): 0.9599796819804313
          SVR Mean (R^2): 0.974325844746916
          Random Forest Mean (R^2): 0.9176625482870522
The best model r2 is: SVR with value: 0.974325844746916
          Linear Regression Mean (MSE): 34519.293368156315
          Elastic Net Mean (MSE): 33097.89373457309
          Ridge Mean (MSE): 33865.560670244384
          Lasso Mean (MSE): 34516.895795481345
          SVR Mean (MSE): 25604.457666635746
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 25604.457666635746
seed 747
[0.98540906 0.96121084 0.94767164 0.94784166 0.96893596]
          Linear Regression Mean (R^2): 0.9622138342635213
          Elastic Net Mean (R^2): 0.9672124567859081
          Ridge Mean (R^2): 0.9646852065982363
          Lasso Mean (R^2): 0.9622191168590352
          SVR Mean (R^2): 0.9706291614863121
          Random Forest Mean (R^2): 0.9118476358962779
The best model r2 is: SVR with value: 0.9706291614863121
          Linear Regression Mean (MSE): 31463.31560708335
          Elastic Net Mean (MSE): 29818.72500878909
          Ridge Mean (MSE): 30551.60582900634
          Lasso Mean (MSE): 31460.971634131834
          SVR Mean (MSE): 28474.274301608355
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 28474.274301608355
seed 748
[0.98952239 0.96367107 0.93093314 0.97389285 0.96235596]
          Linear Regression Mean (R^2): 0.964075082906277
          Elastic Net Mean (R^2): 0.9662440522134702
          Ridge Mean (R^2): 0.9652337906727425
          Lasso Mean (R^2): 0.9640790186342436
          SVR Mean (R^2): 0.9693228691317675
          Random Forest Mean (R^2): 0.9050877084718284
```

```
The best model r2 is: SVR with value: 0.9693228691317675
          Linear Regression Mean (MSE): 34734.99936670791
          Elastic Net Mean (MSE): 32992.60540015982
          Ridge Mean (MSE): 33728.16356161528
          Lasso Mean (MSE): 34732.40910365927
          SVR Mean (MSE): 27043.86780861313
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 27043.86780861313
seed 749
[0.96464814 0.94885197 0.97589321 0.92691711 0.97927186]
          Linear Regression Mean (R^2): 0.9591164581629709
          Elastic Net Mean (R^2): 0.9623886517137331
          Ridge Mean (R^2): 0.9606571492817642
          Lasso Mean (R^2): 0.9591241659722731
          SVR Mean (R^2): 0.9650560852326839
          Random Forest Mean (R^2): 0.886920538180408
The best model r2 is: SVR with value: 0.9650560852326839
          Linear Regression Mean (MSE): 33917.7557101066
          Elastic Net Mean (MSE): 32720.89131621083
          Ridge Mean (MSE): 33381.09321449342
          Lasso Mean (MSE): 33914.50811921414
          SVR Mean (MSE): 28381.460117760173
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 28381.460117760173
seed 750
[0.95496326 0.96780063 0.96585513 0.99104032 0.92974125]
          Linear Regression Mean (R^2): 0.9618801168374764
          Elastic Net Mean (R^2): 0.9675482483404931
          Ridge Mean (R^2): 0.9646192890217442
          Lasso Mean (R^2): 0.961890366929483
          SVR Mean (R^2): 0.9722506085906545
          Random Forest Mean (R^2): 0.9272556970262305
The best model r2 is: SVR with value: 0.9722506085906545
          Linear Regression Mean (MSE): 36630.58743274384
          Elastic Net Mean (MSE): 33439.05827377216
          Ridge Mean (MSE): 35144.685097188936
          Lasso Mean (MSE): 36624.00729671192
          SVR Mean (MSE): 27871.977008478996
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 27871.977008478996
seed 751
[0.94867056 0.98103435 0.97063512 0.94429098 0.98713013]
          Linear Regression Mean (R^2): 0.966352227946844
          Elastic Net Mean (R^2): 0.9713048928611748
          Ridge Mean (R^2): 0.9690506847372726
          Lasso Mean (R^2): 0.9663579704116575
          SVR Mean (R^2): 0.9733578735771363
          Random Forest Mean (R^2): 0.9349699185193066
The best model r2 is: SVR with value: 0.9733578735771363
          Linear Regression Mean (MSE): 31367.83742276253
          Elastic Net Mean (MSE): 28822.936832500745
          Ridge Mean (MSE): 29663.64618476297
          Lasso Mean (MSE): 31364.87074877423
          SVR Mean (MSE): 25947.738133671388
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 25947.738133671388
seed 752
[0.96301983 0.95136093 0.9319628  0.99710777 0.96190824]
          Linear Regression Mean (R^2): 0.9610719143847136
          Elastic Net Mean (R^2): 0.9682174266982985
          Ridge Mean (R^2): 0.9649732173048873
          Lasso Mean (R^2): 0.9610783896364946
          SVR Mean (R^2): 0.969963274095076
          Random Forest Mean (R^2): 0.8991816175669023
The best model r2 is: SVR with value: 0.969963274095076
          Linear Regression Mean (MSE): 34827.05962849044
          Elastic Net Mean (MSE): 30474.37757427867
          Ridge Mean (MSE): 32505.837105259387
          Lasso Mean (MSE): 34822.85218114066
          SVR Mean (MSE): 27051.949461503682
          Random Forest Mean (MSE):
```

```
The best model MSE is: SVR with value: 27051.949461503682
seed 753
[0.94771981 0.98112728 0.99118478 0.94195635 0.96699527]
          Linear Regression Mean (R^2): 0.9657966980368613
          Elastic Net Mean (R^2): 0.9681695018796954
          Ridge Mean (R^2): 0.9669529994728862
          Lasso Mean (R^2): 0.965800620290999
          SVR Mean (R^2): 0.972655256097229
          Random Forest Mean (R^2): 0.9070231056496197
The best model r2 is: SVR with value: 0.972655256097229
          Linear Regression Mean (MSE): 31377.81543637195
          Elastic Net Mean (MSE): 30336.970155668667
          Ridge Mean (MSE): 30900.976224363352
          Lasso Mean (MSE): 31374.948576932296
          SVR Mean (MSE): 25713.666442685266
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 25713.666442685266
seed 754
[0.97235962 0.92751384 0.96785062 0.98689556 0.92384156]
          Linear Regression Mean (R^2): 0.9556922397058848
          Elastic Net Mean (R^2): 0.959587407807662
          Ridge Mean (R^2): 0.9573225390672506
          Lasso Mean (R^2): 0.9556964385600628
          SVR Mean (R^2): 0.9672284470908068
          Random Forest Mean (R^2): 0.906858457071311
The best model r2 is: SVR with value: 0.9672284470908068
          Linear Regression Mean (MSE): 35424.200139594555
          Elastic Net Mean (MSE): 33772.42957385484
          Ridge Mean (MSE): 34658.17120281133
          Lasso Mean (MSE): 35422.17941663207
          SVR Mean (MSE): 26466.010444616026
          Random Forest Mean (MSE):
The best model MSE is: SVR with value: 26466.010444616026
```

APÊNDICE C – Detalhamento dos resultados comparativos dos algoritmos

Produto: 13M1S1

A Tabela 38, apresenta os valores de predição de venda obtidos após aplicação dos algoritmos ML para o produto 13M1S1, utilizando a *seed* 746. A coluna *y_test* apresenta os valores reais contidos no *dataset* e utilizados no modelo como base de teste. A coluna "Média Aritmética Simples" representa a média aritmética simples referente ao total de vendas no índice correspondente e será comparada juntamente com os demais modelos, mantendo a premissa desta pesquisa em selecionar o modelo mais simples seja em termos de desempenho ao aplicar as métricas R^2 e RMSE, complexidade ou tempo de execução.

Tabela 38 - Valores de predição de vendas para o produto 13M1S1

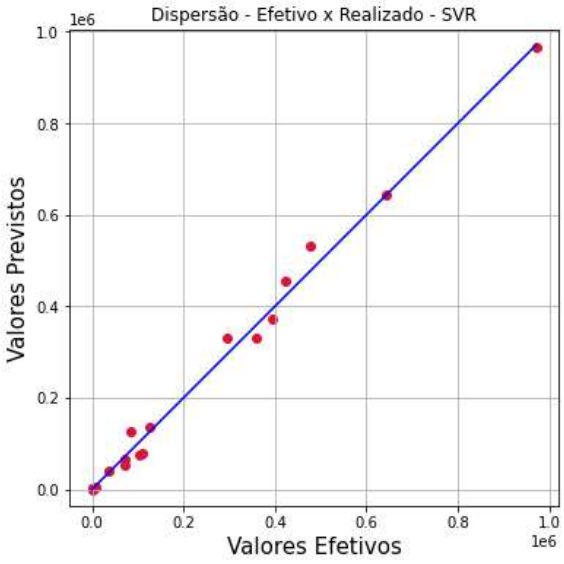
Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
4	72.655	95.141	60.693	52.624	95.182	95.454	95.743	32.799
7	1.870	55.668	1.917	1.897	55.726	55.319	54.390	655
10	71.367	86.360	65.922	67.094	86.351	86.615	87.025	39.359
11	1.704	51.033	1.796	1.762	51.100	51.132	50.983	655
15	6.098	39.309	6.079	5.015	39.333	38.869	37.960	3.279
17	102.736	112.366	83.605	74.754	112.378	111.781	110.605	62.319
19	108.890	115.579	86.036	77.160	115.583	114.480	112.435	70.847
20	85.420	60.344	109.795	127.715	60.210	60.090	60.348	119.391
28	1.187	28.373	1.317	1.184	28.417	28.743	29.231	655
31	37.681	35.105	33.888	41.105	35.069	35.122	35.329	29.519
34	394.172	374.823	397.583	371.233	374.831	375.329	376.162	355.551
36	358.162	329.785	339.398	330.440	329.822	329.858	329.939	351.615
50	476.771	494.532	576.822	532.379	494.602	493.751	492.498	631.727
55	294.454	288.121	364.639	331.968	288.198	289.072	290.853	406.719
58	641.883	612.697	669.655	645.277	612.890	612.855	612.944	797.039
59	127.111	107.861	160.688	134.657	107.854	107.838	108.033	163.999
65	971.111	954.988	839.010	966.460	955.077	954.854	954.411	1.016.799
66	422.304	432.939	512.505	454.681	432.963	433.272	433.918	478.879
74	433	-77.201	5.504	-1.224	-77.345	-77.524	-77.285	655

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared* (R^2), os resíduos e o gráfico de dispersão do modelo SVR são apresentados na Tabela 39, juntamente com a métrica de erro RMSE para o produto 13M1S1, evidenciando o melhor resultado, ou seja, coeficiente de determinação: 99,14% e a menor variação no índice de erro RMSE: 24.002.

Tabela 39 – Resíduos do modelo SVR para o produto 13M1S1

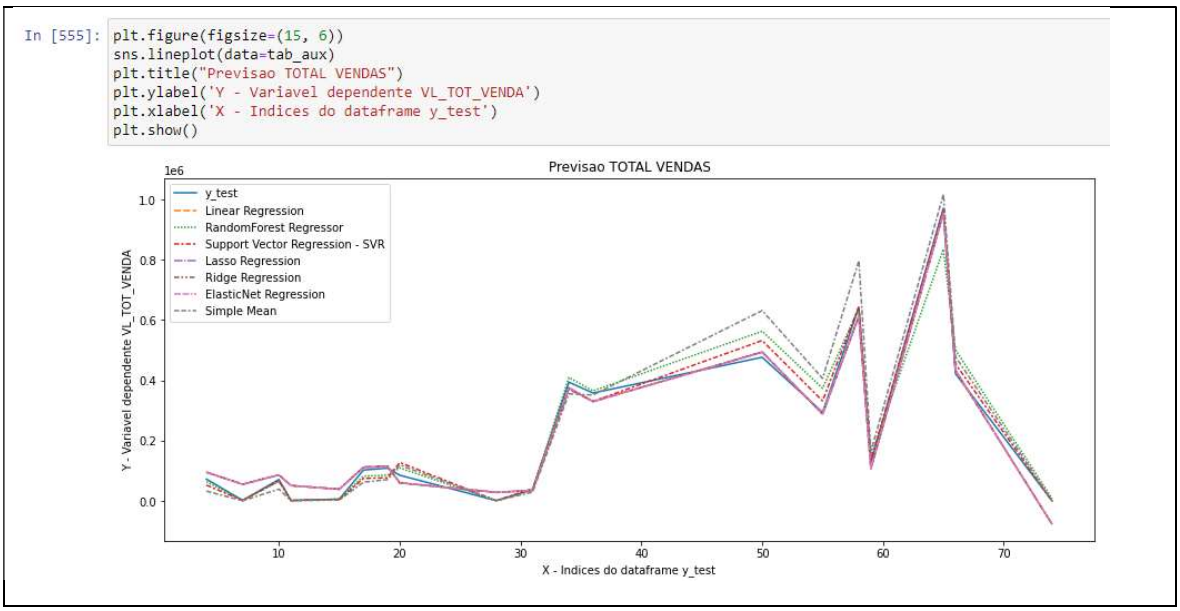
Idx	*y_test*	Support Vector Regression	Resíduos	R^2 = 0.9913870280036559 RMSE = 24.002
4	72.655	52.623	-20.032	
7	1.870	1.896	26	
10	71.367	67.093	-4.274	
11	1.704	1.761	57	
15	6.098	5.015	-1.083	
17	102.736	74.754	-27.982	
19	108.890	77.159	-31.731	
20	85.420	127.714	42.294	
28	1.187	1.184	-3	
31	37.681	41.104	3.423	
34	394.172	371.233	-22.939	
36	358.162	330.440	-27.722	
50	476.771	532.379	55.608	
55	294.454	331.967	37.513	
58	641.883	645.276	3.393	
59	127.111	134.657	7.546	
65	971.111	966.459	-4.652	
66	422.304	454.681	32.377	
74	433	-1.223	-1.656	



Fonte: Resultados da Pesquisa

Na Figura 54, é apresentado o comparativo do coeficiente de determinação *R-squared* (R^2), além da variação no índice de erro RMSE e o respectivo gráfico de todos os modelos aplicados no *dataset* referente ao produto 13M1S1.

Figura 54 - Comparativo entre os modelos para o produto 13M1S1



	Linear Regression	Random Forest Regressor	**Support Vector Regression**	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
R^2	0.9859703	0.9676439	**0.9913870**	0.9859438	0.9859974	0.9862212	0.9403611
RMSE	30.634	46.522	**24.002**	30.663	30.604	30.359	63.161

Fonte: Resultados da Pesquisa

Os resultados evidenciam que, para 10 divisões diferentes de conjuntos de dados, usando CV externa ao longo de dez execuções experimentais no conjunto de dados 13M1S1, o modelo SVR apresentou o melhor coeficiente de determinação e o menor índice de erro. De acordo com o teste de *Shapiro-Wilk* com a hipótese nula de que os dados seguem uma distribuição normal, obtendo um *p-valor* 2.10683^{e-05} a hipótese nula foi rejeitada, desta forma foi aplicado o teste de *Spearman* para avaliar o nível de correlação entre as várias dependentes e independentes, evidenciando que o conjunto de dados atende a premissa de relação de independência, na qual a relação entre a variável dependente e independente é linear, não havendo correlação entre as variáveis independentes.

Para atender a premissa de regressão linear referente a variância dos resíduos, foi aplicado o teste paramétrico de *Breusch-Pagan*, cuja hipótese nula foi aceita com *p-valor* 0.05633, indicando homocedasticidade na distribuição dos resíduos.

Produto: 0-0001

A Tabela 40, apresenta os valores de predição de venda obtidos para as primeiras 19 observações, após aplicação dos algoritmos ML para o produto 0-0001, utilizando a *seed* 748. A coluna *y_test* apresenta os valores reais contidos no *dataset* e utilizados no modelo como base de teste. A coluna "Média Aritmética Simples" representa a média aritmética simples referente ao total de vendas no índice correspondente e será comparada juntamente com os demais modelos, mantendo a premissa desta pesquisa em selecionar o modelo mais simples seja em termos de desempenho ao aplicar as métricas R^2 e RMSE, complexidade ou tempo de execução.

Tabela 40 - Valores de predição de vendas para o produto 0-0001

Idx	**y_test**	**Linear Regression**	**Random Forest Regressor**	**Support Vector Regression**	**Lasso Regression**	**Ridge Regression**	**ElasticNet Regression**	**Média Aritmética Simples**
0	276	198	307	135	221	197	187	435
2	206	-221	285	-266	-194	-232	-308	362
6	503	513	517	484	535	517	549	652
8	2.000	2.002	2.191	2.105	2.008	2.007	2.041	2.394
14	1.030	752	1.210	939	778	750	737	1.305

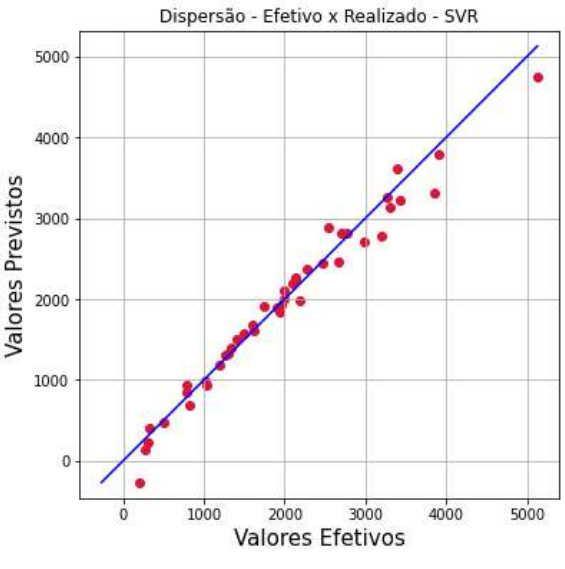
Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
15	823	575	810	689	602	577	595	1.088
19	1.015	949	1.144	988	967	949	951	1.378
34	1.190	1.205	1.282	1.191	1.214	1.197	1.138	1.886
38	1.300	1.168	1.277	1.325	1.194	1.173	1.216	2.176
49	3.392	3.428	3.158	3.607	3.422	3.433	3.472	4.716
51	2.135	2.177	2.144	2.257	2.180	2.179	2.193	2.902
55	2.141	2.220	2.144	2.268	2.220	2.219	2.206	2.902
59	1.504	1.524	1.461	1.572	1.534	1.524	1.524	2.104
72	3.296	3.134	3.128	3.140	3.122	3.134	3.133	4.135
73	1.613	1.660	1.598	1.619	1.664	1.662	1.678	2.031
80	1.269	1.475	853	1.305	1.470	1.470	1.437	1.378
81	1.937	2.081	1.620	1.845	2.062	2.068	1.967	2.104
82	2.663	2.639	2.801	2.457	2.615	2.626	2.533	2.902
92	3.433	3.236	3.397	3.221	3.204	3.227	3.160	2.539

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared* (R^2), os resíduos e o gráfico de dispersão do modelo SVR são apresentados na Tabela 41, juntamente com a métrica de erro RMSE para o produto 0-0001, evidenciando o melhor resultado, ou seja, coeficiente de determinação: 97,17% e a menor variação no índice de erro RMSE: 185.

Tabela 41 - Resíduos do modelo LR para o produto 0-0001

Idx	*y_test*	SVR	Resíduos	**R^2 = 0.9716643815242761**
				RMSE = 185
0	276	135	-141	
2	206	-265	-471	
6	503	483	-20	
8	2.000	2.104	104	
14	1.030	938	-92	
15	823	689	-134	
19	1.015	988	-27	
34	1.190	1.190	0	
38	1.300	1.325	25	
49	3.392	3.607	215	
51	2.135	2.256	121	
55	2.141	2.267	126	
59	1.504	1.571	67	
72	3.296	3.139	-157	
73	1.613	1.618	5	
80	1.269	1.304	35	
81	1.937	1.844	-93	

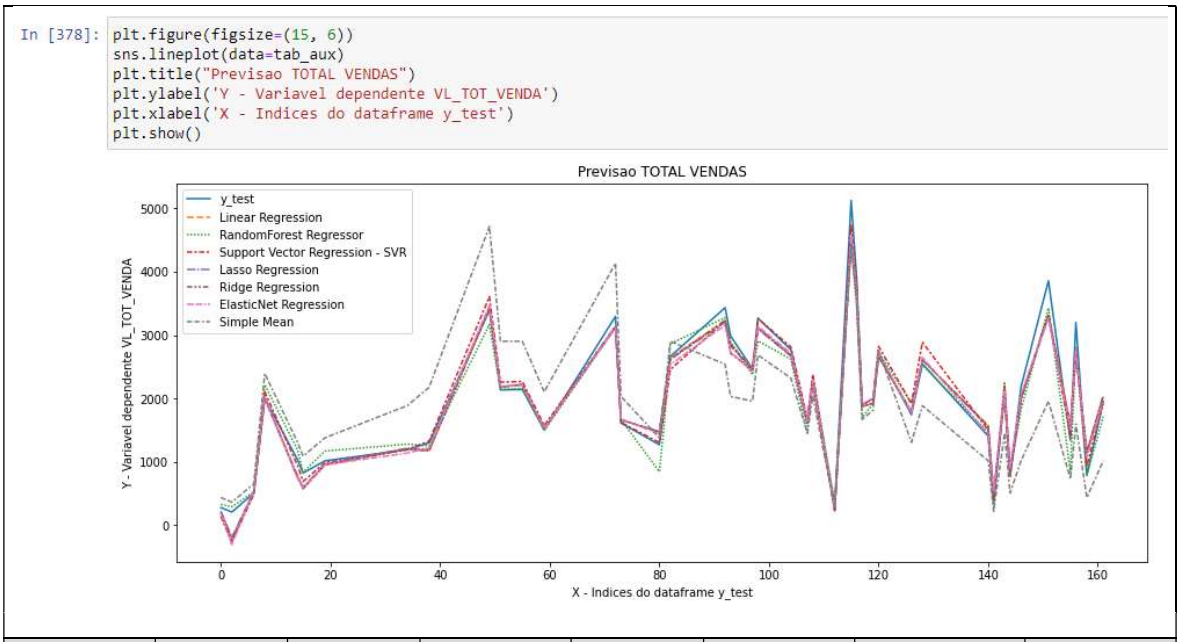


Idx	*y_test*	SVR	Resíduos	R^2 = 0.9716643815242761
				RMSE = 185
82	2.663	2.456	-207	
92	3.433	3.220	-213	

Fonte: Resultados da Pesquisa

Na Figura 55, é apresentado o comparativo do coeficiente de determinação *R-squared* (R^2), além da variação no índice de erro RMSE e o respectivo gráfico de todos os modelos aplicados no *dataset* referente ao produto 0-0001.

Figura 55 - Comparativo entre os modelos para o produto 0-0001



	Linear Regression	Random Forest Regressor	**Support Vector Regression**	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
R^2	0.9655112	0.9558832	**0.9716644**	0.9643137	0.9659043	0.9658657	0.6257375
RMSE	204	231	**185**	208	203	203	673

Fonte: Resultados da Pesquisa

Os resultados evidenciam que, para 10 divisões diferentes de conjuntos de dados, usando CV externa ao longo de dez execuções experimentais no conjunto de dados 0-0001, o modelo SVR apresentou o melhor coeficiente de determinação e o menor índice de erro. De acordo como o teste de *Shapiro-Wilk* com a hipótese nula de que os dados seguem uma distribuição normal, obtendo um *p-valor* 0.015922 a hipótese nula foi rejeitada, desta forma foi aplicado o teste de *Spearman* para avaliar o nível de correlação entre as várias dependentes e independentes, evidenciando que o conjunto de dados atende a

premissa de relação de independência, na qual a relação entre a variável dependente e independente é linear, não havendo correlação entre as variáveis independentes.

Para atender a premissa de regressão linear referente a variância dos resíduos, foi aplicado o teste paramétrico de *Breusch-Pagan*, cuja hipótese nula foi rejeitada com *p-valor* 0.00597, sendo necessário aplicar o teste paramétrico de *Bartlett*, na qual o *dataset* foi divido em 12 grupos representando o faturamento mensal, neste caso a hipótese nula foi aceita, indicando homocedasticidade na distribuição dos resíduos quando analisados por grupo.

Produto: 0-B051

A Tabela 42, apresenta os valores de predição de venda obtidos para as primeiras 19 observações, após aplicação dos algoritmos ML para o produto 0-B051, utilizando a *seed* 750. A coluna *y_test* apresenta os valores reais contidos no *dataset* e utilizados no modelo como base de teste. A coluna "Média Aritmética Simples" representa a média aritmética simples referente ao total de vendas no índice correspondente e será comparada juntamente com os demais modelos, mantendo a premissa desta pesquisa em selecionar o modelo mais simples seja em termos de desempenho ao aplicar as métricas R^2 e RMSE, complexidade ou tempo de execução.

Tabela 42 - Valores de predição de vendas para o produto 0-B051

Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
6	412.978	433.221	412.908	411.421	433.204	432.904	431.327	563.873
9	247.583	244.432	243.591	212.598	244.424	244.279	243.415	264.533
16	300.095	284.739	284.949	251.833	284.719	284.386	282.401	276.135
20	70.533	114.685	58.520	53.381	114.672	114.531	113.431	44.088
26	97.813	121.679	97.356	76.707	121.675	121.709	121.653	67.293
30	153.958	154.774	162.867	133.731	154.778	154.897	155.479	129.946
35	266.329	259.386	282.160	255.567	259.386	259.454	259.816	266.853
36	106.217	120.108	112.819	88.365	120.114	120.396	121.760	81.216
37	10.965	-8.155	36.674	19.921	-8.109	-7.907	-6.186	11.602
39	71.318	71.759	78.718	71.420	71.792	72.574	76.898	64.973
41	292.249	285.550	276.849	292.239	285.552	285.824	287.269	294.699
44	226.582	219.487	239.069	227.607	219.497	219.683	220.795	227.405
50	294.783	296.947	326.916	261.638	296.921	295.919	290.566	331.826
51	66.739	77.079	76.750	67.280	77.099	77.264	78.341	74.254
53	299.152	306.865	314.424	305.022	306.862	306.628	305.519	364.313
60	479.118	498.027	421.440	483.184	497.995	497.549	495.022	580.117
67	269.913	275.416	314.998	296.116	275.421	275.256	274.640	313.263
68	315.847	317.268	323.335	307.154	317.247	316.473	312.367	329.506

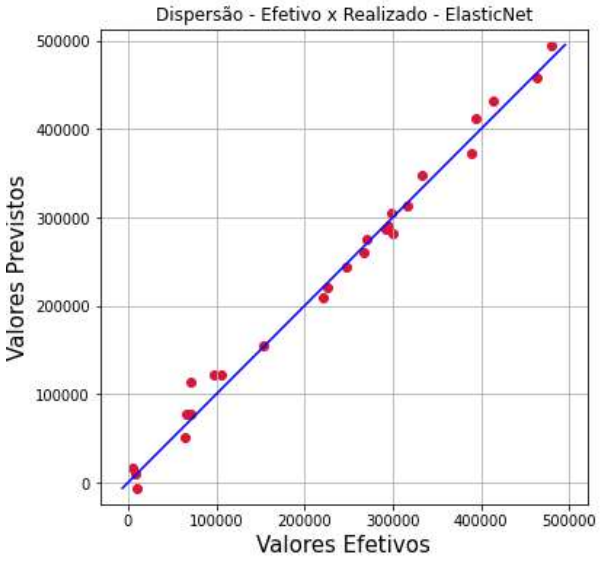
Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
72	64.957	46.327	80.665	74.213	46.375	47.153	51.771	69.614

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared* (R^2), os resíduos e o gráfico de dispersão do modelo ElasticNET são apresentados na Tabela 43, juntamente com a métrica de erro RMSE para o produto 0-B051, evidenciando o melhor resultado, ou seja, coeficiente de determinação: 98,97% e a menor variação no índice de erro RMSE: 14.507.

Tabela 43 - Resíduos do modelo ElasticNET para o produto 0-B051

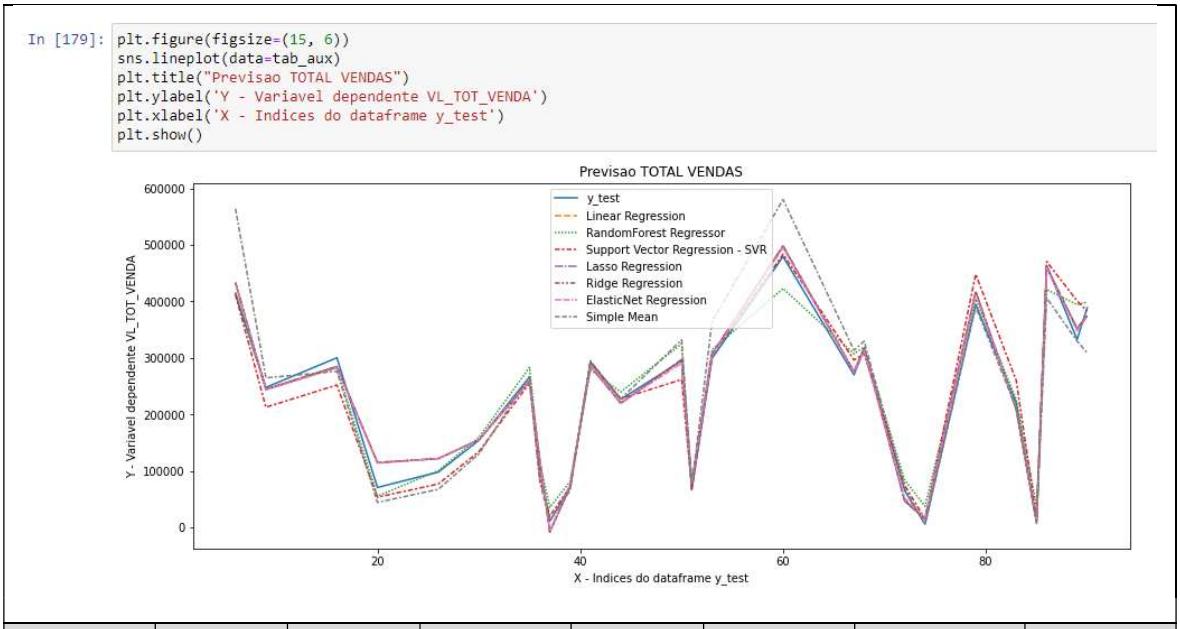
Idx	*y_test*	ElasticNet Regression	Resíduos	R^2 score = 0.989692257246217
				RMSE = 14.507
6	412.978	431.326	18.348	
9	247.583	243.414	-4.169	
16	300.095	282.400	-17.695	
20	70.533	113.430	42.897	
26	97.813	121.653	23.840	
30	153.958	155.479	1.521	
35	266.329	259.816	-6.513	
36	106.217	121.760	15.543	
37	10.965	-6.185	-17.150	
39	71.318	76.898	5.580	
41	292.249	287.269	-4.980	
44	226.582	220.795	-5.787	
50	294.783	290.565	-4.218	
51	66.739	78.340	11.601	
53	299.152	305.519	6.367	
60	479.118	495.022	15.904	
67	269.913	274.640	4.727	
68	315.847	312.366	-3.481	
72	64.957	51.770	-13.187	



Fonte: Resultados da Pesquisa

Na Figura 56, é apresentado o comparativo do coeficiente de determinação *R-squared* (R^2), além da variação no índice de erro RMSE e o respectivo gráfico de todos os modelos aplicados no *dataset* referente ao produto 0-B051.

Figura 56 - Comparativo entre os modelos para o produto 0-B051



	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	**ElasticNet Regression**	Média Aritmética Simples
R^2 score	0.9884350	0.9700913	0.9687610	0.9884473	0.9886798	**0.9896923**	0.8977438
RMSE	15.366	24.711	25.254	15.358	15.202	**14.507**	45.692

Fonte: Resultados da Pesquisa

Os resultados evidenciam que, para 10 divisões diferentes de conjuntos de dados, usando CV externa ao longo de dez execuções experimentais no conjunto de dados 0-B051, o modelo ElasticNET apresentou o melhor coeficiente de determinação e o menor índice de erro. De acordo como o teste de *Shapiro-Wilk* com a hipótese nula de que os dados seguem uma distribuição normal, obtendo um *p-valor* 0.085142 a hipótese nula foi aceita, desta forma foi aplicado o teste de *Pearson* para avaliar o nível de correlação entre as várias dependentes e independentes, evidenciando que o conjunto de dados atende a premissa de relação de independência, na qual a relação entre a variável dependente e independente é linear, não havendo correlação entre as variáveis independentes.

Para atender a premissa de regressão linear referente a variância dos resíduos, foi aplicado o teste paramétrico de *Breusch-Pagan*, cuja hipótese nula foi rejeitada com *p-valor* 4.5575^{e-05}, sendo necessário aplicar o teste paramétrico de *Bartlett*, na qual o *dataset* foi divido em 12 grupos representando o faturamento mensal, neste caso a hipótese nula foi aceita, indicando homocedasticidade na distribuição dos resíduos quando analisados por grupo.

Produto: 8-K011

A Tabela 44, apresenta os valores de predição de venda obtidos para as primeiras 19 observações, após aplicação dos algoritmos ML para o produto 8-K011, utilizando a *seed* 747. A coluna *y_test* apresenta os valores reais contidos no *dataset* e utilizados no modelo como base de teste. A coluna "Média Aritmética Simples" representa a média aritmética simples referente ao total de vendas no índice correspondente e será comparada juntamente com os demais modelos, mantendo a premissa desta pesquisa em selecionar o modelo mais simples seja em termos de desempenho ao aplicar as métricas R^2 e RMSE, complexidade ou tempo de execução.

Tabela 44 - Valores de predição de vendas para o produto 8-K011

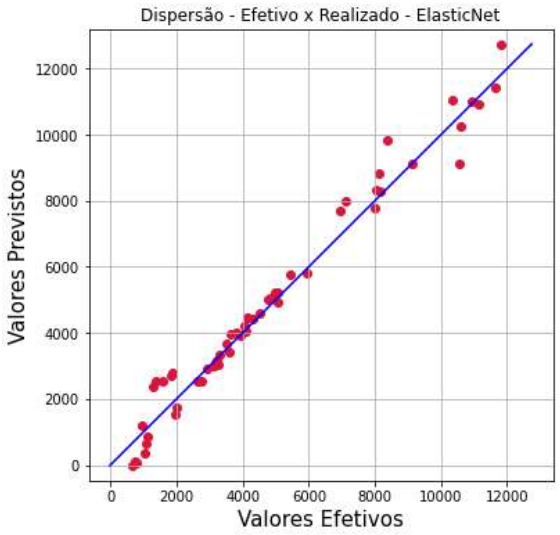
Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
0	659	208	678	460	222	182	-17	610
4	1.076	782	1.257	1.021	798	767	655	916
5	1.021	575	1.251	827	589	551	368	916
6	3.297	3.400	3.586	3.299	3.402	3.392	3.333	2.442
9	4.091	4.107	4.366	4.085	4.112	4.101	4.066	3.969
14	3.251	3.208	3.103	3.240	3.213	3.189	3.048	3.358
18	1.941	1.536	2.205	1.864	1.557	1.534	1.523	2.137
19	1.992	1.751	2.073	1.991	1.769	1.748	1.737	2.137
20	2.649	2.458	2.774	2.688	2.477	2.466	2.541	2.748
21	3.118	2.857	3.187	3.136	2.878	2.870	2.981	3.358
22	3.158	2.839	3.368	3.188	2.866	2.868	3.100	3.053
24	8.136	8.731	9.616	8.638	8.732	8.742	8.847	9.160
25	4.049	4.094	3.927	4.015	4.104	4.106	4.203	3.969
27	4.531	4.565	5.327	4.587	4.575	4.568	4.605	5.191
28	755	167	955	594	192	162	126	916
29	3.212	3.164	3.234	3.249	3.176	3.162	3.152	3.664
34	773	45	973	583	76	48	84	916
35	5.044	5.150	5.148	5.259	5.163	5.155	5.206	6.412
47	2.939	2.940	3.244	3.011	2.953	2.935	2.899	4.275

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared* (R^2), os resíduos e o gráfico de dispersão do modelo ElasticNET são apresentados na Tabela 45, juntamente com a métrica de erro RMSE para o produto 8-K011, evidenciando o melhor resultado, ou seja, coeficiente de determinação: 97,18% e a menor variação no índice de erro RMSE: 538.

Tabela 45 - Resíduos do modelo ElasticNet para o produto 8-K011

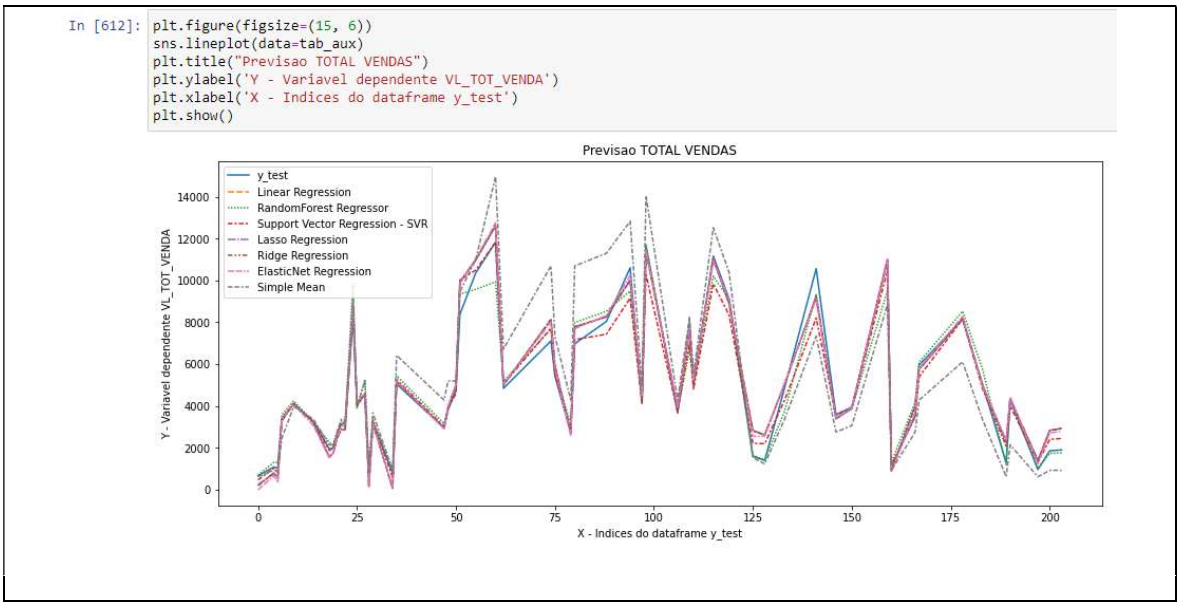
Idx	*y_test*	ElasticNet Regression	Resíduos	R^2 score = 0.9717601146446426
				RMSE = 538
0	659	-16	-675	
4	1.076	655	-421	
5	1.021	368	-653	
6	3.297	3.333	36	
9	4.091	4.066	-25	
14	3.251	3.047	-204	
18	1.941	1.522	-419	
19	1.992	1.737	-255	
20	2.649	2.541	-108	
21	3.118	2.981	-137	
22	3.158	3.100	-58	
24	8.136	8.847	711	
25	4.049	4.202	153	
27	4.531	4.605	74	
28	755	125	-630	
29	3.212	3.151	-61	
34	773	84	-689	
35	5.044	5.206	162	
47	2.939	2.898	-41	



Fonte: Resultados da Pesquisa

Na Figura 57 é apresentado o comparativo do coeficiente de determinação *R-squared* (R^2), além da variação no índice de erro RMSE e o respectivo gráfico de todos os modelos aplicados no *dataset* referente ao produto 8-K011.

Figura 57 - Comparativo entre os modelos para o produto 8-K011



	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	**ElasticNet Regression**	Média Aritmética Simples
R^2 score	0.9698459	0.9626484	0.9639821	0.9702801	0.9702140	**0.9717601**	0.7912402
RMSE	556	619	608	552	553	**538**	1.464

Fonte: Resultados da Pesquisa

Os resultados evidenciam que, para 10 divisões diferentes de conjuntos de dados, usando CV externa ao longo de dez execuções experimentais no conjunto de dados 8-K011, o modelo LR apresentou o melhor coeficiente de determinação e o menor índice de erro. De acordo com o teste de *Shapiro-Wilk* com a hipótese nula de que os dados seguem uma distribuição normal, obtendo um *p-valor* 3.281^{e-07} a hipótese nula foi rejeitada, desta forma foi aplicado o teste de *Spearman* para avaliar o nível de correlação entre as várias dependentes e independentes, evidenciando que o conjunto de dados atende a premissa de relação de independência, na qual a relação entre a variável dependente e independente é linear, não havendo correlação entre as variáveis independentes.

Para atender a premissa de regressão linear referente a variância dos resíduos, foi aplicado o teste paramétrico de *Breusch-Pagan*, cuja hipótese nula foi rejeitada com *p-valor* 3.7762^{e-05}, sendo necessário aplicar o teste paramétrico de *Bartlett*, na qual o *dataset* foi divido em 12 grupos representando o faturamento mensal, neste caso a hipótese nula foi aceita, indicando homocedasticidade na distribuição dos resíduos quando analisados por grupo.

Produto: 1-B301

A Tabela 46, apresenta os valores de predição de venda obtidos após aplicação dos algoritmos ML para o produto 1-B301, utilizando a *seed* 754. A coluna *y_test* apresenta os valores reais contidos no dataset e utilizados no modelo como base de teste. A coluna "Média Aritmética Simples" representa a média aritmética simples referente ao total de vendas no índice correspondente e será comparada juntamente com os demais modelos, mantendo a premissa desta pesquisa em selecionar o modelo mais simples seja em termos de desempenho ao aplicar as métricas R^2 e RMSE, complexidade ou tempo de execução.

Tabela 46 - Valores de predição de vendas para o produto 1-B301

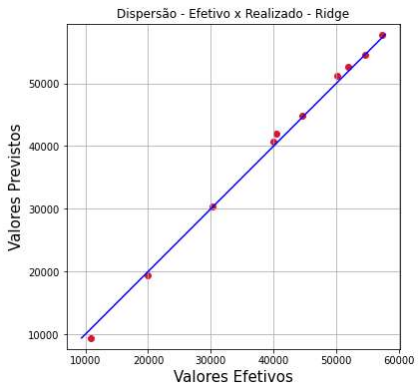
Idx	**y_test**	**Linear Regression**	**Random Forest Regressor**	**Support Vector Regression**	**Lasso Regression**	**Ridge Regression**	**ElasticNet Regression**	**Média Aritmética Simples**
6	49.380	48.601	52.241	39.170	48.572	48.185	48.193	47.364
13	48.391	49.360	52.859	45.212	49.360	49.396	49.378	49.423
15	12.052	12.783	11.455	12.063	12.777	12.655	12.752	12.355

Idx	y_test	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	Ridge Regression	ElasticNet Regression	Média Aritmética Simples
16	40.403	40.815	35.279	36.350	40.827	41.139	41.206	41.186
17	44.496	45.249	43.737	39.735	45.253	45.356	45.302	45.304
26	51.900	52.142	54.522	51.922	52.114	51.798	52.243	51.482
29	51.948	50.449	52.425	53.758	50.463	51.215	52.271	51.482
30	42.830	41.160	44.009	45.307	41.206	42.386	43.155	43.245
31	72.661	70.860	69.754	74.752	70.869	72.054	74.025	72.075
34	10.435	9.561	11.980	11.774	9.587	9.532	8.805	10.296

Fonte: Resultados da Pesquisa

O resultado do coeficiente de determinação *R-squared* (R^2), os resíduos e o gráfico de dispersão do modelo *Ridge Regression* são apresentados na Tabela 47, juntamente com a métrica de erro RMSE para o produto 1-B301, evidenciando o melhor resultado, ou seja, coeficiente de determinação: 99,81% e a menor variação no índice de erro RMSE: 775.

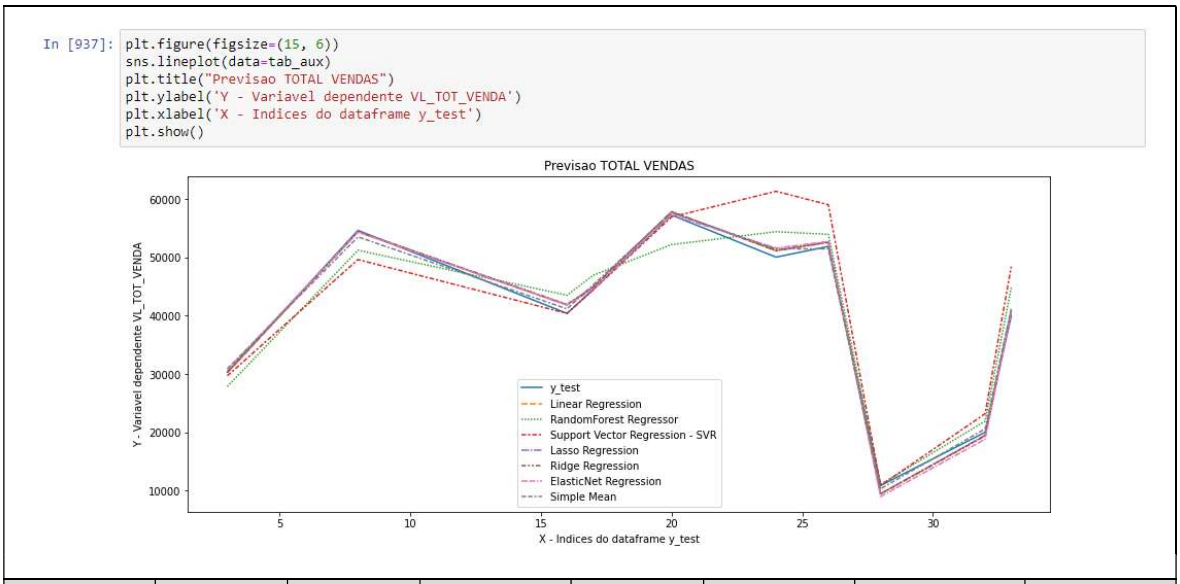
Tabela 47 - Resíduos do modelo RR para o produto 1-B301

Idx	*y_test*	Ridge Regression	Resíduos	**R^2 score = 0.9980810955285695**
				RMSE = 775
6	49.380	48.184	-1.196	
13	48.391	49.396	1.005	
15	12.052	12.654	602	
16	40.403	41.138	735	
17	44.496	45.355	859	
26	51.900	51.797	-103	
29	51.948	51.215	-733	
30	42.830	42.386	-444	
31	72.661	72.053	-608	
34	10.435	9.531	-904	

Fonte: Resultados da Pesquisa

Na Figura 58Figura 58, é apresentado o comparativo do coeficiente de determinação *R-squared* (R^2), além da variação no índice de erro RMSE e o respectivo gráfico de todos os modelos aplicados no *dataset* referente ao produto 1-B301.

Figura 58 - Comparativo entre os modelos para o produto 1-B301



	Linear Regression	Random Forest Regressor	Support Vector Regression	Lasso Regression	**Ridge Regression**	ElasticNet Regression	Média Aritmética Simples
R^2 score	0.9961951	0.9699347	0.9460529	0.9962689	**0.9980811**	0.9971229	0.9976336
RMSE	1.091	3.069	4.111	1.081	**775**	949	861

Fonte: Resultados da Pesquisa

Os resultados evidenciam que, para 10 divisões diferentes de conjuntos de dados, usando CV externa ao longo de dez execuções experimentais no conjunto de dados 0-B301, o modelo RR apresentou o melhor coeficiente de determinação e o menor índice de erro. De acordo com o teste de *Shapiro-Wilk* com a hipótese nula de que os dados seguem uma distribuição normal, obtendo um *p-valor* 0.293697 a hipótese nula foi aceita, desta forma foi aplicado o teste de *Pearson* para avaliar o nível de correlação entre as várias dependentes e independentes, evidenciando que o conjunto de dados atende a premissa de relação de independência, na qual a relação entre a variável dependente e independente é linear, não havendo correlação entre as variáveis independentes.

Para atender a premissa de regressão linear referente a variância dos resíduos, foi aplicado o teste paramétrico de *Breusch-Pagan*, cuja hipótese nula foi rejeitada com *p-valor* 0.0223, sendo necessário aplicar o teste paramétrico de *Bartlett*, na qual o *dataset* foi divido em 12 grupos representando o faturamento mensal, sendo que a hipótese nula foi rejeitada com *p-valor* nulo, desta forma o *dataset* foi divido em 2 grupos representando o faturamento semestral e aplicado o teste *T*, sendo que a hipótese nula foi aceita com *p-valor* 0.9439, indicando homocedasticidade na distribuição dos resíduos.

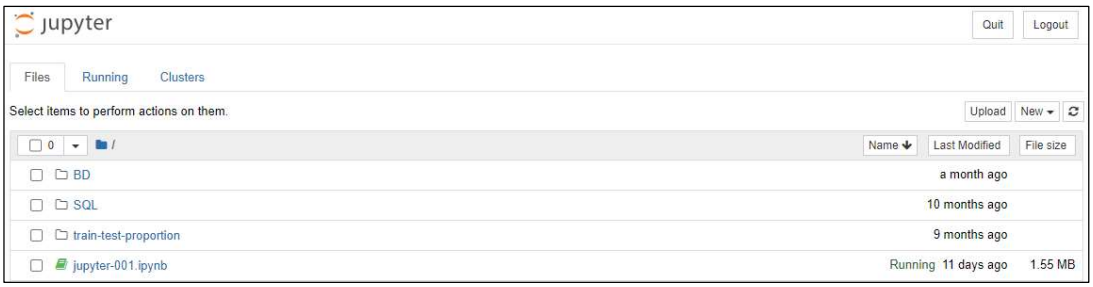
APÊNDICE D – Roteiro de uso do software

Para que seja possível executar a solução computacional é necessário que o ambiente esteja com a versão do Python 3.9 ou superior, cujo *download* está disponível em https://www.python.org/downloads, sendo necessário também a instalação do *Jupyter Notebook*, cuja instalação é realizada de forma on-line através da ferramenta de gerenciamento de pacotes PIP por meio do comando "*python -m pip install notebook*". Maiores informações sobre a instalação e o projeto *Jupyter*, podem ser obtidas por meio do site https://jupyter.org/install.

Adicionalmente será necessário realizar *download* do arquivo "jupyter-001.ipynb" disponível no site *github* conforme URL destacada na seção 3.3.4.

Ao executar o comando "*python Jupyter notebook*", será apresentada a tela inicial conforme Figura 59 - Tela inicial do *Jupyter Notebook*Figura 59.

Figura 59 - Tela inicial do *Jupyter Notebook*

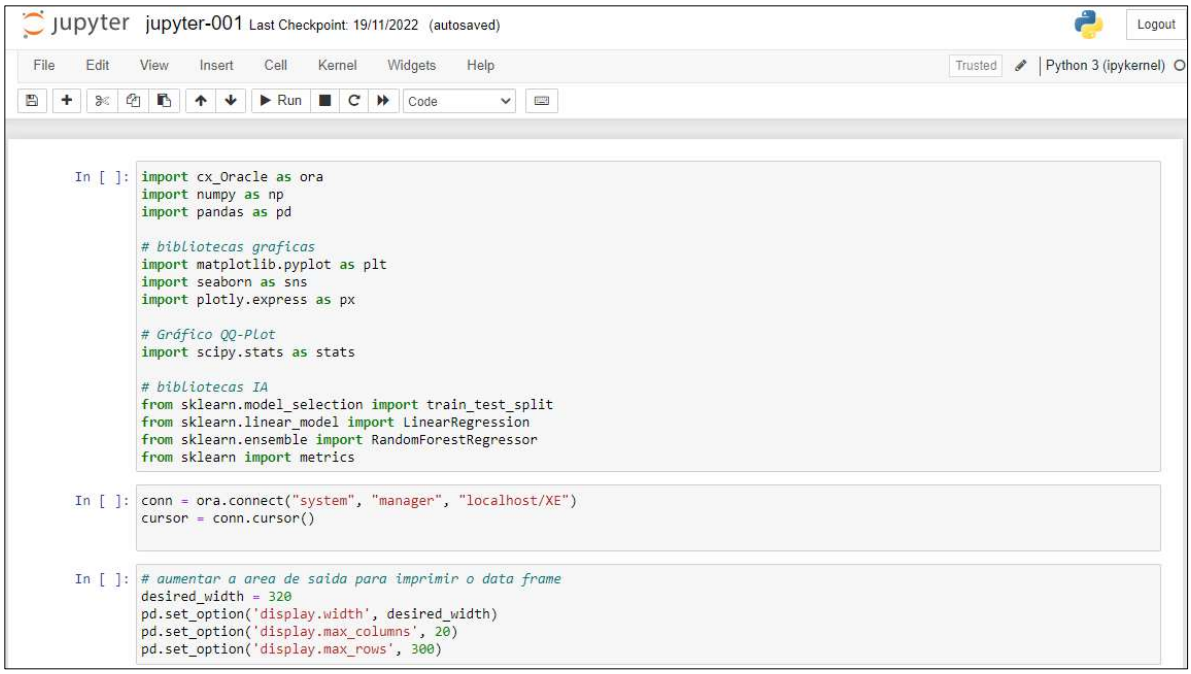


Fonte: Resultados da Pesquisa

Para ter acesso ao código fonte da aplicação, utilize a opção de menu "Files" e selecione o arquivo "jupyter-001.ipynb".

Após abrir o arquivo selecionado, será apresentada a tela conforme Figura 60, destacando-se o segundo quadro, por meio do qual deve ser informada a *string* de conexão com o bando de dados. Neste protótipo foi importada a biblioteca *cx_Oracle* devido a utilização do banco de dados *Oracle*, porém o pacote pode ser alterado conforme o banco de dados que precise conectar.

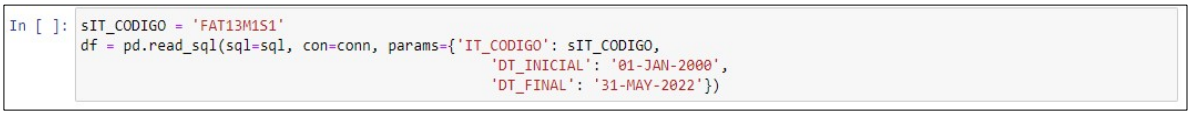
Figura 60 - String de conexão com o banco de dados



Fonte: Resultados da Pesquisa

No próximo passo, devem ser preenchidos o código do produto que deseja analisar, assim como o período de análise conforme destacado na Figura 61.

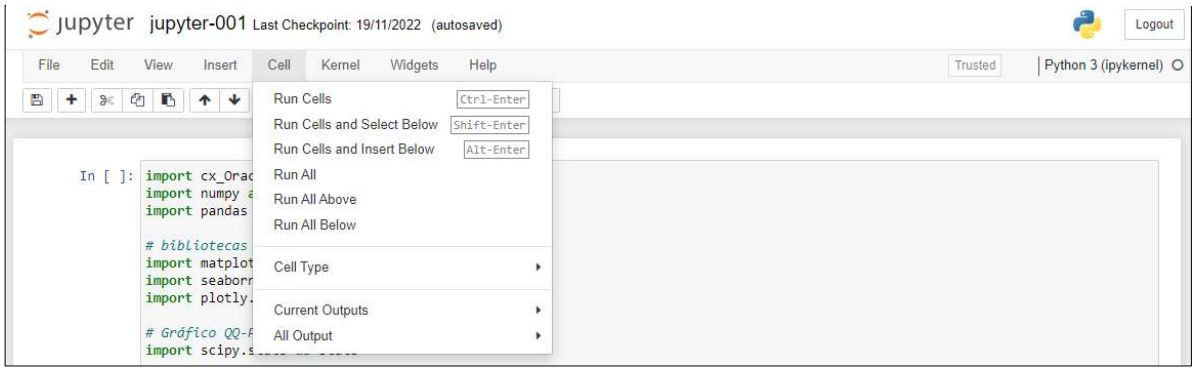
Figura 61 - Parâmetros de filtro do *Dataset*



Fonte: Resultados da Pesquisa

Uma vez estabelecido a conexão com o banco dados e preenchido os parâmetros de filtro, selecione a opção de menu *Cell* e a opção do submenu *Run All* conforme Figura 62.

Figura 62 - Opção de menu *Run All*



Fonte: Resultados da Pesquisa

A partir deste momento a solução computacional realiza a carga dos dados em seu estado bruto com os respectivos parâmetros de filtro. Na fase de pré-processamento dos dados os valores nulos são removidos e o diagrama de caixa boxplot é aplicado para identificar e remover valores *outliers*, na sequência as escalas das variáveis são padronizadas para tornar os dados comparáveis como pré-requisito para o treinamento dos modelos, na próxima etapa é aplicado a redução da dimensionalidade para escolha das variáveis mais significativas para o modelo, na etapa seguinte as variáveis independentes correlacionadas são removidas para evitar multicolinearidade, na sequência os dados são particionados utilizando a técnica k-fold para aplicação da validação cruzada para diminuir a possibilidade de *Overfitting* e *Underfitting*, após o treinamento, os resíduos de cada modelo são avaliados quanto a sua homocedasticidade e caso apresentem indícios de heterocedasticidade, são aplicados os testes paramétrios: *Breusch-Pagan*, *Bartlett* e Teste *t*. Finalmente é aplicado a validação cruzada externa, no qual o modelo com a melhor desempenho é obtido por meio do coeficiente de determinação R^2 mais alto e o menor índice de erro RMSE. Todas estas etapas estão detalhadas na seção 3.4.