MACHINE LEARNING COM BANCO DE DADOS DATA WAREHOUSE BIGQUERY: criação de um modelo para precificação do valor de um imóvel

Rhaan Dias de Oliveira
Graduando em Banco de Dados pela Fatec Bauru
rhaan.oliveira@fatec.sp.gov.br

Thiago Gabriel de Souza Alves de Carvalho
Graduando em Banco de Dados pela Fatec Bauru
thiago.carvalho42@fatec.sp.gov.br

Wagner Leitão de Oliveira
Graduando em Banco de Dados pela Fatec Bauru
wagner.oliveira29@fatec.sp.gov.br

Orientador: Luis Alexandre da Silva
Docente na Fatec Bauru
luis.silva51@fatec.sp.gov.br

RESUMO

O mercado imobiliário é um setor de destaque dentro da economia brasileira, sendo responsável por gerar empregos e renda no país. Apesar de ser um setor bastante rentável, a precificação do imóvel é uma tarefa complexa que envolve diversos aspectos como localização, vizinhança, tipo de construção, segurança, estado de conservação, entre outros aspectos. O objetivo desta pesquisa é de criar um modelo de *machine learning* para precificar o valor de um imóvel a partir de dados históricos e públicos de vendas de casas armazenados em um banco de dados *data warehouse*. O modelo foi construído via linguagem *Structured Query Language* (SQL) no *data warehouse* BigQuery do Google, utilizando o BigQuery ML, uma ferramenta de construção de modelos de *machine learning* em nuvem. Dessa forma, possibilitando treinar um modelo que possa auxiliar na precificação de novos imóveis. Conclui-se com este trabalho que o modelo foi criado com sucesso utilizando o BigQuery ML, e a sua aplicação pode auxiliar na tarefa de precificação de imóveis.

Palavras-chave: precificação; *machine learning*; *data warehouse*; BigQuery; SQL.

1 INTRODUÇÃO

O mercado imobiliário é um dos setores de destaque da economia brasileira. De acordo com os indicadores imobiliários nacionais, fornecidos pelo site oficial da Câmara Brasileira da Indústria da Construção (CBIC), só no ano de 2022, o setor foi responsável por movimentar R$ 144 bilhões de Valor Geral de Vendas (VGV), gerando empregos e renda para o país (CBIC, 2022).

De acordo com o site oficial da empresa Vista, referência nacional no desenvolvimento de tecnologia voltada ao mercado imobiliário, um dos maiores desafios desse setor é precificar corretamente o valor do imóvel, isso porque o mercado é bastante dinâmico e exige um acompanhamento constante das principais tendências e perspectivas. O referido texto cita ainda o especialista Gilberto Yogui, vice-presidente do Conselho Regional de Corretores de Imóveis do Estado de São Paulo (CRECISP), que explica sobre os principais fatores que influenciam diretamente no valor do imóvel, que são: localização e vizinhança, tipo de construção, infraestrutura de tecnologia e lazer, proximidade com lojas e serviços, segurança,

estrutura do terreno, idade da construção, acabamento e estado de conservação. É por isso que a precificação correta se torna uma tarefa tão complexa, são muitas variáveis envolvidas e algumas delas podem até ser subjetivas (VISTA, 2023).

Uma das principais características do mercado imobiliário é a sua baixa liquidez, isso porque o tempo para encontrar um comprador em uma negociação imobiliária é desconhecido, podendo variar entre meses ou até mesmo anos, e em momentos de crise, essa baixa liquidez se torna ainda mais acentuada. Segundo dados da Associação Brasileira das Incorporadoras Imobiliárias (ABRAINC, 2022 apud INVEST NEWS, 2022), o tempo médio para a venda de um imóvel é de cerca de 1 ano e 4 meses.

Apesar de ser uma tarefa complexa, a precificação correta do imóvel é também extremamente importante, uma vez que pode diminuir ainda mais a sua liquidez, exclusivamente nos casos em que há uma supervalorização.

Diante do exposto, o objetivo principal deste trabalho é a construção de um modelo de *machine learning* para precificar o valor de um imóvel, utilizando dados históricos e públicos de vendas de casas adicionados em banco de dados. Para realizar a construção do modelo, utilizou-se o BigQuery ML do Google, uma ferramenta que permite usar consultas em SQL padrão para criar e aplicar modelos de *machine learning* no *data warehouse* BigQuery. O algoritmo escolhido foi a regressão linear, devido à variável resposta ser contínua.

2 FUNDAMENTAÇÃO TEÓRICA

O referido item apresenta as principais referências bibliográficas utilizadas como embasamento teórico para este trabalho.

2.1 *Machine learning*

De acordo com Najafabadi, Hesami e Eskandari (2023), aprendizado de máquina, do inglês, *machine learning* é um subcampo da inteligência artificial que desenvolve algoritmos capazes de aprender com dados, identificar padrões ocultos e tomar decisões. Já Murphy (2012), define *machine learning* como um conjunto de métodos que podem detectar automaticamente padrões nos dados e, em seguida, usar os padrões descobertos para prever dados futuros.

O termo *machine learning* refere-se à detecção automática de padrões relevantes nos dados. Nas últimas duas décadas, tornou-se uma ferramenta muito utilizada em tarefas que exigem a extração de informações de grandes conjuntos de dados. Diversas tecnologias utilizam *machine learning*, como, por exemplo, mecanismos de pesquisa que aprendem como trazer os melhores resultados, *softwares antispam* que aprendem a filtrar mensagens de *e-mail*, transações com cartão de crédito protegidas por um *software* que aprende a detectar fraudes, câmeras digitais que aprendem a detectar rostos, aplicativos de *smartphones* que aprendem a reconhecer comandos de voz e carros equipados com sistemas de prevenção de acidentes (SHALEV-SHWARTZ; BEM-DAVID, 2014).

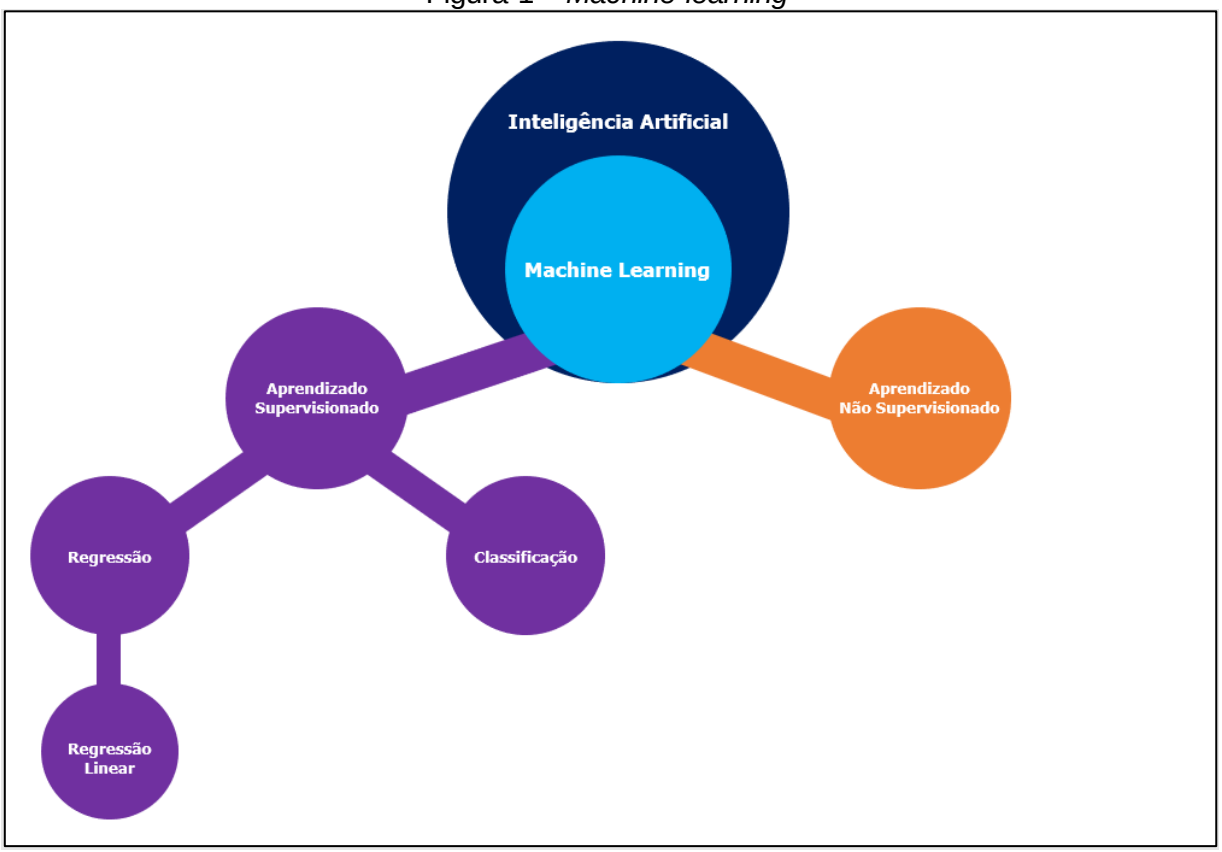
O aprendizado de máquina geralmente é dividido em dois tipos principais: aprendizado supervisionado e aprendizado não supervisionado. No aprendizado supervisionado, os dados possuem um rótulo de saída ou uma variável resposta, e o objetivo é aprender um mapeamento entre as variáveis de entrada e a variável resposta. Quando a variável resposta é do tipo categórica, como masculino ou feminino, o problema é chamado de classificação; quando a variável resposta é um valor contínuo, como o valor da renda, o problema é chamado de regressão. No

aprendizado não supervisionado, não há uma variável resposta, e o objetivo é encontrar padrões interessantes nos dados (MURPHY, 2012).

Segundo Khaire e Kuntawar (2019), a regressão linear é um algoritmo de *machine learning* baseado em aprendizagem supervisionada, sendo utilizada em problemas onde a variável resposta é de natureza contínua, como, por exemplo, a quantidade de vendas de uma rede varejista. A regressão linear pode ser do tipo simples ou múltipla, a diferença é que enquanto a regressão linear simples analisa a relação entre uma variável de entrada e a variável resposta, a regressão linear múltipla analisa a relação entre duas ou mais variáveis de entrada e a variável resposta.

Na Figura 1, é ilustrado a divisão existente entre os dois tipos principais de aprendizado, o supervisionado e o não supervisionado. É ilustrado, também, que o aprendizado de máquina é um subcampo da inteligência artificial.

Figura 1 – *Machine learning*



Fonte: Os autores (2023). Adaptado de https://www.researchgate.net/figure/Main-machine-learning-algorithms_fig3_340114090. Acesso em: 05 set. 2023.

2.2 BigQuery

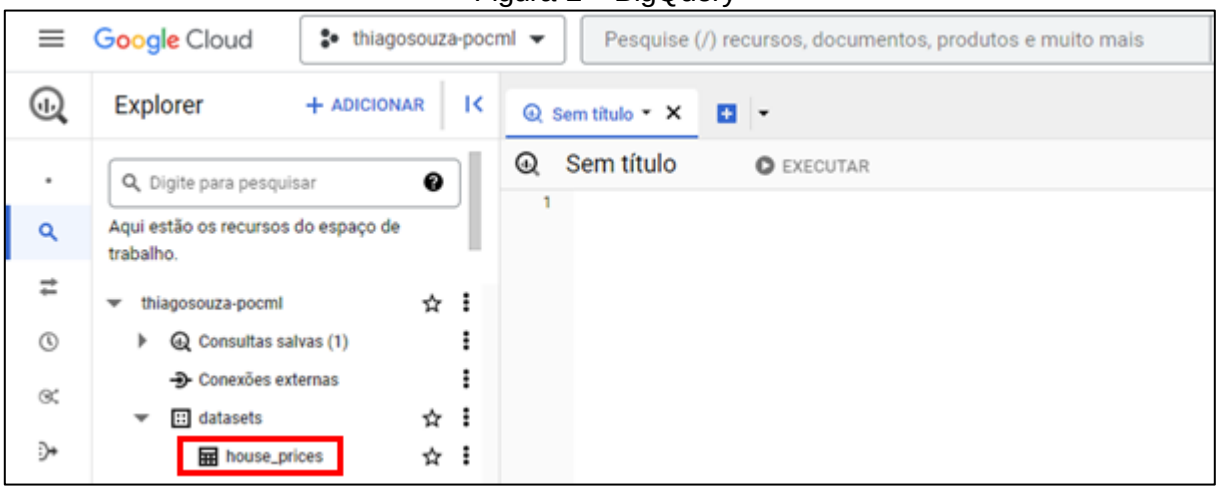
Tigani e Naidu (2014) explicam que o BigQuery, assim como muitas ferramentas, começou com um problema. Os engenheiros do Google estavam com dificuldades em acompanhar o crescimento exponencial dos seus dados. Esse problema com o crescimento dos dados levou ao desenvolvimento de uma ferramenta interna chamada Dremel, que permitia a execução de consultas SQL em grandes conjuntos de dados de forma extremamente rápida. Em 2012, o Google lançou publicamente o BigQuery, permitindo que usuários de fora da empresa aproveitassem o poder e o desempenho da ferramenta Dremel. Desde então, o BigQuery expandiu para um provedor de armazenamento em nuvem.

De acordo com a documentação oficial do Google (2023), o BigQuery é um *data warehouse* corporativo em nuvem que ajuda a gerenciar e analisar dados com recursos integrados, como *machine learning*, análise geoespacial e *Business Intelligence* (BI). O mecanismo de análise distribuída e escalonável do BigQuery permite consultar *terabytes* de dados em segundos e *petabytes* em minutos.

Machado (2013), explica que o *data warehouse* é um armazém de dados históricos com a finalidade de apresentar as informações que permitam identificar indicadores e a evolução de valores ao longo de uma grande janela temporal. Ele representa uma grande base de dados capaz de integrar informações interessantes para a empresa, que se encontram espalhadas pelos sistemas operacionais e até mesmo em fontes externas, para, posteriormente, serem utilizadas como apoio à tomada de decisão.

A Figura 2 demonstra a *User Interface* (UI) do BigQuery, do lado esquerdo é possível notar a tabela chamada “house_prices”, que foi criada para armazenar os dados desta pesquisa.

Figura 2 – BigQuery



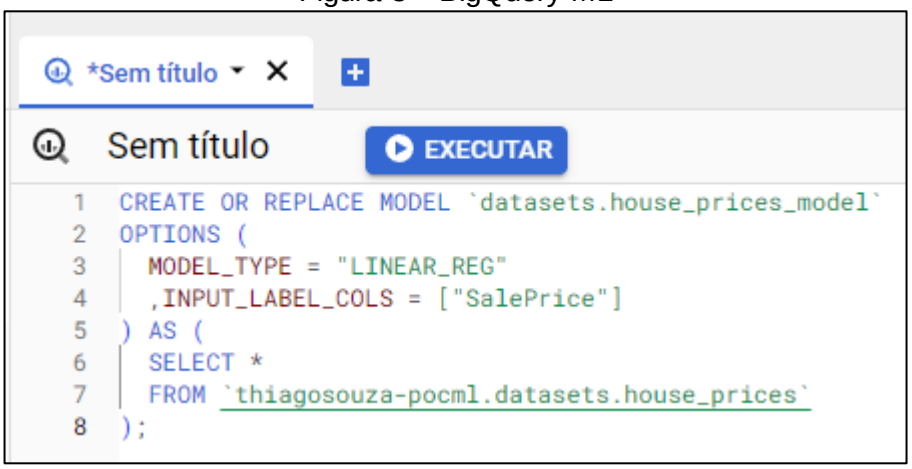
Fonte: Os autores (2023).

2.3 BigQuery ML

De acordo com a documentação oficial, o BigQuery ML é uma ferramenta integrada ao BigQuery que possibilita a criação e execução de modelos de *machine learning* diretamente no BigQuery usando a linguagem SQL. O desenvolvimento desses modelos em grandes conjuntos de dados demanda programação detalhada em linguagens como Python e Java, além do conhecimento em bibliotecas específicas de *machine learning* como PyTorch, TensorFlow, Keras e Scikit-Learn. O BigQuery ML simplifica e agiliza esse processo ao permitir a criação de modelos de *machine learning* utilizando apenas a linguagem SQL. Isso significa que até profissionais com conhecimento limitado em programação podem construir modelos desse tipo. O BigQuery ML suporta os seguintes modelos: regressão linear, regressão logística, *clustering k-means*, fatoração de matrizes, série temporal, árvore otimizada em XGBoost, Deep Neural Network (DNN) e importação de modelos do TensorFlow (GOOGLE, 2023).

Na Figura 3, é demonstrado um exemplo de código para criação de um modelo de *machine learning* utilizando a sintaxe do BigQuery ML. O parâmetro “MODEL_TYPE” serve para definir o tipo do modelo e o parâmetro “INPUT_LABEL_COLS” serve para indicar a variável resposta.

Figura 3 – BigQuery ML



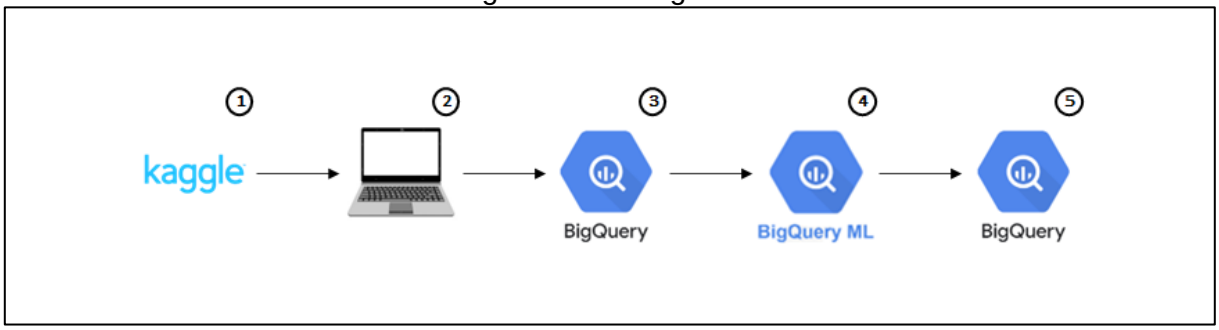
Fonte: Os autores (2023).

3 MATERIAIS E MÉTODOS

Este item apresenta todas as ferramentas e etapas utilizadas no desenvolvimento desta pesquisa. Para atingir o objetivo proposto, foi necessário, primeiramente, coletar um conjunto de dados de vendas de casas, com atributos possivelmente relevantes, como: tamanho do imóvel, tipo de garagem, número de quartos, cozinhas e banheiros, idade da construção, entre outros. Esses dados foram armazenados e processados no BigQuery. Em seguida, treinou-se o modelo com 80% dos dados, e avaliou-se sua performance por meio de métricas de regressão com os outros 20%, como conjunto de teste.

O fluxograma da Figura 4 ilustra de forma enumerada as principais etapas na ordem correta de execução. Na primeira etapa, os dados são extraídos do Kaggle. Em seguida, na segunda etapa, são armazenados localmente em um *desktop*. Na terceira etapa, são carregados e armazenados no BigQuery. A quarta etapa é dedicada ao treinamento do modelo por meio do BigQuery ML. Concluído o treinamento, o modelo é salvo no BigQuery, gerando a quinta e última etapa.

Figura 4 – Fluxograma



Fonte: Os autores (2023).

3.1 Dados utilizados

A fonte dos dados utilizados nesta pesquisa foi o Kaggle, uma plataforma renomada para aprendizado em ciência de dados. Com 1.460 linhas e 81 variáveis, eles representam informações sobre a venda de casas na cidade de Ames, situada

no estado de Iowa, Estados Unidos. O preço de venda é a variável resposta, que é representada pela variável “SalePrice”, que foi depositada no banco de dados.

No Quadro 1, é possível analisar o nome das 81 variáveis e uma breve descrição de cada uma delas.

Quadro 1 – Descrição dos dados utilizados

Variável	Descrição	Variável	Descrição
Id	Identificação dos registros	HeatingQC	Qualidade do aquecimento
SalePrice	Preço de venda	CentralAir	Ar-condicionado central
MSSubClass	Classe da construção	Electrical	Sistema elétrico
MSZoning	Classificação de zoneamento	FlrSF1st	Área do primeiro andar
LotFrontage	Distância da rua	FlrSF2nd	Área do segundo andar
LotArea	Tamanho do lote	LowQualFinSF	Área acabada de baixa qualidade
Street	Tipo de acesso à estrada	GrLivArea	Área habitável acima do solo
Alley	Tipo de acesso à viela	BsmtFullBath	Banheiros completos no porão
LotShape	Formato da propriedade	BsmtHalfBath	Banheiros parciais no porão
LandContour	Planura da propriedade	FullBath	Banheiros completos acima solo
Utilities	Utilidades disponíveis	HalfBath	Banheiros parciais acima solo
LotConfig	Configuração do lote	BedroomAbvGr	Número de quartos acima solo
LandSlope	Inclinação da propriedade	KitchenAbvGr	Número de cozinhas
Neighborhood	Localizações físicas	KitchenQual	Qualidade da cozinha
Condition1	Proximidade principal	TotRmsAbvGrd	Total de quartos
Condition2	Proximidade secundária	Functional	Funcionalidade da casa
BldgType	Tipo da habitação	Fireplaces	Número de lareiras
HouseStyle	Estilo da habitação	FireplaceQu	Qualidade da lareira
OverallQual	Qualidade geral de material	GarageType	Localização da garagem
OverallCond	Classificação de condição	GarageYrBlt	Ano de construção da garagem
YearBuilt	Data da construção	GarageFinish	Acabamento interior da garagem
YearRemodAdd	Data da remodelação	GarageCars	Capacidade de carro na garagem
RoofStyle	Tipo de telhado	GarageArea	Área da garagem
RoofMatl	Material do telhado	GarageQual	Qualidade da garagem
Exterior1st	Revestimento exterior	GarageCond	Condição da garagem
Exterior2nd	Revestimento exterior 2	PavedDrive	Calçada pavimentada
MasVnrType	Revestimento de alvenaria	WoodDeckSF	Área do deck em madeira
MasVnrArea	Área de revestimento	OpenPorchSF	Área aberta da varanda
ExterQual	Qualidade material exterior	EnclosedPorch	Área fechada da varanda
ExterCond	Condição material exterior	snPorch3S	Área da varanda
Foundation	Tipo de fundação	ScreenPorch	Área da varanda com tela
BsmtQual	Altura do porão	PoolArea	Área da piscina
BsmtCond	Condição geral do porão	PoolQC	Qualidade da piscina
BsmtExposure	Paredes do porão	Fence	Qualidade da cerca
BsmtFinType1	Qualidade da área do porão	MiscFeature	Outros recursos
BsmtFinSF1	Área acabada do tipo 1	MiscVal	Valor dos outros recursos
BsmtFinType2	Qualidade da segunda área	MoSold	Mês da venda
BsmtFinSF2	Área acabada do tipo 2	YrSold	Ano da venda
BsmtUnfSF	Área não acabada do porão	SaleType	Tipo da venda
TotalBsmtSF	Área total do porão	SaleCondition	Condição da venda
Heating	Tipo de aquecimento		

Fonte: Os autores (2023).

3.2 BigQuery

Os dados foram extraídos do Kaggle e salvos localmente no formato ".csv". Antes de serem carregados no BigQuery, foi necessário realizar uma limpeza dos valores iguais à "NA", *Not Available* (NA), que representam valores ausentes. Essa limpeza foi importante, pois sem ela os valores NA seriam carregados e interpretados erroneamente como texto. Utilizou-se o bloco de notas do Windows para realizar essa etapa, substituindo os valores NA por vazio.

Durante o processo de *upload* dos dados no BigQuery, optou-se por editar manualmente o *schema* da tabela, garantindo a criação de cada variável com o tipo correto de dado.

Na Figura 5, é possível ver um trecho da edição manual do *schema* da tabela. O parâmetro "description" insere uma descrição para cada coluna, "name" serve para nomear as colunas, "type" define o tipo de dado de cada coluna e "mode" igual à "NULLABLE", permite a gravação de valores ausentes.

Figura 5 – *Schema* da tabela

```
[
    {
        "description": "Id",
        "name": "Id",
        "type": "INTEGER",
        "mode": "NULLABLE"
    },
    {
        "description": "Identifies the type of dwelling involved in the sale",
        "name": "MSSubClass",
        "type": "STRING",
        "mode": "NULLABLE"
    },
```

Fonte: Os autores (2023).

A Figura 6 demonstra parte dos dados já carregados no BigQuery, na tabela "house_prices".

Figura 6 – Dados carregados no BigQuery

Linha	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
1	0	10	2007	WD	Normal	152000
2	600	8	2008	WD	Normal	79500
3	0	9	2009	WD	Normal	133000
4	0	4	2010	WD	Normal	260000

Fonte: Os autores (2023).

3.3 Valores ausentes

Com os dados devidamente carregados no BigQuery, utilizou-se uma consulta SQL para contabilizar a quantidade de valores ausentes em cada variável. Joel, Doorsamy e Paul (2022) explicam que os valores ausentes afetam o desempenho dos algoritmos de *machine learning*, e precisam ser tratados para ajudar no desempenho

desses algoritmos. Isso ocorre porque a precisão dos modelos de *machine learning* depende da qualidade dos dados utilizados, e os valores ausentes diminuem essa qualidade.

Das 81 variáveis, apenas 19 apresentaram valores ausentes, sendo 6 delas com mais de 10%. Dessas 6 variáveis, 5 são categóricas e, de acordo com a descrição do Kaggle, os valores ausentes representam uma categoria adicional. Um exemplo é a variável "PoolQC", que indica a qualidade da piscina, para essa variável os valores ausentes significam que não existe piscina. A única variável numérica com mais de 10% de valores ausentes é a "LotFrontage", com 17,74%.

A fase de tratamento de valores ausentes é conhecida como pré-processamento. No BigQuery ML, há duas possibilidades: manual e automático. Nesta pesquisa, optou-se pelo pré-processamento automático. No pré-processamento automático, o BigQuery ML lida com os valores ausentes de acordo com os tipos de dados das variáveis. Para variáveis numéricas, os valores ausentes são substituídos pela média, e para variáveis categóricas, os valores ausentes são mapeados para uma categoria extra.

Foi escolhido o pré-processamento automático porque a quantidade de variáveis com valores ausentes não é alta. Além disso, a forma com o BigQuery ML trata os valores ausentes, tanto para variáveis categóricas quanto numéricas, pareceu ser adequada para os dados desta pesquisa.

A Tabela 1 apresenta o resultado da consulta SQL utilizada, exibindo o nome das 19 variáveis e os seus respectivos percentuais de valores ausentes, dispostos em ordem decrescente.

Tabela 1 – Percentual de valores ausentes por variável

Linha	Variável	% Ausente	Linha	Variável	% Ausente
1	PoolQC	99,52	11	GarageType	5,55
2	MiscFeature	96,30	12	BsmtFinType2	2,60
3	Alley	93,77	13	BsmtExposure	2,60
4	Fence	80,75	14	BsmtCond	2,53
5	FireplaceQu	47,26	15	BsmtQual	2,53
6	LotFrontage	17,74	16	BsmtFinType1	2,53
7	GarageCond	5,55	17	MasVnrType	0,55
8	GarageFinish	5,55	18	MasVnrArea	0,55
9	GarageYrBlt	5,55	19	Electrical	0,07
10	GarageQual	5,55			

Fonte: Os autores (2023).

3.4 Divisão dos dados

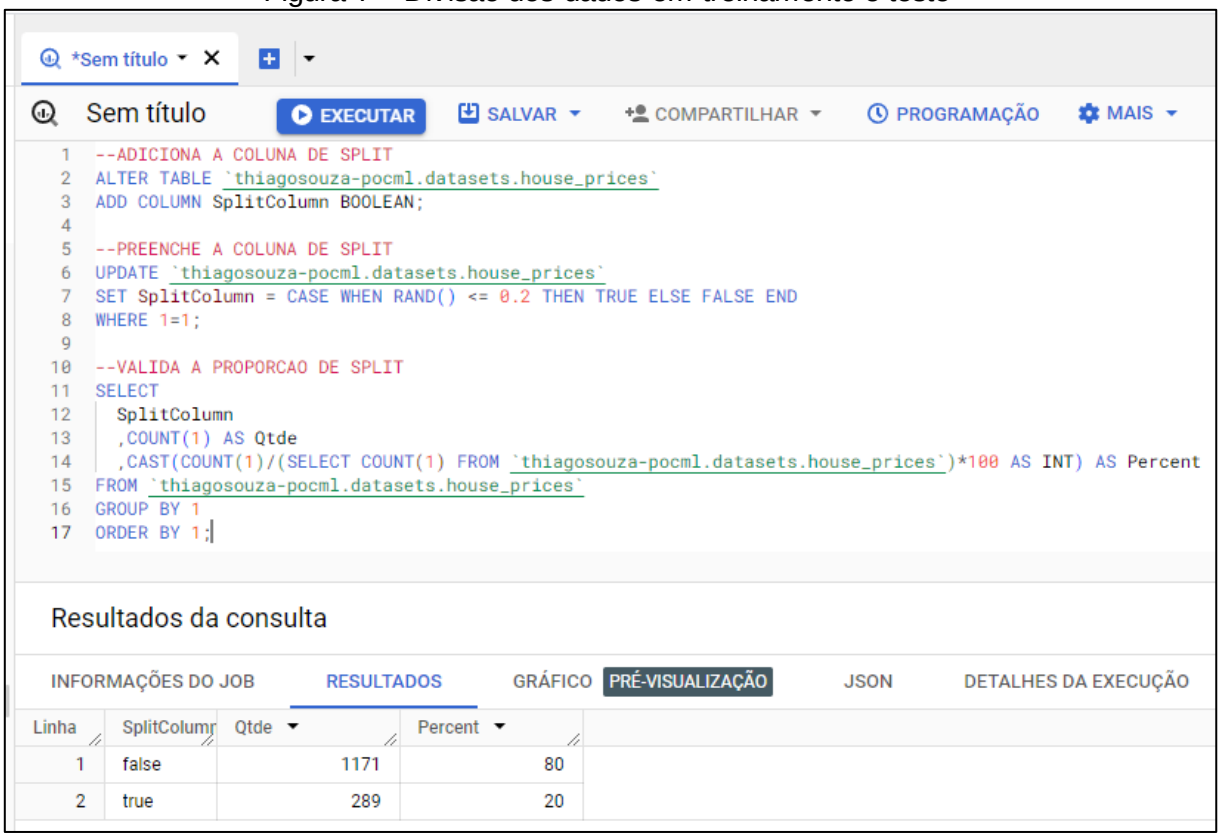
Um dos requisitos de um modelo de *machine learning* é a sua capacidade de generalização, ou seja, para dados novos apresentar resultados semelhantes aos obtidos com dados de treinamento. Quando um modelo apresenta bons resultados em treinamento, mas em dados novos apresenta um resultado muito inferior, diz-se que esse modelo sofreu de *overfitting*. Em outras palavras, ele aprendeu demais com os dados de treinamento e perdeu a sua capacidade de generalização. Uma técnica comum para avaliar a capacidade de generalização do modelo é dividir os dados em duas partes: uma para treinamento e outra para teste (YING, 2019).

Para garantir um acompanhamento melhor dos resultados, optou-se por dividir os dados manualmente. Primeiramente, criou-se uma coluna vazia chamada

“SplitColumn”, em seguida, ela foi preenchida de forma randômica com valores booleanos na proporção de 80% para treinamento e 20% para teste.

Na figura 7, é possível analisar os códigos utilizados nessa etapa. A primeira parte do código foi utilizada para criar a coluna “SplitColumn” de forma vazia, e a segunda parte para adicionar valores a ela. Na terceira e última parte do código está o *select* utilizado para validar a proporção de 80/20 entre treinamento e teste.

Figura 7 – Divisão dos dados em treinamento e teste



Fonte: Os autores (2023).

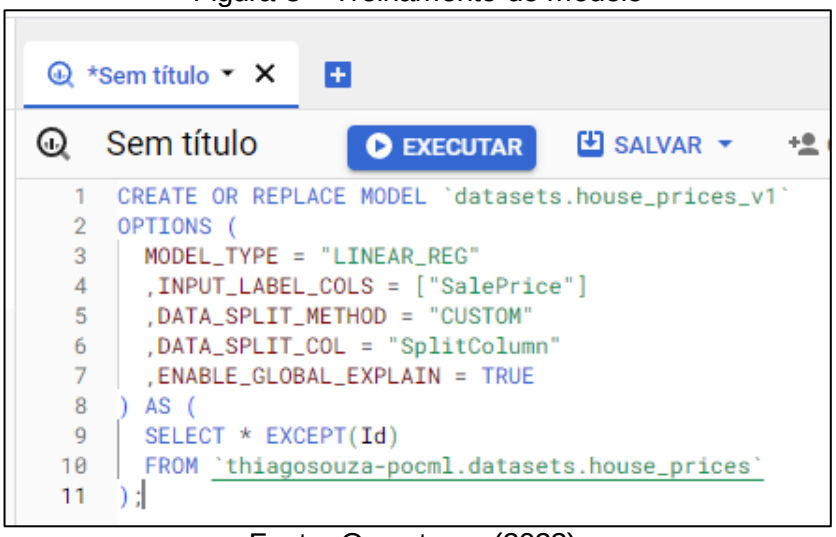
3.5 Treinamento do modelo

Para realizar o treinamento do modelo, utilizou-se o código representado pela Figura 8. Parâmetros utilizados:

a) MODEL_TYPE: serve para definir o tipo de modelo. Utilizou-se a regressão linear, representada pelo valor “LINEAR_REG”;
b) INPUT_LABEL_COLS: serve para definir a variável resposta, ou seja, a variável que se deseja prever. Considerou-se o preço de venda, representado pela variável “SalePrice”;
c) DATA_SPLIT_METHOD: serve para definir o método de divisão dos dados. Utilizou-se o tipo “CUSTOM”, caracterizado pela divisão manual a partir de uma coluna booleana específica. Valores iguais a *“true*” são usados para teste, e valores iguais a “*false*” são usados para treinamento;
d) DATA_SPLIT_COL: quando o método de divisão dos dados é do tipo “CUSTOM”, esse parâmetro serve para especificar o nome da coluna a ser considerada na divisão dos dados. Considerou-se a coluna “SplitColumn”;
e) ENABLE_GLOBAL_EXPLAIN: serve para fornecer explicações do modelo treinado. Por meio desse parâmetro é possível saber a importância de cada variável no modelo. Essa importância é calculada através de um método

chamado Shapley *values*, que representa a média ponderada de como cada variável influencia a predição do modelo quando combinada com todas as outras. Valores maiores indicam uma maior influência da variável na predição do modelo.

Figura 8 – Treinamento do modelo



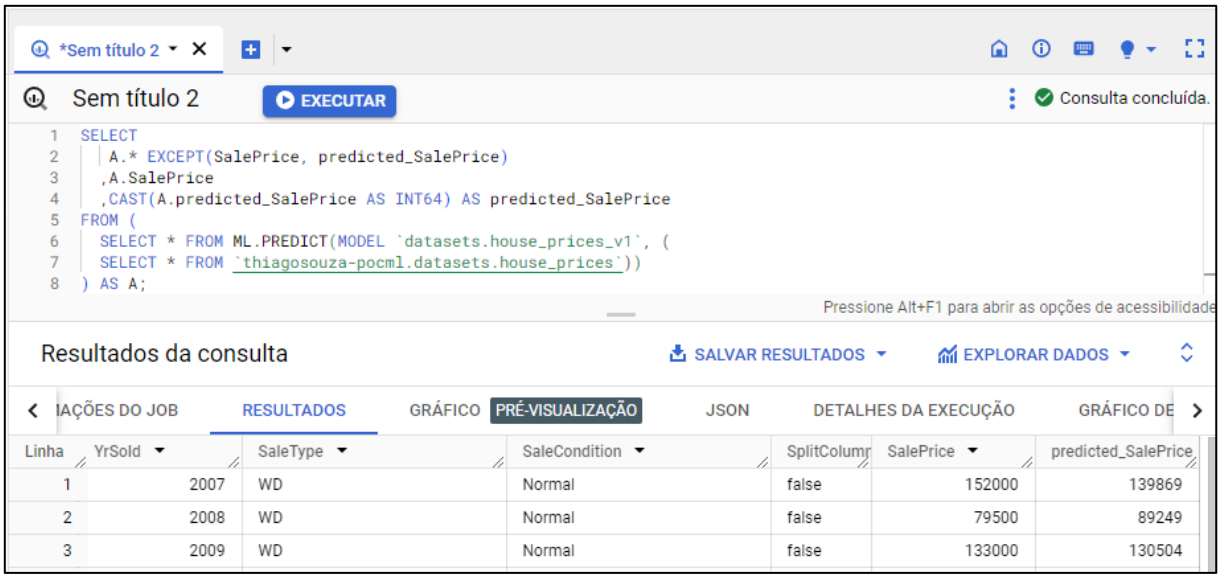
Fonte: Os autores (2023).

3.6 Aplicação do modelo

Após concluída a etapa de treinamento, aplicou-se o modelo em todos os 1.460 registros. A função usada para isso foi a “ML.PREDICT”, que retornou todas as variáveis da tabela, mais uma variável chamada “predicted_SalePrice”, com os valores das predições do modelo.

A Figura 9 apresenta o resultado obtido após a aplicação do modelo. No primeiro registro, o valor de venda foi de $ 152.000 e a previsão do modelo foi de $ 139.869, ou seja, um erro absoluto de $ 12.131 ou 7,98%.

Figura 9 – Aplicação do modelo



Fonte: Os autores (2023).

4 RESULTADOS E DISCUSSÃO

Na Tabela 2, são apresentados os valores das métricas de regressão, utilizadas para avaliação do modelo. Uma observação importante é que elas são calculadas nos dados de teste e não nos dados de treinamento. Métricas principais:

a) Erro médio absoluto: quantifica a média dos valores absolutos das diferenças entre as previsões do modelo e os valores reais. Quanto menor, melhor é o desempenho do modelo. Em resumo, é uma medida da magnitude média dos erros das previsões, independentemente de serem positivos ou negativos;
b) Erro absoluto mediano: semelhante ao anterior, a diferença é que a média é substituída pela mediana. Por considerar a mediana, é menos sensível a valores extremos;
c) Erro quadrático médio: representa a média dos quadrados das diferenças entre as previsões do modelo e os valores reais. Penaliza de forma mais significativa os erros maiores, sendo sensível a valores extremos. Quanto menor, melhor é o desempenho do modelo;
d) R^2: chamado de coeficiente de determinação, é uma métrica estatística que avalia o quão bem um modelo de regressão se ajusta aos dados. Mede a porcentagem da variação na variável resposta que pode ser explicada pelas variáveis de entrada. Quanto mais próximo de 1, melhor o modelo se ajusta aos dados.

Tabela 2 – Métricas de regressão do primeiro modelo

Erro médio absoluto	18.884,15
Erro quadrático médio	1.335.803.573,61
Erro de registro quadrático médio	0,023
Erro absoluto mediano	12.531,86
R^2	0,8022

Fonte: Os autores (2023).

O erro médio absoluto foi $ 18.884,15, a unidade de medida é a mesma da variável resposta. Isso significa que, em média, as previsões do modelo estão

desviando em $ 18.884,15, acima ou abaixo, do preço real de venda. Considerando que o preço médio de venda nos dados de teste é de $ 175.601,17, o erro médio absoluto é de aproximadamente 10,75% desse valor, o que parece ser um erro razoável.

A Tabela 3 apresenta a importância das 20 primeiras variáveis do modelo em ordem decrescente, calculada por meio do parâmetro “ENABLE_GLOBAL_EXPLAIN”, definido na etapa de treinamento. Para retornar a importância, utilizou-se a função “ML.GLOBAL_EXPLAIN”. É importante destacar que sem a escolha do parâmetro “ENABLE_GLOBAL_EXPLAIN” igual à “TRUE”, a função “ML.GLOBAL_EXPLAIN” não funcionaria.

Das 79 variáveis iniciais, desconsiderando a variável resposta e a variável de identificação “Id”, 68 foram utilizadas para o treinamento do modelo. Isso significa que 11 variáveis não foram selecionadas para o treinamento por não terem importância significativa.

Tabela 3 – Importância das 20 primeiras variáveis

Linha	Variável	Atribuição	Linha	Variável	Atribuição
1	BsmtQual	19.974,60	11	BsmtFinType1	8.493,82
2	RoofMatl	17.243,21	12	LotConfig	8.456,63
3	Neighborhood	11.688,82	13	LandSlope	8.195,02
4	SaleType	10.626,07	14	ExterQual	7.960,38
5	SaleCondition	10.115,49	15	GarageFinish	7.928,61
6	LandContour	9.697,31	16	GrLivArea	7.890,73
7	FullBath	9.178,53	17	KitchenQual	7.706,67
8	RoofStyle	8.883,15	18	Fireplaces	7.512,25
9	MasVnrArea	8.854,34	19	OverallQual	7.287,71
10	BsmtExposure	8.662,37	20	TotalBsmtSF	7.111,07

Fonte: Os autores (2023).

As duas variáveis com maior importância foram a “BsmtQual”, variável categórica que avalia a altura do porão, e a “RoofMatl”, variável categórica que representa o material do telhado, respectivamente.

Na Tabela 4, é possível notar a relação linear existente entre a variável “BsmtQual” e a variável resposta nos dados de teste, à medida que a altura do porão aumenta, o preço de venda também aumenta. Na coluna “SalePrice”, considerou-se o valor médio. As casas que não possuem porão, possuem o menor preço médio de venda.

Tabela 4 – Análise da variável “BsmtQual”

Linha	BsmtQual	Qtde	SalePrice
1	1. Excelente (100+ inches)	25	329.223
2	2. Good (90-99 inches)	109	201.667
3	3. Typical (80-89 inches)	133	135.442
4	4. Fair (70-79 inches)	11	119.464
5	5. No Basement	10	103.297

Fonte: Os autores (2023).

A Tabela 5 mostra as métricas de um segundo modelo treinado somente com as 11 variáveis de baixa importância. O erro médio absoluto foi $ 48.963,02, 159% superior ao primeiro modelo. O coeficiente de determinação foi 0,24, ou seja, muito

próximo de zero. Comparando essas métricas, é possível concluir que essas 11 variáveis realmente não possuem uma importância significativa, e, portanto, faz sentido não terem sido selecionadas no primeiro modelo.

Tabela 5 – Métricas de regressão do segundo modelo

Erro médio absoluto	48.963,02
Erro quadrático médio	5.095.653.147,44
Erro de registro quadrático médio	0,1413
Erro absoluto mediano	37.748,14
R^2	0,2454

Fonte: Os autores (2023).

5 CONSIDERAÇÕES FINAIS

Cabe ressaltar que os dados utilizados na construção do modelo foram provenientes de casas americanas, o que implica em particularidades que devem ser levadas em consideração ao extrapolar as previsões para o contexto brasileiro. Variáveis como localização, características arquitetônicas e fatores socioeconômicos podem variar substancialmente, exigindo um novo treinamento com dados nacionais, para uma aplicação precisa no mercado nacional. Outro ponto é com relação a quantidade pequena de registros, apenas 1.460, para se ter um modelo mais generalista, o treinamento deve ser realizado com uma quantidade maior de registros.

A escolha do algoritmo de regressão linear, se deu por conta da sua simplicidade, interpretabilidade e explicabilidade. A regressão linear é um modelo simples que fornece uma boa interpretabilidade dos dados. No entanto, é importante reconhecer a existência de algoritmos mais robustos, como *Random Forest*, *Gradient Boosting* e *Support Vector Machines*, que poderiam ser explorados em trabalhos futuros para aprimorar ainda mais a precisão das previsões.

Em relação aos resultados obtidos, é importante destacar que a ausência de uma base de parâmetro (*baseline*) dificulta a avaliação direta da performance do modelo. No entanto, comparando o erro médio absoluto com o preço médio de venda, a performance mostrou-se razoável, podendo auxiliar na tarefa de precificação de um imóvel.

Diante do exposto, conclui-se que o presente trabalho atingiu sucesso na criação de um modelo para precificação de imóveis usando o BigQuery ML. Os resultados obtidos fornecem uma base para pesquisas futuras e reforçam a importância de inovações tecnológicas na otimização dos processos de avaliação imobiliária.

6 REFERÊNCIAS

CÂMARA BRASILEIRA DA INDÚSTRIA DA CONSTRUÇÃO (CBIC). **Indicadores imobiliários nacionais - 4º trimestre/2022**. Disponível em: http://www.cbicdados.com.br/media/anexos/MERCADO_IMOBILI%C3%81RIO_NACIONAL_4_TRIMESTRE_2022.pdf. Acesso em: 24 mar. 2023.

GOOGLE. **Cloud data warehouse to power your data-driven innovation**. 2023. Disponível em: https://cloud.google.com/bigquery. Acesso em: 22 abr. 2023.

_____. **What is BigQuery?** 2023. Disponível em: https://cloud.google.com/bigquery/docs/introduction. Acesso em: 22 abr. 2023.

_____. **What is BigQuery ML?** 2023. Disponível em: https://cloud.google.com/bigquery/docs/bqml-introduction. Acesso em: 28 abr. 2023.

INVEST NEWS. **Como calcular a valorização anual de um imóvel?** 2022. Disponível em: https://investnews.com.br/colunistas/sos-financas/como-calcular-a-valorizacao-anual-de-um-imovel/. Acesso em: 07 abr. 2023.

JOEL, L. O.; DOORSAMY, W.; PAUL, B. S. A Review of Missing Data Handling Techniques for Machine Learning. **International Journal of Innovative Technology and Interdisciplinary Sciences**, 2022. Disponível em: https://www.researchgate.net/publication/363454985_A_Review_of_Missing_Data_Handling_Techniques_for_Machine_Learning. Acesso em: 19 nov. 2023.

KHAIRE, P. S.; KUNTAWAR, S. V. Predictive model for device identification using machine learning algorithm. **International Journal of Engineering Applied Sciences and Technology**, 2019. Disponível em: http://www.ijeast.com/papers/105-114_old%20file,Tesma404,IJEAST.pdf. Acesso em: 24 abr. 2023.

MACHADO, F. N. R. **Tecnologia e projeto de Data Warehouse**. São Paulo: Editora Érica - Sob Demanda, 2013.

MURPHY, K. P. **Machine learning a probabilistic perspective**. United States of America: Mit Press, 2012.

NAJAFABADI, M. Y.; HESAMI, M.; ESKANDARI, M. **Machine Learning-Assisted Approaches in Modernized Plant Breeding Programs**. Canada: University of Guelph, 2023.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning from theory to algorithms**. United States of America: Cambridge University Press, 2014.

TIGANI, J.; NAIDU, S. **Google BigQuery Analytics**. Indiana: John Wiley & Sons, Inc., 2014.

VISTA. **Precificação:** como saber o valor do imóvel? 2023. Disponível em: https://www.vistasoft.com.br/precificacao-valor-do-imovel/. Acesso em: 24 mar. 2023.

YING, X. An Overview of Overfitting and its Solutions. **Journal of Physics: Conference Series**, 2019. Disponível em: https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/meta. Acesso em: 20 nov. 2023.