

Desafio Técnico - Construindo um Modelo de Regressão Linear para Previsão de Preços de Airbnbs no Rio de Janeiro .

Primeiros vamos começar tratando os dados utilizando o Python.

- Importação das bibliotecas necessárias para a análise e tratamento da base de dados

```
import pandas as pd # necessária para manipulação de dados em dataframes
from scipy.stats.mstats import winsorize # Tratamento de outliers substituindo valores nos extremos
```

✓ 0.0s Python

- Lendo a tabela CSV e incorporando ela a um dataframe através do pandas

```
database = pd.read_csv("listings.csv")
```

✓ 2.4s Python

- Primeira impressão da tabela para simples visualização de dados.

database.head()

✓ 0.0s Python

	id	listing_url	scrape_id	last_scraped	source	name	description	neighborhood_overview	
0	231497	https://www.airbnb.com/rooms/231497	20230922043705	2023-09-22	city scrape	Rental unit in Rio de Janeiro · ★4.73 · 1 bedr...	This is a big studio at the end of Copacabana...	NaN	https://a0.muscache.com/pictures/358231
1	231516	https://www.airbnb.com/rooms/231516	20230922043705	2023-09-22	city scrape	Rental unit in Rio de Janeiro · ★4.71 · 1 bedr...	Special location of the building, on Copacaban...	NaN	https://a0.muscache.com/pictures/367168
2	236991	https://www.airbnb.com/rooms/236991	20230922043705	2023-09-23	city scrape	Rental unit in Rio de Janeiro ·	Aconchegante, amplo, básico, arcelado	Copacabana, apelidada a princesinha do mar, fa	https://a0.muscache.com/pictures/5725a5

database.describe()

✓ 0.2s Python

	id	scrape_id	host_id	host_listings_count	host_total_listings_count	neighbourhood_group_cleansed	latitude	lon
count	3.196400e+04	3.196400e+04	3.196400e+04	31961.000000	31961.000000	0.0	31964.000000	31964
mean	3.514699e+17	2.023092e+13	1.663077e+08	16.112450	27.273020	NaN	-22.967574	-43
std	3.999653e+17	0.000000e+00	1.684315e+08	89.082309	134.207472	NaN	0.035182	0
min	1.787800e+04	2.023092e+13	1.671000e+03	1.000000	1.000000	NaN	-23.073276	-43
25%	2.184739e+07	2.023092e+13	2.258631e+07	1.000000	1.000000	NaN	-22.984820	-43
50%	5.261496e+07	2.023092e+13	8.693849e+07	2.000000	3.000000	NaN	-22.972860	-43
75%	7.814279e+17	2.023092e+13	3.004090e+08	5.000000	7.000000	NaN	-22.956165	-43
max	9.855551e+17	2.023092e+13	5.379850e+08	1311.000000	1803.000000	NaN	-22.749690	-43

8 rows × 41 columns

- Visto que temos algumas inconsistências dos dados na nossa base EX: Valores nulos ou ausentes, teremos que fazer os devidos tratamentos.
- Formatando os valores da coluna “price” e transformando preço do apartamento do tipo Object para Float64

```
database['price'] = database['price'].replace('$', '', regex=True).astype(float)
```

✓ 0.0s Python

- Decidi formatar os valores da coluna “Bathrooms_text” visto que a coluna “bathroom” estava com a maioria dos valores nulos, também substitui os valores nulos ou ausentes desta coluna pela mediana da coluna e transformei esses valores em float.

```
database['bathrooms_text'] = database['bathrooms_text'].str.extract('(\\d+)')
database['bathrooms_text'].fillna(database['bathrooms_text'].median(), inplace=True)
database['bathrooms_text'] = database['bathrooms_text'].astype(float)
```

✓ 0.2s Python

- Formatei também a coluna “host_is_superhost” que estava com os valores [“f” , “t”] para [“0”, “1”] , respectivamente , para facilitar a interpretação algébrica .

```
database['host_is_superhost'] = database['host_is_superhost'].replace({'f': 0, 't': 1})
```

✓ 0.0s Python

- Exclui todas as colunas que tinham todas as suas linhas com valores ausentes .

```
database = database.dropna(axis=1, how='all', inplace=True)
```

✓ 0.0s Python

- Para aproveitar a coluna “neighbourhood_cleansed” busquei dados dos Índices de Desenvolvimento Humano separados por bairros do Rio de Janeiro, esses dados foram disponibilizados pelo IBGE.
- Então criei um dicionário relacionando Bairro e IDH

```
dados = [
    0.97, 0.967, 0.963, 0.962, 0.959, 0.959, 0.959, 0.959, 0.957, 0.957, 0.956, 0.955, 0.952, 0.944, 0.94, 0.938, 0.931, 0.926, 0.922, 0.9
    0.909, 0.909, 0.905, 0.904, 0.904, 0.901, 0.901, 0.9, 0.9, 0.898, 0.894, 0.894, 0.882, 0.878, 0.877, 0.876, 0.876, 0.873, 0.867, 0.861
    0.86, 0.859, 0.859, 0.858, 0.858, 0.857, 0.857, 0.857, 0.856, 0.855, 0.853, 0.851, 0.85, 0.85, 0.849, 0.845, 0.839, 0.839, 0.83
    0.835, 0.833, 0.833, 0.833, 0.831, 0.831, 0.831, 0.829, 0.828, 0.826, 0.825, 0.822, 0.822, 0.817, 0.815, 0.814, 0.812, 0.81, 0.81, 0.8
    0.806, 0.804, 0.804, 0.804, 0.803, 0.802, 0.802, 0.802, 0.8, 0.798, 0.794, 0.792, 0.791, 0.79, 0.788, 0.78, 0.778, 0.773, 0.766
    0.763, 0.762, 0.761, 0.759, 0.753, 0.751, 0.751, 0.75, 0.747, 0.746, 0.745, 0.744, 0.742, 0.732, 0.731, 0.726, 0.722, 0.72, 0.713, 0.7
]

bairros = [
    'Gávea', 'Leblon', 'Jardim Guanabara', 'Ipanema', 'Lagoa', 'Flamengo', 'Humaitá', 'Barra da Tijuca, Joá', 'Laranjeiras', 'Jardim Botâ
    'Botafogo, Urca', 'Maracanã', 'Glória', 'Grajaú', 'Méier', 'Tijuca, Alto da Boa Vista', 'Todos os Santos', 'Anil', 'Vila da Penha', 'A
    'Campinho, Vila Valqueire', 'Moneró, Portuguesa', 'Catete', 'Vila Isabel', 'Cachambi', 'Pechincha', 'Freguesia', 'Recreio dos Bandeir
    'Santa Teresa, Cosme Velho', 'Água Santa, Encantado', 'Taquara', 'Vila Cosmos', 'Vidigal, São Conrado', 'Cidade Nova, Praça da Bandeir
    'Ribeira, Cacuia', 'Lins de Vasconcelos', 'Engenho Novo', 'Zumbi, Pitangueiras, Praia da Bandeira', 'Ramos', 'Engenho de Dentro', 'Abc
    'Oswaldo Cruz', 'Olaria', 'Bento Ribeiro', 'Piedade', 'Quintino Bocaiúva', 'Rio Comprido', 'Praça Seca', 'Jardim América', 'Jacaré, R
    'Engenho da Rainha', 'Brás de Pina', 'São Cristóvão, Vasco da Gama', 'Cascadura', 'Parque Anchieta', 'Madureira', 'Pilares', 'Tanque',
    'Benfica', 'Paqueta', 'Itanhangá', 'Tauá', 'Rocha Miranda', 'Marechal Hermes', 'Turiaçu', 'Guadalupe', 'Inhaúma', 'Campo Grande', 'Cav
    'Coelho Neto', 'Padre Miguel', 'Penha', 'Honório Gurgel', 'Realengo', 'Senador Vasconcelos', 'Tomás Coelho', 'Magalhães Bastos', 'Catu
    'Saúde, Gamboa, Santo Cristo', 'Cordovil', 'Pavuna', 'Anchieta', 'Santíssimo', 'Fundão, Galeão', 'Vicente de Carvalho', 'Jacarepaguá',
    'Sepetiba', 'Cosmos', 'Caju', 'Paciência', 'Cidade de Deus', 'Barros Filho', 'Inhoaíba',
]

# Criando o dicionário
dicionario_bairros_idh = dict(zip(bairros, dados))
```

✓ 0.0s Python

- Lógica para relacionar Bairros com a criação de uma nova coluna com seus respectivos IDH's

```

lista_de_idhs = []

for bairro in database['neighbourhood_cleansed']:
    if (bairro) in (dicionario_bairros_idh):
        lista_de_idhs.append(dicionario_bairros_idh[bairro])
    else:
        lista_de_idhs.append(None)

database['idh'] = lista_de_idhs

```

✓ 0.0s

Python

- Criação de uma lista que contém as colunas numéricas que irei utilizar como base para o aprendizado

```

numerics_columns = ["idh",
                    "host_is_superhost",
                    "bathrooms_text",
                    "accommodates",
                    "bedrooms",
                    "beds",
                    "review_scores_rating",
                    "review_scores_cleanliness",
                    "review_scores_communication",
                    "review_scores_checkin",
                    "review_scores_accuracy",
                    "review_scores_location",
                    "review_scores_value",
                    "number_of_reviews_l30d",
                    "number_of_reviews_ltm",
                    "number_of_reviews",
                    "reviews_per_month",
                    'calculated_host_listings_count_private_rooms',
                    "price"]

```

✓ 0.0s

Python

- Aplicando Winsorizing com limites nos dois extremos para remover outliers das colunas selecionadas

```

database['price'] = winsorize(database['price'], limits=[0.25, 0.3], inplace=True)

database['beds'] = winsorize(database['beds'], limits=[0.25, 0.25], inplace=True)

database['bedrooms'] = winsorize(database['bedrooms'], limits=[0.1, 0.25], inplace=True)

database['accommodates'] = winsorize(database['accommodates'], limits=[0.1, 0.20], inplace=True)

database['bathrooms_text'] = winsorize(database["bathrooms_text"] , limits=[0.1, 0.25], inplace=True)

database['reviews_per_month'] = winsorize(database["reviews_per_month"] , limits=[0.1, 0.25], inplace=True)

database['calculated_host_listings_count_private_rooms'] = winsorize(database["calculated_host_listings_count_private_rooms"] , limits=[0.1, 0.25], inplace=True)

```

✓ 0.0s

Python

- Transformando o nosso database para arquivo csv e exportando

```

database.to_csv('airbnbTest.csv', index=True)

```

✓ 8.1s

Python

Agora vamos trabalhar com o Google Cloud Plataform ...

1º - Acessei a Cloud Storage > Buckets

Google Cloud | My First Project | Pesquise (/) recursos, documentos, produtos e muito mais | Pesquisa

Cloud Storage | Blocos | CRIAR | ATUALIZAR | SAIBA MAIS

Buckets

- Monitoramento
- Configurações

Revise seus buckets do Autoclass

O Autoclass vai passar por mudanças em breve. Revise seus buckets e prepare-se.

SAIBA MAIS

Segurança

Conferir recomendações de segurança

Aumente a proteção dos seus buckets aplicando recomendações de segurança a eles. A coluna "Insirir segurança" na tabela descreve quais buckets têm permissões em excesso.

CONFERIR NA TABELA | SAIBA MAIS

Filtro | Filtrar intervalos

	Nome ↑	Criado em	Tipo de local	Local	Classe de armazenamento padrão	Última modificação
Nenhuma linha a ser exibida						

Marketplace

Notas de lançamento

2º - Criei o bucket necessário para armazenar a base de dados.

Google Cloud | My First Project | Pesquise (/) recursos, documentos, produtos e muito mais | Pesquisa

Cloud Storage | Detalhes do bucket | ATUALIZAR | SAIBA MAIS

bucket_sauter

Local: us (várias regiões nos Estados Unidos) | Classe de armazenamento: Standard | Acesso público: Não público | Proteção: Nenhum

OBJETOS | CONFIGURAÇÃO | PERMISSÕES | PROTEÇÃO | CICLO DE VIDA | OBSERVABILIDADE | RELATÓRIOS DE INVENTÁRIO

Intervalos > bucket_sauter

FAZER UPLOAD DE ARQUIVOS | CARREGAR PASTA | CRIAR PASTA | TRANSFERIR DADOS | GERENCIAR RETENÇÕES | FAZER O DOWNLOAD EXCLUIR

Filtrar apenas pelo prefixo do nome | Filtro | Filtrar objetos e pastas | Mostrar dados excluídos

	Nome	Tamanho	Tipo	Criado	Classe de armazenamento	Última modificação	Acesso público	Histórico de versões	Criptografia
Nenhuma linha a ser exibida									

Marketplace

Notas de lançamento

Bucket criado bucket_sauter

3º - Fiz o Upload da base de dados Desafio_sauter.csv para o Bucket. (Processo de armazenamento finalizado)

Google Cloud | My First Project | Pesquise (/) recursos, documentos, produtos e muito mais | Pesquisa

Cloud Storage | Detalhes do bucket

bucket_sauter

Local: us (várias regiões nos Estados Unidos) | Classe de armazenamento: Standard | Acesso público: Não público | Proteção: Nenhum

OBJETOS | CONFIGURAÇÃO | PERMISSÕES | PROTEÇÃO | CICLO DE VIDA | OBSERVABILIDADE | RELATÓRIOS DE INVENTÁRIO

Intervalos > bucket_sauter

FAZER UPLOAD DE ARQUIVOS | CARREGAR PASTA | CRIAR PASTA | TRANSFERIR DADOS | GERENCIAR RETENÇÕES | FAZER O DOWNLOAD

EXCLUIR

Filtrar apenas pelo prefixo do nome | Filtro | Filtrar objetos e pastas | Mostrar dados excluídos

<input type="checkbox"/>	Nome	Tamanho	Tipo	Criado	Classe de armazenamento	Última modificação	
<input type="checkbox"/>	Desafio_sauter.csv	49,5 MB	text/csv	28 de nov. de 2023 18:24:05	Standard	28 de nov. de 2023 18:24:05	📄 ⋮

1 arquivo enviados com sucesso

4º - Precisei criar um conjunto de dados (dataset_sauter) para armazenar a base de dados (table_sauter) que está no Bucket.

Criar tabela

Origem

Criar tabela de: Google Cloud Storage

Selecione o arquivo do bucket do GCS ou use um padrão de URI *
☒ bucket_sauter/Desafio_sauter.csv | PROCURAR

Formato do arquivo: CSV

☐ Particionamento de dados de origem

Destino

Projeto *: warm-particle-406411 | PROCURAR

Conjunto de dados *: dataset_sauter

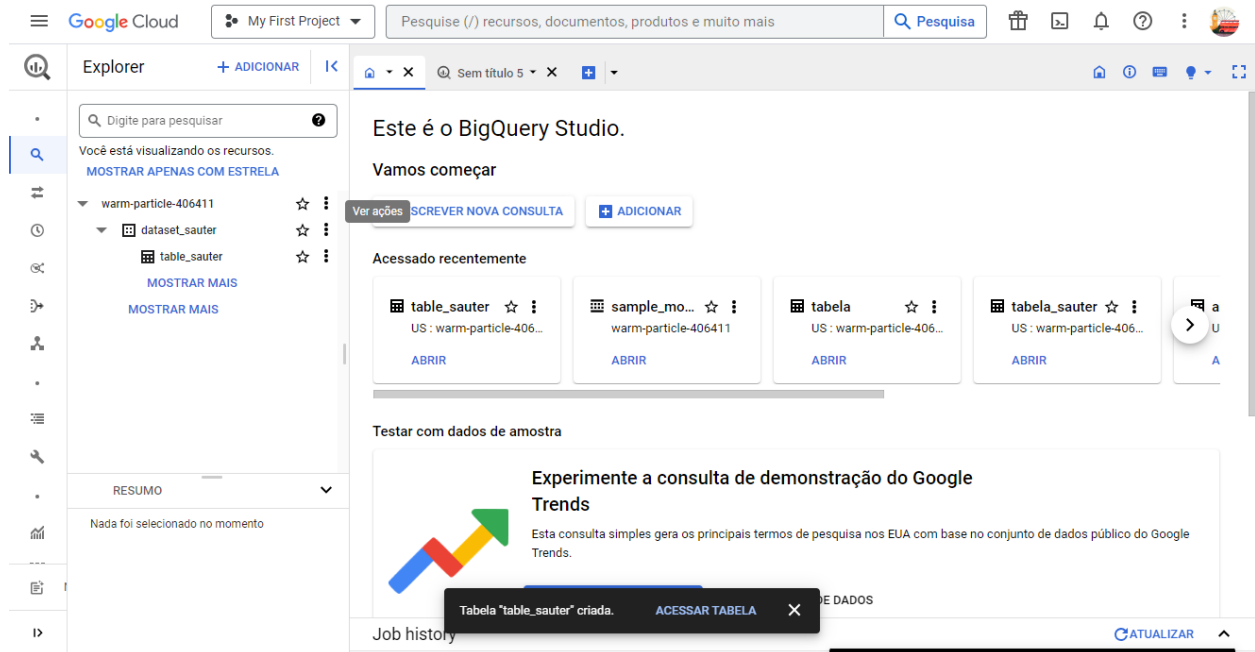
Tabela *: table_sauter

Letras Únicode, marcas, números, conectores, traços ou espaços são permitidos.

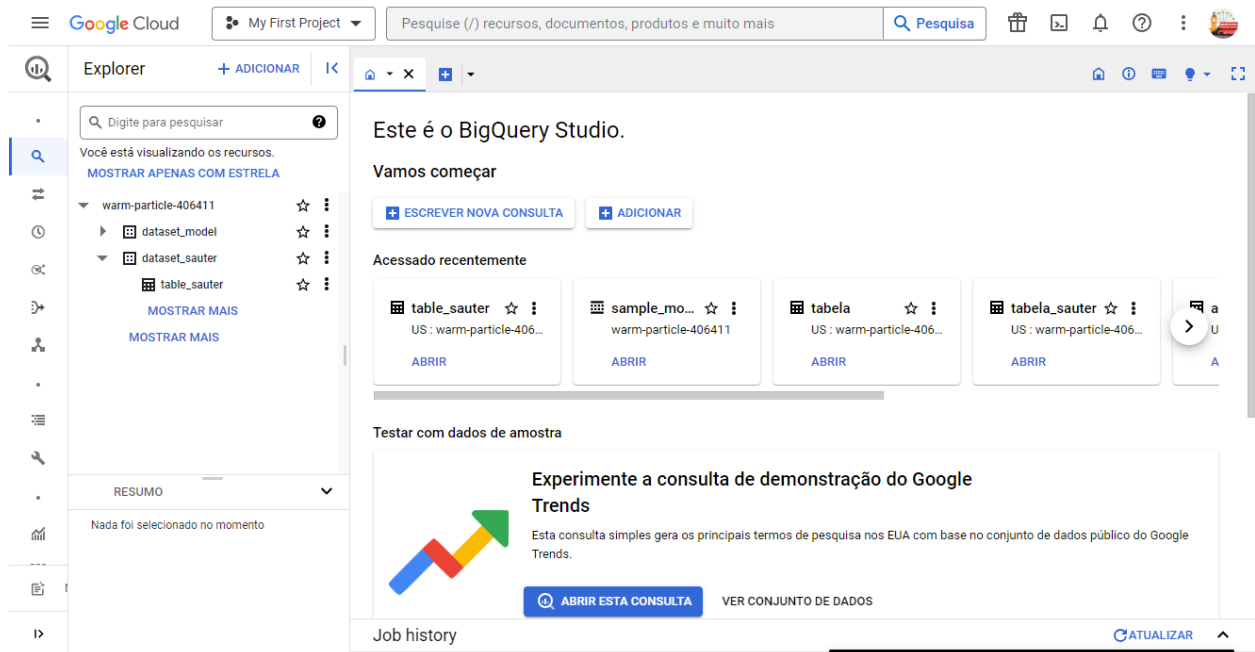
Tipo de tabela: Tabela nativa

CRIAR TABELA | CANCELAR

6º - Tabela (table_sauter) criada !

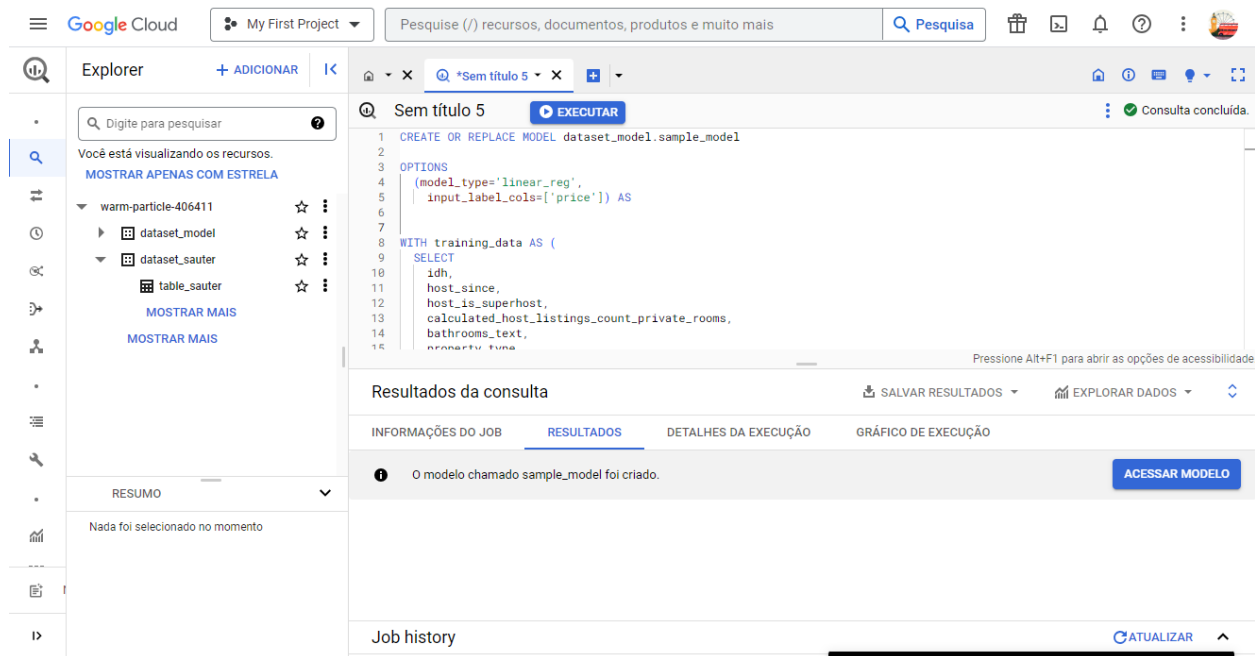


7º - Precisei criar um conjunto de dados (dataset_model) para armazenar os dados de treinamento que serão utilizados.



Criação de tabela e conjuntos de dados finalizado, vamos para a Execução das Querys ...

8º - Nesta execução vamos criar um modelo de regressão linear através da função **CREATE OR REPLACE MODEL** que será treinado para prever os valores de *“price”* com dados extraídos da *“table_sauter”* .



The screenshot shows the Google Cloud console interface. On the left, the 'Explorer' pane displays a project named 'warm-particle-406411' with a folder 'dataset_model' containing a table 'table_sauter'. The main editor shows a SQL query titled 'Sem título 5' with the following code:

```
1 CREATE OR REPLACE MODEL dataset_model.sample_model
2
3 OPTIONS
4   (model_type='linear_reg',
5    input_label_cols=['price']) AS
6
7
8 WITH training_data AS (
9   SELECT
10    idh,
11    host_since,
12    host_is_superhost,
13    calculated_host_listings_count_private_rooms,
14    bathrooms_text,
15    property_type,
```

The 'Resultados da consulta' pane shows a message: 'O modelo chamado sample_model foi criado.' (The model named sample_model was created.) with an 'ACESSAR MODELO' button. The 'Job history' pane is visible at the bottom.

```
CREATE OR REPLACE MODEL dataset_model.sample_model

OPTIONS
  (model_type='linear_reg',
   input_label_cols=['price']) AS

WITH training_data AS (
  SELECT
    idh,
    host_since,
    host_is_superhost,
    calculated_host_listings_count_private_rooms,
    bathrooms_text,
    property_type,
    room_type,
    accommodates,
    bedrooms,
    beds,
    amenities,
```



```

        review_scores_rating,
        review_scores_accuracy,
        review_scores_cleanliness,
        review_scores_checkin,
        review_scores_communication,
        review_scores_location,
        review_scores_value,
        number_of_reviews_130d,
        number_of_reviews_ltm,
        number_of_reviews,
        reviews_per_month,
        price
    FROM
        dataset_sauter.table_sauter
)

SELECT
    idh,
    host_since,
    calculated_host_listings_count_private_rooms,
    host_is_superhost,
    bathrooms_text,
    property_type,
    room_type,
    accommodates,
    amenities,
    review_scores_rating,
    review_scores_accuracy,
    review_scores_cleanliness,
    review_scores_checkin,
    review_scores_communication,
    review_scores_location,
    review_scores_value,
    number_of_reviews,
    number_of_reviews_ltm,
    number_of_reviews_130d,
    reviews_per_month,
    price,
    bedrooms,
    beds
FROM
    training_data;

```

9º -Utilizei a função `ML.PREDICT` para fazer previsões usando a tabela gerada pelo modelo “*sample_model*” criado anteriormente.

Google Cloud My First Project Pesquise (/) recursos, documentos, produtos e muito mais Pesquisa

Explorer + ADICIONAR

Você está visualizando os recursos. MOSTRAR APENAS COM ESTRELA

- warm-particle-406411
 - dataset_model
 - dataset_sauter
 - table_sauter

MOSTRAR MAIS

RESUMO

Nada foi selecionado no momento

Sem título 9 EXECUTAR

```

1 SELECT
2 *
3 FROM
4 ML.PREDICT(MODEL dataset_model.sample_model,
5 (
6 SELECT
7 *
8 FROM
9 dataset_sauter.table_sauter)))

```

Pressione Alt+F1 para abrir as opções de acessibilidade.

Resultados da consulta SALVAR RESULTADOS EXPLORAR DADOS

Linha	predicted_price	int64_field_0	id	listing_url	scrape_id	last_scraped	source
1	477.9147215895...	1637	2776035	https://www.airbnb.com/room...	20230922043705	2023-09-22	city scrapi
2	217.6957655217...	22855	7552547629290...	https://www.airbnb.com/room...	20230922043705	2023-09-22	city scrapi

Resultados por página: 50 1 - 50 de 21726

Job history ATUALIZAR

```

SELECT
*
FROM
ML.PREDICT(MODEL dataset_model.sample_model,
(
SELECT
*
FROM
dataset_sauter.table_sauter)))

```

- 10º - Utilizei a função **ML.EVALUATE** para avaliar a tabela gerada pelo modelo que criei e assim visualizar seus resultados;

Google Cloud My First Project Pesquise (/) recursos, documentos, produtos e muito mais Pesquisa

Explorer + ADICIONAR

Você está visualizando os recursos. MOSTRAR APENAS COM ESTRELA

- warm-particle-406411
 - dataset_model
 - dataset_sauter
 - table_sauter

MOSTRAR MAIS

RESUMO

Nada foi selecionado no momento

Sem título 10 EXECUTAR

```

1 SELECT
2 *
3 FROM
4 ML.EVALUATE(MODEL dataset_model.sample_model,
5 (
6 SELECT
7 *
8 FROM
9 dataset_sauter.table_sauter))

```

Pressione Alt+F1 para abrir as opções de acessibilidade.

Resultados da consulta SALVAR RESULTADOS EXPLORAR DADOS

INFORMAÇÕES DO JOB	RESULTADOS	GRÁFICO	PRÉ-VISUALIZAÇÃO	JSON	DETALHES DA EXECUÇÃO	GRÁFICO
Linha	mean_absolute_error	mean_squared_error	mean_squared_log_e	median_absolute_err	r2_score	explained_variance
1	45.22553097215...	4341.830013842...	0.040973268290...	30.86982151655...	0.731538994910...	0.741523351431...

Job history ATUALIZAR

```

SELECT
*
FROM
ML.EVALUATE(MODEL dataset_model.sample_model,
(
SELECT
*
FROM
dataset_sauter.table_sauter))

```

Resultados finais

mean_absolute_error	mean_squared_error	median_absolute_error	r2_score
45.22553	4341.8300	30.8698	0.7315

• Questões

1º -

Durante o desenvolvimento do projeto, enfrentei desafios relacionados à falta de documentação necessária, à formatação inadequada dos dados e à presença de

valores ausentes ou nulos. Para resolver a questão dos valores ausentes, recorri à biblioteca pandas do Python, substituindo esses valores pela mediana correspondente. Essa abordagem foi escolhida para assegurar maior precisão nas colunas utilizadas. Adicionalmente, utilizei a mesma biblioteca para reformatar e converter o tipo de dado da coluna 'price' de string para float64, promovendo consistência e alinhamento dos dados com os requisitos do projeto.

1.2º -

analisar os itens que os clientes mais avaliam em um apartamento

2º -

Pré-processamento dos dados > Treinamento do Modelo > Teste do Modelo > Implantação do Modelo > Monitoramento do Modelo > Atualização Automática do Modelo > Backup e Versionamento > Logs e Monitoramento de Erros

Automatizar a solução de regressão linear para prever preços de aluguéis no Airbnb seria conduzida por um pipeline eficiente no Google Cloud Platform (GCP). A preparação dos dados, inicialmente realizada com Pandas, seria substituída por ferramentas nativas do GCP, como o Dataflow. No treinamento do modelo, as ferramentas Scikit-learn ou TensorFlow seriam substituídas pelo AutoML. A implantação ocorreria no Google Cloud AI Platform, utilizando o Kubernetes e Cloud Build para um pipeline de CI/CD automatizado. O monitoramento contínuo seria feito com Stackdriver.

A atualização automática do modelo seria incorporada ao Cloud Build, utilizando estratégias de implementação contínua. A documentação automática seria gerada utilizando ferramentas do GCP. Medidas de segurança seriam implementadas utilizando as ferramentas de controle de acesso do GCP. A escalabilidade do pipeline seria garantida com o uso de serviços escaláveis do GCP.

Integração com sistemas externos seria realizada através de serviços como Cloud Functions ou Cloud Endpoints. O backup regular de modelos e dados seria assegurado pelo Cloud Storage, com o controle de versão nativo do GCP.

Por fim, logs detalhados seriam implementados utilizando o Stackdriver Logging. Essa abordagem abrangente no GCP combinaria ferramentas nativas e práticas

recomendadas para garantir a consistência, eficiência e segurança ao longo de todas as fases do pipeline de Machine Learning.

3º -

Para otimizar o pipeline de Machine Learning no Google Cloud Platform (GCP) para acomodar grandes conjuntos de dados, algumas estratégias eficazes podem ser implementadas. O uso do Google Cloud Storage (GCS) e Google Cloud BigQuery para armazenamento e consulta eficiente de dados em larga escala é essencial. O Google Cloud Dataflow pode ser empregado para processamento paralelo e distribuído, enquanto ferramentas como AutoML Tables e BigQuery ML permitem treinamento e inferência diretamente nos dados armazenados.

A escalabilidade pode ser aprimorada usando o Kubernetes Engine para alocação dinâmica de recursos e Apache Beam no Google Cloud Dataflow para distribuição eficiente de tarefas. Estratégias como caching de resultados intermediários e pré-processamento eficiente reduzem a carga durante treinamento e inferência. A utilização de serviços gerenciados, como Cloud AI Platform e Cloud Dataflow, simplifica operações e garante otimização automática.

A monitorização contínua e otimização dos recursos, juntamente com a análise de desempenho, são práticas essenciais para garantir um pipeline escalável e eficiente no GCP. Essas abordagens combinadas asseguram que o pipeline possa lidar com grandes volumes de dados sem comprometer o desempenho, proporcionando uma execução consistente e otimizada.