

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS**  
**ÁREA III**  
**ENGENHARIAS**

**RELATÓRIO V DESAFIO EM CIÊNCIAS DE DADOS**

**Alunos:**

Arthur Ragazzo  
Gustavo Palmier  
Milton Ribeiro  
Pedro Lucas  
Thales Pasqualetto

Arthur Ragazzo  
Gustavo Palmier  
Milton Ribeiro  
Pedro Lucas  
Thales Pasqualetto

### **Relatório de Estágio Supervisionado**

Relatório apresentado como parte  
dos materiais a ser entregues na  
plataforma Teams do V Desafio de Dados  
da PUC Goiás

## Sumário

1.	INTRODUÇÃO.....	4
2.	OBJETIVOS .....	5
3.	DESENVOLVIMENTO .....	5
3.1	Preparação dos dados (Data Preparation).....	5
3.2	Tratamento dos dados .....	5
3.3	Exploratory Data Analysis (EDA) .....	5
3.4	Machine Learning.....	14
3.5	Avaliação de Modelos .....	16
3.6	MODELO GLM .....	17
4.	CONCLUSÕES .....	20
	REFERÊNCIAS .....	22

## 1. INTRODUÇÃO

O Airbnb, originado em 2007, revolucionou a indústria da hospitalidade ao oferecer uma plataforma inovadora que permite que indivíduos aluguem espaços ou propriedades inteiras diretamente de outros indivíduos, normalmente por períodos curtos. Esse modelo de economia compartilhada não só ofereceu aos viajantes uma alternativa mais personalizada e acessível ao alojamento tradicional, mas também proporcionou aos anfitriões uma nova fonte de renda, permitindo-lhes alugar espaços não utilizados ou propriedades inteiras.

O Rio de Janeiro, conhecido mundialmente como a "Cidade Maravilhosa", é um dos destinos turísticos mais procurados do mundo. Com suas praias icônicas, montanhas majestosas, cultura vibrante e eventos de renome mundial como o Carnaval, o Rio atrai milhões de turistas a cada ano. Esta imensa afluência de visitantes requer uma infraestrutura de hospedagem robusta e diversificada para atender às variadas necessidades e orçamentos dos turistas.

Neste contexto, o Airbnb encontrou um terreno fértil no Rio de Janeiro. A plataforma permitiu que moradores locais oferecessem suas residências ou quartos a visitantes, proporcionando uma experiência mais autêntica da cidade. Além disso, os visitantes ganharam uma vasta gama de opções de hospedagem, desde apartamentos aconchegantes em bairros tradicionais até luxuosas vilas à beira-mar.

Dada a popularidade do Airbnb e a rica dinâmica turística do Rio de Janeiro, há uma enorme quantidade de dados gerados sobre as listagens, os anfitriões e os visitantes. A análise desses dados pode oferecer insights valiosos sobre padrões de hospedagem, preferências dos viajantes, dinâmica de preços e muito mais. Neste desafio, nos aprofundamos nesses dados, buscando entender as tendências e extrair informações que possam beneficiar tanto os anfitriões quanto os visitantes, otimizando a experiência do Airbnb na cidade.

Ao longo deste relatório, exploraremos os dados do Airbnb (PUC GO, 2023) no Rio de Janeiro de abril de 2018 a maio de 2020, buscando padrões, identificando oportunidades e propondo soluções baseadas em dados para melhorar ainda mais a experiência de hospedagem na cidade. Neste relatório apresenta os passos desenvolvidos e conclusões obtidas, para identificar o código utilizado recomendo acesso ao github do nosso projeto.

## 2. OBJETIVOS

O objetivo principal deste trabalho é desenvolver uma calculadora baseada em modelo estatístico para previsão de preços de hospedagens no Airbnb no Rio de Janeiro. Esta ferramenta servirá como um guia tanto para anfitriões quanto para hóspedes:

**Para os Anfitriões:** A calculadora permitirá que eles insiram detalhes e características de seus imóveis e recebam uma estimativa de preço de diária com base nas tendências atuais do mercado e nas informações históricas. Isso ajudará os anfitriões a definir um preço competitivo para seus imóveis, garantindo uma boa ocupação e um retorno adequado.

**Para os Hóspedes:** Ao inserir os detalhes do imóvel que desejam alugar, os hóspedes poderão usar a calculadora para verificar se o preço oferecido está alinhado com a média de mercado para acomodações similares. Isso oferece uma ferramenta valiosa para negociar preços ou buscar alternativas mais econômicas.

## 3. DESENVOLVIMENTO

### 3.1 Preparação dos dados (Data Preparation)

Os dados estão disponíveis na plataforma Teams, dentro da pasta "V Desafio de Ciência de Dados da PUC Goiás". Essa pasta, em formato ZIP, contém planilhas referentes a cada mês do período de abril de 2018 a maio de 2020, além de uma planilha consolidada denominada "total\_data", que reúne todos esses registros.

Para esclarecer o significado das informações em cada coluna, o edital do desafio forneceu um dicionário. Através deste, é possível discernir o tipo de dado apresentado na maioria das colunas. Enquanto algumas colunas têm seu significado facilmente compreendido, outras, como as da série "availability\_30", "availability\_60", "availability\_90" e "availability\_365", podem gerar dúvidas. Vale destacar que existem colunas, como "guests\_included", que não foram mencionadas no dicionário de dados.

### 3.2 Tratamento dos dados

A planilha "total\_data.csv" tem um tamanho aproximado de 2,5 GB, o que pode tornar seu processamento desafiador para determinados computadores. Os dados coletados provêm de raspagens ao longo de vários meses. No entanto, é

importante notar que muitas acomodações na plataforma Airbnb já não estão disponíveis ou não têm sido reservadas há um longo período.

Optamos por filtrar os dados pela coluna "last\_review", considerando o intervalo de junho de 2018 a maio de 2020. Qualquer dado fora deste período foi descartado. Durante esse processo, acomodações cujos preços vinham diminuindo até se tornarem inexistentes foram eliminadas, resultando em uma seleção de dados mais representativos e confiáveis.

Posteriormente, os valores monetários, apresentados originalmente em formato de texto com o símbolo "\$" (indicando reais, R\$), foram devidamente formatados. Além disso, observou-se que as colunas referentes a camas, banheiros e quartos apresentavam valores em múltiplos de 10; tais valores foram corrigidos (Figura 1).

Figura 1. Planilha durante o tratamento dos dados.

	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY
	price	weekly_price	monthly_price	security_deposit	cleaning_fee	guests	extra_people	minimum_nights	maximum_nights	calendar_last_scraped	has_availability	availability_30	availability_60	availability_90	availability_365	calendar_last_scraped_nur
ter	R\$ 179,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		20	50	80	355	14/09/2018
ter	R\$ 180,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	11	41	316	12/10/2018
ter	R\$ 178,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		15	40	70	345	15/11/2018
ter	R\$ 182,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		20	50	80	355	14/12/2018
ter	R\$ 138,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		10	10	10	20	11/02/2019
ter	R\$ 138,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	40	13/03/2019
ter	R\$ 140,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	75	17/04/2019
ter	R\$ 139,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	110	22/05/2019
ter	R\$ 139,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	139	20/06/2019
ter	R\$ 139,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	165	16/07/2019
ter	R\$ 139,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 today	t		0	0	0	195	15/08/2019
ter	R\$ 137,00		R\$ 2.298,00		R\$ 119,00	2	R\$ 30,00	2	1125 2 weeks a	t		0	0	0	235	24/09/2019

Fonte: PUC GO, 2023.

Acomodações que requeriam uma estadia mínima de mais de 30 noites foram removidas do conjunto de dados. O foco da análise são as acomodações de curto prazo. Em geral, acomodações de longo prazo oferecem preços diários mais acessíveis devido à extensão da estadia, assemelhando-se à dinâmica entre inquilinos e locatários. Essa perspectiva não se alinha ao objetivo primário desta análise, especialmente quando se considera uma metrópole como o Rio de Janeiro.

Os dados referentes à latitude e longitude, anteriormente em notação científica, foram reformatados para formato numérico convencional, utilizando um ponto "." após o segundo número depois do sinal negativo.

A análise foca acomodações localizadas no estado do Rio de Janeiro. No entanto, ao avaliar a coluna "smart\_location", que determina a localização com base nas coordenadas de latitude e longitude, identificaram-se algumas entradas que não condiziam com a área de interesse. Para identificar esses desvios, a coluna

"smart\_location" foi transferida para uma planilha separada, as entradas duplicadas foram removidas, e as cidades que não pertencem ao estado do Rio de Janeiro foram identificadas:

- São Paulo, Brazil
- Niterói, Brazil
- Angra dos Reis, Brazil
- Teresopolis, Brazil
- Nova Iguaçu, Brazil
- Paraty, Brazil
- Duque De Caxias, Brazil
- Рио-де-Жанейро, Brazil

Observou-se a presença de alguns registros em japonês, bem como acomodações voltadas para esse público específico. Estes dados, desalinhados com o foco da análise, foram excluídos.

Por fim, iniciou-se a identificação de colunas que não apresentavam dados relevantes após o tratamento inicial, sem variação de informação ou que eram consideradas irrelevantes para a análise, essas foram excluídas:

- |                                |                                    |
|--------------------------------|------------------------------------|
| • cancellation_policy          | • thumbnail_url                    |
| • country                      | • require_guest_phone_verification |
| • instant_bookable             | • host_listings_count              |
| • space                        | • has_availability                 |
| • monthly_price                | • is_location_exact                |
| • host_url                     | • name                             |
| • host_thumbnail_url           | • house_rules                      |
| • picture_url                  | • host_location                    |
| • notes                        | • xl_picture_url                   |
| • scrape_id                    | • square_feet                      |
| • host_id                      | • street                           |
| • jurisdiction_names           | • host_picture_url                 |
| • neighbourhood_group_cleansed | • is_business_travel_ready         |
| • license                      | • host_has_profile_pic             |
| • host_neighbourhood           | • country_code                     |
| • reviews_per_month            | • interaction                      |
| • host_verifications           | • listing_url                      |
| • neighborhood_overview        | • host_since                       |

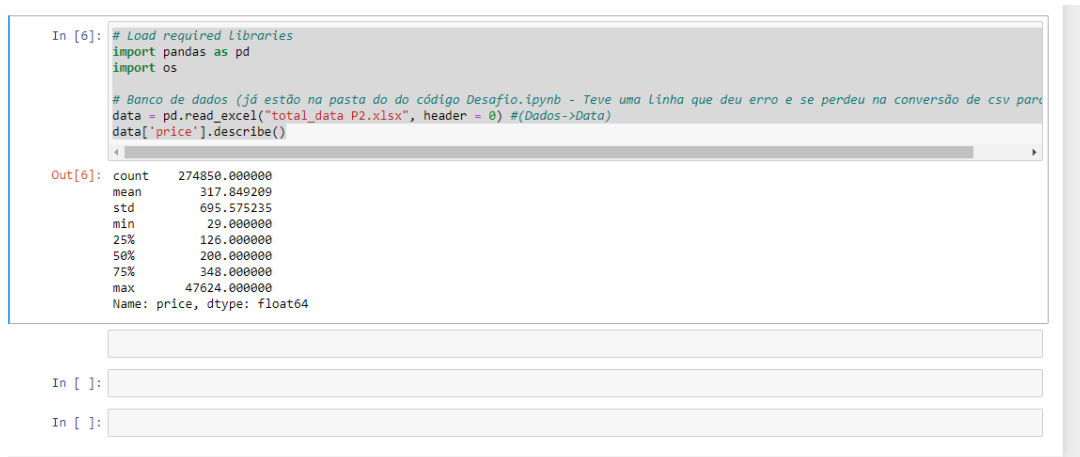
- require\_guest\_profile\_picture
- transit
- host\_name
- id
- host\_about
- host\_response\_time
- market
- description
- requires\_license
- neighbourhood
- host\_total\_listings\_count
- city
- access
- maximum\_nights
- medium\_url
- weekly\_price
- experiences\_offered
- calculated\_host\_listings\_count
- summary

É importante destacar que, dentre as colunas removidas, está a coluna "neighbourhood". No entanto, na planilha, temos a coluna "neighbourhood\_cleansed", que é mais completa. Esta última identifica o bairro de maneira geocodificada usando as coordenadas de latitude e longitude, fornecendo assim uma informação mais precisa e confiável, sem o problema de células vazias.

### 3.3 Exploratory Data Analysis (EDA)

No início da exploração dos dados, os mesmos foram importados para o Jupyter Notebook (PROJECT JUPYTER, 2023), uma ferramenta que facilita o trabalho com a linguagem Python. Depois de importar as bibliotecas essenciais, cuja lista pode ser consultada no arquivo disponível no GitHub, executamos o primeiro código para observar o comportamento da coluna de interesse, denominada "price" (Figura 2).

Figura 2. Função *describe()* executada antes de dropar dados de preço.



```
In [6]: # Load required Libraries
import pandas as pd
import os

# Banco de dados (já estão na pasta do código Desafio.ipynb - Teve uma linha que deu erro e se perdeu na conversão de csv para excel)
data = pd.read_excel("total_data P2.xlsx", header = 0) #(Dados->Data)
data['price'].describe()
```

```
Out[6]: count    274850.000000
       mean      317.849209
       std       695.575235
       min        29.000000
       25%       126.000000
       50%       200.000000
       75%       348.000000
       max      47624.000000
       Name: price, dtype: float64
```

Fonte: PUC GO, 2023.



Ao notar um desvio padrão bastante elevado, surgiu a curiosidade de examinar os valores altos de algumas acomodações. Veja as etapas do processo:

**1. Acomodações acima de R\$ 40.000:**

- Inicialmente, identificaram-se 10 acomodações nessa faixa de preço.
- Ao eliminar registros com links duplicados, restaram 4 links.
- Dos 4, apenas 1 link estava ativo, e o anúncio correspondente tinha um valor de R\$ 100.
- Esses registros foram removidos, especialmente porque o "property\_type" indicava "apartamento", o que era incompatível com o valor originalmente registrado.

**2. Acomodações acima de R\$ 30.000:**

- Foram encontradas 17 linhas nessa faixa.
- Depois de eliminar duplicatas, sobraram 4 registros.
- Em 14/10/2023, nenhum dos links estava ativo. Decidiu-se por sua exclusão.

**3. Acomodações acima de R\$ 24.000:**

- Existiam 52 linhas nesse intervalo.
- Após a remoção de duplicatas, restaram 30.
- Em 14/10/2023, 7 links ainda estavam ativos, mas os valores e tipos de acomodações não coincidiam com os da planilha. Por exemplo, havia listagens de "hostel" e "apartamento", incompatíveis com os valores apresentados. Portanto, também foram excluídos.

Para os dados restantes, com valores acima de R\$ 20.000:

- Alguns registros eram consistentes com os valores apresentados, e o "property\_type" foi identificado como "house". Além disso, as coordenadas de latitude e longitude dos 5 registros que restaram nesse intervalo de preço apontavam para localizações privilegiadas. Os valores podem ser justificados, considerando que os 5 dados foram coletados em uma época de baixa temporada.

Figura 3. Função describe() executada depois de dropar dados de preço.

```
In [4]: # Load required Libraries
import pandas as pd
import os

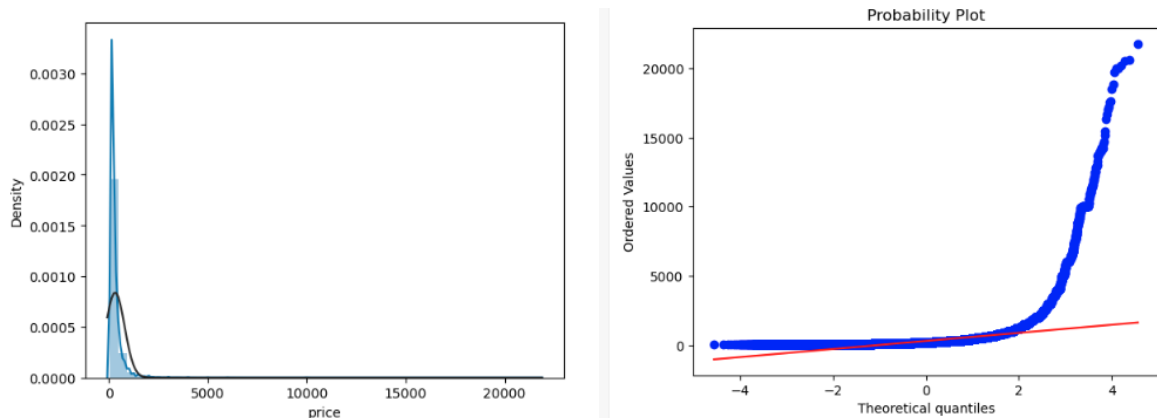
# Banco de dados (já estão na pasta do código Desafio.ipynb - Teve uma linha que deu erro e se perdeu na conversão de csv para
data = pd.read_excel("total_data P2 T.xlsx", header = 0) #(Dados->Data)
data['price'].describe()
```

```
Out[4]: count    274771.000000
       mean      309.436571
       std       474.262292
       min        29.000000
       25%       126.000000
       50%       200.000000
       75%       348.000000
       max      21778.000000
       Name: price, dtype: float64
```

Fonte: PUC GO, 2023.

Na análise da segunda rodada (Figura 3 e Gráfico 1), os dados mostraram-se mais equilibrados. Contudo, eles não se alinham perfeitamente a uma distribuição normal, conforme evidenciado pelo histograma e pelo gráfico Q-Q. Os indicadores obtidos foram:

Gráfico 1. Histograma e Gráfico Q-Q, price.



Fonte: PUC GO, 2023.

- **Skewness:** 12.834379
- **Kurtosis:** 302.009680

Para contextualizar:

- **Skewness** mede o grau de assimetria da distribuição em relação à média. Um valor próximo de 0 indica simetria. Valores positivos apontam uma cauda à direita da distribuição, enquanto negativos indicam uma cauda à esquerda.

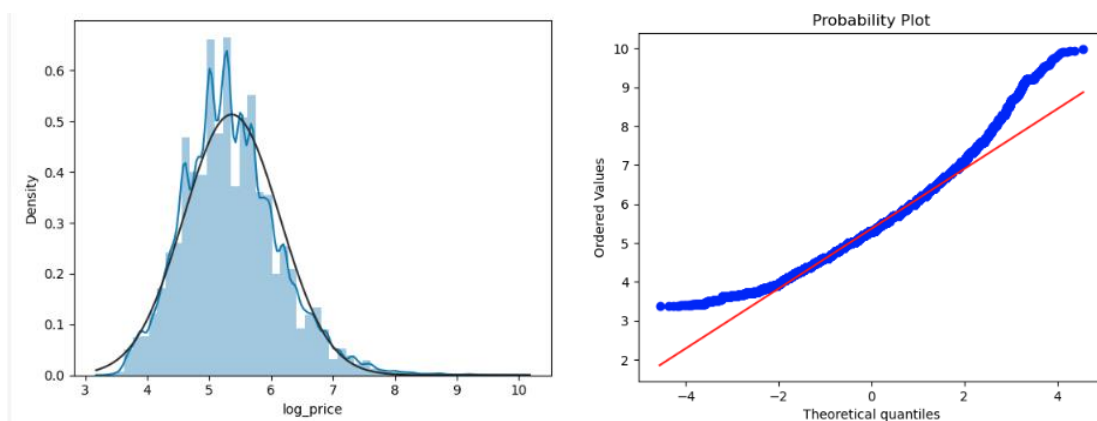
- **Kurtosis** avalia a "pontagudez" e a extensão das caudas da distribuição. Uma alta kurtosis indica presença de outliers ou caudas pesadas. Um valor próximo de 0 sugere uma cauda semelhante à de uma distribuição normal.

Uma estratégia para lidar com tal assimetria é a transformação dos dados em logaritmos naturais. A escolha de usar logaritmos, ou qualquer outra transformação, deve ser embasada em critérios analíticos e no conhecimento da natureza dos dados. Transformações logarítmicas são particularmente úteis quando enfrentamos distribuições fortemente assimétricas e o objetivo é modelar efeitos multiplicativos em vez de aditivos. Contudo, para algumas variáveis, essa abordagem pode não ser a mais adequada, especialmente para aquelas que já se aproximam da distribuição normal ou que contêm muitos valores zero.

No contexto da nossa análise, a transformação logarítmica demonstrou eficácia em tornar a distribuição de preços mais simétrica, evidenciado pela skewness reduzida. Apesar de uma curtose ainda elevada, ela sugere uma presença não tão marcante de outliers.

Em geral, as métricas indicam que a transformação logarítmica beneficiou a distribuição dos preços, aproximando-a de uma distribuição normal — condição frequentemente preferida para análises estatísticas e modelagem (Gráfico 2).

Gráfico 2. Histograma e Gráfico Q-Q, log\_price.



Fonte: PUC GO, 2023.

Nesta análise, os indicadores revelaram:

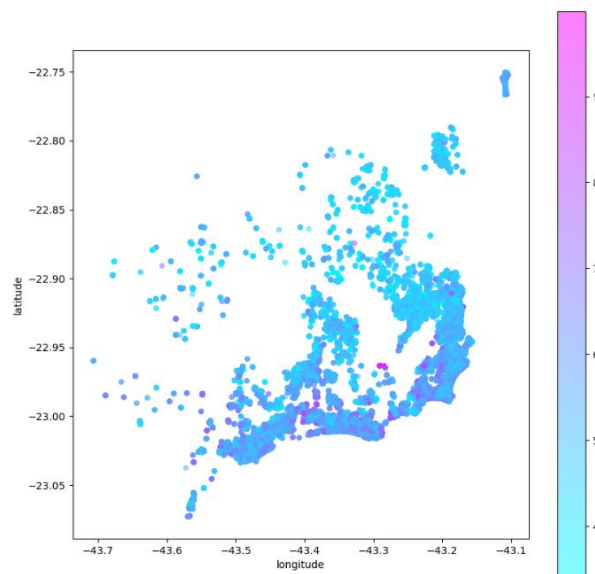
- **Skewness:** 0.596246
- **Kurtosis:** 0.921316

A aplicação da transformação logarítmica melhorou a simetria da distribuição de preços, refletido por uma skewness reduzida. Embora a curtose esteja levemente elevada, sugere-se a existência de alguns outliers, mas não em quantidade alarmante.

Em resumo, as métricas mostram que a transformação logarítmica foi efetiva em aproximar a distribuição dos preços de uma distribuição normal, condição frequentemente buscada em análises estatísticas e modelagem.

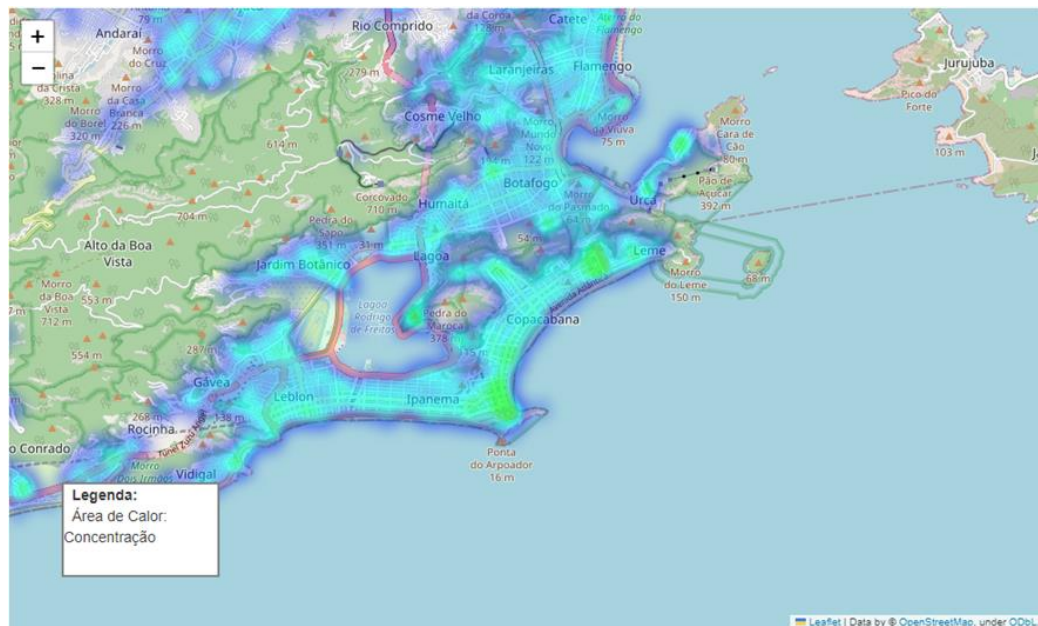
Posteriormente, buscamos compreender a distribuição dos preços no Rio de Janeiro. Ao mapear as latitudes e longitudes, identificamos visualmente uma concentração de valores mais altos ao longo do litoral. Utilizando a biblioteca Folium, também foi possível criar um mapa colorido para visualizar as regiões com maior concentração de acomodações (Figura 4 e Gráfico 3).

Gráfico 3. Latitude e longitude em função do preço.



Fonte: PUC GO, 2023.

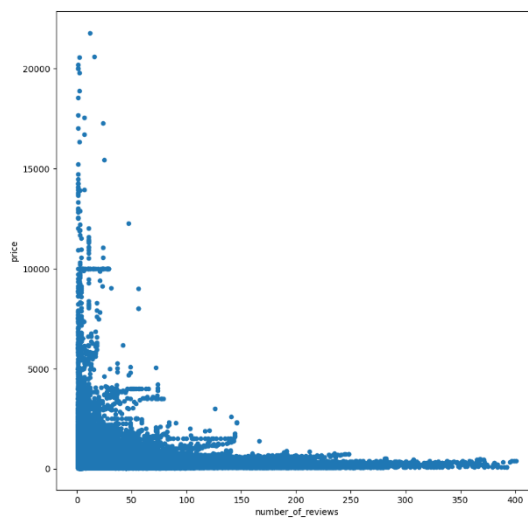
Figura 4. Mapa de calor da concentração dos dados de latitude e longitude.



Fonte: PUC GO, 2023.

Em seguida, iniciamos uma exploração mais detalhada dos dados e suas correlações. Começamos traçando um Gráfico 4 comparativo entre o número de reviews e o preço.

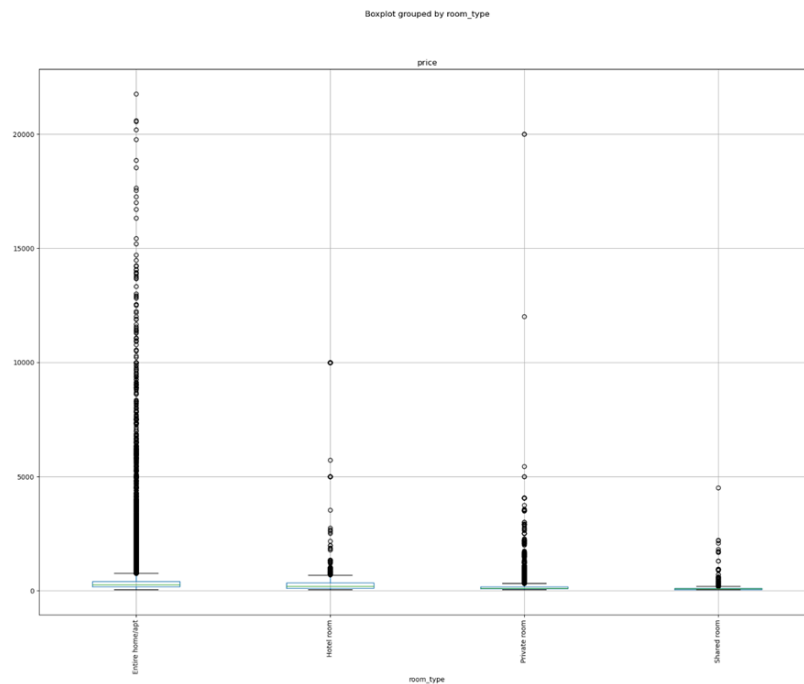
Gráfico 4. Número de reviews vs preço



Fonte: PUC GO, 2023.

Observou-se que as acomodações mais caras tendem a ter um número menor de reviews. No entanto, essa variação pode impactar nosso modelo. Assim, procedemos com a criação de um boxplot para comparar o tipo de quarto com o preço.

Gráfico 5. Tipo de acomodação vs preço

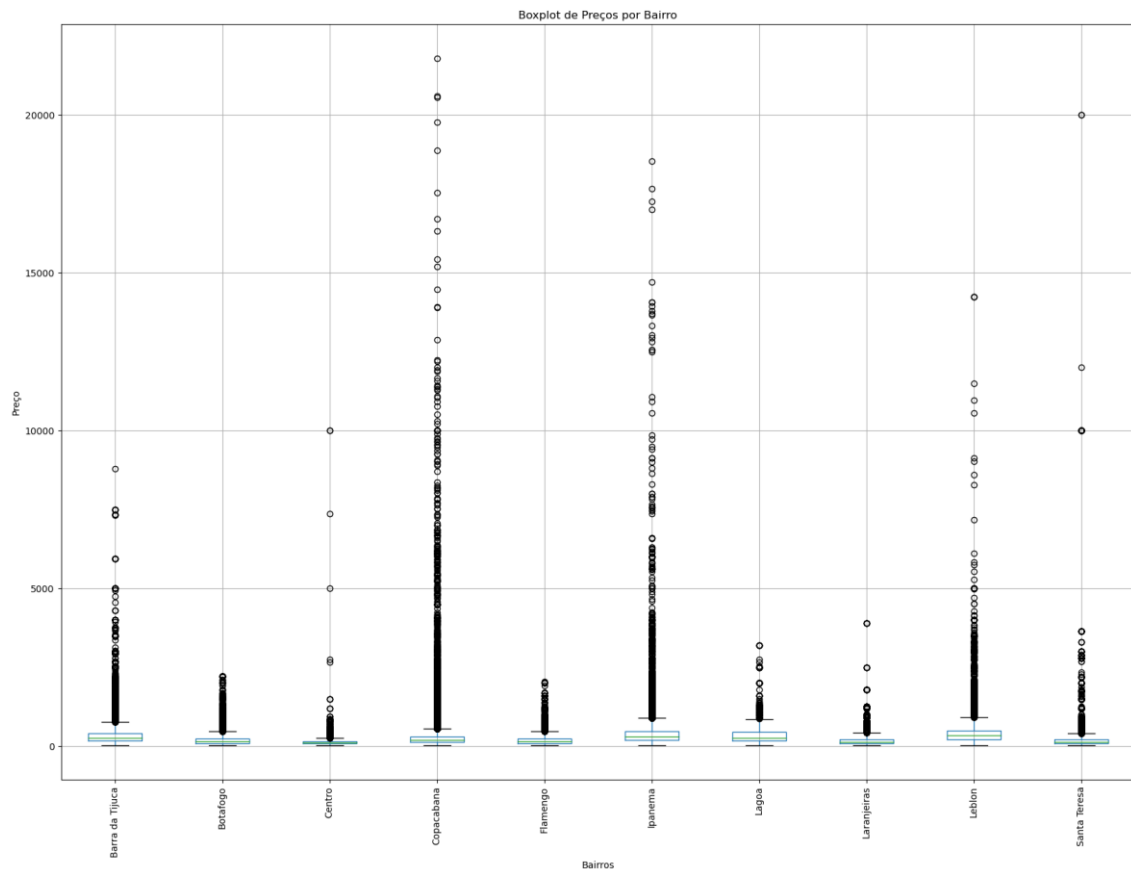


Fonte: PUC GO, 2023.

No Gráfico 5, fica evidente que acomodações particulares apresentam preços mais elevados, mas também exibem uma maior variação de preços. Por outro lado, quartos de hotel, privados e compartilhados demonstram média, mediana e quartis em faixas de preço mais estreitas, respectivamente. É particularmente notável que os quartos compartilhados têm uma variação de preço menos acentuada, gravitando mais consistentemente em torno da média.

Posteriormente, elaboramos um gráfico comparando bairros com preços. No entanto, devido à vasta quantidade de bairros, optamos por focar nos 10 principais bairros do Rio de Janeiro.

Gráfico 6. Top 10 Bairros vs preço



Fonte: PUC GO, 2023.

Foi possível identificar os bairros com maior variabilidade de preços e também aqueles com médias mais elevadas. Esta observação é crucial para direcionar nosso foco a essas áreas específicas (Gráfico 6).

Considerando que o Rio de Janeiro é uma cidade com períodos de alta demanda ao longo do ano, decidimos analisar a variação de preços em relação aos bimestres, focando especificamente nesses 10 bairros principais (Figura 5).

Figura 5. Resultado função *describe()* para o conjunto dos 10 bairros por bimestre.

```

Contagem de registros para meses 12 e 1: 38381
Contagem de registros para meses 2 e 3: 45437
Contagem de registros para meses 4 e 5: 46930
Contagem de registros para meses 6 e 7: 20591
Contagem de registros para meses 8 e 9: 28509
Contagem de registros para meses 10 e 11: 32517

```

	Meses 12 e 1	Meses 2 e 3	Meses 4 e 5	Meses 6 e 7	Meses 8 e 9 \
count	38381.000000	45437.000000	46930.000000	20591.000000	28509.000000
mean	321.463172	349.811497	344.849584	309.852557	297.380827
std	452.431534	576.456468	564.027333	401.014097	398.298558
min	31.000000	30.000000	31.000000	31.000000	29.000000
25%	138.000000	145.000000	144.000000	135.000000	129.000000
50%	209.000000	220.000000	221.000000	201.000000	199.000000
75%	352.000000	380.000000	379.000000	351.000000	329.000000
max	14063.000000	21778.000000	20562.000000	12934.000000	13938.000000

```

Meses 10 e 11
count    32517.000000
mean      306.202079
std       434.193348
min        30.000000
25%       130.000000
50%       202.000000
75%       342.000000
max       14068.000000

```

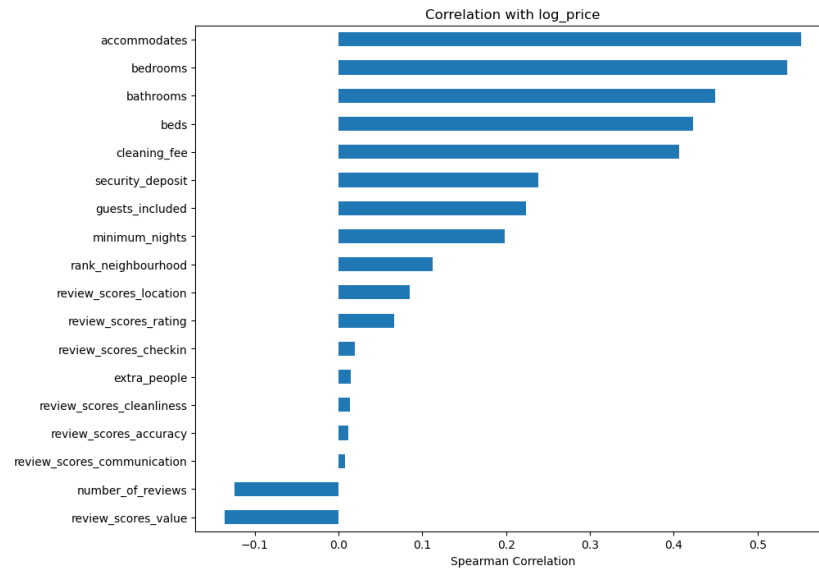
Fonte: PUC GO, 2023.

Antecipávamos uma variação significativa nas médias de preços durante períodos de pico, como o final do ano. De fato, observamos que os meses de fevereiro e março apresentam as médias mais elevadas.

No entanto, o segundo semestre tende a mostrar preços mais baixos. Embora uma análise bimestral possa oferecer insights mais detalhados, ela também complexifica o modelo. Por isso, optamos por manter uma perspectiva global (Gráfico 7).



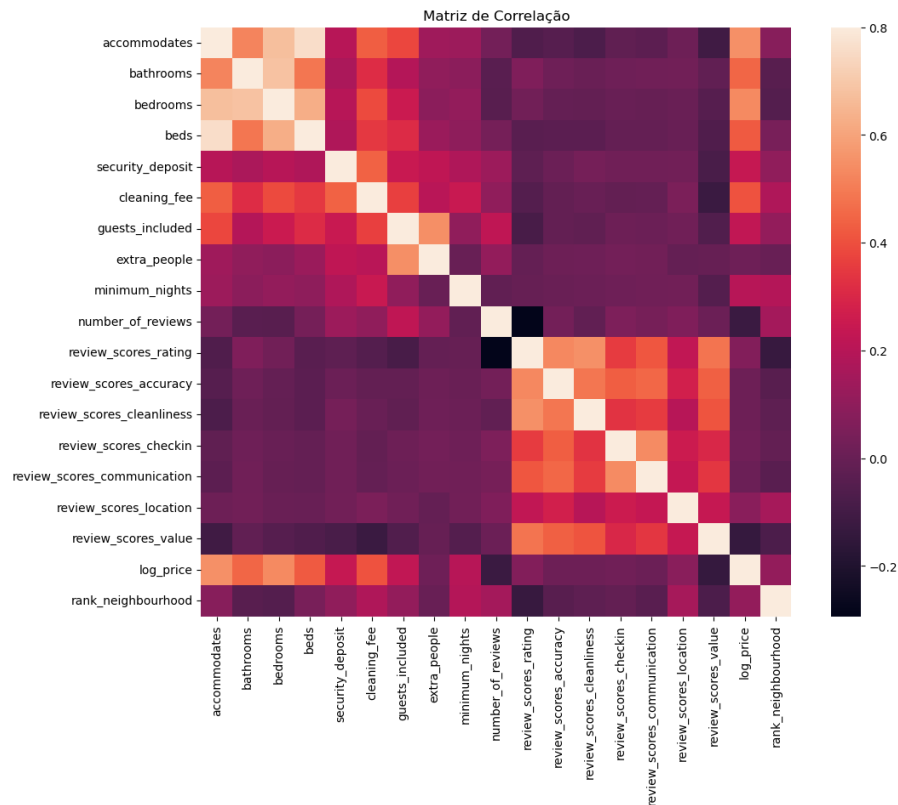
Gráfico 7. Correlação das principais variáveis para a variável preço.



Fonte: PUC GO, 2023.

Também desenvolvemos uma matriz de correlação, representada por um heatmap, para facilitar a visualização (Gráfico 8).

Gráfico 8. Heatmap das principais variáveis para a variável preço.



Fonte: PUC GO, 2023.

Percebemos então que as variáveis de interesse são: 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'cleaning\_fee', 'security\_deposit', 'guests\_included', 'minimum\_nights', 'review\_scores\_location', 'review\_scores\_rating', 'neighbourhood\_cleansed\_', 'property\_type\_', 'room\_type\_' 'bed\_type\_'.

### 3.4 Machine Learning

Para cada modelo, ele é treinado no conjunto de treinamento e avaliado no conjunto de treinamento e teste. A métrica usada para avaliação é o Erro Quadrático Médio da Raiz (RMSE).

Interpretação do modelo Lasso.

Os coeficientes do modelo Lasso são examinados. O modelo Lasso tem a propriedade de fazer seleção de características, definindo alguns coeficientes como zero. Os coeficientes são ordenados em ordem decrescente de magnitude e impressos.

Visualização dos resíduos para o modelo Lasso.

Os resíduos (a diferença entre os valores verdadeiros e previstos) para o modelo Lasso são visualizados em um gráfico de dispersão disponível no código no github.

#### 1. Regressão Linear (LinearRegression)

A regressão linear é um dos métodos mais simples e amplamente utilizados para modelar a relação entre uma variável dependente (neste caso, price) e uma ou mais variáveis independentes (as características, como accommodates, bathrooms, etc.). A ideia principal é encontrar a melhor linha (em 2D), plano (em 3D) ou hiperplano (em mais de 3 dimensões) que se ajuste aos dados.

#### 2. Ridge Regression (RidgeCV)

A regressão Ridge é uma técnica de regularização que tenta evitar o ajuste excessivo (overfitting) penalizando coeficientes grandes. Ela adiciona uma penalidade à soma dos quadrados dos coeficientes. A magnitude dessa penalidade é controlada por um parâmetro, frequentemente denotado como  $\alpha$ .

### 3. Lasso Regression (LassoCV)

A regressão Lasso (Least Absolute Shrinkage and Selection Operator) é outra técnica de regularização que, como a regressão Ridge, adiciona uma penalidade aos coeficientes. No entanto, a penalidade do Lasso é a soma dos valores absolutos dos coeficientes. Isso tem a propriedade interessante de forçar alguns coeficientes a serem exatamente zero, o que equivale a uma forma de seleção automática de características.

### 4. ElasticNet Regression (ElasticNetCV)

ElasticNet é uma combinação das regularizações Ridge e Lasso. Ela adiciona ambas as penalidades (soma dos quadrados e soma dos valores absolutos) aos coeficientes. Isso pode ser útil quando há muitas características correlacionadas.

#### Construção e Avaliação dos Modelos:

Uma lista chamada `models` é definida. Cada item da lista é uma tupla contendo:

Um objeto do modelo (por exemplo, `LinearRegression()`, `RidgeCV(...)`, etc.)

Uma string que representa o nome do modelo (por exemplo, "LinearRegression", "Ridge", etc.)

```
In [ ]: models = [
    (LinearRegression(), "LinearRegression"),
    (RidgeCV(alphas=[1000, 100, 50, 20, 10, 1, 0.1, 0.01]), "Ridge"),
    (LassoCV(alphas=[1000, 100, 50, 20, 10, 1, 0.1, 0.01], max_iter=10000), "Lasso"),
    (ElasticNetCV(alphas=[1000, 100, 50, 20, 10, 1, 0.1, 0.01], l1_ratio=[0.001, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9]), "Elastic")
]
```

O código, em seguida, itera em cada tupla (listas) na lista `models`

```
In [ ]: for model, name in models:
```

Dentro deste loop:

O modelo é treinado no conjunto de treinamento usando o método `fit`:

```
In [ ]: model.fit(X_train_scaled, y_train)
```

O modelo é usado para fazer previsões no conjunto de treinamento e no conjunto de teste:

```
In [ ]: y_pred_train = model.predict(X_train_scaled)
        y_pred_test = model.predict(X_test_scaled)
```

O desempenho do modelo é avaliado usando o Erro Quadrático Médio da Raiz (RMSE) para ambos os conjuntos de treinamento e teste:

```
In [ ]: rmse_train = np.sqrt(mean_squared_error(y_pred=y_pred_train, y_true=y_train))
        rmse_test = np.sqrt(mean_squared_error(y_pred=y_pred_test, y_true=y_test))
```

Os resultados são impressos para cada modelo:

```
In [ ]: print(f"{name} - RMSE Train: {rmse_train}, RMSE Test: {rmse_test}")
```

Dessa forma, o código está treinando e avaliando cada um dos quatro modelos no conjunto de dados fornecido. Ele fornece uma avaliação de desempenho para cada modelo, permitindo que você compare como eles se saem em relação uns aos outros.

### 3.5 Avaliação de Modelos

Baseline RMSE:

Este é um modelo de referência simples que prevê o preço mediano do conjunto de treinamento para todas as observações no conjunto de teste. O RMSE é 511.81, que serve como uma linha de base para comparar o desempenho dos outros modelos.

Regressão Linear:

O RMSE de treinamento é 391.01 e o RMSE de teste é um valor extremamente alto. Isso sugere que o modelo de regressão linear está tendo um problema de "overfitting" extremo ou pode haver um erro no código ou nos dados.

Ridge:

A regressão Ridge produziu um RMSE de treinamento de 388.59 e um RMSE de teste de 420.19. A performance no conjunto de teste é próxima do baseline, mas melhor.

Lasso:

A regressão Lasso produziu um RMSE de treinamento de 388.59 e um RMSE de teste de 420.18. A performance é muito semelhante ao Ridge.

ElasticNet:

A regressão ElasticNet produziu um RMSE de treinamento de 388.59 e um RMSE de teste de 420.22. Novamente, a performance é semelhante ao Ridge e ao Lasso.

Interpretação dos Coeficientes do Modelo Lasso:

Os coeficientes listados representam a mudança prevista na variável dependente (neste caso, o preço) para um aumento de uma unidade na variável independente correspondente, mantendo todas as outras variáveis constantes.

Por exemplo:

bathrooms: 102.66053538470577

Isso sugere que, para cada banheiro adicional em uma propriedade, espera-se que o preço aumente em aproximadamente 102.66 unidades monetárias (seja qual for a moeda), mantendo todas as outras variáveis constantes.

accommodates: 77.16479969885451

Por cada pessoa adicional que uma propriedade pode acomodar, o preço previsto aumenta em cerca de 77.16 unidades monetárias.

security\_deposit: 68.45064485235392

Para cada unidade monetária adicional exigida como depósito de segurança, o preço previsto aumenta em 68.45 unidades monetárias.

O modelo Lasso tem a propriedade de fazer seleção de características, o que significa que ele pode definir coeficientes de algumas características para serem exatamente zero. Isso sugere que essas características são consideradas "irrelevantes" para a previsão. No entanto, neste caso, muitas características têm coeficientes não nulos, indicando sua relevância na previsão do preço.

No código, após treinar o modelo Lasso, ele extrai e lista os coeficientes em ordem decrescente de magnitude (valor absoluto). Isso ajuda a entender rapidamente quais características têm a maior influência (positiva ou negativa) na previsão do preço.

O modelo de regressão linear parece ter sérios problemas, possivelmente devido à multicolinearidade ou algum outro problema nos dados ou no código.

Os modelos Ridge, Lasso e ElasticNet têm desempenhos semelhantes e são melhores do que o modelo de linha de base, mas não por uma margem significativa.

Os coeficientes do modelo Lasso dão uma ideia das características mais importantes ao prever o preço. Por exemplo, o número de banheiros, a capacidade de acomodação e o tipo de propriedade são algumas das características mais influentes.

Posteriormente, realizamos uma correlação, utilizando o método de Spearman, entre as principais variáveis numéricas e o preço para entender suas inter-relações.

### **3.6 MODELO GLM**

RMSE da validação cruzada:  $683.26 \pm 575.10$

O RMSE (Erro Quadrático Médio da Raiz) é uma medida popular para avaliar o desempenho de modelos de regressão. Valores menores indicam melhores desempenhos.

"Validação cruzada" é uma técnica utilizada para avaliar a capacidade de generalização de um modelo. O conjunto de dados é dividido em 'k' subconjuntos, e o modelo é treinado em 'k-1' destes e testado no subconjunto restante. Isso é repetido 'k' vezes, de modo que cada subconjunto seja usado como conjunto de teste exatamente uma vez.

O valor "683.26" é o RMSE médio obtido dessas 'k' iterações.

O valor " $\pm 575.10$ " indica o desvio padrão do RMSE ao longo das 'k' iterações. Isso sugere que houve uma variação considerável no desempenho do modelo nas diferentes iterações da validação cruzada. Uma variação tão grande pode indicar instabilidade no modelo ou no conjunto de dados, talvez devido à presença de outliers ou à natureza do próprio conjunto de dados.

RMSE de teste: 412.59

Este é o RMSE quando o modelo foi avaliado em um conjunto de teste separado (não utilizado durante o treinamento ou a validação cruzada).

Um RMSE de 412.59 indica que, em média, o modelo erra em 412.59 unidades monetárias ao prever o preço. Comparado ao RMSE da validação cruzada, o desempenho no conjunto de teste parece ser melhor.

Conclusões:

O modelo GLM tem um desempenho razoável, com um RMSE de teste de 412.59, que é melhor do que o RMSE da linha de base que você forneceu anteriormente (511.81) e também melhor do que o RMSE da validação cruzada.

A grande variação no RMSE durante a validação cruzada pode ser motivo de preocupação e merece uma investigação mais aprofundada. Pode ser útil verificar a distribuição dos dados e a presença de outliers, ou considerar outras formas de pré-processamento ou modelagem.

No geral, o GLM parece ser um modelo promissor para este conjunto de dados, com base no RMSE do conjunto de teste, mas a instabilidade observada na validação cruzada pode precisar ser abordada.

GLM (Modelo Linear Generalizado): O GLM oferece uma extensão flexível da regressão linear tradicional, permitindo respostas com diferentes modelos de distribuição de erro além da normal. Este modelo amplia a regressão linear relacionando-a com a variável resposta por meio de uma função de ligação, e também permite que a variância de cada observação dependa do valor previsto.

Quando o GLM utiliza a distribuição gaussiana, ele se assemelha muito à regressão de mínimos quadrados ordinários, assumindo que a variável alvo tem uma distribuição normal. Esta é a forma mais comum de GLM.

Importante lembrar que os dados de número de reviews interferem nos ruídos dos preços.

Ao aplicar um filtro para considerar apenas reviews superiores a 50, obtivemos os seguintes resultados:

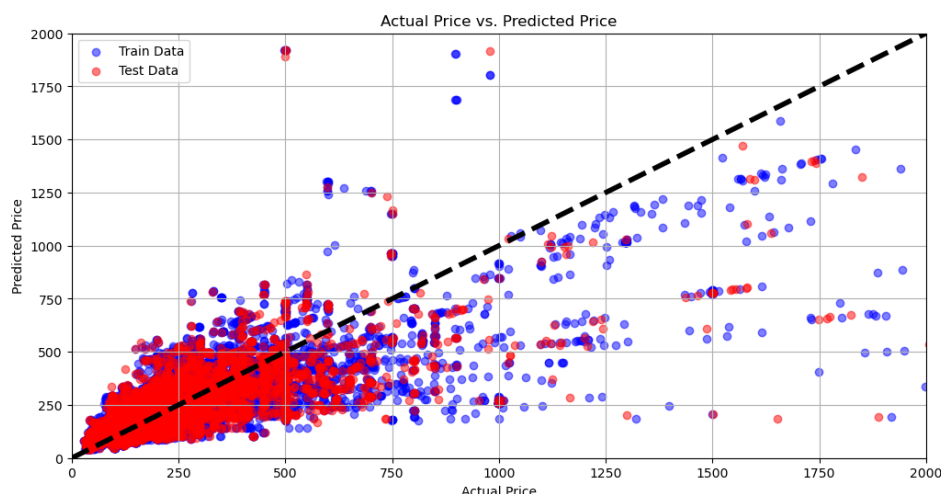
RMSE proveniente da validação cruzada:  $151.29 \pm 24.30$

RMSE no conjunto de teste: 136.31

Isso indica que o modelo teve um desempenho um pouco melhor no conjunto de teste em comparação com a validação cruzada. O valor RMSE (Root Mean Square Error) é uma medida de quão bem o modelo prevê os dados; valores mais baixos indicam melhores previsões. O intervalo  $\pm 24,30$  na validação cruzada dá uma ideia da variabilidade do desempenho do modelo em diferentes subconjuntos de dados.

Percebe-se que o Gráfico 9 de preços previsto versus atual está com a previsão mais assertiva (vide código github).

Gráfico 9. Preço atual vs Preço Previsto.



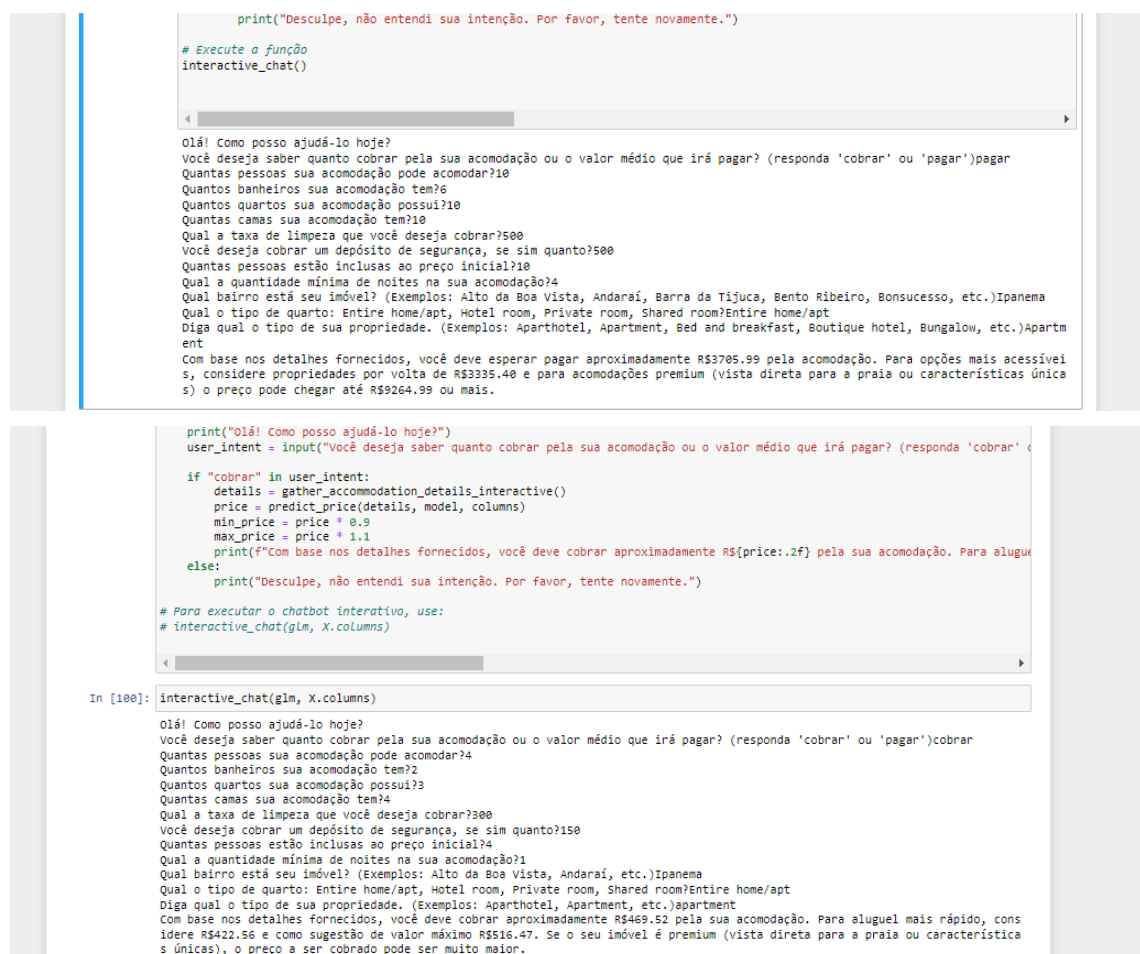
Fonte: PUC GO, 2023.

Ao tentar implementar um chatbot utilizando a API da OpenAI, nos deparamos com um obstáculo: recebemos a mensagem "Você excedeu sua cota atual. Verifique seu plano e detalhes de cobrança." Suspeitamos que o tamanho do nosso banco de dados possa ter sido o causador dessa limitação. Diante disso, optamos por uma solução alternativa, desenvolvendo a funcionalidade diretamente no Jupyter (PROJECT JUPYTER, 2023), a qual operou eficazmente dentro do ambiente Python. Para testar e verificar nosso trabalho, convidamos você a acessar o código completo disponível em nosso repositório no GitHub.

#### 4. CONCLUSÕES

Teste para acomodações com perfil alto padrão e perfil médio padrão (Figura 6).

Figura 6. Resultados da calculadora em dois testes.



```

print("Desculpe, não entendi sua intenção. Por favor, tente novamente.")

# Execute a função
interactive_chat()

Olá! Como posso ajudá-lo hoje?
Você deseja saber quanto cobrar pela sua acomodação ou o valor médio que irá pagar? (responda 'cobrar' ou 'pagar')pagar
Quantas pessoas sua acomodação pode acomodar?10
Quantos banheiros sua acomodação tem?6
Quantos quartos sua acomodação possui?10
Quantas camas sua acomodação tem?10
Qual a taxa de limpeza que você deseja cobrar?500
Você deseja cobrar um depósito de segurança, se sim quanto?500
Quantas pessoas estão inclusas ao preço inicial?10
Qual a quantidade mínima de noites na sua acomodação?4
Qual bairro está seu imóvel? (Exemplos: Alto da Boa Vista, Andaraí, Barra da Tijuca, Bento Ribeiro, Bonsucesso, etc.)Ipanema
Qual o tipo de quarto: Entire home/apt, Hotel room, Private room, Shared room?Entire home/apt
Diga qual o tipo de sua propriedade. (Exemplos: Aparthotel, Apartment, Bed and breakfast, Boutique hotel, Bungalow, etc.)Apartment
Com base nos detalhes fornecidos, você deve esperar pagar aproximadamente R$3705.99 pela acomodação. Para opções mais acessíveis, considere propriedades por volta de R$3335.40 e para acomodações premium (vista direta para a praia ou características únicas) o preço pode chegar até R$9264.99 ou mais.

print("Olá! Como posso ajudá-lo hoje?")
user_intent = input("Você deseja saber quanto cobrar pela sua acomodação ou o valor médio que irá pagar? (responda 'cobrar' ou 'pagar')")

if "cobrar" in user_intent:
    details = gather_accommodation_details_interactive()
    price = predict_price(details, model, columns)
    min_price = price * 0.9
    max_price = price * 1.1
    print(f"Com base nos detalhes fornecidos, você deve cobrar aproximadamente R${price:.2f} pela sua acomodação. Para aluguel mais rápido, considere propriedades por volta de R${min_price:.2f} e para acomodações premium (vista direta para a praia ou características únicas) o preço pode chegar até R${max_price:.2f} ou mais.")
else:
    print("Desculpe, não entendi sua intenção. Por favor, tente novamente.")

# Para executar o chatbot interativo, use:
# interactive_chat(glm, X.columns)

In [100]: interactive_chat(glm, X.columns)

Olá! Como posso ajudá-lo hoje?
Você deseja saber quanto cobrar pela sua acomodação ou o valor médio que irá pagar? (responda 'cobrar' ou 'pagar')cobrar
Quantas pessoas sua acomodação pode acomodar?4
Quantos banheiros sua acomodação tem?2
Quantos quartos sua acomodação possui?3
Quantas camas sua acomodação tem?4
Qual a taxa de limpeza que você deseja cobrar?300
Você deseja cobrar um depósito de segurança, se sim quanto?150
Quantas pessoas estão inclusas ao preço inicial?4
Qual a quantidade mínima de noites na sua acomodação?1
Qual bairro está seu imóvel? (Exemplos: Alto da Boa Vista, Andaraí, etc.)Ipanema
Qual o tipo de quarto: Entire home/apt, Hotel room, Private room, Shared room?Entire home/apt
Diga qual o tipo de sua propriedade. (Exemplos: Aparthotel, Apartment, etc.)apartment
Com base nos detalhes fornecidos, você deve cobrar aproximadamente R$469.52 pela sua acomodação. Para aluguel mais rápido, considere R$422.56 e como sugestão de valor máximo R$516.47. Se o seu imóvel é premium (vista direta para a praia ou características únicas), o preço a ser cobrado pode ser muito maior.

```

Fonte: PUC GO, 2023.



Após uma série de testes rigorosos, podemos concluir que a calculadora funcionou de maneira eficaz e eficiente. Os resultados obtidos foram consistentes e satisfatórios, evidenciando a confiabilidade e precisão do instrumento. Esta avaliação positiva ressalta o sucesso do dispositivo em cumprir sua função proposta.

Link para GitHub: <https://github.com/TPasqualetto/V-Desafios-de-Dados---PUC-Goi-s---AIRBNB.git>

## REFERÊNCIAS

PROJECT JUPYTER. Jupyter Notebook.. Berkeley, CA, 2020. Disponível em: <<https://jupyter.org/>>. Acesso em: 16 out. 2023.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS (PUC GOIÁS). V Desafio em Ciências de Dados: Predição de Preços no Mercado de Hospedagem Virtual. Goiânia, 2023. Base de Dados: Airbnb 2018 a 2020 Rio de Janeiro. Organizado por: 9 CCTI PUC GO 2023/2, Mestrado em Engenharia de Produção e Sistemas (PPGEPS), Núcleo de Matemática do NEPE.

PASQUALETTO, Thales. V-Desafios-de-Dados---PUC-Goi-s---AIRBNB. GitHub. 2023. Disponível em: < <https://github.com/TPasqualetto/V-Desafios-de-Dados---PUC-Goi-s---AIRBNB.git> >.