Desafio Cientista de Dados

LH_CD_GABRIEL VINÍCIUS DE FIGUEIREDO

Introdução

Neste desafio o objetivo foi desenvolver um modelo de previsão de preços de aluguéis temporários a partir de dados de um concorrente com o intuito de, após realizar Análise Exploratória dos Dados (EDA), responder perguntas ao longo do relatório e realizar a predição de preços considerando o modelo de análise escolhido.

Informações do Dataset

Inicialmente prepara-se os dados fornecidos em arquivo .csv nomeado de "teste_indicium_precificacao", com **dataset** que contém 16 colunas com informações sobre anúncios, incluindo informações de localização, características e informações do imóvel e informações do anfitrião e disponibilidade.

Código

```
# Carregamento do dataset
data = pd.read_csv('teste_indicium_precificacao.csv', encoding='utf-
8', on_bad_lines='skip')

# Informações gerais do dataset
print(data.head())
print(data.info())
```

Informações gerais do Dataset		
1	id	Informações do imóvel
2	nome	Informações do imóvel
3	host_id	Informações do anfitrião e disponibilidade
4	host_name	Informações do anfitrião e disponibilidade
5	bairro_group	Informações de localização
6	bairro	Informações de localização
7	latitude	Informações de localização
8	longitude	Informações de localização
9	room_type	Características do imóvel
10	price	Características do imóvel
11	minimo_noites	Características do imóvel
12	numero_de_reviews	Informações sobre anúncios
13	ultima_review	Informações sobre anúncios
14	reviews_por_mes	Informações sobre anúncios
15	alculado_host_listings_count	Informações sobre anúncios
16	disponibilidade_365	Informações do anfitrião e disponibilidade

Diagnóstico e tratamento dos Dados

Foram identificadas algumas inconsistências no banco de dados, comprovando a necessidade de tratamento, dentre os erros estão:

- Erros estruturais – Linhas com número incorreto de colunas.

Código

```
# Definição do número de colunas esperado conforme arquivo .csv
num_colunas_esperado = 16

# Verificação do número de colunas existentes em cada linha
with open('teste_indicium_precificacao.csv', 'r', encoding='utf-8') as
f:
    for i, linha in enumerate(f):
        num_colunas = len(linha.split(','))
        if num_colunas != num_colunas_esperado:
            print(f"Linha {i+1} tem {num_colunas} colunas")
```

- Valores ausentes – Colunas “reviews_por_mes”, “nome” e “host_name” apresentaram valores nulos;
    - As colunas “nome” e “host_name” demonstraram poucos valores ausentes; então podemos preencher com "Sem Nome" e "Desconhecido";
    - As colunas “ultima_review” e “reviews_por_mes” possuem **10.052** valores ausentes, indicando que muitas listagens nunca receberam avaliações;

Código

```
# Contagem de valores ausentes em cada coluna
missing_values = data.isnull().sum()

# Exibição de colunas que possuem valores ausentes
missing_values = missing_values[missing_values > 0]
print("\nValores Ausentes no Dataset:")
print(missing_values)
```

Valores ausentes no Dataset		
1	nome	16
2	host_name	21
3	ultima_review	10052
4	reviews_por_mes	10052

- Outliers – Valores discrepantes em “price” e “minimo_noites”.

- Outliers em “price” – A média de preços fora identificado como **$152,00**, porém com algumas discrepâncias, como por exemplo o valor máximo de **$10.000,00**;
- Outliers em “minimo_noites” – Por volta de **75%** dos imóveis exigem no máximo 5 noites, logo, valores como o máximo de **1.250** noites devem ser evitados para que não ocorra distorções significativas na análise dos dados.

Código

```
# Estatísticas descritivas para verificação de valores discrepantes
print("\nEstatísticas Descritivas:")
print(data[['price', 'minimo_noites']].describe())

# Boxplot do preço para visualização de outliers
plt.figure(figsize=(8, 5))
sns.boxplot(x=data['price'])
plt.title("Boxplot de Preços")
plt.xlabel("Preço ($)")
plt.show()

# Boxplot do número mínimo de noites
plt.figure(figsize=(8, 5))
sns.boxplot(x=data['minimo_noites'])
plt.title("Boxplot do Número Mínimo de Noites")
plt.xlabel("Mínimo de Noites")
plt.show()
```

Estatísticas Descritivas		
Estatísticas	price	mínimo_noites
count	48894.000000	48894.000000
mean	152.720763	7.030085
std	240.156625	20.510741
min	0.000000	1.000000
25%	69.000000	1.000000
50%	106.000000	3.000000
75%	175.000000	5.000000
max	10000.000000	1250.000000

Ao identificar as inconsistências, logo se dá seguimento ao devido tratamento dos dados.

1. Garantir que todas as linhas têm o número correto de colunas:

Código

```
# Carregamento do dataset
data = pd.read_csv('teste_indicium_precificacao.csv', encoding='utf-
8', on_bad_lines='skip')
```

2. Substituição de valores ausentes:
    a. Colunas “nome” e “host_name” – Preenchimento de dados nulos por "Sem Nome" e "Desconhecido";
    b. Colunas “ultima_review” e “reviews_por_mes” – Preenchimento de dados ausentes com 0 para representar listagens sem avaliações;

Código

```
# Preencher valores ausentes
data.loc[:, 'nome'] = data['nome'].fillna('Sem Nome')
data.loc[:, 'host_name'] = data['host_name'].fillna('Desconhecido')
data.loc[:, 'reviews_por_mes'] = data['reviews_por_mes'].fillna(0)
data['ultima_review'] = pd.to_datetime(data['ultima_review'])
```

3. Remoção de outliers:
    a. Valores em “price” acima de $1.000,00;
    b. Valores em “minimo_noites” acima de 365, limitando em anual.

Código

```
# Remoção de outliers
outliers_price = 1000
data = data[data['price'] <= outliers_price]
data = data[data['minimo_noites'] <= 365]
```

Análise Exploratória de Dados (EDA)

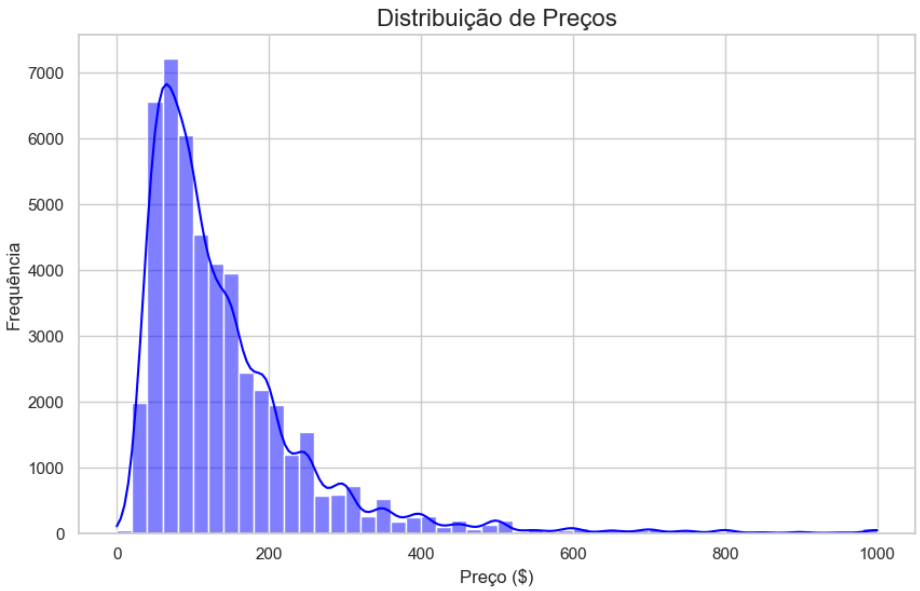
Após limpeza e tratamento dos dados deu-se seguimento a obtenção de estatísticas Descritivas, obtendo os seguintes dados relevantes para análise:

Código

```
# Estatísticas descritivas
print("Estatísticas Descritivas Após Limpeza:")
print(data.describe())


# Gráficos Exploratórios
# Gráfico 1 - Distribuição de preços
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.histplot(data['price'], bins=50, kde=True, color='blue')
plt.title("Distribuição de Preços", fontsize=16)
plt.xlabel("Preço ($)")
plt.ylabel("Frequência")
plt.show()
```

- Preço Médio (price) – O preço médio obtido foi de $141,28, com um desvio padrão de $116,74 e uma mediana de $105, indicando assimetria à direita, com um grande número de anúncios concentrados abaixo de $200,00, se mostrando consistente com a expectativa de preços em uma plataforma de aluguel a curto prazo;



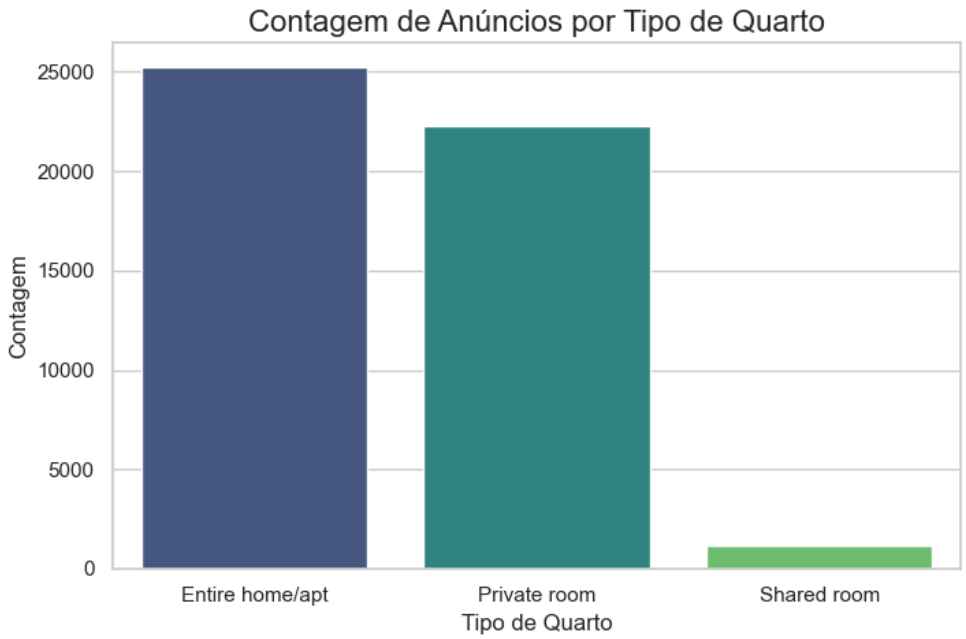
- Mínimo de Noites (minimo_noites) – Obteve-se uma média de 6,78 noites, porém, grande parcela dos anúncios, 75%, exigem até 5 noites;
- Disponibilidade Anual (disponibilidade_365) – Média de 112 dias, mas 25% dos anúncios não possuem disponibilidade, ou seja, 0 dias disponíveis;
- Número de Avaliações (numero_de_reviews) – Em média são 23 avaliações por anúncio, com um máximo de 629 avaliações apresentadas em um único anúncio.

Para melhor análise, além do gráfico de distribuição de preços obtido anteriormente, faz-se também avaliação por outros tipos de grupamento como:

- Contagem de Anúncios por Tipo de Quarto – Onde A maioria dos anúncios correspondem a 'Entire home/apt', seguido por 'Private room' são as categorias predominantes, com maior participação no mercado e preços médios superiores às outras categorias;

Código

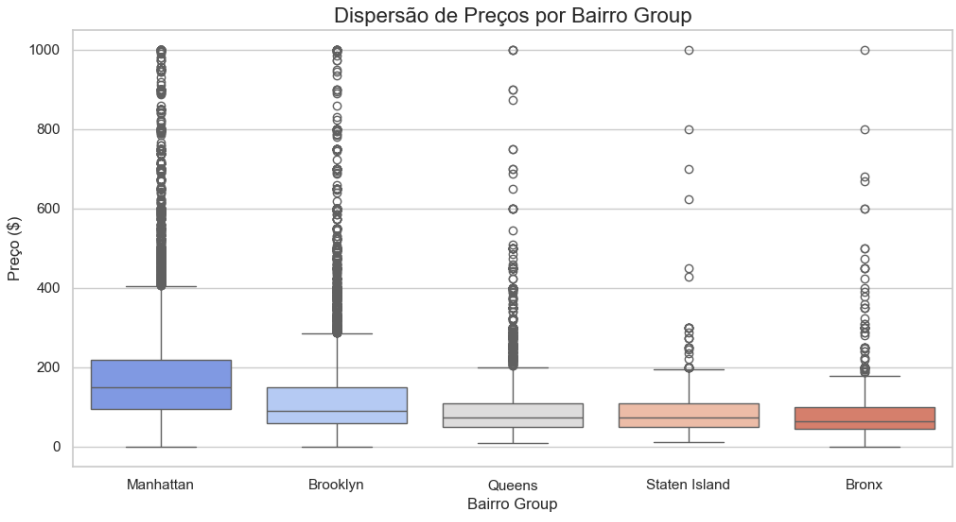
```
# Gráfico 2 - Contagem de anúncios por tipo de quarto
plt.figure(figsize=(8, 5))
sns.countplot(data=data, x='room_type', palette='viridis',
hue='room_type', dodge=False)
plt.title("Contagem de Anúncios por Tipo de Quarto", fontsize=16)
plt.xlabel("Tipo de Quarto")
plt.ylabel("Contagem")
plt.legend([], [], frameon=False)
plt.show()
```



- Boxplot de Preços por Bairro – Compara a dispersão de preços entre os bairros, onde pode-se observar que os preços em Manhattan são significativamente mais altos do que em outros bairros, com preços médios de $178,93, enquanto áreas periféricas apresentam preços menores, como é o caso de Brooklyn com preços médios de $117,81;

Código

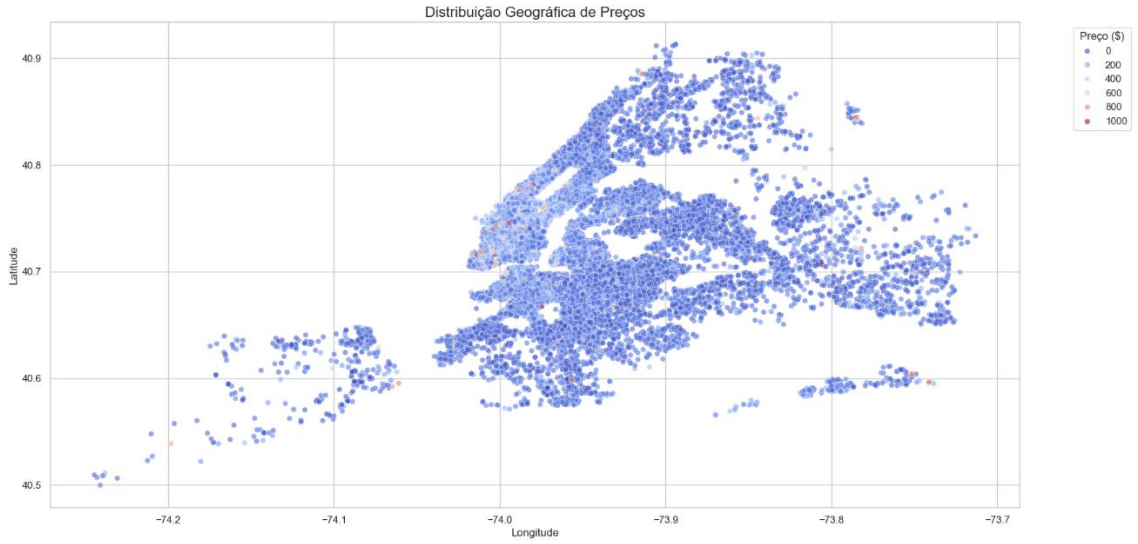
```
# Gráfico 3 - Boxplot de preços por bairro_group
plt.figure(figsize=(12, 6))
sns.boxplot(data=data, x='bairro_group', y='price',
hue='bairro_group', palette='coolwarm', legend=False)
plt.title("Dispersão de Preços por Bairro Group", fontsize=16)
plt.xlabel("Bairro Group")
plt.ylabel("Preço ($)")
plt.show()
```



- Mapa de Preços Geográfico – Visualiza a distribuição de preços pela distribuição geográfica, observando-se que os preços mais altos estão concentrados em Manhattan e área turísticas próximas, enquanto áreas periféricas apresentam preços menores.

Código

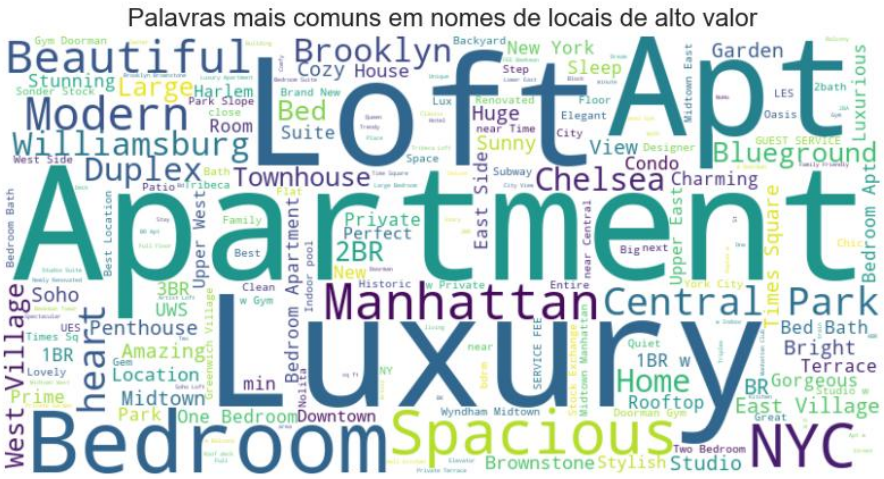
```
# Gráfico 4 - Mapa de preços geográfico
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='longitude', y='latitude', hue='price',
palette='coolwarm', alpha=0.6)
plt.title("Distribuição Geográfica de Preços", fontsize=16)
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.legend(title='Preço ($)', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.show()
```



- Padrões em locais de alto valor – Ao utilizar uma WordCloud, identifica-se palavras comuns como "Luxury", "Loft", "Brownstone", e "Penthouse", sendo esses recorrentes em anúncios mais caros como top 10%.

Código

```
# WordCloud para locais de alto valor
high_price = data[data['price'] > data['price'].quantile(0.90)]
text = " ".join(high_price['nome'].dropna())
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Palavras mais comuns em nomes de locais de alto valor",
fontsize=16)
plt.show()
```



Desenvolvimento de modelos preditivos

Essa etapa consiste em desenvolver modelos preditivos capazes de estimar o preço de um aluguel temporário com base nas características dos imóveis a fim de responder a seguinte questão:

- Supondo que uma pessoa esteja pensando em investir em um apartamento para alugar na plataforma, onde seria mais indicada a compra?

Inicialmente é feito a preparação dos dados para a modelagem, seguido do treinamento de diferentes modelos de machine learning, avaliação dos modelos utilizando métricas de erro, comparação e seleção do melhor modelo e salvamento deste modelo para uso futuro.

Preparação dos Dados para modelagem

Antes de treinar os modelos, deve-se garantir que os dados estejam no formato correto para machine learning, seguindo os passos:

- Selecionar variáveis relevantes, explicativas (x) e alvo (y);
- Transformar variáveis categóricas (“bairro_group”, “room_type”) em numéricas;
- Dividir os dados em conjunto de treino (80%) e teste (20%) para avaliar o desempenho do modelo com dados nunca vistos.

```
# Seleção das variáveis explicativas e variáveis alvo
X = data[['bairro_group', 'room_type', 'latitude', 'longitude',
'minimo_noites',
          'numero_de_reviews', 'reviews_por_mes',
'calculado_host_listings_count',
          'disponibilidade_365']]
y = data['price']

# Transformação de variáveis categóricas (One-Hot Encoding)
```

```
X = pd.get_dummies(X, drop_first=True)

# Divisão dos dados em treino (80%) e teste (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Treinamento de diferentes modelos de Machine Learning

Após a preparação dos dados, foram testados três métodos diferentes para comparar qual apresenta o melhor desempenho:

- **Regressão Linear –** Modelo simples e com fácil interpretação, porém pode demonstrar inconsistência e erros na captura de relações não-lineares complexas que podem estar presente nos dados. Neste código, a regressão linear, tentará encontrar uma relação linear entre as variáveis preditoras e o preço do aluguel;

Código

```
# Regressão Linear
lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
y_pred_lin = lin_model.predict(X_test)
mae_lin = mean_absolute_error(y_test, y_pred_lin)
rmse_lin = mean_squared_error(y_test, y_pred_lin) ** 0.5
print(f"Regressão Linear - MAE: {mae_lin:.2f}, RMSE: {rmse_lin:.2f}")
```

- **Random Forest –** Modelo baseado em árvores de decisão, possui resistência a outliers, além de ser eficiente para detectar padrões contidos nos dados.

Código

```
# Random Forest
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
mae_rf = mean_absolute_error(y_test, y_pred_rf)
rmse_rf = mean_squared_error(y_test, y_pred_rf) ** 0.5
print(f"Random Forest - MAE: {mae_rf:.2f}, RMSE: {rmse_rf:.2f}")
```

- **Gradient Boosting –** Modelo mais avançado que ajusta iterativamente suas previsões, dessa forma, melhorando o desempenho.

Código

```
# Gradient Boosting
gb_model = GradientBoostingRegressor(random_state=42)
gb_model.fit(X_train, y_train)
```

```
y_pred_gb = gb_model.predict(X_test)
mae_gb = mean_absolute_error(y_test, y_pred_gb)
rmse_gb = mean_squared_error(y_test, y_pred_gb) ** 0.5
print(f"Gradient Boosting - MAE: {mae_gb:.2f}, RMSE: {rmse_gb:.2f}")
```

Avaliação dos modelos utilizando métricas de erro

Feito o treinamento dos modelos, compara-se os resultados das métricas MAE (Mean Absolute Error) e RMSE (Root Mean Squared Error), determinando-se assim, qual o melhor modelo a ser utilizado.

Modelo	MAE (Erro Médio Absoluto)	RMSE (Erro Quadrático Médio)
Regressão Linear	mae_lin	rmse_lin
Random Forest	mae_rf	rmse_rf
Gradient Boosting	mae_gb	rmse_gb

- MAE indica o erro médio absoluto nas previsões, medindo a média da diferença absoluta entre os valores previstos pelo modelo e os valores observados, dessa forma, quanto menor o valor, melhor;
- A raiz do erro quadrático médio RMSE, calcula a raiz quadrada da média dos quadrados dos erros, onde o erro bruto é a diferença entre o valor previsto pelo modelo e o valor real, dessa forma, penalizando grandes erros, então quanto menor, melhor.

Após os resultados dos modelos, visualizados abaixo, o modelo com menor RMSE, pois ele minimiza os grandes erros, que fora escolhido para prever preços de aluguel foi Random Forest, por seu melhor equilíbrio entre precisão e complexidade.

Métrica de Avaliação	Regressão Linear	Random Forest	Gradient Boosting
MAE	$57,01	$49,74	$51,21
RMSE	$96,57	$87,55	$90,14

Código

```
# Melhor modelo a ser utilizado, ou seja, menor RMSE
if rmse_rf < rmse_gb and rmse_rf < rmse_lin:
    best_model = rf_model
    model_name = 'Random Forest'
elif rmse_gb < rmse_rf and rmse_gb < rmse_lin:
    best_model = gb_model
    model_name = 'Gradient Boosting'
else:
    best_model = lin_model
    model_name = 'Regressão Linear'

joblib.dump(best_model, 'melhor_modelo_precificacao.pkl')
```

```
print(f"Modelo escolhido: {model_name}. Salvo como
'melhor_modelo_precificacao.pkl'")
```

Análise Estatística e Correlações

Por fim, faz-se valer dos insights finais para se obter dados relevantes para explorar a relação entre as variáveis do Dataset e identificar padrões que possam impactar a precificação dos imóveis.

Código - Matriz de Correlação

```
# Matriz de correlação
numeric_data = data.select_dtypes(include=[float, int])
correlation_matrix = numeric_data.corr()
print("Matriz de Correlação:")
print(correlation_matrix['price'].sort_values(ascending=False))
```

Matriz de Correlação	
price	1.000000
calculado_host_listings_count	0.130702
disponibilidade_365	0.117880
latitude	0.063031
host_id	0.032482
minimo_noites	0.023814
id	0.021388
reviews_por_mes	-0.055909
numero_de_reviews	-0.057882
longitude	-0.258819

1. **Matriz de Correlação –** A correlação mede o grau de relação entre duas variáveis, ou seja, quanto mais próximo do valor 1 positivo, mais forte a correlação positiva, dessa forma, quando uma variável aumenta, a outra também tende a aumentar, enquanto mais próximo do valor 1 negativo, maior a correlação negativa, assim, quando uma variável aumenta, a outra tende a diminuir, agora, valores que tendem a 0 costumam indicar que não há relação entre as variáveis. Dessa forma, pode-se analisar alguns insights relevantes, como:
    - **I.** A variável “calculado_host_listings_count” contém a maior correlação positiva com o preço (+0.13), indicando que anfitriões com maiores listagens tendem a cobrar mais;
    - **II.** A disponibilidade do imóvel ao longo do ano “disponibilidade_365” (+0.12) também tem uma leve correlação positiva com o preço;
    - **III.** A “longitude” (-0.26) tem a correlação negativa mais forte, sugerindo que imóveis mais a oeste da cidade tendem a ser mais baratos;

Código - Preços médios por bairro

```
# Preços médios por bairro
mean_prices = data.groupby('bairro_group')['price'].mean()
print(f"Preços médios por bairro: {mean_prices}")
```

Preços médios por bairro	
Manhattan	$178,93
Brooklyn	$117,82
Queens	$94,98
Staten Island	$98,58
Bronx	$85,28

2. **Preços médios por bairro –** Média de preços por região para identificação de quais bairros possuem os aluguéis mais caros, auxiliando investidores a entenderem onde o retorno financeiro pode ser maior.

Código - Correlações específicas

```
# Correlações específicas
corr_min_nights = data['minimo_noites'].corr(data['price'])
corr_availability = data['disponibilidade_365'].corr(data['price'])
print(f"Correlação entre mínimo de noites e preço: {corr_min_nights}")
print(f"Correlação entre disponibilidade e preço:
{corr_availability}")
```

3. **Correlações específicas –** Mínimo de Noites e Disponibilidade, onde:
    a. A correlação entre mínimo de noites e preço foi de 0.0238, ou seja, praticamente inexistente, concluindo que um não afeta o outro;
    b. A correlação entre disponibilidade do imóvel e preço foi de 0.1178, ou seja, fraca, concluindo que há um leve impacto, mas outros fatores, como localização e tipo de imóvel, são mais relevantes;

Código - Palavras comuns em locais de alto valo

```
# Palavras comuns em locais de alto valor
common_words = " ".join(high_price['nome'].dropna())
print("Palavras comuns em locais de alto valor:")
print(common_words)
```

4. **Palavras comuns em locais de alto valor –** palavras como "Luxury", "Penthouse", "Loft", "Spacious", "Elegant", "Chelsea", "Park", "Duplex", "Terrace", "Designer" aparecem com maior frequência em anúncios de imóveis mais caros;

Código - Previsão de preço para novos apartamentos

```
# Previsão de preço para novos apartamentos
novo_apartamento = pd.DataFrame([{
    'bairro_group': 'Manhattan',
    'room_type': 'Entire home/apt',
    'latitude': 40.75362,
    'longitude': -73.98377,
    'minimo_noites': 1,
    'numero_de_reviews': 45,
    'reviews_por_mes': 0.38,
    'calculado_host_listings_count': 2,
    'disponibilidade_365': 355
}])
```

5. **Previsão de preço para novos apartamentos –** Utilizando o modelo escolhido para prever o preço de um apartamento fictício baseado em características especificadas no documento “[Lighthouse] Desafio Ciência de Dados 2025-3”;

Código - Transformação de variáveis categóricas em dummies

```
# Transformação de variáveis categóricas em dummies
novo_apartamento_dummies = pd.get_dummies(novo_apartamento,
drop_first=True)
# Garantia de que todas as colunas necessárias estejam presentes
for col in X_train.columns:
    if col not in novo_apartamento_dummies.columns:
        novo_apartamento_dummies[col] = 0
```

6. **Transformação de variáveis categóricas em dummies –** Transformação das variáveis categóricas em One-Hot Encoding para evitar erros de incompatibilidade com o modelo treinado;

Código - Reorganização de colunas para corresponder ao treinamento

```
# Reorganização de colunas para corresponder ao treinamento
novo_apartamento_dummies = novo_apartamento_dummies[X_train.columns]
```

7. **Reorganização de colunas para corresponder ao treinamento –** Para que o novo DataFrame tenha exatamente a mesma estrutura dos dados usados no treinamento, garantindo que o modelo leia os dados corretamente.

Código - Previsão do Preço

```
# Previsão do Preço
price_pred = rf_model.predict(novo_apartamento_dummies)
print(f"Preço sugerido: ${price_pred[0]:.2f}")
```

8. **Previsão do Preço –** Por fim, encontra-se, utilizando o melhor modelo, Random Forest, a precificação conforme caracterização do imóvel desejado, que para as características estabelecidas foi de $274,25.

Conclusão

a) Supondo que uma pessoa esteja pensando em investir em um apartamento para alugar na plataforma, onde seria mais indicada a compra?
R: Manhattan, pois possui os preços médios mais altos $178.93, dentre as outras localidades, trazendo maior retorno. Porém, caso busque facilidade em encontrar investidores que buscam por preços mais acessíveis, há a alternativa de investir no Brooklyn, com preços médios em torno de $117.81, trazendo maior economia.

b) O número mínimo de noites e a disponibilidade ao longo do ano interferem no preço?
R: As correlações entre mínimo de noites e preço sendo de 0.0238 e entre disponibilidade do imóvel e preço sendo de 0.1178, destaca que ambas as variáveis têm impacto limitado no preço.

c) Existe algum padrão no texto do nome do local para lugares de mais alto valor?
R: Sim, pois palavras como "Luxury", "Penthouse", "Loft", "Spacious", "Elegant", "Chelsea", "Park", "Duplex", "Terrace", "Designer" aparecem com maior frequência em anúncios de imóveis mais caros, ou seja, usar descrições atrativas e destacadas no título pode aumentar o valor percebido.