Título do Trabalho:

"Análise das Listagens e Reservas do Airbnb em Nova York (2019): Tendências de Hospedagem e Impacto no Mercado Imobiliário".

ALEX JUNIOR MOURA DA SILVA	10415336
TIAGO UEDA	10274779
WAGNER DE MENDONÇA TRINDADE	10407917
DANILO BRITO DA SILVA	10415882
LARISSA SAYURI ITIMURA	10414911

Sumário

Contexto do Estudo: 4
Referências de Aquisição dos Dados: 4
Descrição da Origem dos Dados: 4
Descrição do Dataset: 5
Metadados 5
Cronograma do Processo Análitico 6
Bilbliotecas 7
Análise Exploratória dos Dados 7
Definição e descrição das bases teóricas dos métodos e acurácia 8
Scripts 9
Resultados 16
Conclusão 21

Contexto do Estudo:

- Este estudo busca compreender as dinâmicas do mercado imobiliário de Nova York, com foco nas listagens e reservas do Airbnb durante o ano de 2019. O Airbnb tem se tornado uma parte significativa do setor de hospedagem da cidade, e entender seu impacto é crucial para profissionais do ramo imobiliário, turismo e políticas urbanas.
- O objetivo deste estudo é analisar as tendências de hospedagem no Airbnb em Nova York, identificando padrões de preços, ocupação e avaliações dos hóspedes, além de explorar possíveis correlações com o mercado imobiliário local.

Referências de Aquisição dos Dados:

- Os dados foram obtidos a partir do conjunto de dados público do Airbnb para Nova York em 2019.
- Fonte Kaggle

Descrição da Origem dos Dados:

- Os dados foram coletados e disponibilizados pelo próprio Airbnb, uma plataforma de hospedagem online que permite que as pessoas listem, descubram e reservem acomodações em todo o mundo.
- O Airbnb fornece regularmente conjuntos de dados abertos para pesquisadores e profissionais interessados em realizar análises sobre suas operações e impacto.
- Esses conjuntos de dados são fornecidos de forma transparente e incluem informações sobre listagens de propriedades, reservas, avaliações de hóspedes e outras métricas relevantes para entender o funcionamento da plataforma.
- Para garantir a privacidade e a segurança dos usuários, o Airbnb anonimiza os dados antes de disponibilizá-los publicamente, removendo informações pessoais identificáveis.
- O acesso aos dados é geralmente concedido mediante concordância com os termos de uso estabelecidos pelo Airbnb e pode estar sujeito a restrições adicionais quanto ao seu uso e divulgação.

Descrição do Dataset:

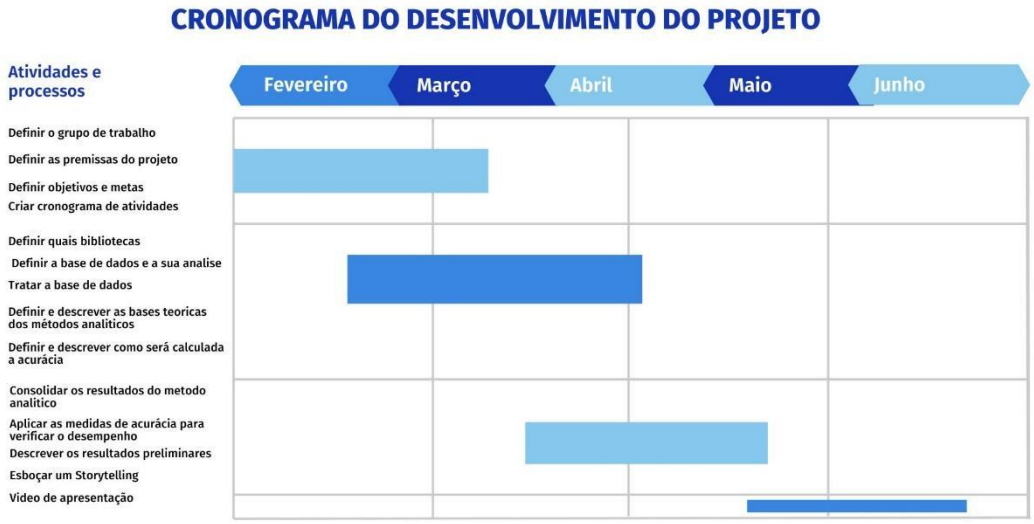
- O dataset contém informações detalhadas sobre as propriedades listadas no Airbnb em Nova York em 2019, incluindo características das propriedades, informações sobre os anfitriões, detalhes das reservas e avaliações dos hóspedes.
- O conjunto de dados é composto por múltiplas variáveis, como tipo de propriedade, número de quartos, preço de hospedagem, datas de check-in e check-out, avaliações dos hóspedes, entre outros.

Esses são alguns dos metadados comuns que podem ser encontrados no conjunto de dados "New York City Airbnb Open Data". Eles são essenciais para entender as características e atributos das listagens de propriedades de aluguel na cidade de Nova York.

Metadados

- ID da Listagem (listing ID): Identificador único atribuído a cada propriedade listada no Airbnb.
- Nome da Propriedade (name): Título ou nome da propriedade fornecido pelo proprietário ou anfitrião.
- ID do Anfitrião (host_id): Identificador único atribuído a cada anfitrião no Airbnb.
- Nome do Anfitrião (host_name): Nome do anfitrião responsável pela propriedade listada.
- Grupo do Bairro (neighbourhood_group): Localização geral da propriedade, agrupada por bairro ou distrito.
- Bairro (neighbourhood): Área específica ou bairro onde a propriedade está localizada.
- Latitude: Coordenadas de latitude da localização da propriedade.
- Longitude: Coordenadas de longitude da localização da propriedade.
- Tipo de Quarto (room_type): Tipo de espaço de hospedagem oferecido, como apartamento inteiro, quarto privado, etc.
- Preço (price): Preço por noite para alugar a propriedade, em dólares.
- Número Mínimo de Noites (minimum_nights): Quantidade mínima de noites que um hóspede deve reservar ao optar por ficar nesta propriedade.
- Número de Avaliações (number_of_reviews): Total de avaliações que a propriedade recebeu de usuários do Airbnb.
- Última Avaliação (last_review): Data da última avaliação recebida pela propriedade.
- Avaliações por Mês (reviews_per_month): Número médio de avaliações recebidas pela propriedade por mês.
- Número de Listagens do Anfitrião (calculated_host_listings_count): Quantidade total de listagens que um anfitrião possui.
- Disponibilidade Anual (availability_365): Número de dias em que a propriedade está disponível para reserva durante o ano.
- Esses são alguns dos metadados comuns que podem ser encontrados no conjunto de dados "New York City Airbnb Open Data". Eles são essenciais para entender as características e atributos das listagens de propriedades de aluguel na cidade de Nova York.

Cronograma do Processo Analítico



Bilbliotecas

Para análise de dados em Python, as bibliotecas fundamentais incluem Pandas para manipulação e análise de dados, Matplotlib para visualização e NumPy para operações numéricas eficientes.

- **Pandas**: Facilita a importação, limpeza e manipulação de conjuntos de dados.
- **Matplotlib**: Permite criar uma variedade de gráficos de alta qualidade para comunicar insights.

Juntas, essas bibliotecas fornecem uma base sólida para análise de dados em Python, permitindo explorar, analisar e visualizar dados de maneira eficaz e eficiente.

Análise Exploratória dos Dados

Na etapa inicial da minha análise de dados, comecei por examinar os tipos de variáveis presentes no conjunto de dados. Posteriormente, identifiquei a presença de valores nulos e implementei estratégias para lidar com eles. Para garantir a relevância dos dados para o objetivo da análise, removi as colunas que não contribuíam para o contexto em questão.

A fim de obter uma compreensão abrangente das acomodações em Nova York, empreguei técnicas de agrupamento para extrair informações tanto em níveis macro quanto micro. Este processo permitiu uma análise detalhada, revelando padrões e tendências importantes.

Além disso, utilizei gráficos visuais para representar os insights obtidos de forma mais intuitiva. Essas visualizações auxiliaram na identificação de padrões de comportamento e na comunicação eficaz dos resultados.

Por fim, realizei uma análise de correlação entre as variáveis numéricas. Esse procedimento revelou relações e dependências entre os diferentes atributos, proporcionando uma compreensão mais profunda da estrutura dos dados.

Essas etapas combinadas formam a base sólida da minha abordagem exploratória de dados, destacando tanto a amplitude quanto a profundidade da análise realizada.

Definição e descrição das bases teóricas dos métodos e acurácia

Modelo de regressão para prever os preços das acomodações com base nos atributos selecionados.

Treinamento do Modelo: Dividir os dados em conjuntos de treinamento e teste, treinar o modelo nos dados de treinamento e avaliar seu desempenho nos dados de teste.

Avaliação do Modelo: Avaliar o desempenho do modelo utilizando métricas apropriadas para problemas de regressão, como erro médio absoluto (MAE), erro médio quadrático (MSE) ou coeficiente de determinação (R^2).

Scripts

```
import pandas as pd

def tab_dist(dados):

    # Dados de exemplo

    # Determinar os limites dos intervalos de classe
    limite_inferior = min(dados) - 0.5
    limite_superior = max(dados) + 0.5
    largura_intervalo = 10
    intervalos = [i for i in range(int(limite_inferior),
int(limite_superior) + largura_intervalo, largura_intervalo)]

    # Classificar os dados nos intervalos de classe
    frequencias = pd.cut(dados, bins=intervalos,
right=False).value_counts().sort_index()

    # Criar a tabela de distribuição de frequência
    tabela_distribuicao = pd.DataFrame({'Intervalo de Classe':
frequencias.index,
                                        'Frequência':
frequencias.values})

    # Adicionar coluna de frequência relativa
    tabela_distribuicao['Frequência Relativa (%)'] =
(tabela_distribuicao['Frequência'] / len(dados)) * 100

    # Adicionar coluna de frequência acumulada
    tabela_distribuicao['Frequência Acumulada'] =
tabela_distribuicao['Frequência'].cumsum()

    return tabela_distribuicao
```

```
# %%
#importando bibliotecas
import pandas as pd
import matplotlib.pyplot as plt
import funcoes
import numpy as np


# %%
#importando dataset já com as colunas selecionadas
df = pd.read_csv(r"AB_NYC_2019.csv",
usecols=[0,2,4,5,6,7,8,9,10,11,14,15])

# %%
display(df.head())

# %%
#Alteração do tipo das variavéis de id
df.id = df.id.astype(str)
df.host_id = df.host_id.astype(str)
df.dtypes

# %%
#estatística principais das variavéis númerica
df.iloc[:,6:].describe()

# %%
#dataset com exclusão das hospedagem sem disponibilidade
df["is_open"] =  list(map(lambda x : x!=0, df["availability_365"]))
df_base = df.copy()
df = df.loc[df["is_open"] == True]
display(df.head())

# %%
#Acomodações sem disponibilidade de reservas
df_base["is_open"].value_counts()

# %%
#dataset agrupado por bairro
df_bairro =
df.groupby(["neighbourhood_group","neighbourhood","room_type"]).agg({"pri
ce":"mean","id":"count","number_of_reviews":"sum"}).reset_index()
df_bairro.rename(columns={"price":"price_average","id":"qtd_id"},
inplace=True)
df_bairro.sort_values(by=["price_average","room_type","neighbourhood"],
inplace=True)
display(df_bairro.head())

# %%
```

```
#dataset agrupado por distrito
df_distrito =
df.groupby(["neighbourhood_group","room_type"]).agg({"price":"mean","id":
"count","number_of_reviews":"sum"}).reset_index()
df_distrito.rename(columns={"price":"price_average","id":"qtd_id"},
inplace=True)
df_distrito["percentual"] = round((df_distrito["qtd_id"] /
df_distrito["qtd_id"].sum())*100 , 2)
display(df_distrito.head())

# %%
#Gráfico de preço médio por distrito e com divisão do tipo de instalagem
x = np.arange(5)
y1 = np.array(df_distrito.loc[df_distrito["room_type"] == "Entire
home/apt"]["price"])
y2 = np.array(df_distrito.loc[df_distrito["room_type"] == "Private
room"]["price"])
y3 = np.array(df_distrito.loc[df_distrito["room_type"] == "Shared
room"]["price"])
width = 0.2

fig = plt.figure(figsize = (5,3))
plt.bar(x-0.2, y1, width, color = "b")
plt.bar(x, y2, width, color = "r")
plt.bar(x+0.2, y3, width, color = "y")
plt.xticks(x, df_distrito["neighbourhood_group"].unique())
plt.legend(["Entire home/apt","Private room","Shared room"])
plt.title("Média de preço por distrito")
plt.xlabel("Distritos")
plt.ylabel("Opções de Locações")

# %%
#Gráfico de disponibilidade por distrito com divisão do tipo de
instalagem
x = df_distrito["neighbourhood_group"].unique()
y1 = np.array(df_distrito.loc[df_distrito["room_type"] == "Entire
home/apt"]["percentual"])
y2 = np.array(df_distrito.loc[df_distrito["room_type"] == "Private
room"]["percentual"])
y3 = np.array(df_distrito.loc[df_distrito["room_type"] == "Shared
room"]["percentual"])

fig = plt.figure(figsize = (5,3))
plt.bar(x, y1, color = "b")
plt.bar(x, y2, bottom = y1, color = "r")
plt.bar(x, y3, bottom = y1+y2, color = "y")
plt.title("Distribuição por distrito")
plt.xlabel("Distritos")
plt.ylabel("Opções de Locações")
```

```
plt.legend(["Entire home/apt","Private room","Shared room"])
plt.show()

# %%
#Tabela de distribuição de frequência com classe de intervalos para a
quantidade de opções por host
df_host =
df[["host_id","calculated_host_listings_count"]].drop_duplicates()
tab_dist = funcoes.tab_dist(df_host["calculated_host_listings_count"])
tab_dist.loc[tab_dist["Frequência"] != 0].iloc[:,:-1]

# %%
#Faturamento do host por carteira de acomodações
df_host = df.groupby(["host_id",
"neighbourhood_group"]).agg({"id":"count","price":"sum"}).sort_values(by=
["price"], ascending=False).reset_index()
df_host.rename(columns={"id":"nro_acomodações","price":"faturamento"},
inplace=True)
df_host

# %%
#Quantidade de host por distrito
df_id =
df_host.groupby(["neighbourhood_group"]).agg({"host_id":"count"}).sort_va
lues(by=["host_id"], ascending=False).reset_index()
df_id.rename(columns={"host_id":"total_host"})
df_id

# %%
#Gráfico para quantidade de host por distrito
fig = plt.figure(figsize = (5,3))
plt.bar(df_id["neighbourhood_group"], df_id["host_id"])
plt.title("Host por distrito")
plt.xlabel("Distritos")
plt.ylabel("Quantidade de Locações")
plt.show()

# %%
#Média das estadias por distrito e tipo de hospedagem
df_estadia = df.groupby(["neighbourhood_group",
"room_type"]).agg({"minimum_nights":"mean",
"price":"mean"}).reset_index()
df_estadia.rename(columns={"price":"price_mean","minimum_nights":"minimum
_nights_mean"}, inplace=True)
df_estadia
```

```
# %%
#Média das estadias por distrito e tipo de hospedagem
df_estadia = df.groupby(["availability_365",
"neighbourhood_group","room_type"]).agg({"minimum_nights":"mean",
"price":"mean"}).reset_index()
df_estadia

# %%
#Tabela de distribuição de frequência com classe de intervalos para a
quantidade de opções por host
df_estadia = df.groupby(["availability_365",
"neighbourhood_group","room_type"]).agg({"minimum_nights":"mean",
"price":"mean"}).reset_index()
tab_dist = funcoes.tab_dist(df_estadia["availability_365"])
tab_dist = tab_dist.loc[tab_dist["Frequência"] != 0].iloc[:,:-1]


# %%
# Gráfico de histograma da distribuição de dias disponivéis
fig = plt.figure(figsize = (5,3))
plt.hist(df_estadia["availability_365"], bins = 10)
plt.title("Histograma de dias disponíveis")
plt.show()

# %%
#Correlações para variaveis númericas
df_cor = df.astype({"id":"str","host_id":str})
df_cor = df_cor.select_dtypes(exclude = ["object","bool"])
df_cor.corr().style.background_gradient(cmap = "Blues")

# %%
# Configurações do gráfico
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

# Adicionar título ao canvas
fig.suptitle('Tipos de Acomodações em Nova York', fontsize=16)

# Plotar os pontos no gráfico - Entire home/apt
longitudes_entire = df["longitude"].loc[df["room_type"] == "Entire
home/apt"]
latitudes_entire = df["latitude"].loc[df["room_type"] == "Entire
home/apt"]
axs[0].scatter(longitudes_entire, latitudes_entire, color='g', marker =
".")
axs[0].set_title('Entire home/apt')
axs[0].set_xlabel('Longitude')
axs[0].set_ylabel('Latitude')
axs[0].grid(True)
```

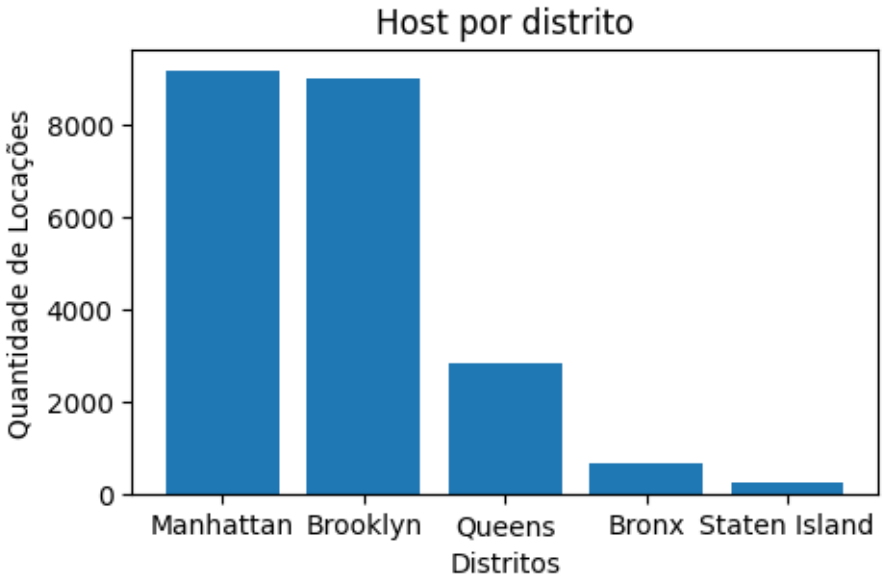
```
# Plotar os pontos no gráfico - Private
longitudes_private = df["longitude"].loc[df["room_type"] == "Private
room"]
latitudes_private = df["latitude"].loc[df["room_type"] == "Private room"]
axs[1].scatter(longitudes_private, latitudes_private, color='b', marker =
".")
axs[1].set_title('Private')
axs[1].set_xlabel('Longitude')
axs[1].set_ylabel('Latitude')
axs[1].grid(True)

# Plotar os pontos no gráfico - Shared room
longitudes_shared = df["longitude"].loc[df["room_type"] == "Shared room"]
latitudes_shared = df["latitude"].loc[df["room_type"] == "Shared room"]
axs[2].scatter(longitudes_shared, latitudes_shared, color='r', marker =
".")
axs[2].set_title('Shared room')
axs[2].set_xlabel('Longitude')
axs[2].set_ylabel('Latitude')
axs[2].grid(True)

# Ajustar layout
plt.tight_layout()

# Exibir o gráfico
plt.show()
# Regressão linear para prever
# Supondo que 'dados' é o nome do seu dataframe
# Defina as variáveis independentes (X) e a variável dependente (y)
X = df_sub_distrito[['neighbourhood', 'room_type',
'minimum_nights_mean']]
y = df_sub_distrito['price_mean']

# Codificar variáveis categóricas, se necessário
X = pd.get_dummies(X)

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Inicializar e ajustar o modelo de regressão linear
modelo = LinearRegression()
modelo.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
previsoes = modelo.predict(X_test)

# Avaliar o desempenho do modelo usando o erro quadrático médio (MSE)
mse = mean_squared_error(y_test, previsoes)
```

```
print("Erro Quadrático Médio (MSE):", mse)
```

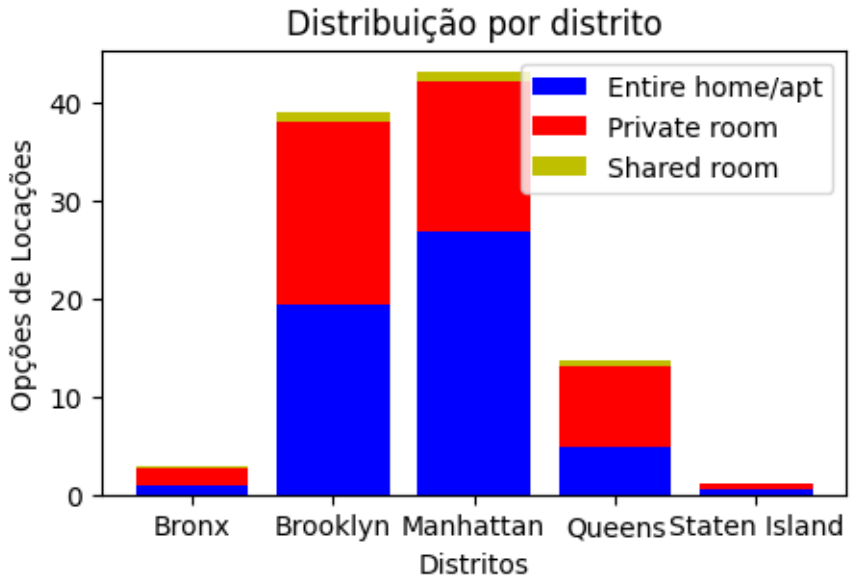
Resultados

A análise exploratória dos dados revelou insights significativos sobre o conjunto de dados em questão. Inicialmente, identificamos que o dataset original consistia em 48.895 registros. No entanto, para aprimorar nossa compreensão dos dados, optamos por remover as 17.533 unidades sem disponibilidade de locação, resultando em um conjunto mais conciso e relevante para nossas análises subsequentes.

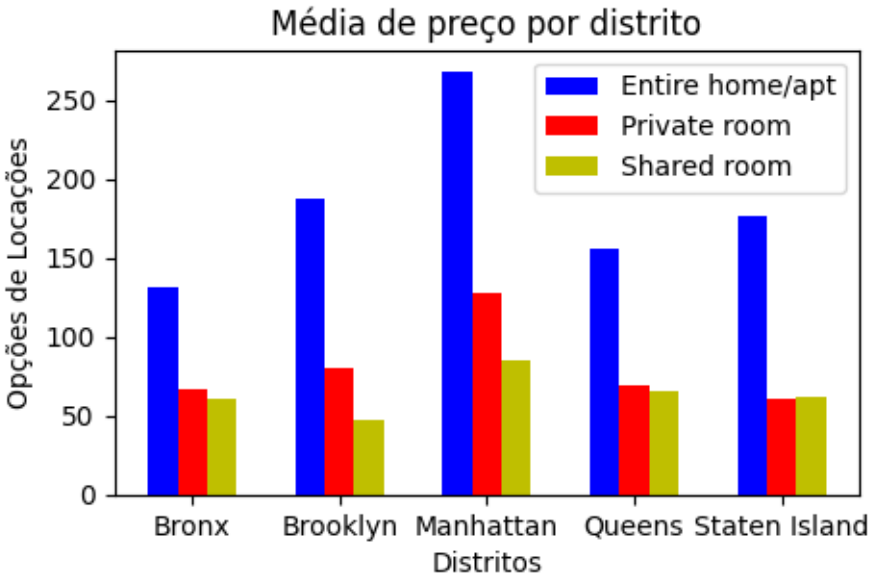
Em seguida, procedemos com o agrupamento dos dados por distrito e tipo de locação, uma estratégia que facilitou a obtenção de valores médios específicos para cada categoria de locação.



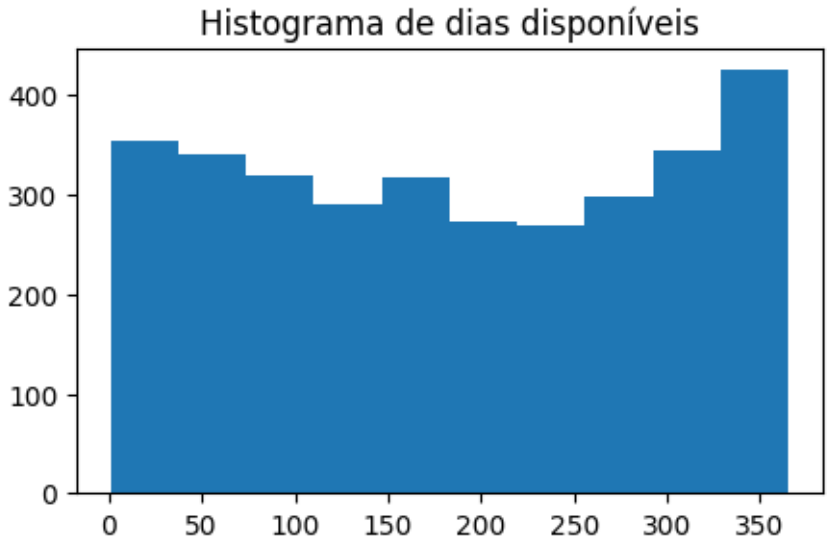
Ao explorar a distribuição do tipo de locação em cada distrito, destacou-se o distrito de Brooklyn como um caso particular. Aqui, encontramos uma distribuição equilibrada entre locações completas e apenas quartos disponíveis, sugerindo uma dinâmica de mercado única nessa região específica.



Outro aspecto examinado foi a frequência de propriedades por locatário, analisando sua distribuição geográfica por distrito. Essa análise pode fornecer insights valiosos sobre padrões de investimento imobiliário e a densidade de locadores em áreas específicas da cidade.



A seguir, exploramos o histograma de frequência dos dias disponíveis para locação, revelando padrões de. Essas informações são cruciais para estratégias de precificação e gerenciamento de estoque no setor de hospedagem.



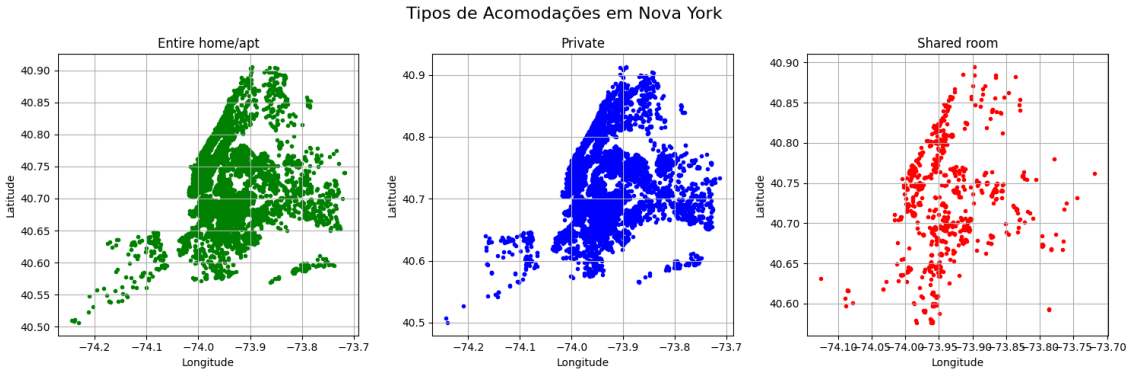
Em Manhattan, o potencial de faturamento máximo por anfitrião é excepcionalmente alto devido à demanda constante por hospedagem na região central da cidade, proporcionando oportunidades lucrativas com preços médios elevados por noite.

	host_id	neighbourhood_group	nro_acomodações	faturamento
0	219517861	Manhattan	327	82795
1	107434423	Manhattan	230	69741
2	205031545	Manhattan	49	35294
3	30283594	Manhattan	121	33581
4	156158778	Manhattan	6	26071
...	...	...	...	...
21876	197169969	Queens	1	10
21877	205820814	Bronx	1	10
21878	97001292	Queens	1	10
21879	167570251	Brooklyn	1	10
21880	10132166	Brooklyn	1	0

Após examinar as correlações entre as variáveis numéricas, embora não tenhamos identificado relações significativas, esse processo nos forneceu insights sobre a natureza dos dados e as possíveis interações entre as variáveis.

	latitude	longitude	price	minimum_nights	number_of_reviews	calculated_host_listings_count	availability_365
latitude	1.000000	0.078610	0.033803	0.035607	-0.013481	0.025341	-0.003714
longitude	0.078610	1.000000	-0.158940	-0.081527	0.033989	-0.146085	0.027964
price	0.033803	-0.158940	1.000000	0.039449	-0.072919	0.060828	0.074509
minimum_nights	0.035607	-0.081527	0.039449	1.000000	-0.116086	0.124207	0.125418
number_of_reviews	-0.013481	0.033989	-0.072919	-0.116086	1.000000	-0.115415	0.009881
calculated_host_listings_count	0.025341	-0.146085	0.060828	0.124207	-0.115415	1.000000	0.187949
availability_365	-0.003714	0.027964	0.074509	0.125418	0.009881	0.187949	1.000000

Por fim, ao plotar o tipo de locação no mapa da cidade de Nova York, pudemos visualizar a disponibilidade de hospedagem em toda a região metropolitana, destacando áreas de alta concentração e demanda.



Em resumo, essa análise exploratória nos proporcionou uma compreensão mais profunda do mercado de locação em Nova York, fornecendo insights valiosos para tomadas de decisão estratégicas no setor de hospedagem e turismo.

	neighbourhood_group	room_type	minimum_nights_mean	price_mean
0	Bronx	Entire home/apt	6.295597	131.682390
1	Bronx	Private room	4.117216	66.699634
2	Bronx	Shared room	3.800000	61.200000
3	Brooklyn	Entire home/apt	7.364460	187.711133
4	Brooklyn	Private room	6.468027	80.701190
5	Brooklyn	Shared room	8.525974	46.964286
6	Manhattan	Entire home/apt	13.575643	268.215614
7	Manhattan	Private room	6.345253	127.971560
8	Manhattan	Shared room	7.857143	84.517857
9	Queens	Entire home/apt	5.769032	155.343871
10	Queens	Private room	5.023202	69.043697
11	Queens	Shared room	3.660494	65.419753
12	Staten Island	Entire home/apt	6.078947	176.776316
13	Staten Island	Private room	3.865497	61.070175
14	Staten Island	Shared room	2.125000	62.125000

Conclusão

A análise utilizada foi de Regressão Linear. O modelo deve inferir valores de diárias para novas empreendimento com base no bairro e tipo de alocação. No entanto, após termos separado a base, treinamento e teste, e feito todo o procedimento adequado, deparou-se com um MSE, muito alto. Tornando inviável o a utilização do modelo.

A partir disso, será avaliado o motivo, a seguir possíveis motivos listados.

- **Modelo inadequado**: O modelo de regressão linear pode não ser apropriado para os dados em questão. Pode ser necessário explorar modelos mais complexos ou técnicas de modelagem diferentes.
- **Dados de entrada inadequados**: Os preditores podem não estar capturando adequadamente a relação com a variável de resposta. Talvez seja necessário considerar outros preditores ou transformações nos dados.
- **Problemas de escala**: Se os preditores estiverem em diferentes escalas, isso pode afetar a performance do modelo. Normalmente, é uma boa prática padronizar ou normalizar os dados antes de ajustar um modelo de regressão.
- **Presença de outliers**: Outliers nos dados podem distorcer o ajuste do modelo e aumentar o MSE. É importante examinar os dados em busca de valores atípicos e considerar maneiras de lidar com eles, como remoção ou transformação.

Overfitting ou underfitting: O modelo pode estar sofrendo de overfitting (ajuste excessivo) ou underfitting (ajuste insuficiente). Overfitting ocorre quando o modelo se ajusta muito bem aos dados de treinamento, mas não generaliza bem para novos dados. Underfitting ocorre quando o modelo é muito simples para capturar a relação nos dados.

Diante dos valores apresentados não foi possível inferir e comprovar nossas hipóteses a respeitos do dataset em questão. Pois destaca-se a ausência de informações pertinentes como: histórico de alocações, avaliação dos locais e descrição dos locais.