**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](#_Toc166937473)

[Referências de Aquisição dos Dados: 4](#_Toc166937474)

[Descrição da Origem dos Dados: 4](#_Toc166937475)

[Descrição do Dataset: 5](#_Toc166937476)

[Metadados 5](#_Toc166937477)

[Cronograma do Processo Análitico 6](#_Toc166937478)

[Bilbliotecas 7](#_Toc166937479)

[Análise Exploratória dos Dados 7](#_Toc166937480)

[Definição e descrição das bases teóricas dos métodos e acurácia 8](#_Toc166937481)

[Scripts 9](#_Toc166937482)

[Resultados 16](#_Toc166937483)

[Conclusão 21](#_Toc166937484)

# 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](https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data?resource=download)

# 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

**Gráfico, Linha do tempo

Descrição gerada automaticamente**

# 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 (R²).

# 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({"price":"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\_values(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.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

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.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

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.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

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.

Gráfico, Histograma

Descrição gerada automaticamente

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.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

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.

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

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.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

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.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# 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.