

Project Name -AirBnb Bookings Analysis

Project Type - EDA(Exploratory data analysis)

Project Summary -

The Airbnb NYC 2019 dataset contains detailed information on Airbnb listings across New York City, including attributes like location, room type, price, and availability. The dataset includes around 49,000 instances, with 16 different attributes.

The dataset includes the following key columns:

id: Unique listing id.

name: Name of the property.

host_id: Unique identifier for each listed host.

host_name: Name of the host.

neighbourhood_group: Location

neighbourhood: Area

latitude: Latitude coordinates

longitude: Longitude coordinates

room_type: Type of room being rented (e.g., Entire home/apt, Private room).

price: Price per night for renting the property.

minimum_nights: Minimum number of nights required for a booking or stay

number_of_reviews: Number of reviews written for the listing

last_review: Date of the most recent review.

reviews_per_month: Average number of reviews per month.

calculated_host_listings_count: Total no of listings against the host id

availability_365: Number of days when listing is available for booking

The goal of this project is to perform Exploratory Data Analysis (EDA) on the Airbnb NYC 2019 dataset. EDA involves examining the dataset to find useful insights and patterns. By analyzing this data, we aim to discover how different factors affect property prices, understand which neighborhoods are more popular, and identify trends that can be useful for Airbnb hosts and potential guests.

Data Analysis: Price Distribution: We examined how prices are distributed across different listings. This analysis showed that most properties are priced between 75 dollars and 500 dollars per night. There are fewer properties with very high or very low prices.

Room Type Analysis: We explored the different types of rooms available. We found that entire apartments or homes are the most common, followed by private rooms. Shared rooms are less common.

Neighborhood Insights: By analyzing the locations of the properties, we discovered which neighborhoods are most popular and which have the highest prices. Manhattan generally has higher prices compared to other boroughs, while some neighborhoods in Brooklyn and Queens offer more affordable options.

Correlation Analysis: We looked at how different factors like price, number of reviews, and availability relate to each other. For example, properties with higher prices tend to have fewer reviews, possibly indicating that high-end properties are booked less frequently but at a higher rate.

The exploratory data analysis of the Airbnb NYC 2019 dataset provided valuable insights into the rental market in New York City. We learned about the distribution of prices, the popularity of different room types, and the variation in property prices across neighborhoods. These insights can help Airbnb hosts price their properties more effectively and guide guests in choosing accommodations based on their budget and preferences. Overall, this analysis helps in understanding the dynamics of the Airbnb market in NYC and can inform future decisions for both hosts and guests.

GitHub Link -

Provide your GitHub Link here.

Problem Statement

Since 2008, Airbnb has revolutionized travel by offering personalized experiences, with data analysis of millions of listings crucial for business decisions and platform insights. The dataset, containing around 49,000 observations, helps analyze host behavior, customer preferences, and market trends to guide strategic actions.

Define Your Business Objective?

The objective of analyzing the Airbnb NYC 2019 dataset through EDA is to uncover insights that optimize listings, inform investment decisions, and guide city planning. This data-driven approach aims to enhance profitability, improve occupancy rates, and support sustainable urban development.

Let's Begin !

1. Know Your Data

Import Libraries

```
# Import Libraries
import numpy as np
import pandas as pd
from numpy import math
from scipy.stats import *
import seaborn as sns
import matplotlib.pyplot as plt
import sys
```

Dataset Loading

```
# Load Dataset
df = pd.read_csv('/content/Airbnb NYC 2019.csv')
```

Dataset First View

```
# Dataset First Look
df.head()

df.tail()
```

Dataset's list of column names

```
print(df.columns)
```

Dataset Rows & Columns count

```
# Dataset Rows & Columns count
print(df.shape)
```

Size of Dataset

```
print(sys.getsizeof(df))
```

Dataset Information

```
# Dataset Info
print(df.info())
```

Data types of the columns of Dataset

```
print(df.dtypes)
```

Duplicate Values

```
print(df.duplicated().sum())
```

Missing Values/Null Values

```
# Missing Values/Null Values Count
print(df.isnull().sum())

# Visualizing the missing values
print("Visualization of the missing values : \n")
plt.figure(figsize=(10,6))
df.isnull().sum().plot(kind='bar')
plt.title('\nMissing Values\n')
plt.xlabel('\nVariables\n')
plt.ylabel('\nCount\n')
plt.plot()
plt.show()
```

What did you know about your dataset?

The Airbnb dataset contains detailed information on Airbnb listings across New York City, including attributes like location, room type, price, and availability. The dataset includes around 49,000 instances, with 16 different attributes.

2. Understanding Your Variables

```
# Dataset Columns
print(df.columns)
```

Dataset Description

```
# Dataset Describe
print(df.describe(include='all'))
```

Variables Description

The variable description for the Airbnb dataset :-

id: Unique identifier for each listing in the dataset.

name: Title or name of the listing as provided by the host. Text field that often includes a brief description or key features of the listing.

neighbourhood_group: Broad geographic region where the listing is located. Categories include Manhattan, Brooklyn, Queens, The Bronx, and Staten Island.

neighbourhood: Specific neighborhood within the region. Examples include Williamsburg, Harlem, and SoHo.

room_type: Type of room being offered. Categories include Entire home/apt, Private room, and Shared room.

price: Price per night for the listing, in USD. Ranges from 10 to 10000.

minimum_nights: Minimum number of nights required for booking the listing. Ranges from 1 to 1250.

number_of_reviews: Total number of reviews the listing has received. Ranges from 0 to 1000.

last_review: Date when the last review was posted. Format is YYYY-MM-DD.

reviews_per_month: Average number of reviews per month for the listing. Ranges from 0 to 100.

calculated_host_listings_count: Number of listings that the host has in total. This value represents how many different properties a host manages, indicating their level of involvement on the platform. Numeric value.

host_id: Unique identifier for the host of the listing. Numeric value.

host_name: Name of the host. Text field.

latitude: Latitude coordinate of the listing's location. Ranges from 40.5 to 45.0.

longitude: Longitude coordinate of the listing's location. Ranges from -74.2 to -73.7.

availability_365: Number of days the listing is available for booking in a year. Ranges from 0 to 365.

Check Unique Values for each variable.

```
# Check Unique Values for each variable.  
df.nunique()
```

3. *Data Wrangling*

Filling Missing Values

```
df['reviews_per_month'].fillna(df['reviews_per_month'].mean(), inplace=  
True)
```

Reviews per month column had a lot of missing rows but is important for analysis, hence missing values will be replaced with the mean of that column

Capitalize First Letter of Each Column Name

```
df.columns = [col.capitalize() for col in df.columns]
```

Dropping columns with a lot of missing values

```
df.drop(columns=['Last_review'], axis=1, inplace=True)
```

Revenue

```
# Calculating 'Revenue' as Price * Minimum_nights
df['Revenue'] = df['Price'] * df['Minimum_nights']
#sum the revenue
total_revenue=df['Revenue'].sum()
# Display the updated DataFrame
print("Updated DataFrame with 'Revenue' column:")
print(df.info())
print (total_revenue)
```

last_review have many missing values and they are not of any importance for analysis,hence they will be deleted

Filling empty rows with Unknown

```
df.fillna('Unknown', inplace=True)
```

Checking the total no. of null values in the dataset

```
df.isnull().sum()
```

Dataset Rows & Columns count after data wrangling

```
print(df.shape)
df.head()
df.tail()
```

What all manipulations have you done and insights you found?

Missing data in the reviews_per_month column was significant but crucial for analysis, so it was replaced with the mean value to maintain the dataset's completeness.

By capitalizing the first letter of each column name, the dataset is more consistent and easier to work with.

Dropping columns with too many missing values helps improve the quality of the data, focusing on the most relevant features for analysis.

4. Data Vizualization, Storytelling & Experimenting with charts : Understand the relationships between variables

1.Most common room types

```
# Plotting the Pie chart for showing the proportion of each room type in the whole area .
print("Pie Chart of the Most common room types : \n\n")
room_type = df.Room_type.value_counts()
room_type.plot(kind='pie', figsize=(8,8), title='Most frequent room types', fontsize=10, explode=(0,0,0), startangle=90, colors=['green','orange','red'], autopct='%1.1f%%')
plt.title('\nPie chart of room type\n', fontweight='bold', pad=10)
plt.legend()
plt.ylabel(" ")
plt.show()
```

This pie chart shows the proportion percentage of different room types in the dataset. This makes it easy to compare and see which room type is most common at a glance.

percentage of "Entire home/apt" room type = 52.0 %

percentage of "Private room" room type = 45.7 %

percentage of "Shared room" room type = 2.4 %

52.0 % customers demand "Entire home" and 45.7 % customers demand the "Private room" and there are very less customers (2.4 %) who have no problem of having "Shared room". This insight information can help hosts and investors focus on what customers prefer, leading to better business decisions.

2.The most common neighbourhood_group

```
# Plotting the Pie Chart for showing the proportion of the different neighbourhood groups in the whole area.
print("Pie Chart of the neighbourhood group : \n\n")
neighbourhood_Group = df.Neighbourhood_group.value_counts()
neighbourhood_Group.plot(kind='pie', figsize=(8,8), title='Most frequent neighbourhood_group types', fontsize=10, explode=(0,0,0.1,0.1,0), startangle=90, colors=['green','orange','red'], autopct='%1.1f%%')
plt.title('\nPie chart of neighbourhood_group types\n', fontweight='bold', pad=10)
plt.legend()
plt.ylabel(" ")
plt.show()
```

This pie chart visually represents the share percentage of each neighborhood group in the dataset, making it simple to understand which areas are most popular for Airbnb listings.

This pie chart shows that manahatten (44.3 %) and Brooklyn (41.1 %) are the prime locations for customers (Tourists) and some customers also prefer Queens (11.6 %) , but there are very fewer customers who prefer Bronx and Staten Island as their prime location.

These insights guide investors and hosts to focus on high-demand areas, ensuring better returns. This visualization highlights the growth opportunities in popular neighborhoods, while less popular areas might require different strategies.

3.Top 10 Host IDs

```
# Assuming `df` is already defined and contains the data
top10_hi = df.Host_id.value_counts().head(10)
print("Top 10 Host IDs: ", top10_hi)
print("\n\nThe Horizontal Bar Chart between host IDs and the number of
Airbnbs under each of the hosts:\n\n")

# Plot the horizontal bar chart using Matplotlib's default color cycle
top10_hi.plot(kind='barh', color=plt.cm.tab10.colors)

# Add title and labels
plt.title('\nTop 10 Host IDs\n')
plt.xlabel('\nNumber of Listings\n')
plt.ylabel('\nHost ID\n')

# Display the chart
plt.show()
```

This horizontal bar chart clearly shows the top 10 hosts with the most listings, which makes it easy to compare their relative dominance in the market.

This chart shows that a few hosts control a significant number of listings, indicating a concentrated market where certain hosts have a strong presence and potential influence over pricing and availability.

These insights can help in targeting collaborations with top hosts for strategic partnerships or marketing efforts. If these key hosts decide to leave the platform, the market concentration could indicate a potential risk which might reduce listing availability in popular areas.

4.Distribution of Airbnb Listings based on Neighbourhood group

```
# Plotting the Count Plot of distribution of Airbnb Listings based on
Neighbourhood group
print("Count Plot of distribution of Airbnb Listings based on
Neighbourhood group : \n\n")
plt.figure(figsize=(10,5))
sns.countplot(x='Neighbourhood_group',
data=df,palette="Set1",width=0.3)
plt.title('\nDistribution of Airbnb Listings based on Neighbourhood
group\n')
plt.xlabel('\nneighbourhood_group\n')
plt.ylabel('\nCount\n')
plt.show()
```

This count plot shows the distribution of Airbnb listings across different neighborhood groups, which makes it easy to compare how many listings each area has.

This chart presents that the certain neighborhood groups, like Manhattan and Brooklyn, have a much higher concentration of Airbnb listings, which indicates that these are the most popular areas for hosts and guests.

These insights can guide investment decisions by highlighting areas with high demand, ensuring better returns.

Shortcoming : Oversaturation in popular areas could lead to increased competition, potentially lowering prices and profitability for hosts. Diversifying into less saturated areas might balance this risk.

5. Distribution of Airbnb Listings based on room type

```
# Plotting count plot for showing the distribution of Airbnb Listings based on room type
print("Count Plot of distribution of Airbnb Listings based on room type : \n\n")
plt.figure(figsize=(6,4))
sns.countplot(x='Room_type', data=df,width=0.4,palette="Set1")
plt.title('\nDistribution of Airbnb Listings based on room type\n')
plt.xlabel('\nroom_type\n')
plt.ylabel('\nCount\n')
plt.show()
```

This count plot visually represents the distribution of different room types across Airbnb listings, which makes it easy to see which types are most and least common.

The chart shows that "Entire home/apt" is the most common room type, which indicates a strong demand for whole apartments or homes.

This chart suggests tourists prefer privacy and space (Private room), while other room types like "Shared room" are less frequent.

Understanding that "Entire home/apt" is preferred can guide hosts and investors to focus on offering whole apartments to meet demand.

Shortcoming : An oversupply of popular room types could lead to increased competition and potentially lower prices if not managed properly.

6. TOP 10 Host names

```
# Assuming `df` is already defined and contains the data
top10_hosts = df.Host_name.value_counts().head(10)

# Print the top 10 host names
print("Top 10 Host Names:\n", top10_hosts)

# Plot the horizontal bar chart
print("\n\nThe Horizontal Bar Chart between host names and the number of Airbnbs under each host:\n\n")
plt.figure(figsize=(8, 6)) # Adjust the figure size to reduce space
```

```
sns.barplot(x=top10_hosts.values, y=top10_hosts.index,
palette="viridis")

# Set title and labels with reduced margins
plt.title('\nTop 10 Hosts by Number of Listings\n', fontsize=14)
plt.xlabel('\nNumber of Listings\n', fontsize=12)
plt.ylabel('\nHost Name\n', fontsize=12)

# Adjust layout and invert the y-axis to display the highest value at the top
plt.tight_layout() # This helps in reducing the space and improving layout
plt.gca().invert_yaxis()
plt.show()
```

Michael leads the Airbnb market with the highest number of listings, managing over 400 properties. David follows closely as the second most active host, holding a slightly smaller number of listings than Michael. In third place is Sonder (NYC), a professional hosting organization, emphasizing its significant role as a large-scale operator on the platform.

Other hosts, like John, Alex, and Blueground, manage substantial numbers of properties, suggesting they may be property management companies or professional operators rather than individual hosts. Jessica and Maria, while ranked at the lower end of the top 10, still manage around 150-200 listings each, contributing notably to the overall distribution.

The top hosts dominate a large portion of the market, revealing that a small number of hosts control a significant share of listings. This concentration underscores the strong presence of professional operators, such as property management companies, in Airbnb's ecosystem, alongside individual hosts. The landscape is competitive, with both large-scale operators and individual hosts coexisting, albeit with varying levels of influence.

7.TOP 10 HOST NAMES BY REVENUE

```
# Calculate total revenue per host
Host_revenue = df.groupby('Host_name')['Revenue'].sum().reset_index()

# Display the top 10 hosts by revenue
print(Host_revenue.nlargest(10, 'Revenue'))

# Create a bar chart for the top 10 hosts by revenue
plt.figure(figsize=(10, 6))
sns.barplot(y='Host_name', x='Revenue', data=Host_revenue.nlargest(10, 'Revenue'), palette="Spectral") # Using 'Spectral' for a vibrant look
plt.title('Top 10 Host Names by Revenue')
plt.xlabel('Total Revenue')
plt.ylabel('Host Name')
plt.show()
```

The chart reveals which neighbourhood groups generate the highest and lowest average revenue, identifying areas with significant earnings and those with potential for growth.

8.Top 10 Neighbourhood

```
# Top 10 neighborhoods
top10_neighborhoods = df.Neighbourhood.value_counts().head(10)

# Plot the horizontal bar chart using Matplotlib's default color cycle
top10_neighborhoods.plot(kind='barh', color=plt.cm.tab10.colors)

# Add title and labels
plt.title('\nTop 10 Neighborhoods\n', fontsize=14)
plt.xlabel('\nNumber of Listings\n', fontsize=12)
plt.ylabel('\nNeighborhood\n', fontsize=12)

# Display the chart
plt.show()
```

The horizontal bar chart displays the top 10 neighborhoods ranked by the number of listings.

The y-axis represents neighborhoods, while the x-axis shows the number of listings.

Williamsburg has the highest number of listings, close to 4,000.

Bedford-Stuyvesant follows as the second highest with approximately 3,500 listings.

Harlem and Bushwick have moderate numbers, each with over 2,000 listings.

Upper West Side, Hell's Kitchen, East Village, and Upper East Side have similar listing counts, ranging between 1,000 and 1,500.

Crown Heights and Midtown have the fewest listings among the top 10, each under 1,000.

The chart highlights listing distribution across key neighborhoods.

Average revenue by neighbourhood group

```
# Calculate the average revenue by neighborhood group
neighbourhood_group_revenue = df.groupby('Neighbourhood_group')
['Revenue'].mean().reset_index()

# Create a bar chart using Seaborn's default color palette
plt.figure(figsize=(10, 6))
sns.barplot(
    x='Neighbourhood_group',
    y='Revenue',
    data=neighbourhood_group_revenue,
    palette='tab10' # Use the default `tab10` color palette
)

# Add title and labels
plt.title('Average Revenue by Neighbourhood Group', fontsize=16)
plt.xlabel('Neighbourhood Group', fontsize=12)
plt.ylabel('Average Revenue', fontsize=12)
```

```
# Display the chart
plt.show()
```

The chart reveals which neighbourhood groups generate the highest and lowest average revenue, identifying areas with significant earnings and those with potential for growth.

9. Category count plot between room type and neighbourhood group

```
# Plotting the Category count plot between room type and neighbourhood group
print("Category count plot between room type and neighbourhood group : \n\n")
sns.catplot(x='Room_type', kind='count', hue='Neighbourhood_group', data=df)
plt.title('\nCategory count plot between room type and neighbourhood group\n')
plt.xlabel('\nroom_type\n')
plt.ylabel('\nCount\n')
plt.show()
```

This catplot compares the distribution of room types across different neighborhood groups, providing insights into which room types are popular in each area.

The chart shows variations in room type preferences by neighborhood. For instance, "Entire home/apt" and "Private room" are more common in central areas like "Manhattan" and "Brooklyn", while other room types could be more prevalent in suburban neighborhoods.

These insights can help tailor listings to meet specific neighborhood preferences by optimizing the appeal and profitability of Airbnb offerings.

Shortcoming : If certain room types are overrepresented in specific areas, it might lead to market saturation, affecting pricing and competition negatively.

10. Correlation among features of Airbnb Bookings

```
# Plotting the Correlation Heatmap of features of Airbnb Bookings
print("Correlation Heatmap of features of Airbnb Bookings : \n\n")
numeric_df = df.select_dtypes(include=[np.number])
corr_matrix = numeric_df.corr()
mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
plt.figure(figsize=(12, 8))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.heatmap(corr_matrix, mask=mask, cmap='coolwarm', vmax=0.3, center=0, square=True, linewidths=.5, annot=True, fmt='.2f')
plt.title('\nCorrelation Heatmap of features of Airbnb Bookings\n')
plt.xlabel('\nFeatures\n')
plt.ylabel('\nFeatures\n')
plt.show()
```

The heatmap visually represents the correlation between numeric variables, allowing us to easily identify which features are strongly related or independent from each other.

The heatmap shows relationships between numeric features, such as how variables like price and number of reviews might be correlated. For example : Correlation heatmap shows a high correlation between number of reviews and reviews per month..

Understanding these correlations can guide strategic decisions, such as pricing strategies or focusing on features that drive more reviews.

Shortcoming : If some features are highly correlated, it might suggest redundancy, which could lead to inefficiencies if not managed properly.

11.Relation between neighbourhood group and availability

```
# Plotting the Violin Plot between neighbourhood_group and
availability_365
print("Violin Plot between neighbourhood_group and availability_365 :
\n\n")
plt.figure(figsize=(10,10))
ax = sns.violinplot(data=df, x="Neighbourhood_group",
y="Availability_365")
plt.title('\nRelation between neighbourhood group and availability\n')
plt.xlabel('\nneighbourhood_group\n')
plt.ylabel('\navailability_365\n')
plt.plot()
plt.show()
```

The violin plot shows the distribution of "availability_365" across different neighborhood groups and highlights both the spread and density of availability of rooms for 365 days a year within each neighborhood group.

This plot shows the variations in listing availability across neighborhoods, such as whether some areas have more listings available year-round compared to others, and how this availability is distributed.

Understanding availability trends by neighborhood can guide investment decisions and pricing strategies. For instance, Staten Island has the highest mean availability value around 220-250 days compared to others .

Shortcoming : Areas with very high availability could indicate potential oversupply, affecting market dynamics.

12.Relation between neighbourhood group and price

```
# Plotting the barplot between Neighbourhood_group and Price
print("Barplot between neighbourhood group and price:\n\n")
plt.figure(figsize=(12, 8))

# Create the barplot using Seaborn's default color palette
```

```

sns.barplot(
    data=df,
    x='Neighbourhood_group',
    y='Price',
    palette='tab10' # Use the default `tab10` color palette
)

# Add title and labels
plt.title('\nRelation between Neighbourhood Group and Price\n',
    fontsize=16)
plt.xlabel('\nNeighbourhood Group\n', fontsize=12)
plt.ylabel('\nPrice\n', fontsize=12)

# Display the chart
plt.show()

```

The bar plot compares the average prices of Airbnb listings across different neighborhoods, making it easy to see which areas have higher or lower average rental costs.

This chart shows variations in average listing prices between neighborhoods, displaying which areas are more expensive or affordable. For example : neighborhoods like Manhattan might have higher average prices compared to others.

Understanding price trends across neighborhoods can help set competitive pricing strategies and target areas with higher potential revenue.

Shortcoming : If Neighborhoods with high average prices are not managed well , they might also indicate higher competition, potentially affecting profitability .

13.Airbnb Listings by Neighborhood Group

```

# Plotting the Scatter plot with longitude and latitude, colored by
neighbourhood_group
print("Scatter Plot between latitude and longitude of each Airbnb
listing by Neighbour group : \n\n")
plt.figure(figsize=(10, 6))
# Scatter plot with longitude and latitude, colored by
neighbourhood_group
sns.scatterplot(x='Longitude', y='Latitude',
    hue='Neighbourhood_group', data=df)

# Turn off the interactive mode to prevent plots from displaying
twice
plt.ioff()
plt.title('\nScatter Plot between latitude and longitude of each
Airbnb listing\n')
plt.xlabel('\nLongitude\n')
plt.ylabel('\nLatitude\n')
plt.legend()

```

```
plt.show()
```

The scatter plot visually displays the geographic distribution of Airbnb listings across different neighborhoods and shows how listings are spread out and grouped geographically.

This chart reveals how listings are concentrated in various neighborhood groups, which can popular areas (Manhattan and Brooklyn) for Airbnb rentals and highlight any spatial patterns or clusters of high-density listings.

Understanding the distribution of listings can guide towards strategic decisions, such as targeting areas with high demand or avoiding oversaturated neighborhoods. Insights into geographic patterns can help in optimizing pricing and marketing strategies.

Shortcoming : An oversupply in certain areas might lead to increased competition and potentially lower prices if not managed carefully.

14. Classification of rooms based on price

```
# Function to categorize room prices
def categorise(Hotel_Price):
    if Hotel_Price <= 60:
        return 'Low'
    elif 60 < Hotel_Price <= 400:
        return 'Medium'
    else:
        return 'High'

# Apply categorization
df['Room_Category'] = df['Price'].apply(categorise)

# Plot the bar chart using Seaborn's default color palette
room_category_counts = df['Room_Category'].value_counts()
room_category_counts.plot(kind='bar', color=sns.color_palette("Set2",
n_colors=3))

# Add title and labels
plt.title("\nClassification of Rooms Based on Price\n", fontsize=16)
plt.xlabel("\nRoom Category Based on Price\n", fontsize=12)
plt.ylabel("\nCount\n", fontsize=12)

# Display the chart
plt.show()
```

This bar chart displays the distribution of Airbnb listings categorized by price range, allowing us to see how many listings fall into each price category (Low, Medium, High).

This chart shows the frequency of listings in each price category, revealing whether most listings are budget-friendly, mid-range, or high-end. This helps identify the market segments most represented in the dataset.

Understanding the distribution of listings by price range can help to tailor the marketing and pricing strategies to target the most common price segments, optimizing revenue potential.

Shortcoming : If a large portion of listings is in the 'Low' category, it might indicate a saturated budget market, potentially leading to lower margins and higher competition.

5. Solution to Business Objective

The exploratory data analysis have focussed on understanding key metrics like price distribution, room types, neighborhood popularity, and availability patterns in the Airbnb NYC 2019 dataset. By visualizing and analyzing these factors, we can identify trends and correlations that influence guest booking behavior and host success. This analysis has provided a guide to Airbnb in optimizing pricing strategies, enhancing customer satisfaction, and tailoring marketing efforts. Additionally, insights gained can help in improving platform features and introducing new services, ultimately driving growth and maintaining a competitive edge in the market.

Project Summary

Project Overview

Dataset: Contains 49,000 entries with 16 attributes related to Airbnb listings in New York City.

Objective: Perform Exploratory Data Analysis (EDA) to uncover insights for optimizing listings, investment decisions, and urban planning.

Key Findings

Price Distribution:

Most properties are priced between \$75 and \$500 per night. Outliers exist at both high and low ends of the price spectrum.

Room Type Insights:

Entire homes/apartments dominate (52%), followed by private rooms (45.7%). Shared rooms are the least common (2.4%).

Neighborhood Analysis:

Manhattan and Brooklyn are the most popular neighborhoods, accounting for 85.4% of listings. Listings in Manhattan generally have higher prices.

Correlation Analysis:

Higher-priced properties have fewer reviews, suggesting high-end properties may be less frequently booked.

Availability Trends:

Staten Island has the highest average availability, while other boroughs show varied patterns.

Host Concentration:

A few hosts dominate the market with multiple listings, indicating a concentrated host presence.

Visual Insights

Pie Charts: Show distribution of room types and neighborhood popularity.

Bar Charts: Highlight top hosts, neighborhoods, and room categories by price.

Scatter Plots: Display geographic distribution of listings.

Heatmaps: Show correlations between numeric features.

Business Recommendations

For Hosts:

Focus on offering entire homes/apartments in popular neighborhoods. Competitive pricing is essential for high-demand areas like Manhattan. Consider unique offerings in underrepresented room types or areas.

For Airbnb:

Optimize platform features to highlight high-demand locations and room types. Develop promotional strategies during low-demand periods. Collaborate with top-performing hosts for marketing opportunities.

Conclusion

The analysis of the Airbnb NYC 2019 data showed important patterns in prices, room types, and popular neighborhoods.

It helped identify which areas and room types are most in demand.

The pricing trends can guide hosts to set competitive rates that attract more guests.

Room preferences, like entire homes or shared spaces, can help Airbnb and hosts meet customer needs.

Popular neighborhoods can become focus areas for improving services and experiences.

Seasonal trends show when demand is higher or lower, helping with planning and promotions.

This information helps Airbnb make smart decisions to improve customer satisfaction and grow the business.

Overall, understanding these trends ensures Airbnb stays competitive in New York City.

Hurrah! You have successfully completed your EDA Capstone Project !!!