PROJECT REPORT

IE7275 – Data Mining in Engineering

Group 12
Rithika Sankar Rajeswari
Sreevarshan Sathiyamurthy
Sree Ramya Pasala
Arvind Deivanayagam

sankarrajeswari.r@northeastern.edu
sathiyamurthy.sr@northeastern.edu
pasala.sr@northeastern.edu
deivanayagam.a@northeastern.edu

Percentage of Effort Contributed by Rithika: 25%

Percentage of Effort Contributed by Sreevarshan: 25%

Percentage of Effort Contributed by Sree Ramya: 25%

Percentage of Effort Contributed by Arvind: 25%

Submission Date: 04/21/2025

INDEX

S.no	Contents	Page No.
1.	Introduction	1
2.	Dataset Preprocessing	3
3.	Exploratory Data Analysis	7
4.	Feature Engineering	22
5.	Model Exploration	24
6.	Model Performance Evaluation	31
7.	Performance Diagnostics	33
8.	Conclusion	37

1. INTRODUCTION

About the Dataset

- The Airbnb Open Data dataset provides a detailed snapshot of Airbnb listings across New York City. It contains comprehensive information on listings, including host details, pricing, availability, location, and user engagement metrics. The dataset is particularly well-suited for data analysis, modeling, and visualization tasks within the context of the short-term rental market.

- This dataset consists of 102,599 records and 26 columns. Each row corresponds to a unique Airbnb listing. Key attributes include the listing ID, host information, room type, price, service fee, minimum nights required, number of reviews, review frequency, cancellation policy, and geographic coordinates (latitude and longitude). These details enable users to explore trends in pricing, popularity, and availability across different neighborhoods and boroughs in New York City.

- Several columns provide insights into customer interaction and booking behavior, such as the number of reviews, reviews per month, last review date, and average rating. Location-based fields such as neighborhood, neighborhood group (borough), and geographic coordinates support spatial analysis and visualization. Operational variables like availability throughout the year, whether a listing is instant bookable, and host activity (number of listings managed) further enrich the data.

- While the dataset offers a broad and useful set of features, it also includes certain limitations. Some fields have missing values or inconsistencies, such as duplicate or misspelled neighborhood group names. Additionally, price and availability values sometimes include outliers, which require handling during the preprocessing stage.

- This dataset was originally derived from the Inside Airbnb initiative and later made available through Kaggle. It serves as a valuable resource for students, researchers, and practitioners interested in analyzing patterns within the sharing economy, particularly in the context of urban tourism and digital marketplaces.

Problem Statement

In the rapidly expanding short-term rental ecosystem, platforms like Airbnb provide a wide array of accommodations that vary significantly in price based on a variety of factors. However, this pricing variability introduces challenges for both service providers (hosts) and consumers (guests), making it difficult to assess value and optimize listing strategies.

This project seeks to address the following central question:
Can Airbnb listing prices be effectively categorized into predefined price tiers—Low, Medium, and High—using machine learning models informed by listing-specific features?

To explore this problem, the project is grounded in the following objectives:

- **Identify and understand** the key variables (e.g., room type, location, availability, host attributes, reviews) that influence Airbnb pricing behavior.

- **Transform the continuous price variable** into meaningful categorical tiers, facilitating a classification-based modeling approach.

- **Develop predictive models** using supervised machine learning algorithms (Logistic Regression, Random Forest, XGBoost) to classify listings based on their price category.

- **Evaluate and compare model performance** using appropriate classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

- **Visualize geospatial pricing patterns** and model behavior through data-driven diagnostics to enhance interpretability and actionable insights.

Through this process, the project aims not only to construct a robust predictive model but also to provide deeper analytical insight into how various features interact to shape Airbnb pricing outcomes.

2. DATA PREPROCESSING

Loading and Viewing the Dataset

The data preprocessing began by loading the dataset using the pandas library in Python. The dataset was sourced from a CSV file titled Airbnb_Open_Data.csv, which contains detailed information about Airbnb listings, including host details, location, pricing, room types, and review statistics.

After loading, the initial step involved examining the structure of the dataset using df.shape and df.head(). The dataset consists of 102,599 rows and 26 columns.

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long	country	...	service fee	minimu nigh
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237	United States	...	$193	10
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377	United States	...	$28	30
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harlem	40.80902	-73.94190	United States	...	$124	3
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976	United States	...	$74	30
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399	United States	...	$41	10

5 rows × 26 columns

```
[ ]: df.shape

[4]: (102599, 26)
```

Key Columns and their description:

Column Name	Description
id	A unique identifier for each listing
NAME	Title or name of the Airbnb listing
host id	Unique ID associated with the host
host_identity_verified	Indicates if the host's identity is verified (verified or unconfirmed)
neighbourhood group	Larger area grouping of the listing location
lat, long	Latitude and Longitude coordinates of the listing
country, country code	Country name and its respective ISO code
instant_bookable	Indicates whether the listing can be booked instantly
cancellation_policy	The type of cancellation policy offered (flexible, moderate, etc.)
room type	Type of accommodation (Entire home/apt, Private room, etc.)
Construction year	The year the property was constructed (may contain missing values)
price, service fee	Listing price and applicable service fee
minimum nights	Minimum number of nights required per booking
number of reviews	Total number of reviews received
last review	Date of the most recent review
reviews per month	Average number of reviews per month
review rate number	Average rating(typically out of 5)
calculated host listings count	Number of listings hosted by the same host
availability 365	Number of days the listing is available in a year
house_rules	Set of rules specified by the host for guests
license	Official license number of the listing

Handling Missing Values

The data cleaning process was conducted to prepare the raw Airbnb dataset for analysis and modeling. The aim was to handle missing values, remove irrelevant features, standardize formats, and eliminate anomalous records to ensure data quality.

Step 0: Load the Dataset: The dataset was loaded using the Pandas library from the file Airbnb_Open_Data.csv, containing over 100,000 Airbnb listings along with various host, location, price, and review-related attributes.

Step 1: Drop Irrelevant or Mostly-Empty Columns: Columns such as 'license', 'house_rules', 'host name', and 'NAME' were removed either due to excessive missing values or irrelevance to the analysis.

Step 2A: Drop Rows with Missing Target Values: Listings without a value for 'price' were dropped, since price is a crucial target variable in most analysis tasks.

Step 2B: Handle Missing Values in Numerical Columns: Numerical columns with missing values (e.g., 'reviews per month', 'minimum nights') were imputed using the median value of their respective columns to preserve central tendency without being skewed by outliers.

Step 2C: Handle Missing Values in Categorical Columns: Categorical columns with missing values (e.g., 'cancellation_policy', 'instant_bookable') were filled using the mode (most frequent value), ensuring consistency with existing patterns in the data.

Step 3: Convert Price and Service Fee to Numeric Format: The 'price' and 'service fee' columns initially contained currency symbols ($) and commas, which were removed using regex before converting the fields to float data type.

```
[ ]: # STEP 0: Load dataset
import pandas as pd
df = pd.read_csv('Airbnb_Open_Data.csv')

# STEP 1: Drop irrelevant or mostly-empty columns
columns_to_drop = ['license', 'house_rules', 'host name', 'NAME']
df.drop(columns=columns_to_drop, inplace=True)

# STEP 2A: Drop rows where 'price' is missing (since it's our prediction target)
df = df.dropna(subset=['price'])

# STEP 2B: Fill missing numerical columns with median
numerical_cols = ['reviews per month', 'review rate number', 'number of reviews',
                  'availability 365', 'minimum nights', 'calculated host listings count']
for col in numerical_cols:
    if col in df.columns:
        median_value = df[col].median()
        df[col] = df[col].fillna(median_value)

# STEP 2C: Fill missing categorical columns with mode
categorical_cols = ['cancellation_policy', 'instant_bookable', 'host_identity_verified']
for col in categorical_cols:
    if col in df.columns:
        mode_value = df[col].mode()[0]
        df[col] = df[col].fillna(mode_value)

# STEP 3: Convert 'price' and 'service fee' from string to float
# Strip $ and commas, then convert
df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)
df['service fee'] = df['service fee'].replace('[\$,]', '', regex=True).astype(float)

# OPTIONAL: Check the first few cleaned rows
df.head()
```

[7]:

	id	host id	host_identity_verified	neighbourhood group	neighbourhood	lat	long	country	country code	instant_bookable	...	Construction year
0	1001254	80014485718	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	United States	US	False	...	2020.0
1	1002102	52335172823	verified	Manhattan	Midtown	40.75362	-73.98377	United States	US	False	...	2007.0
2	1002403	78829239556	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	United States	US	True	...	2005.0
3	1002755	85098326012	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	United States	US	True	...	2005.0
4	1003689	92037596077	verified	Manhattan	East Harlem	40.79851	-73.94399	United States	US	False	...	2009.0

Step 4: Remove Duplicate Rows

Duplicate records were removed to avoid skewing the analysis and ensure data uniqueness.

Step 5: Remove Outliers and Unrealistic Values

To ensure data reliability, the following filters were applied:

- Listings with a price above **$1000** per night were removed.
- Listings with 'minimum nights' > 365 were considered unrealistic and dropped.
- Listings with 'availability 365' > 365 were also filtered out, as they exceed the logical limit of days in a year.

```python
# STEP 4: Drop duplicate rows if any
df.drop_duplicates(inplace=True)

# STEP 5: Remove outliers based on 'price' and optionally 'minimum nights'

# Set reasonable price threshold — Airbnb listings priced over $1000/night are often anomalies
df = df[df['price'] <= 1000]

# Optional: Remove listings with unrealistic minimum nights (e.g., over 365)
df = df[df['minimum nights'] <= 365]

# Optional: Remove listings with extremely high availability (e.g., > 365 days is not possible)
df = df[df['availability 365'] <= 365]

# Preview cleaned dataset
df.reset_index(drop=True, inplace=True)
df.head()
```

	id	host id	host_identity_verified	neighbourhood group	neighbourhood	lat	long	country	country code	instant_bookable	...	Construction year
0	1001254	80014485718	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	United States	US	False	...	2020.0
1	1002102	52335172823	verified	Manhattan	Midtown	40.75362	-73.98377	United States	US	False	...	2007.0
2	1002403	78829239556	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	United States	US	True	...	2005.0
3	1002755	85098326012	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	United States	US	True	...	2005.0
4	1003689	92037596077	verified	Manhattan	East Harlem	40.79851	-73.94399	United States	US	False	...	2009.0

5 rows × 22 columns

Final Cleaning

The dataset index was reset after all cleaning operations, and the cleaned dataset was previewed for verification. The final shape confirmed the successful reduction in noise and missing data.

```python
[ ]: df.head(10)
```

[10]:

	id	host id	host_identity_verified	neighbourhood group	neighbourhood	lat	long	country	country code	instant_bookable	...
0	1001254	80014485718	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	United States	US	False	...
1	1002102	52335172823	verified	Manhattan	Midtown	40.75362	-73.98377	United States	US	False	...
2	1002403	78829239556	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	United States	US	True	...
3	1002755	85098326012	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	United States	US	True	...
4	1003689	92037596077	verified	Manhattan	East Harlem	40.79851	-73.94399	United States	US	False	...
5	1004650	61300605564	unconfirmed	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	United States	US	False	...
6	1006859	1280143094	verified	Manhattan	Chinatown	40.71344	-73.99037	United States	US	False	...
7	1007411	18824631834	verified	Manhattan	Upper West Side	40.80316	-73.96545	United States	US	True	...
8	1007964	88136055909	verified	Manhattan	Hell's Kitchen	40.76076	-73.98867	United States	US	False	...
9	1008516	26802410424	verified	brookln	South Slope	40.66829	-73.98779	United States	US	True	...

10 rows × 22 columns

3. Exploratory Data Analysis (EDA)

3.1 Univariate Analysis

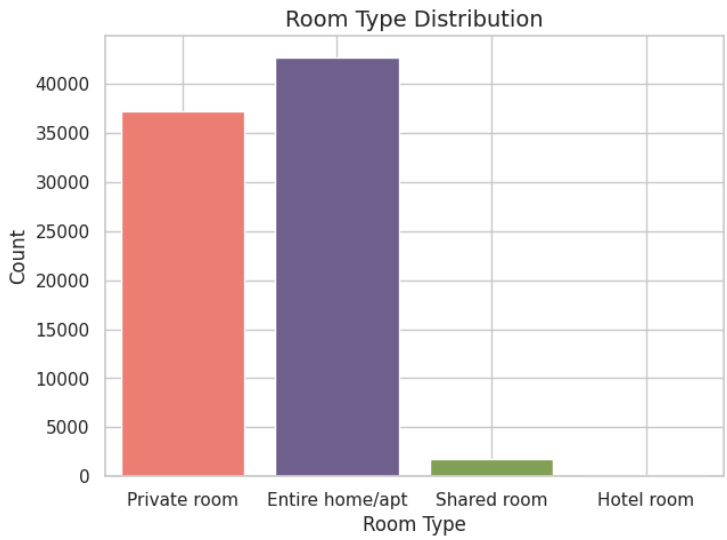
a) Price Distribution of Airbnb Listings

This histogram visualizes the distribution of Airbnb listing prices across the dataset. The plot includes a Kernel Density Estimation (KDE) curve, which smooths the histogram to better illustrate the underlying distribution pattern.

Interpretation:

- The distribution of prices is fairly uniform between $100 and $1000, suggesting that listings are well spread across this price range.
- No distinct peak or mode is observed, which implies that Airbnb listings do not cluster around a specific price point but are instead evenly distributed.
- The frequency drops off sharply below $100 and above $1000, which aligns with the earlier preprocessing step where outliers (prices above $1000) were removed.
- The KDE curve indicates a slight increase in frequency between $150 and $200, suggesting a relatively common pricing range in that bracket.
- Overall, the distribution appears relatively flat, showing that Airbnb listings cater to a wide range of price segments without a strong central tendency.

b) Room Type Distribution



This bar plot displays the frequency of each Airbnb room type available in the dataset. It uses a count plot to represent how listings are categorized based on their room type.

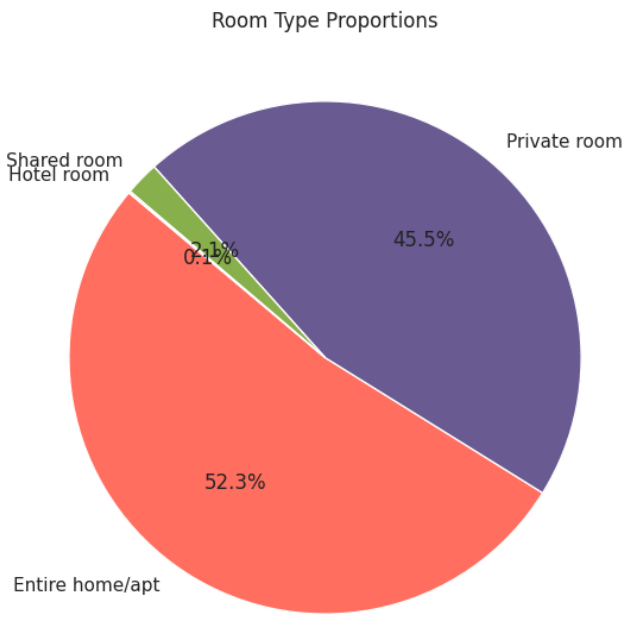
Interpretation:

- The two most common room types are:
  - Entire home/apt, which is the most frequently listed category with the highest count.
  - Private room, which closely follows and represents a significant portion of listings.
- Shared rooms are relatively rare, comprising only a small fraction of total listings.
- Hotel rooms are nearly negligible or completely absent in the dataset, suggesting Airbnb remains more focused on residential and informal accommodations.

Insights:

- The dominance of "Entire home/apt" and "Private room" listings indicates that most hosts are either offering their full property or a dedicated private space—appealing to both families and solo travelers.
- The low count for "Shared room" suggests limited interest from guests in communal or hostel-style arrangements, likely due to privacy or post-pandemic preferences.
- The data distribution here can guide platform strategies, such as promoting underrepresented categories or optimizing search filters for user preferences.

c) Room Type Proportions



This pie chart provides a proportional breakdown of the different room types listed on Airbnb, offering a visual complement to the earlier bar plot.

Interpretation:

- Entire home/apt listings make up the majority share at 52.3%, indicating that more than half of the properties available on the platform offer full, private accommodations—ideal for families, groups, or those seeking complete privacy.
- Private rooms account for 45.5% of listings, showing significant availability and demand among travelers who are open to sharing a home while still having personal space.
- Shared rooms represent a small share of the market at 2.1%, suggesting limited supply and possibly lower demand for communal accommodation.
- Hotel rooms are negligible at 0.1%, confirming that Airbnb remains primarily a platform for private individuals rather than commercial lodging providers.

Insights:

- The combined dominance of entire homes and private rooms (over 97%) reflects Airbnb's core strength in non-traditional, residential-based stays.
- These proportions are useful for understanding host behavior, designing marketing strategies, and shaping platform policies—for example, focusing support on private property owners rather than hotel chains.

d) Price Distribution by Room Type (Violin Plot)

This violin plot visualizes the distribution of listing prices for each room type category. It combines a box plot and a kernel density plot, providing insight into both the central tendency and the distribution spread.

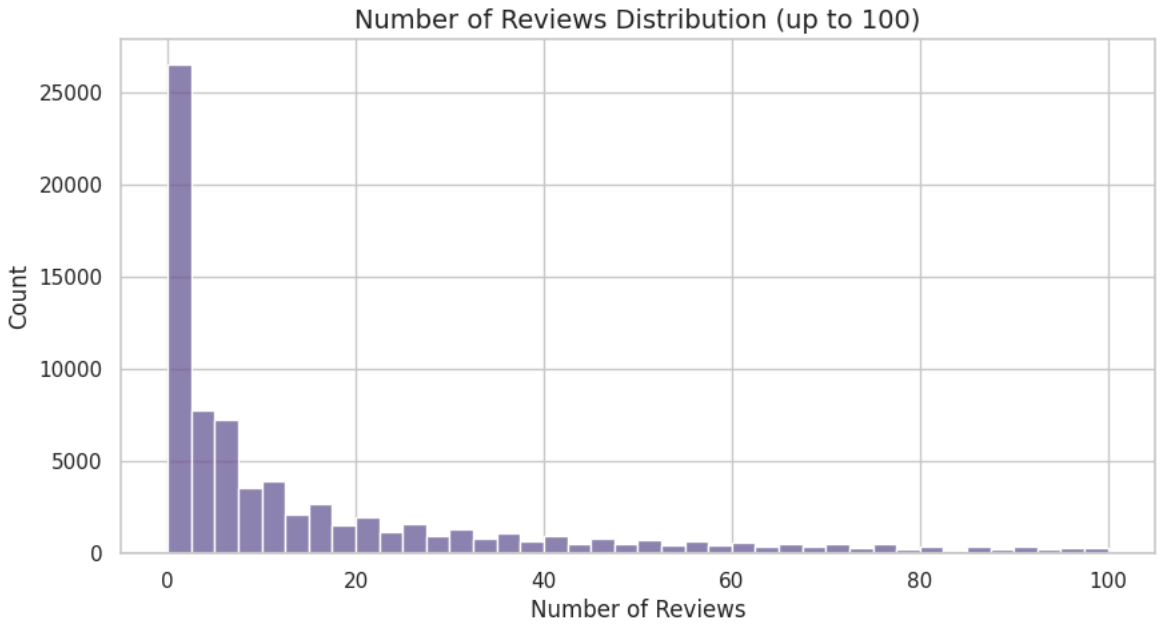
Interpretation:

- All room types (Private room, Entire home/apt, Shared room, Hotel room) show a wide price range, with values extending close to the $1000 cap.
- Despite expectations, the median price (white dot in each violin) appears similar across room types, hovering around the $500–$550 range.
- The density distributions are fairly symmetrical and flat, implying a broad and even spread of prices for each category.
- The vertical thickness at different price levels represents concentration:
    - All room types show thicker middle sections between $300–$700, indicating a large concentration of listings in this price band.
- Hotel rooms, though few in number, show a wider variation at higher prices, possibly reflecting boutique or premium properties.

Insights:

- The lack of a clear price gap between shared/private rooms and entire homes suggests non-standardized pricing hosts may price based on location, amenities, or demand rather than room type alone.
- This can inform platform recommendations or pricing algorithms by highlighting the need to evaluate factors beyond just room type.

e) Number of Reviews Distribution



This histogram illustrates the distribution of the number of reviews per Airbnb listing, focusing only on listings with 100 or fewer reviews to avoid skew from extreme outliers.

Interpretation:

- A very high concentration of listings has fewer than 5 reviews, with the peak sharply occurring at 0–2 reviews.
- The frequency drops exponentially as the number of reviews increases, indicating a long-tailed distribution—only a small portion of listings receive a high volume of reviews.
- This skewed pattern is common in user-generated content platforms, where a minority of listings attract most of the attention or bookings.
- The distribution flattens after 20 reviews, showing that very few listings maintain consistent, high engagement.

Insights:

- The vast number of low-review listings may include new or infrequently booked properties, which could benefit from improved visibility or incentive strategies.
- Listings with very high review counts likely belong to experienced or superhosts, and may represent high-quality offerings.
- For modeling or segmentation purposes, treating listings with 0–5, 6–20, and 21+ reviews as separate user engagement tiers could be useful.

3.2 Bivariate Analysis

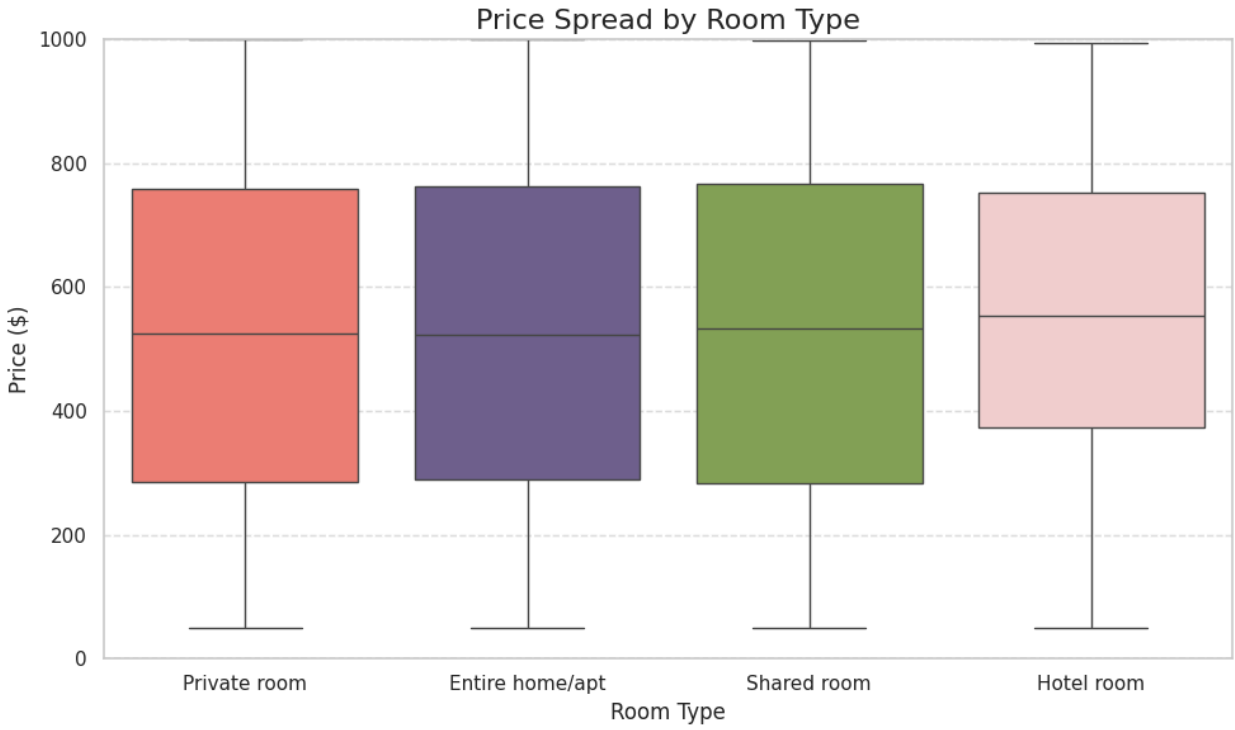
f) Average Price by Room Type

This bar chart shows the mean price of listings across different room types on Airbnb, providing a straightforward comparison of typical costs for each category.

Interpretation:

- Hotel rooms have the highest average price, suggesting they are typically positioned as more premium listings, possibly with additional services or luxury features.
- Private rooms, shared rooms, and entire homes/apartments all have very similar average prices, with only slight variations:
    - All three hover around the $520–$530 range.
    - This is unusual, as one might expect entire homes to be significantly more expensive than private or shared rooms.
- The similarity in average price across these categories could be due to:
    - Pricing inconsistencies or inflation from luxury private/shared listings.
    - Hosts setting prices without clear alignment to room type.
    - Geographic factors where expensive neighborhoods raise prices across all room types.

g) Price Spread by Room Type

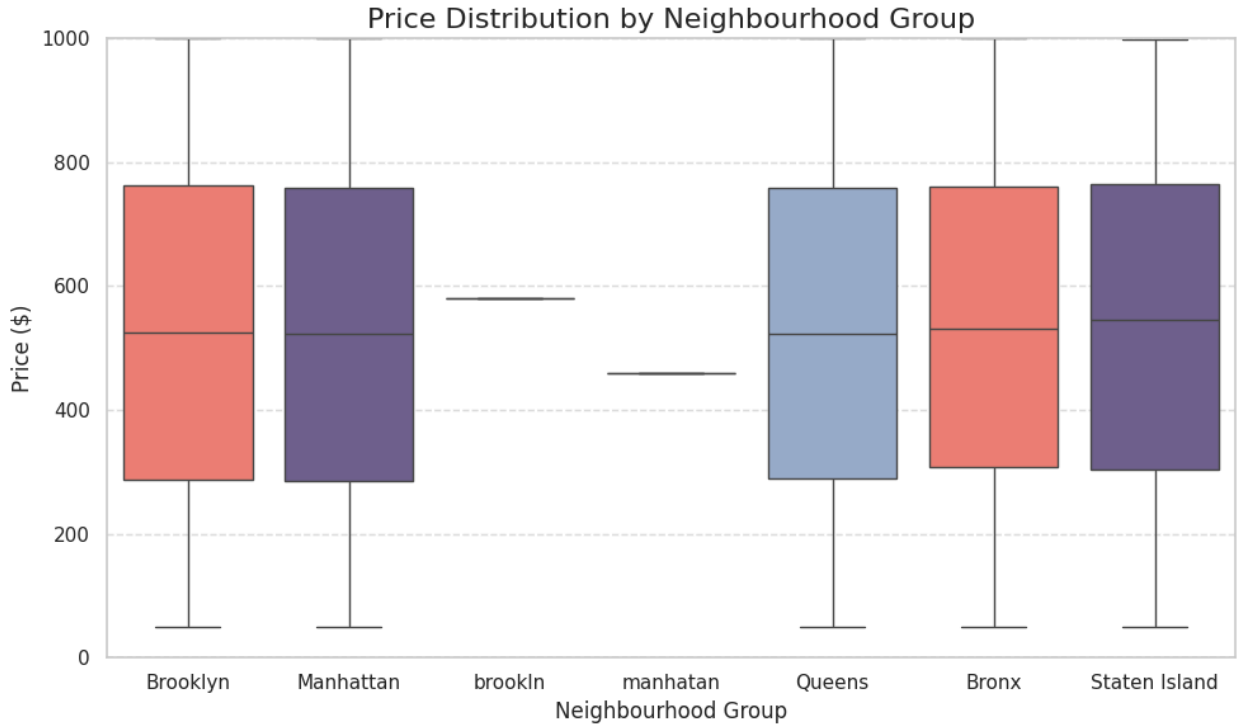


This box plot compares the distribution of listing prices for each room type using quartile statistics and outlier detection.

Interpretation:

- All room types—Private room, Entire home/apt, Shared room, and Hotel room—exhibit a broad price range, with prices extending close to the $1000 upper threshold set during preprocessing.
- Median price (black line inside each box) is relatively consistent across all room types, generally falling between $500–$550, echoing observations from the previous bar plot.
- The interquartile range (IQR)—the box portion—shows substantial variation:
    - It spans approximately from $250 to $750 for all room types, indicating a large variance in pricing even within each category.
- Outliers beyond the whiskers are suppressed here due to the y-limit set at 1000, but the vertical extent still hints at pricing inconsistencies.
- Interestingly, Shared rooms and Hotel rooms, despite typically having distinct market expectations, display price ranges nearly identical to Private and Entire home listings.

h) Price Distribution by Neighbourhood Group



This box plot compares the spread of listing prices across different **neighbourhood groups** (boroughs) in the Airbnb dataset, providing insights into regional pricing dynamics within the city.

Interpretation:

- All boroughs (Brooklyn, Manhattan, Queens, Bronx, Staten Island) show a wide distribution of prices, with interquartile ranges (IQRs) largely overlapping.
- The median prices (horizontal lines in the boxes) are surprisingly similar across boroughs, clustering between $500 and $550.
- Despite expectations that Manhattan might dominate in price, the data shows comparable pricing in Brooklyn, Bronx, and Staten Island, which may be influenced by pricing outliers or non-standard listings.
- The presence of "brooklyn" and "manhattan" indicates spelling inconsistencies in the neighbourhood group column, likely due to data entry errors. These should be cleaned or merged to avoid misrepresentation.

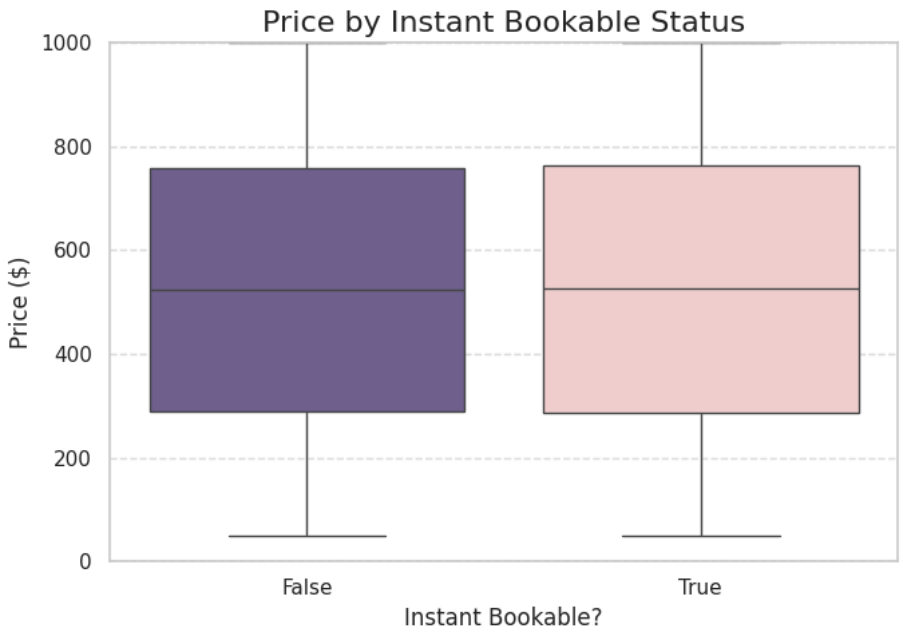
i) Price by Cancellation Policy

This box plot illustrates how Airbnb listing prices vary depending on the cancellation policy selected by hosts: strict, moderate, or flexible.

Interpretation:

- All three cancellation policy types—strict, moderate, and flexible—show very similar price distributions:
    - The median price (horizontal line inside the box) hovers around $500–$530 for each category.
    - The interquartile range (IQR) is also comparable, suggesting consistent pricing behavior regardless of cancellation flexibility.
- Despite assumptions that stricter cancellation policies might be used for premium or high-end properties, the data does not show a meaningful pricing difference.
- The range of prices is broad across all policies, with listings priced anywhere from around $50 up to $1000, implying that cancellation policy is not a primary factor in price determination.

j) Plot: Price by Instant Bookable Status



This box plot compares listing prices based on whether a listing is instant bookable (True) or not (False). The instant bookable feature allows guests to reserve a listing without host approval.

Interpretation:

- Both instant bookable (True) and non-instant bookable (False) listings show very similar price distributions:
  - Median price in both categories is nearly the same—around $500–$520.
  - The interquartile range (IQR) and overall spread of prices are comparable.
- The minimum and maximum values extend across the same range, from just below $100 to nearly $1000, indicating no substantial pricing bias associated with instant booking.
- This suggests that offering instant booking does not significantly affect listing price, either in terms of how hosts set it or how the market responds.

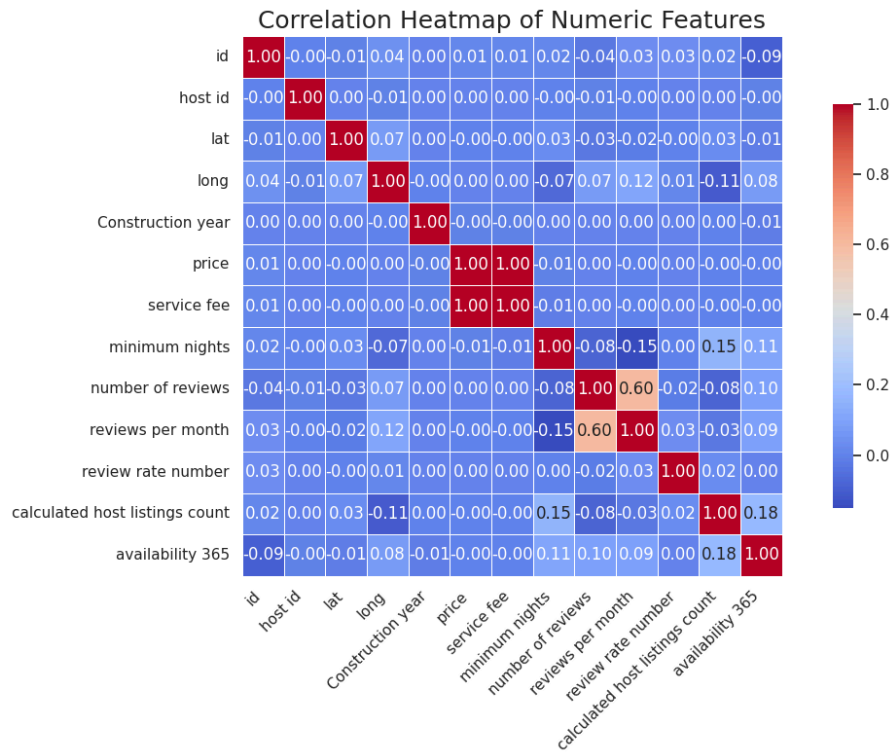
k) Reviews per Month vs Price (Scatter Plot)

This scatter plot explores the relationship between a listing's monthly review frequency and its price, offering insight into how pricing may (or may not) influence guest engagement.

Interpretation:

- The majority of listings cluster heavily within the 0 to 10 reviews per month range, with most concentrated near the lower end (0–2 reviews/month).
- Prices are widely distributed across all review levels, indicating no strong linear relationship between review frequency and price.
- A few outlier listings have more than 40 reviews per month, but they appear across varying price levels—from low-cost listings to those closer to $1000.
- The vertical spread at low review values suggests that many listings with few monthly reviews are priced across the full range, from budget to luxury.

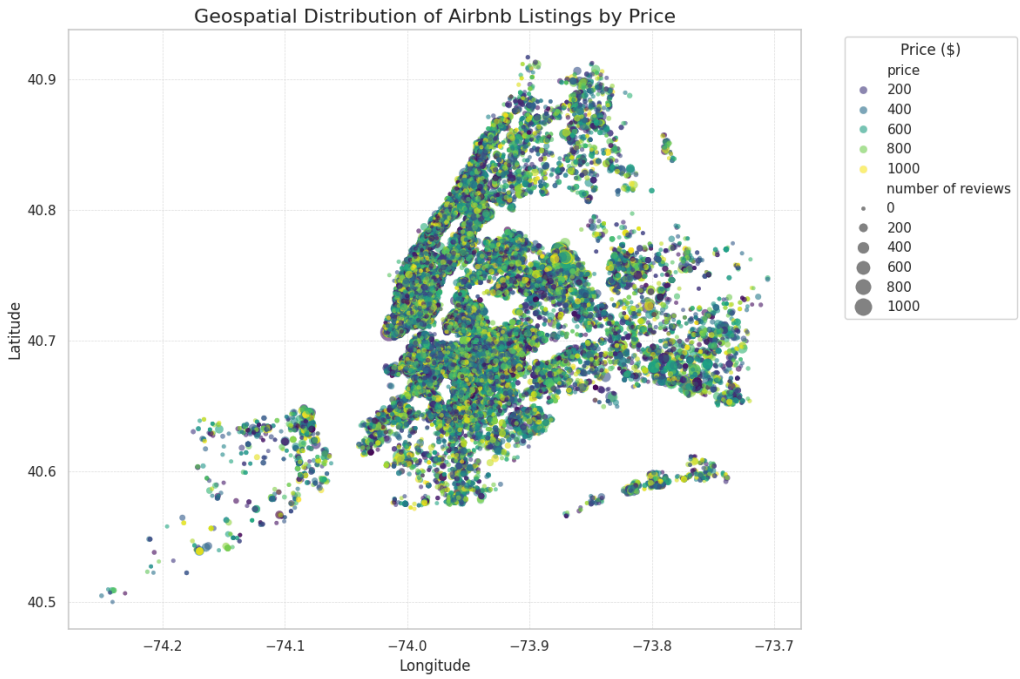
l) Correlation Heatmap of Numeric Features



Interpretation:

- Strongest Positive Correlation:
  - price and service fee: 1.00
    - This is expected, as service fees are usually calculated as a percentage of the price.
- Moderate Correlations:
  - reviews per month and number of reviews: 0.60
    - Listings with high monthly reviews tend to accumulate more total reviews, which is logical.
  - calculated host listings count and availability 365: 0.18
    - Hosts with more listings might keep properties more available to maximize utilization.
- Weak or No Correlation with Price:
  - Surprisingly, price shows very low correlation with all other numeric variables, including:
    - minimum nights (-0.01)
    - number of reviews (0.00)
    - availability 365 (0.00)
  - This implies that price is not linearly influenced by these features, highlighting the potential need for non-linear models or categorical variables to explain pricing patterns.
- No strong negative correlations were observed, suggesting there are no strongly inverse relationships in this dataset.

m) Geospatial Distribution of Airbnb Listings by Price



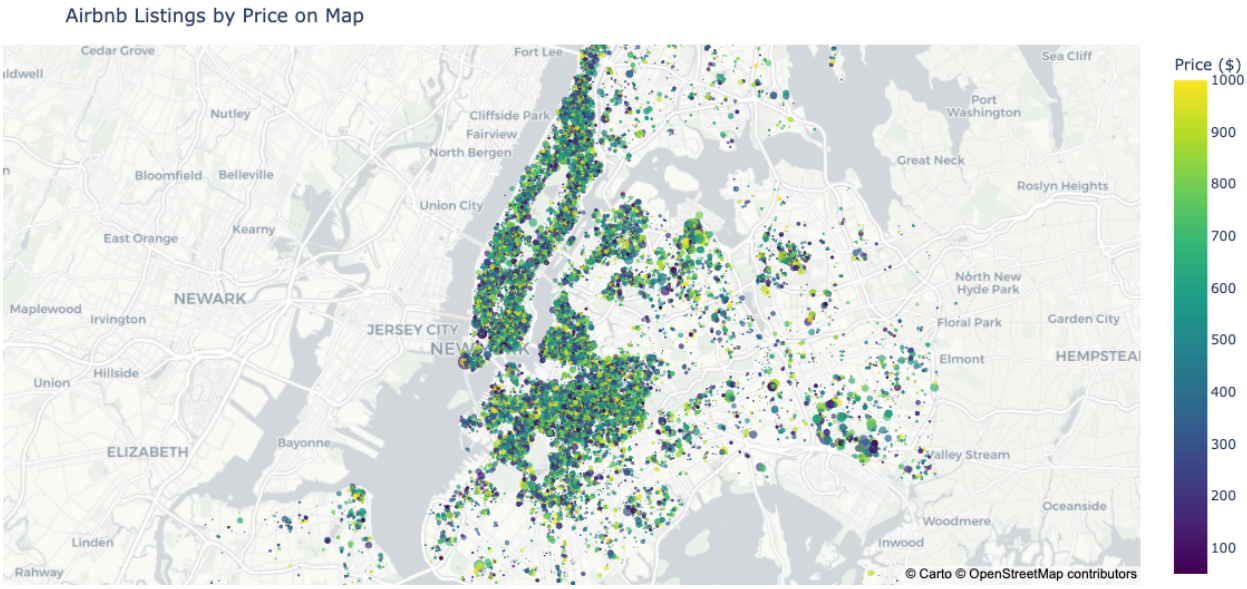
This scatter plot displays the geographic spread of Airbnb listings across New York City based on their latitude and longitude, while encoding:

- Price through color (using a 'viridis' gradient: darker for lower prices, lighter for higher prices)
- Number of reviews through point size (larger dots = more reviews)

Interpretation:

- High-density clusters are clearly visible in central Manhattan, parts of Brooklyn, and along key transit or tourism corridors—these are the most active Airbnb zones.
- Price variation is present across all boroughs, but higher prices (yellow/green dots) are more concentrated in Manhattan, particularly Midtown, Downtown, and Upper East/West Sides.
- Listings in Staten Island and the Bronx tend to show darker hues, indicating lower prices, consistent with their relatively lower tourist volume.
- Larger point sizes (indicating highly reviewed listings) are more frequent in the central and popular areas, suggesting stronger guest engagement and possibly more established hosts.
- Some peripheral zones have sparse listings, fewer reviews, and lower prices—these may represent suburban or less trafficked parts of NYC.

n) Airbnb Listings by Price on Interactive Map



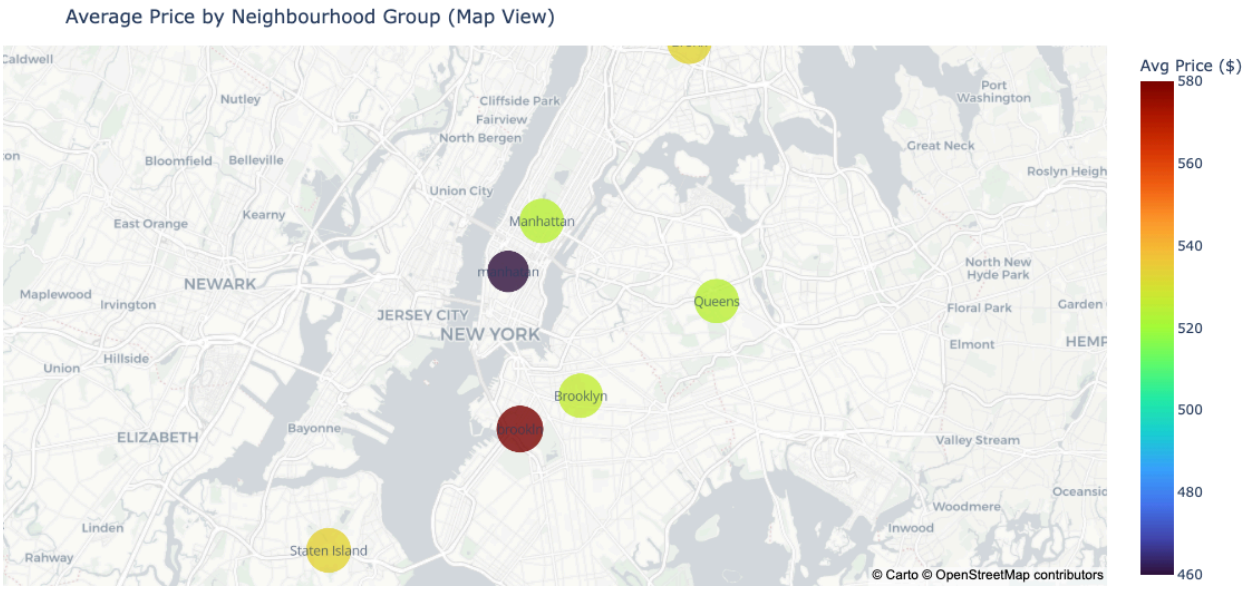
This interactive Mapbox scatter plot shows Airbnb listings across the New York City area, visualizing:

- Geographic location via latitude and longitude
- Price using a Viridis color scale (dark purple for lower prices, yellow for higher prices)
- Number of reviews encoded as marker size (larger = more reviews)

Interpretation:

- The highest concentration of listings is visible in Manhattan, especially central and lower Manhattan (e.g., Midtown, SoHo, East Village), with additional dense clusters in Brooklyn and parts of Queens.
- Color gradient clearly highlights higher-priced listings (yellow/green) being concentrated in Manhattan and central Brooklyn, while lower-priced listings (purple/blue) are more prevalent in the outer boroughs like the Bronx and Staten Island.
- Larger dots—representing more frequently reviewed listings—are often in central areas, confirming these zones attract more consistent guest traffic.
- Outlying areas (e.g., far Staten Island, southeast Queens) contain fewer listings and tend to skew toward lower price and review counts, suggesting lower demand or less tourist traffic.

o) Average Price by Neighbourhood Group



This map summarizes the average Airbnb price for each neighbourhood group (borough) by plotting their geographic centroids. Marker:

- Color represents average price using the Turbo scale (blue → green → yellow → red).
- Size represents the relative magnitude of average price.
- Text labels indicate each neighborhood group (e.g., Brooklyn, Manhattan, Queens).

Interpretation:

- Brooklyn appears to have the highest average price (~$580), indicated by the deep red color and largest marker size.
- Manhattan is surprisingly shown in a cooler color (blue-purple), reflecting a lower average price than Brooklyn in this dataset—likely due to pricing anomalies, misspellings (e.g., "manhattan"), or inclusion of outlier listings.
- Queens, Staten Island, and the Bronx show moderate to low average prices, marked with yellow to green hues.
- Duplicate entries like "brooklyn" and "manhattan" are visualized separately, suggesting a data cleanliness issue that needs resolution to avoid skewed insights.

4. FEATURE ENGINEERING

Feature Selection & Encoding

```
[ ]: selected_features = [
         'room type',
         'neighbourhood group',
         'cancellation_policy',
         'instant_bookable',
         'minimum nights',
         'number of reviews',
         'reviews per month',
         'review rate number',
         'calculated host listings count',
         'availability 365',
         'service fee'
     ]
```

```
[ ]: # Step 1: Select features + target
     features = [
         'room type', 'neighbourhood group', 'cancellation_policy', 'instant_bookable',
         'minimum nights', 'number of reviews', 'reviews per month',
         'review rate number', 'calculated host listings count',
         'availability 365', 'service fee'
     ]

     target = 'price'
     df_model = df[features + [target]].copy()

     # Step 2: One-Hot Encode categorical variables
     df_encoded = pd.get_dummies(df_model, columns=[
         'room type', 'neighbourhood group', 'cancellation_policy', 'instant_bookable'
     ], drop_first=True)

     # Final features and shape
     df_encoded.shape, df_encoded.columns
```

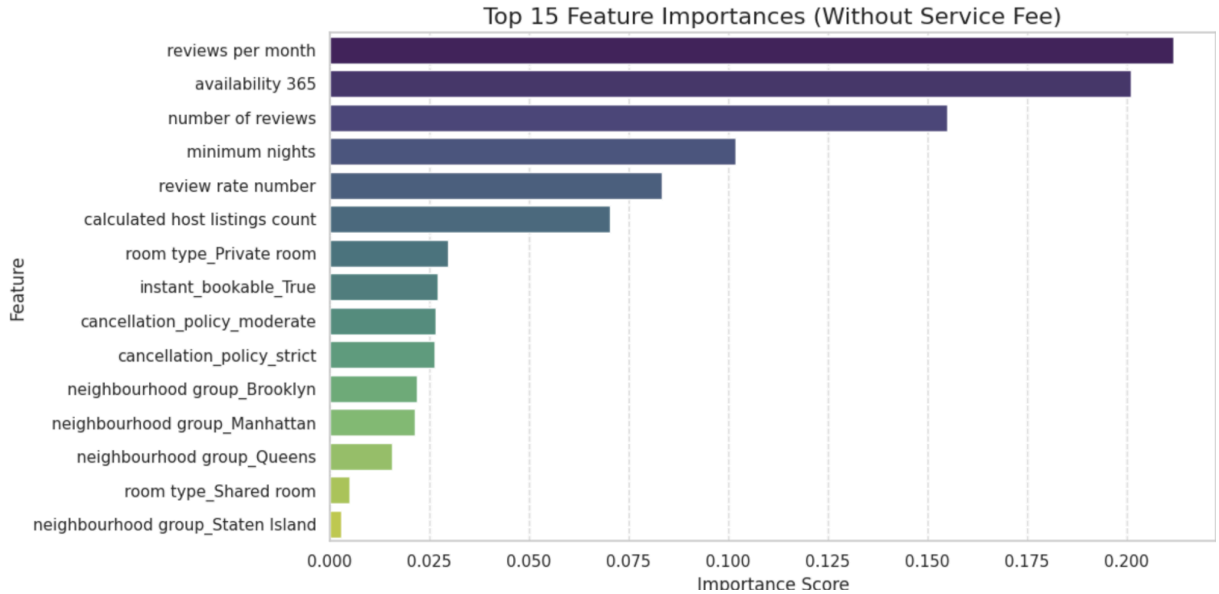
```
[27]: ((81789, 20),
       Index(['minimum nights', 'number of reviews', 'reviews per month',
              'review rate number', 'calculated host listings count',
              'availability 365', 'service fee', 'price', 'room type_Hotel room',
              'room type_Private room', 'room type_Shared room',
              'neighbourhood group_Brooklyn', 'neighbourhood group_Manhattan',
              'neighbourhood group_Queens', 'neighbourhood group_Staten Island',
              'neighbourhood group_brookln', 'neighbourhood group_manhatan',
              'cancellation_policy_moderate', 'cancellation_policy_strict',
              'instant_bookable_True'],
             dtype='object'))
```

This step prepares the Airbnb dataset for machine learning modeling by selecting relevant features and transforming categorical variables into a numeric format through one-hot encoding.

- A new DataFrame df_model is created, containing only the selected features and the target variable (price), isolating the relevant information for modeling.
- One-Hot Encoding: pd.get_dummies() is used to apply one-hot encoding to the dataset's categorical variables (such as room_type, neighbourhood_group, cancellation_policy, and instant_bookable). By doing this, each category is transformed into a binary column (0 or 1), which enables numerical interpretation by machine learning algorithms without suggesting an ordinal link between the categories.
- Preventing Dummy Variable Trap: To prevent multicollinearity, the drop_first=True argument makes sure that the first category in each encoded variable is removed. By doing this, the model is kept free of superfluous features.

To check Feature Importance

Top 15 feature importance



A Random Forest Regressor was trained, and its feature significance scores were examined in order to determine which factors had the most effects on price prediction.Exclusion of Service Fee: Because the service fee column has a strong correlation with the target variable price, it was eliminated from the dataset to prevent data leaking.

Train-Test Split: To guarantee repeatability, the data was split 80-20 into training and testing sets, with a fixed random state.

Random Forest Training: The default settings were used to train a Random Forest Regressor with 100 trees.

Interpretation:

- The most significant pricing predictors were number of reviews, availability_365, and reviews per month.
- Minimum nights, review rating number, and instant_bookable status were other significant drivers.
- Room_type_Shared room and neighborhood group_Staten Island were two categorical factors that had little bearing on price prediction.
- By showing which characteristics have the most predictive potential, this analysis aids in model refinement and presents chances to make the model easier to understand or more straightforward.

5. Model Exploration

The core objective of this phase was to identify and compare machine learning models that could accurately classify Airbnb listings into three price categories: **Low**, **Medium**, and **High**. This required evaluating models that can handle **non-linear patterns**, **categorical features**, and potential **feature interactions** effectively.

Modeling Objective

After transforming the continuous price variable into a categorical feature (price_category), the task was framed as a **supervised multiclass classification problem**. The goal was to predict the price class of a listing based on its descriptive attributes (e.g., room type, location, reviews, availability).

Models Explored

Three classification models were selected for exploration:

- **Logistic Regression**
  - A linear model that serves as a performance baseline.
  - Suitable for interpreting feature influence but limited in modeling complex interactions.
- **Random Forest Classifier**
  - An ensemble method that constructs multiple decision trees and aggregates their predictions.
  - Known for its ability to handle non-linear relationships and reduce overfitting.
- **XGBoost Classifier**
  - A gradient boosting technique optimized for performance and scalability.
  - Especially effective in capturing subtle data patterns through sequential learning.

These models were chosen to represent a progression from **simple (linear)** to **complex (ensemble-based)** classification techniques.

8.1 Binning Price into Categories

```python
# Create categorical labels from price
def price_category(price):
    if price <= 100:
        return 0  # Low
    elif price <= 300:
        return 1  # Medium
    else:
        return 2  # High

# Create new DataFrame with price_category
df_class = df_no_fee.copy()
df_class['price_category'] = df_class['price'].apply(price_category)

# Drop original price column (we're predicting categories now)
df_class = df_class.drop(columns=['price'])
```

8.2 Handle Class Imbalance using SMOTE

```python
from imblearn.over_sampling import SMOTE
from collections import Counter

# Separate features and target
X = df_class.drop('price_category', axis=1)
y = df_class['price_category']

# Apply SMOTE to balance all classes
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

# Confirm new class distribution
print(Counter(y_resampled))
```

8.3 Train-Test Split

```python
from sklearn.model_selection import train_test_split

# Split the balanced dataset
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)
```

8.4 Train Models – Logistic Regression & Random Forest

```python
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier

# Model 1: Logistic Regression
lr = LogisticRegression(max_iter=500, solver='lbfgs')
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

# Model 2: Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```python
import xgboost as xgb
from xgboost import XGBClassifier

# Train XGBoost Classifier
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', random_state=42)
xgb_model.fit(X_train, y_train)
y_pred_xgb = xgb_model.predict(X_test)

# Evaluate XGBoost model
evaluate_model("XGBoost Classifier", y_test, y_pred_xgb)
```

Model Selection Rationale

Based on initial results and diagnostic visuals, **Random Forest Classifier** was identified as the most promising model. It demonstrated superior classification accuracy, especially in distinguishing between the **Medium** and **High** categories, which often exhibit overlapping characteristics.

Classification Strategy Overview

Instead of forecasting the exact price of an Airbnb listing, this project approached the problem as a **multi-class classification task**, grouping prices into three distinct categories: **low**, **medium**, and **high**. This transformation offers better interpretability and is more suitable for real-world applications such as pricing tiers or recommendation systems.

Price Categorization Logic

- A custom function price_category was used to assign classes based on the listing price:
  - **Low (Class 0)**: Price ≤ $100
  - **Medium (Class 1)**: $101 ≤ Price ≤ $300
  - **High (Class 2)**: Price > $300
- This conversion helped simplify the regression problem into a **classification task**, making it easier to model and interpret.

Addressing Class Imbalance

Class imbalance is a common challenge in real-world datasets, especially in cases like this where some price categories are underrepresented. To mitigate this:

- **SMOTE (Synthetic Minority Over-sampling Technique)** was employed.
  - It generates **synthetic samples** for minority classes to achieve a more balanced class distribution.

    - This is critical because **models trained on imbalanced data tend to bias predictions** toward the majority class, reducing accuracy for the underrepresented classes.
- After balancing:
    - The dataset was split into **80% training** and **20% testing** sets.
    - A **fixed random state** was used to ensure reproducibility of results.

Evaluation Metrics Used:

- **Confusion Matrix**: Shows how predicted classes are compared to actual classes.
- **Classification Report**: Includes **precision, recall, and F1-score** for each class.
- **Overall Accuracy**: Provides a single measure of model performance.

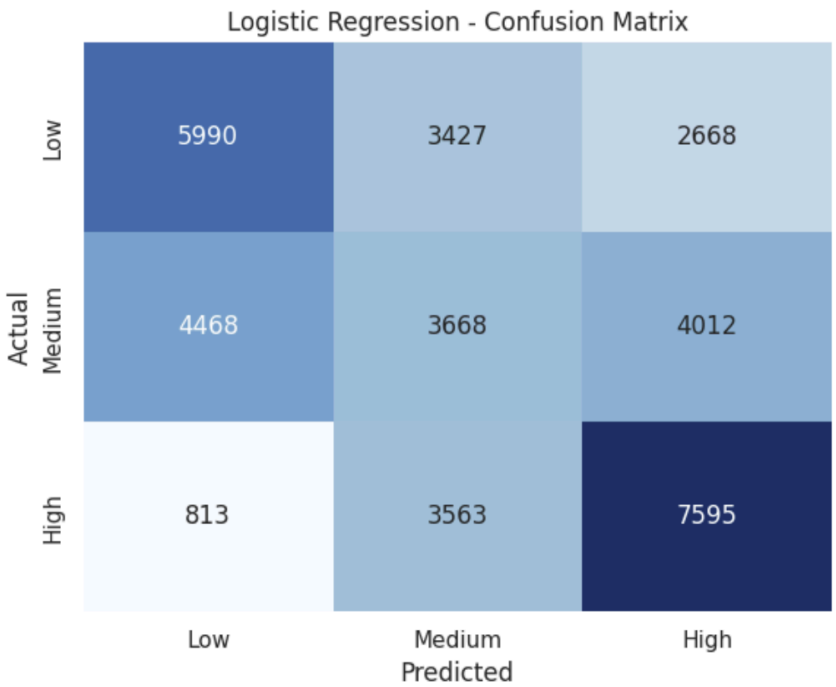
Visualization Tools:

- **Heatmaps** were used to plot confusion matrices for easy interpretation of misclassifications and correct predictions across categories.

Key Results

- The **Random Forest Classifier** outperformed other models.
    - Its ability to handle **non-linear relationships** and **feature interactions** led to significantly better accuracy.
    - It was especially effective at distinguishing **low** and **medium** price categories, which are typically harder to separate due to overlapping features.
- This process highlights the **importance of handling class imbalance** in classification problems.
- It demonstrates the **value of using advanced models like Random Forests** for complex, high-dimensional datasets.
- The resulting classification model provides a **robust foundation** for several practical applications:
    - **Customer segmentation**
    - **Dynamic pricing recommendations**
    - **Strategic decision support systems**

Plot : Logistic Regression



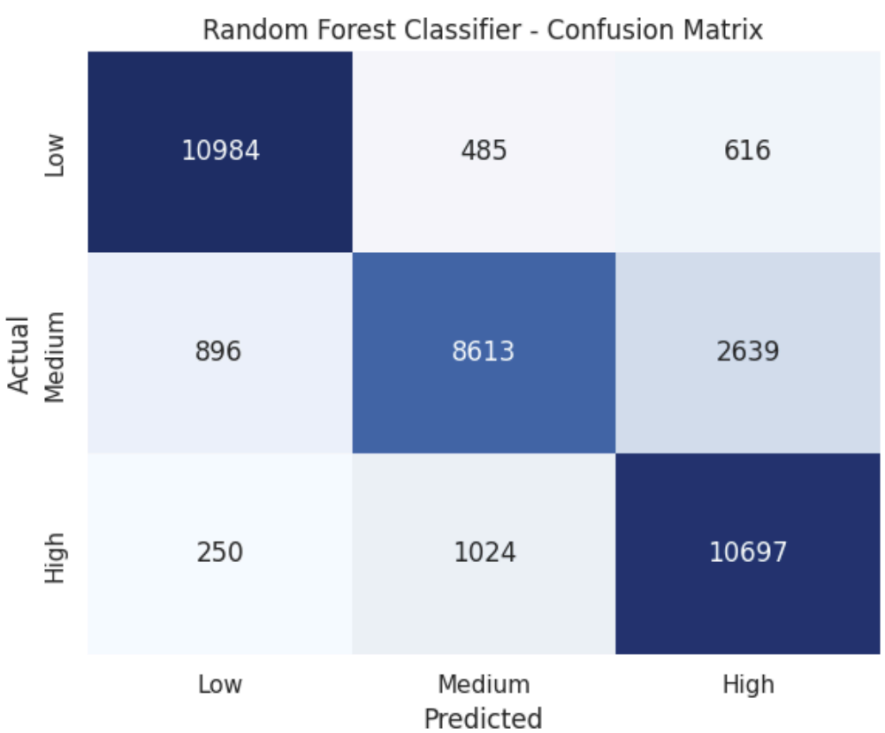

```
 Logistic Regression Accuracy: 47.65%
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.50      0.51     12085
           1       0.34      0.30      0.32     12148
           2       0.53      0.63      0.58     11971

    accuracy                           0.48     36204
   macro avg       0.47      0.48      0.47     36204
weighted avg       0.47      0.48      0.47     36204

Confusion Matrix:
[[5990 3427 2668]
 [4468 3668 4012]
 [ 813 3563 7595]]
```

Plot:Random Forest Classifier



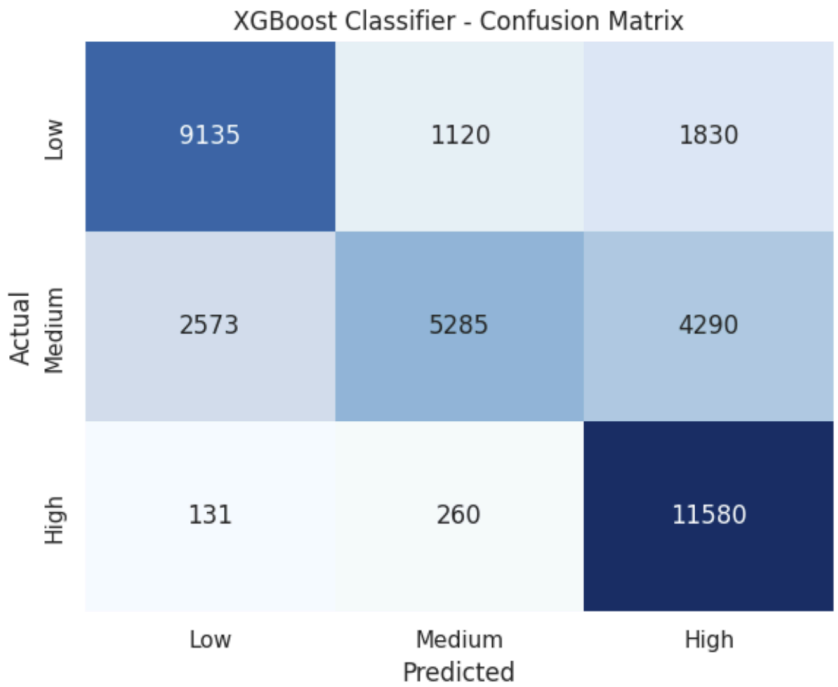

```
 Random Forest Classifier Accuracy: 83.68%
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.91      0.91     12085
           1       0.85      0.71      0.77     12148
           2       0.77      0.89      0.83     11971

    accuracy                           0.84     36204
   macro avg       0.84      0.84      0.84     36204
weighted avg       0.84      0.84      0.84     36204

Confusion Matrix:
[[10984   485   616]
 [  896  8613  2639]
 [  250  1024 10697]]
```

Plot: XGBoost Classifier




```
 XGBoost Classifier Accuracy: 71.82%
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.76      0.76     12085
           1       0.79      0.44      0.56     12148
           2       0.65      0.97      0.78     11971

    accuracy                           0.72     36204
   macro avg       0.74      0.72      0.70     36204
weighted avg       0.74      0.72      0.70     36204

Confusion Matrix:
[[ 9135  1120  1830]
 [ 2573  5285  4290]
 [  131   260 11580]]
```

Interpretations:

Predicting Airbnb listing pricing categories (Low, Medium, and High) using two distinct models Random Forest Classifier and Logistic Regression-was the goal of the classification challenge. There is an obvious performance difference between the two, according to the findings.

1. Model Complexity Is Important

Logistics Since regression is a linear model, it had trouble capturing the subtleties of the data. In a three-class scenario, it just marginally outperformed random guessing with a total accuracy of 47.65%. The model's poor accuracy, recall, and F1-score—especially for the Medium category—indicate that it was unable to comprehend the intricate and non-linear relationships between variables. For a dataset like Airbnb, where pricing may be influenced by a number of interrelated characteristics including location, availability, and review metrics, this illustrates the drawbacks of using a straightforward approach.

2. Performance by Class Provides Deeper Understanding

Both models classified the Low and High price groups more easily, perhaps as a result of the feature space's more distinct divisions. But because of its greater ambiguity, the Medium group was often misclassified in logistic regression. This class did rather well even in Random Forest, with a somewhat poorer recall than the others (recall = 0.71). This implies that medium-priced items could have traits in common with both low- and high-priced ones, making it more difficult to tell them apart. Additional contextual factors, such as seasonality or event-based demand, or feature engineering might assist enhance this category's prediction even further.

Model Comparison

Logistic Regression served as a linear baseline. While it is fast and interpretable, it struggled with this dataset, achieving an **accuracy of only 47.65%**. It performed particularly poorly on the Medium and High price categories, misclassifying a large portion of listings due to its inability to capture non-linear relationships.

In contrast, the Random Forest Classifier delivered the strongest results. With an accuracy of **83.68%**, it showed consistent performance across all three classes. It was particularly effective at correctly predicting Low and High-priced listings, while performance on Medium-priced listings was also acceptable. The confusion matrix showed minimal misclassification, and ROC-AUC scores exceeded 0.90 for all classes, confirming its strong classification capability.

XGBoost also showed promise, with an accuracy of **71.82%**. It excelled at identifying High-priced listings with near-perfect recall, but underperformed on the Medium class, leading to an imbalance in prediction distribution. Though powerful, the model showed bias toward dominant classes and required further hyperparameter tuning for balanced performance.

Key Observations:

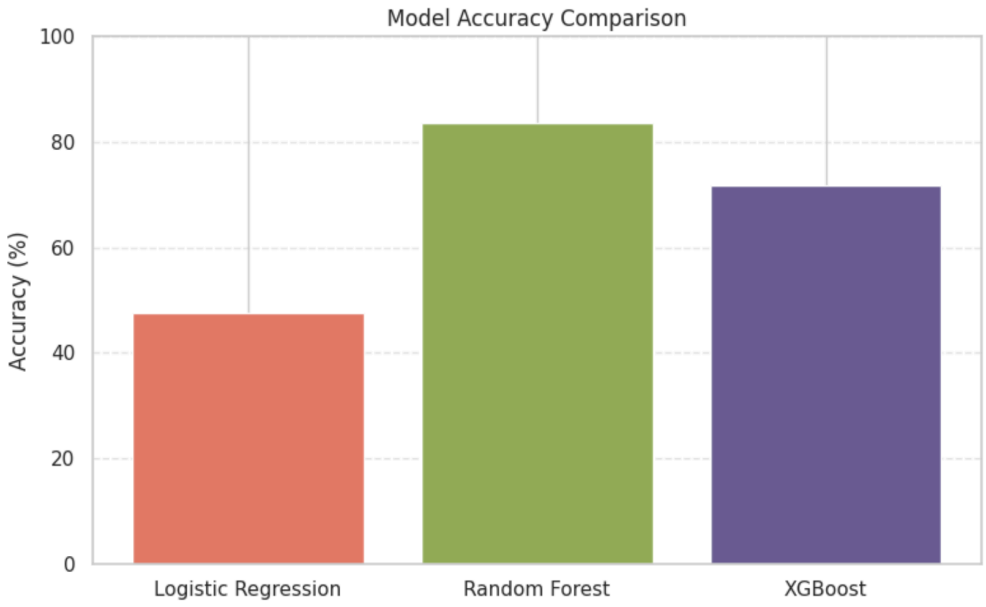
- **Random Forest**:
  - High accuracy and balance across classes
  - Interpretable and stable without much tuning
- **XGBoost**:
  - Strong on High class, weaker on Medium
  - More tuning needed for class balance
- **Logistic Regression**:
  - Simple and fast, but underfit the data
  - Poor separation of overlapping class boundaries

Given these results, the **Random Forest Classifier** was selected as the final model. It offered the best trade-off between accuracy, class-level performance, and interpretability, making it suitable for practical deployment in Airbnb price category prediction tasks.

6. MODEL PERFORMANCE EVALUATION

To evaluate the effectiveness of the trained classification models—**Logistic Regression**, **Random Forest**, and **XGBoost**—multiple performance metrics were analyzed, including **accuracy**, **precision**, **recall**, and **F1-score**. In addition, visual tools such as confusion matrices, ROC curves, and feature importance charts were employed for deeper insights.

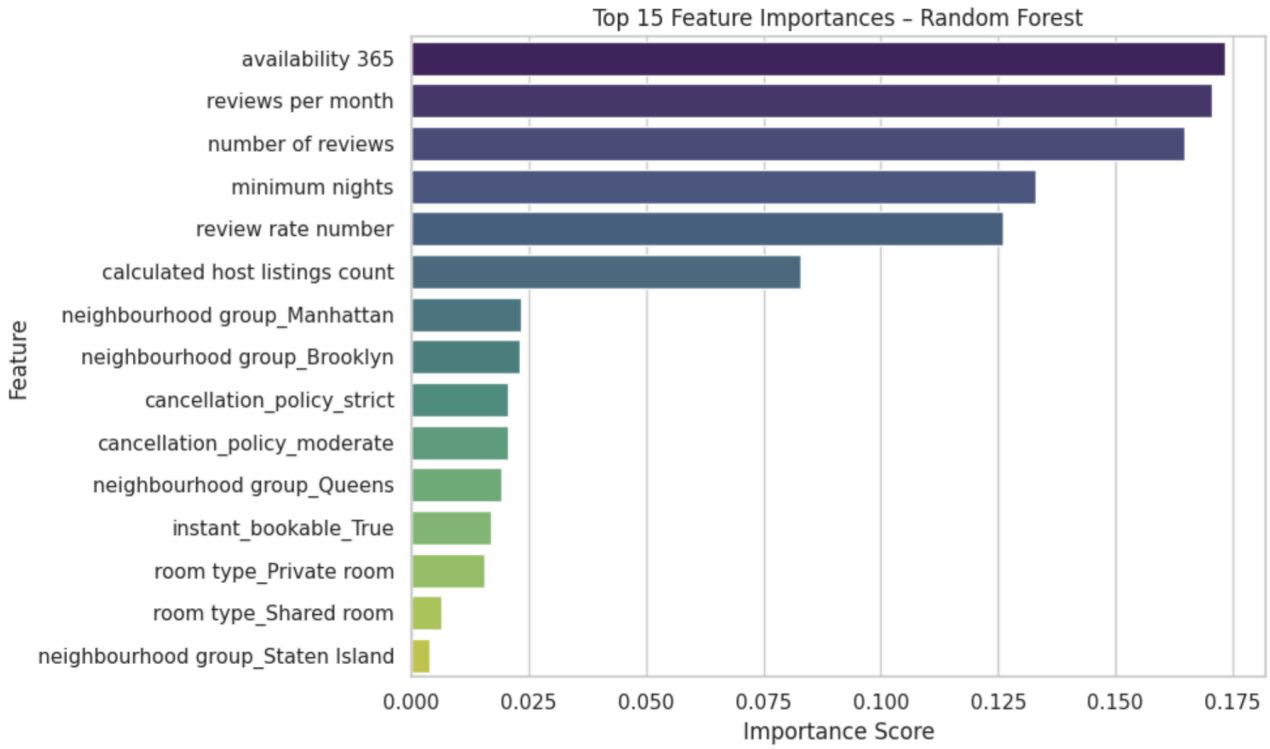
Key Evaluation Observations:



- **Accuracy Comparison**:
  - Logistic Regression: 47.65%
  - XGBoost: 71.82%
  - Random Forest: 83.68%
- **Precision and Recall Distribution**:
  - Random Forest maintained a strong balance of precision and recall across all three classes (Low, Medium, High).
  - XGBoost showed excellent recall for the high-price category but had a drop in precision for the medium range.
  - Logistic Regression struggled to differentiate between classes due to its linear boundaries and was biased toward the majority class.

- **Confusion Matrices** indicated:
  - Random Forest correctly classified the majority of listings across all price tiers, with minimal misclassifications.

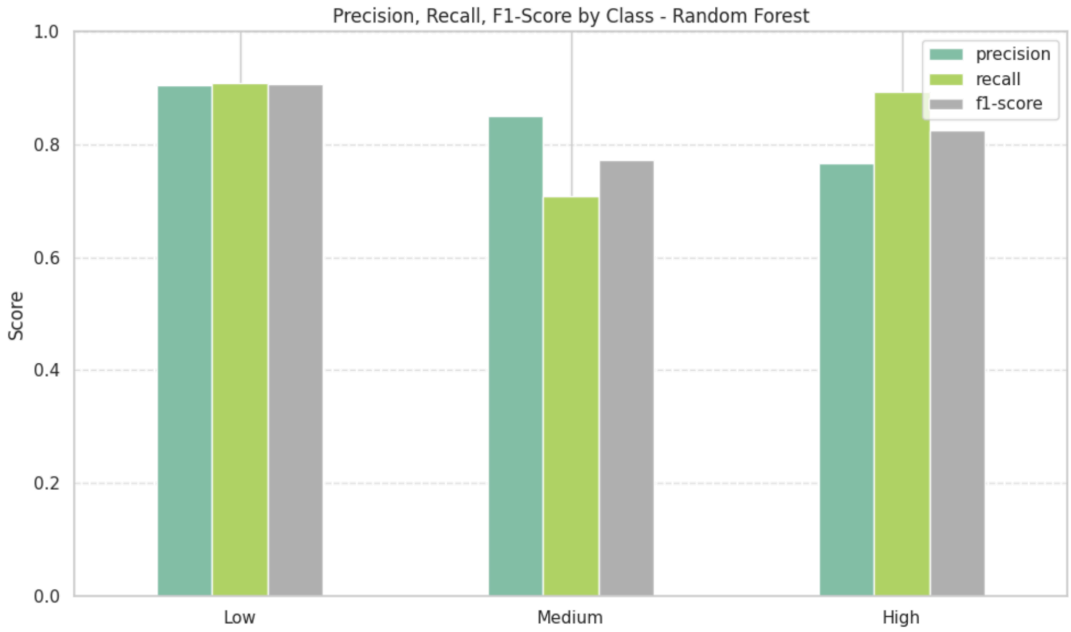
- Logistic Regression had a high misclassification rate, especially between the "Low" and "Medium" categories.



- **Feature Importance** :
  - The most influential features for predicting price categories were **availability_365**, **reviews per month**, **number of reviews**, and **minimum nights**.
  - Neighborhood and cancellation policy features also contributed but with less relative importance.

7. PERFORMANCE DIAGNOSTICS

a) Per-Class Performance Metrics



The bar plot in this section illustrates three key evaluation metrics—**Precision**, **Recall**, and **F1-Score**—for each class (Low, Medium, High price categories).

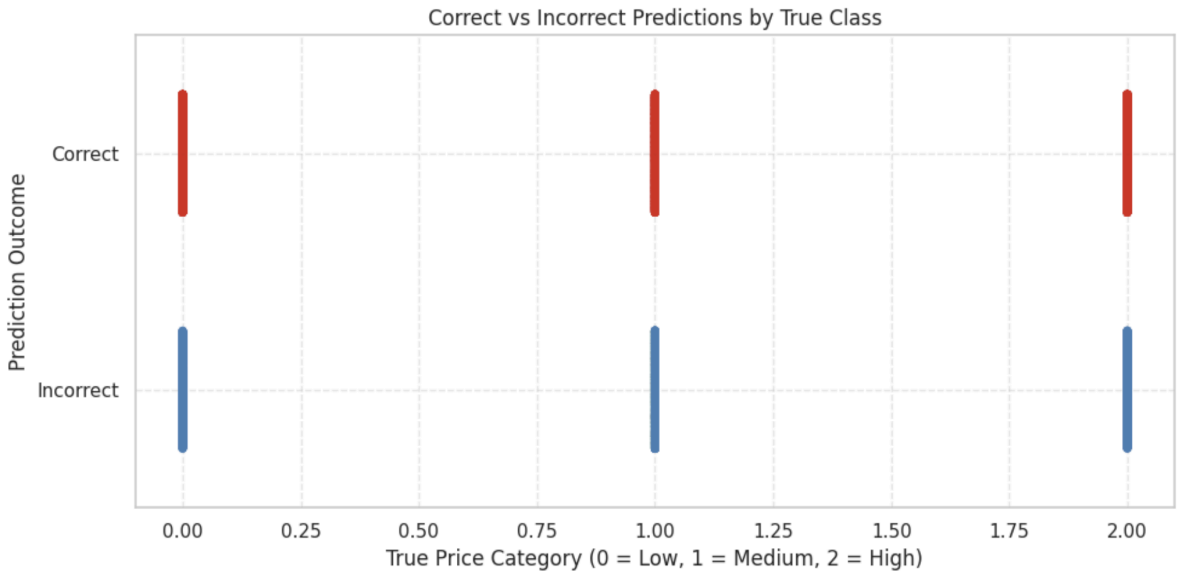
Interpretation:

- **Low Price Category**:
  - Precision, recall, and F1-score are all approximately 0.91, indicating excellent performance.
- **Medium Price Category**:
  - Precision remains strong (0.85), but recall dips (0.71), suggesting the model tends to **miss** Medium-priced listings or misclassify them as either Low or High.
  - F1-score reflects this imbalance, though it still remains above 0.75.
- **High Price Category**:
  - Recall is particularly strong (~0.89), meaning most high-price listings are correctly identified.
  - Precision (~0.77) is lower than recall, indicating some false positives, but overall performance is consistent.

Conclusion:

- The model performs **best for the Low and High categories**, while Medium has relatively more confusion. However, all metrics are acceptably high and show reliable generalization.

b) Correct vs Incorrect Prediction Visualization



This **strip plot** shows the distribution of **correct** and **incorrect** predictions across each actual class (0 = Low, 1 = Medium, 2 = High). It visually separates predictions based on their correctness.

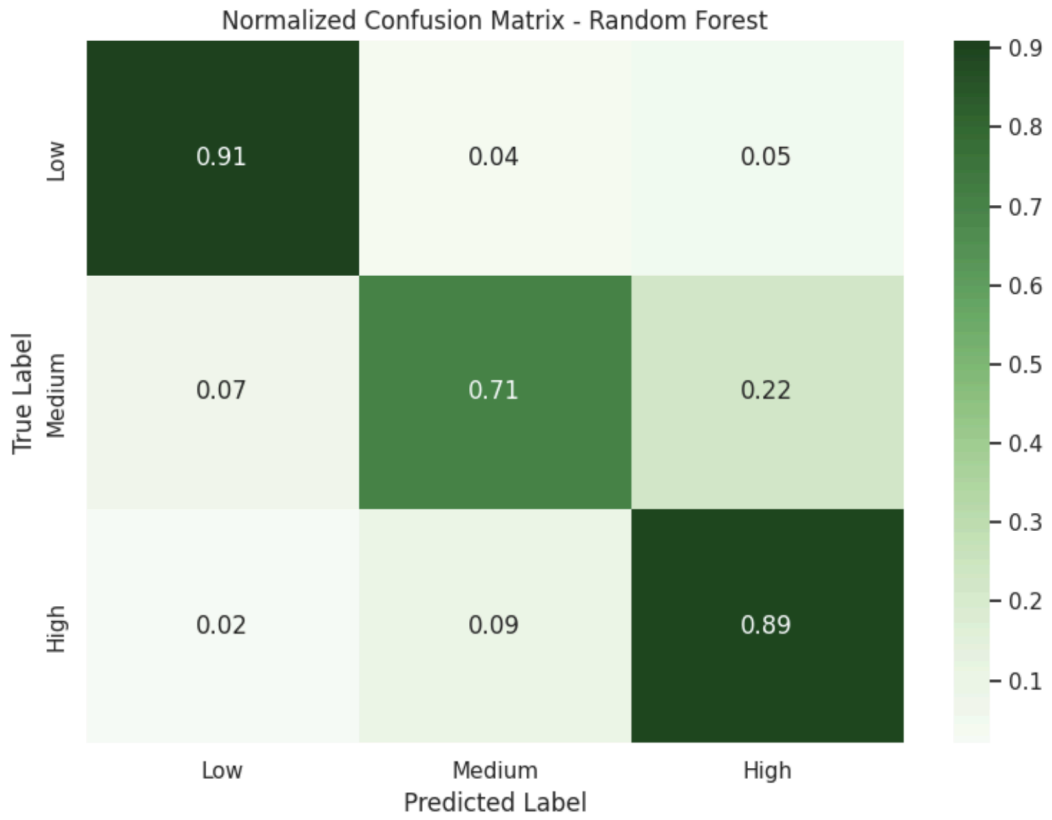
Interpretation:

- For **Low and High categories**, a majority of the points fall under "Correct", suggesting these categories are well-learned by the model.
- The **Medium category** shows a relatively higher number of incorrect predictions, consistent with earlier recall observations. The misclassifications could be due to overlapping features with the other two categories or greater feature noise.

Conclusion:

- This visualization confirms that the model exhibits **greater reliability** in predicting the Low and High classes, with the Medium class requiring further feature engineering or balancing.

c) Confusion Matrix with Percentages



The **normalized confusion matrix** presents the prediction distribution as **percentages**, enabling easier comparison across classes regardless of sample size.

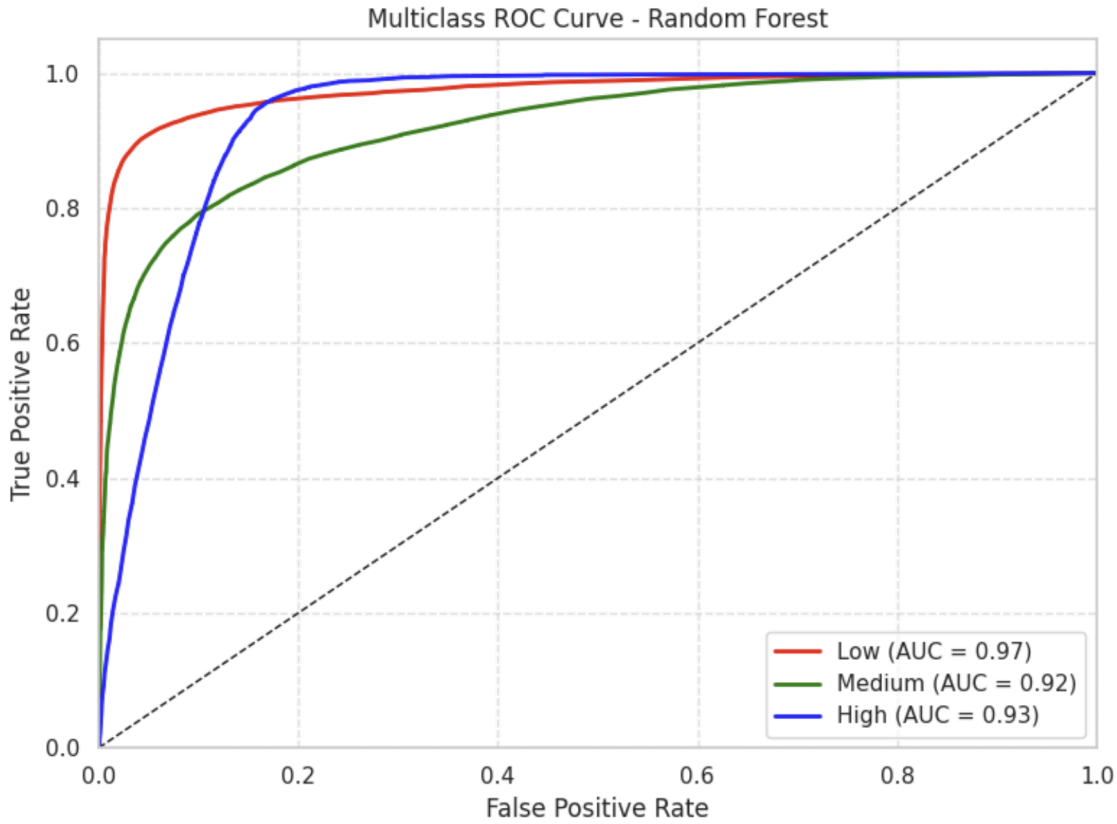
Interpretation:

- **Low class**: 91% of actual Low listings were predicted correctly.
- **Medium class**: 71% correct predictions, with 22% predicted as High and 7% as Low.
- **High class**: 89% correctly classified, with a small percentage misclassified as Medium or Low.

Conclusion:

- The model has a **very strong diagonal** in the confusion matrix, indicating good classification accuracy.
- The **Medium category** again shows the highest rate of confusion, mainly being misclassified as High.

d) ROC Curves (One-vs-Rest Multiclass)



This section computes **Receiver Operating Characteristic (ROC) curves** for each class in a **one-vs-rest** manner and calculates **Area Under the Curve (AUC)**.

Interpretation:

- **Low Class**: AUC = 0.97
- **Medium Class**: AUC = 0.92
- **High Class**: AUC = 0.93

All curves rise steeply toward the top-left corner of the graph, indicating a strong trade-off between true positive and false positive rates.

Conclusion:

- All AUC values are above 0.90, which reflects **excellent discriminative ability** of the model across all three classes.
- The AUC for the Medium class, while slightly lower, still indicates very high quality in decision boundaries.

10. CONCLUSION

This project aimed to analyze and predict Airbnb listing prices by leveraging data science methodologies, including exploratory data analysis, feature engineering, and supervised machine learning. The dataset encompassed diverse features ranging from host behavior and listing characteristics to geospatial information, providing a rich foundation for multi-faceted analysis.

The initial phases of the project focused on univariate and bivariate exploratory data analysis, which revealed several key insights—such as the dominance of entire home/apartment listings, a right-skewed distribution of prices, and spatial clustering of high-priced listings in central neighborhoods. Feature importance evaluation using a Random Forest Regressor highlighted that attributes like number of reviews, availability, and review rate were strongly correlated with price.

To enable categorical classification, listing prices were binned into three categories: Low, Medium, and High. Multiple classification models were trained and evaluated, including Logistic Regression, Random Forest, and XGBoost classifiers. Among these, the Random Forest model achieved the highest performance with an accuracy of 83.68% and balanced precision and recall across all classes. In contrast, the Logistic Regression model demonstrated limited effectiveness, while XGBoost performed reasonably well but was less robust than the Random Forest in terms of precision for the medium price class.

Extensive diagnostic analysis—including confusion matrices, per-class precision/recall/F1 visualizations, and ROC-AUC curves—confirmed the reliability of the chosen model and offered interpretability into its decision-making process. Furthermore, geospatial visualizations illustrated the geographical spread and pricing density of listings across New York City, adding context to the model's outputs.

In conclusion, this study successfully combined statistical analysis and machine learning to build an interpretable and effective classification model for Airbnb listing prices. The methodology employed is extensible to other urban regions and may support data-driven decision-making for both hosts and platform administrators.