Exploratory Data Analysis (EDA) on New York City Airbnb Listings Dataset

Overview

This project performs Exploratory Data Analysis (EDA) on a dataset containing Airbnb listings in New York City. The dataset contains 48,895 rows and 16 columns with various features like listing ID, host details, geographical information, price, and availability. The goal of this analysis is to uncover patterns, insights, and relationships within the dataset, while identifying any anomalies or trends that could help stakeholders understand the data better.

Dataset Overview

The dataset represents Airbnb listings in New York City and contains the following 16 columns:

1. **id**: Unique identifier for each listing (integer).
2. **name**: Name of the Airbnb listing (string, some missing values).
3. **host_id**: Unique identifier for the host (integer).
4. **host_name**: Name of the host (string, some missing values).
5. **neighbourhood_group**: The borough/group in NYC (string).
6. **neighbourhood**: Neighborhood within the borough (string).
7. **latitude**: Latitude of the listing (float).
8. **longitude**: Longitude of the listing (float).
9. **room_type**: Type of room available (string).
10. **price**: Price per night (integer).
11. **minimum_nights**: Minimum number of nights required to book (integer).
12. **number_of_reviews**: Total number of reviews (integer).
13. **last_review**: Date of the last review (string, some missing values).

14. **reviews_per_month**: Average number of reviews per month (float, some missing values).
15. **calculated_host_listings_count**: Total number of listings by the host (integer).
16. **availability_365**: Number of days the listing is available in a year (integer).

Steps in Exploratory Data Analysis (EDA)

1. Data Cleaning and Handling Missing Values

Objective: Clean the dataset by addressing missing or incorrect values.

Action Steps:

- Identify columns with missing values (e.g., name, host_name, last_review, reviews_per_month).
- Handle missing values through appropriate strategies:
    - Drop rows if missing values are not essential.
    - Impute missing values if necessary (e.g., imputing mean values for reviews_per_month).

Example Code:

python

Copy code

```
# Checking for missing values

missing_values = df.isnull().sum()

print(missing_values)


# Dropping rows with missing values in essential columns

df_cleaned = df.dropna(subset=['name', 'host_name'])
```

Imputing missing values in 'reviews_per_month'

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

2. Data Type Conversion

Objective: Ensure that all columns have the correct data types to facilitate analysis.

Action Steps:

- Convert last_review to a datetime type for date-based operations.

Example Code:

python

Copy code

df['last_review'] = pd.to_datetime(df['last_review'], errors='coerce')

3. Descriptive Statistics

Objective: Get an overall sense of the data through summary statistics.

Action Steps:

- Generate summary statistics for numerical columns (e.g., price, minimum_nights, number_of_reviews).
- Examine distributions of categorical variables (e.g., neighbourhood_group, room_type).

Example Code:

python

Copy code

Summary statistics for numerical columns

print(df.describe())

Distribution of categorical variables

print(df['neighbourhood_group'].value_counts())

print(df['room_type'].value_counts())

4. Data Visualization

Objective: Create visual representations of the data to better understand its structure and distribution.

Action Steps:

- Use histograms and boxplots to visualize the distribution of price and minimum_nights.
- Use bar charts or count plots to visualize room_type and neighbourhood_group.

Example Code:

python

Copy code

Price distribution

sns.histplot(df['price'], bins=50, kde=True)

plt.title('Price Distribution')

plt.show()

Count plot of room types

sns.countplot(x='room_type', data=df)

plt.title('Room Types in NYC Listings')

plt.show()

5. Geographical Analysis

Objective: Explore the geographical distribution of the listings.

Action Steps:

- Plot listings on a scatter map based on latitude and longitude.

- Create a density map to visualize concentrations of listings.

Example Code:

python

Copy code

```
# Scatter plot of listings based on location

sns.scatterplot(x='longitude', y='latitude', data=df, hue='neighbourhood_group', palette='Set1')

plt.title('Geographical Distribution of Listings')

plt.show()


# Density plot for geographical data

sns.kdeplot(x=df['longitude'], y=df['latitude'], fill=True, cmap='viridis')

plt.title('Density of Listings in NYC')

plt.show()
```

6. Outlier Detection

Objective: Identify extreme values that may indicate outliers in the dataset.

Action Steps:

- Use the Interquartile Range (IQR) method to detect outliers in price and minimum_nights.

Example Code:

python

Copy code

```
# Outlier detection using IQR

def detect_outliers(df, column):

    Q1 = df[column].quantile(0.25)
```

```
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] < lower_bound) | (df[column] > upper_bound)]

# Example for price
outliers_price = detect_outliers(df, 'price')
print(outliers_price)
```

7. Time Series Analysis

Objective: Analyze the trends in customer reviews over time.

Action Steps:

- Use the last_review column to analyze review trends over time.
- Aggregate data to understand how review frequency has changed over the months.

Example Code:

python

Copy code

```
# Time series analysis for reviews
df['year_month'] = df['last_review'].dt.to_period('M')
monthly_reviews = df.groupby('year_month')['reviews_per_month'].sum().reset_index()

# Plotting the trend of reviews per month
sns.lineplot(x='year_month', y='reviews_per_month', data=monthly_reviews)
```

```
plt.xticks(rotation=45)

plt.title('Trend of Reviews Per Month')

plt.show()
```

Conclusion

The EDA process on this New York City Airbnb listings dataset has provided insights into the structure of the data and revealed key patterns, such as price distributions, geographical listing trends, and customer review dynamics. The analysis helped identify missing values, outliers, and review trends over time, and it sets the foundation for further data-driven insights.