# Airbnb Case Study - Methodology Document

### Firstly, before starting our analysis on Raw data, we must clean it on Jupiter Notebook by performing Data Cleaning, EDA.

# Problem Statement

Working as a data analyst at Airbnb, for the past few months, Airbnb has seen a major decline in revenue. Now that the covid restrictions have started lifting and people have started to travel more, Airbnb wants to make sure that it is fully prepared for this change.

The different leaders at Airbnb want to understand some important insights based on various attributes in the dataset to increase the revenue.

### So, before proceeding further with our Analysis, let's Clean the Data that we have received.

**Disabling Python Warnings**

In [1]:

*#Ignore Warnings*

**import** warnings **as** war war.filterwarnings('ignore')

**Importing Libraries**

In [2]:

*# Importing all important libraries*

**import** numpy **as** np **import** pandas **as** pd **import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**import** os

### Changing current working directory

Changing current working directory to where our data is saved

In [3]:

os.chdir('C:\\Users\\DELL\\Downloads\\UpGrad\\Tableau & Power\_BI\\Dataset'

## Importing required Dataset for EDA

In [6]:

*# Loading the required dataset and Checking its top 5 rows*

airbnb\_df **=** pd.read\_csv('AB\_NYC\_2019.csv') airbnb\_df.head()

Out[6]: **id name host\_id host\_name neighbourhood\_group neighbourhood latit**

**0** 2539

Clean & quiet apt home by the

park

2787 John Brooklyn Kensington 40.64

Skylit Midtown

**1** 2595

Castle 2845 Jennifer Manhattan Midtown 40.75

**2** 3647

THE VILLAGE

OF HARLEM....NEW

YORK !

4632 Elisabeth Manhattan Harlem 40.80

**3** 3831

Cozy Entire

Floor of Brownstone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4869 | LisaRoxanne | Brooklyn | Clinton Hill | 40.68 |
| 7192 | Laura | Manhattan | East Harlem | 40.79 |

**4** 5022

Entire Apt: Spacious Studio/Loft by central park



## Checking various stats of our Dataset

In [7]:

*# Checking Shape of our Dataset*

airbnb\_df.shape

Out[7]: (48895, 16)

**Insight,**

**We can see, there are more than 48K Rows and 16 Columns in our Dataset.**

In [8]:

*# Checking information about the Dataset and its Datatype.*

airbnb\_df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 48895 entries, 0 to 48894 Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | id | 48895 | non-null | int64 |
| 1 | name | 48879 | non-null | object |
| 2 | host\_id | 48895 | non-null | int64 |
| 3 | host\_name | 48874 | non-null | object |
| 4 | neighbourhood\_group | 48895 | non-null | object |
| 5 | neighbourhood | 48895 | non-null | object |
| 6 | latitude | 48895 | non-null | float64 |
| 7 | longitude | 48895 | non-null | float64 |
| 8 | room\_type | 48895 | non-null | object |
| 9 | price | 48895 | non-null | int64 |
| 10 | minimum\_nights | 48895 | non-null | int64 |
| 11 | number\_of\_reviews | 48895 | non-null | int64 |
| 12 | last\_review | 38843 | non-null | object |
| 13 | reviews\_per\_month | 38843 | non-null | float64 |
| 14 | calculated\_host\_listings\_count | 48895 | non-null | int64 |
| 15 | availability\_365 | 48895 | non-null | int64 |

dtypes: float64(3), int64(7), object(6) memory usage: 6.0+ MB

In [9]:

*# lets check the Descriptive statistical summary of all the columns in our*

airbnb\_df.describe(include**=**'all')

Out[9]:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **id** | **name** | **host\_id** | **host\_name** | **neighbourhood\_group** | **neighbourho** |
| **count** | 4.889500e+04 | 48879 | 4.889500e+04 | 48874 | 48895 | 48 |
| **unique** | NaN | 47896 | NaN | 11452 | 5 |  |
| **top** | NaN | Hillside | NaN | Michael | Manhattan | Williamsb |
|  |  | Hotel |  |  |  |  |
| **freq** | NaN | 18 | NaN | 417 | 21661 | 3 |
| **mean** | 1.901714e+07 | NaN | 6.762001e+07 | NaN | NaN | N |
| **std** | 1.098311e+07 | NaN | 7.861097e+07 | NaN | NaN | N |
| **min** | 2.539000e+03 | NaN | 2.438000e+03 | NaN | NaN | N |
| **25%** | 9.471945e+06 | NaN | 7.822033e+06 | NaN | NaN | N |
| **50%** | 1.967728e+07 | NaN | 3.079382e+07 | NaN | NaN | N |
| **75%** | 2.915218e+07 | NaN | 1.074344e+08 | NaN | NaN | N |
| **max** | 3.648724e+07 | NaN | 2.743213e+08 | NaN | NaN | N |



# Data pre-processing

### Removing case mis-match

In [10]:

*# Creating Function to convert alphabetic characters to uppercase*

**def** to\_uppercase\_alpha(val):

**if** isinstance(val, str):

**return** ''.join([char.upper() **if** char.isalpha() **else** char **for** char

**return** val

In [11]:

*# Removing case mis-match by Standardizing all the values of string type t*

airbnb\_df **=** airbnb\_df.map(**lambda** x: to\_uppercase\_alpha(x) **if** isinstance(x,

### Assigning Correct Datatype

Changing Datatype of last\_review to 'Datetime' Datatype, id and host\_id to 'Object' Datatype.

In [12]:

*# Changing Datatype of `last\_review` to Datetime Datatype*

airbnb\_df['last\_review'] **=** pd.to\_datetime(airbnb\_df['last\_review'])

In [13]:

*# Changing Datatype of `id` and `host\_id` to Object Datatype* airbnb\_df['id'] **=** airbnb\_df['id'].astype(object) airbnb\_df['host\_id'] **=** airbnb\_df['host\_id'].astype(object)

In [14]:

*# Checking Info() for Changes*

airbnb\_df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 48895 entries, 0 to 48894 Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | id | 48895 | non-null | object |
| 1 | name | 48879 | non-null | object |
| 2 | host\_id | 48895 | non-null | object |
| 3 | host\_name | 48874 | non-null | object |
| 4 | neighbourhood\_group | 48895 | non-null | object |
| 5 | neighbourhood | 48895 | non-null | object |
| 6 | latitude | 48895 | non-null | float64 |
| 7 | longitude | 48895 | non-null | float64 |
| 8 | room\_type | 48895 | non-null | object |
| 9 | price | 48895 | non-null | int64 |
| 10 | minimum\_nights | 48895 | non-null | int64 |
| 11 | number\_of\_reviews | 48895 | non-null | int64 |
| 12 | last\_review | 38843 | non-null | datetime64[ns] |
| 13 | reviews\_per\_month | 38843 | non-null | float64 |
| 14 | calculated\_host\_listings\_count | 48895 | non-null | int64 |
| 15 | availability\_365 | 48895 | non-null | int64 |

dtypes: datetime64[ns](1), float64(3), int64(5), object(7) memory usage: 6.0+ MB

In [ ]:

# Data Cleaning

### Checking for Duplicates Rows Entires

In [15]:

*# Checking for Duplicate Entires in our Dataset*

airbnb\_df.duplicated().sum()

Out[15]: 0

**Insight,**

There are No Duplicate entries or rows in our database.

### Checking for Missing Values in our Dataset

In [16]:

*# Checking null values in the Dataset*

airbnb\_df.isnull().sum()

|  |  |  |
| --- | --- | --- |
| Out[16]: | id | 0 |
|  | name | 16 |
|  | host\_id | 0 |
|  | host\_name | 21 |
|  | neighbourhood\_group | 0 |
|  | neighbourhood | 0 |
|  | latitude | 0 |
|  | longitude | 0 |
|  | room\_type | 0 |
|  | price | 0 |
|  | minimum\_nights | 0 |
|  | number\_of\_reviews | 0 |
|  | last\_review | 10052 |
|  | reviews\_per\_month | 10052 |
|  | calculated\_host\_listings\_count | 0 |
|  | availability\_365 | 0 |
|  | dtype: int64 |  |

In [17]:

*# Checking percentage of null values*

airbnb\_df.isna().sum()**/**len(airbnb\_df)**\***100

|  |  |  |
| --- | --- | --- |
| Out[17]: | id | 0.000000 |
|  | name | 0.032723 |
|  | host\_id | 0.000000 |
|  | host\_name | 0.042949 |
|  | neighbourhood\_group | 0.000000 |
|  | neighbourhood | 0.000000 |
|  | latitude | 0.000000 |
|  | longitude | 0.000000 |
|  | room\_type | 0.000000 |
|  | price | 0.000000 |
|  | minimum\_nights | 0.000000 |
|  | number\_of\_reviews | 0.000000 |
|  | last\_review | 20.558339 |
|  | reviews\_per\_month | 20.558339 |
|  | calculated\_host\_listings\_count | 0.000000 |
|  | availability\_365 | 0.000000 |
|  | dtype: float64 |  |

**Insight,**

We can see, There are total 4 cloumns having Missing Values last\_review and reviews\_per\_month cloumns has around 20.55% missing values i.e. ~10K null values

host\_name column has around 0.04% missing values i.e. 21 null values

name column has around 0.03% missing values i.e. 16 null values

### Treating Missing Values of reviews\_per\_month

To deal with missing values in reviews\_per\_month , Firstly, lets check its relation with

number\_of\_reviews

In [18]:

*# Checking relation of `number\_of\_reviews` with `reviews\_per\_month`*

airbnb\_df.loc[airbnb\_df['reviews\_per\_month'].isna() , 'number\_of\_reviews'

Out[18]: number\_of\_reviews

0 10052

Name: count, dtype: int64

**Insight,**

As we can see, reviews\_per\_month is related to

number\_of\_reviews ,

Thus we can impute missing values of reviews\_per\_month with '0' (zreo) as number\_of\_reviews is also '0' (zero)

In [19]:

*# Imputing 'reviews\_per\_month' by zero (0))*

airbnb\_df['reviews\_per\_month'].fillna('0',inplace**=True**)

### Treating Missing Values of last\_review

To deal with missing values in last\_review , which has high percentage of null data (around 20.55%), we can drop this column.

This decision was made because form this column we will not be able to get much details, as it only shows the last date on which any property have received the review. By Imputing any value, our data might show wrong analysis.

In [20]:

*# Droping 'last\_review'*

airbnb\_df.drop(['last\_review'], axis**=**1 , inplace**=True**)

### Treating Missing Values of name and host\_name

Through Data definition, we can see id column is associated with name column and host\_id is associated with host\_name . Thus, to deal with missing values in name and host\_name , we can drop both the columns, as we can use id and host\_id respectively in their place for analysis.

In [21]:

*# Droping Column `name` and `host\_name` as they are not required for our a*

airbnb\_df.drop(['name', 'host\_name'], axis**=**1 , inplace**=True**)

### Checking changes in our Dataset

In [22]:

*# Checking missing values for changes*

airbnb\_df.isna().sum()

Out[22]: id 0

host\_id 0

neighbourhood\_group 0

neighbourhood 0

latitude 0

longitude 0

room\_type 0

price 0

minimum\_nights 0

number\_of\_reviews 0

reviews\_per\_month 0

calculated\_host\_listings\_count 0

availability\_365 0

dtype: int64

In [23]:

*# Checking Shape of our cleaned Dataset*

airbnb\_df.shape

Out[23]: (48895, 13)

# Exporting Cleaned Dataset for Visualization

Now, lets export the cleaned Dataset for visualization with powerful tools like Tableau and PowerBI

In [24]:

*# Exporting cleaned Dataset*

airbnb\_df.to\_csv("AB\_NYC\_2019\_Cleaned.csv" , index**=False** )

**Visualization Overview**

### We have performed various Market Overview Analysis:

### By checking listings Distribution: Highlight the concentration in Manhattan and potential growth in Brooklyn and Queens.

### Checking Distribution of Room Types: Emphasize user preferences for Entire Homes/Apartments.

### Then, Pricing Analysis:

### By checking price disparity and discussing the variation in prices across neighborhoods and room types.

### Finding opportunities in affordable areas and potential to attract a broader audience by focusing on lower-cost listings.

### User Experience Insights:

### High-Demand Areas: Focus on enhancing user experience in neighbourhoods like Williamsburg and Harlem.

### Minimum Stay Adjustments: Consider revising minimum stay requirements to cater to short-term travellers.

### Operational Strategies:

### Focus on Premium Services in Manhattan: Maintain high standards to justify premium pricing.

### Expand in Brooklyn and Queens: Increase marketing efforts and potentially adjust service offerings to attract more bookings in these areas

### Recommendations to improve Revenue:

### Neighbourhood-Specific Campaigns: Target marketing and promotional efforts in high-demand neighbourhoods.

### Enhance User Experience: Invest in improving the user experience in areas with high listing concentrations.

### Operational Efficiency: Streamline operations in Manhattan while exploring new opportunities in less saturated areas.