Lab 8: Define and Solve an ML Problem of Your Choosing

In [1]:
```python
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In this lab assignment, you will follow the machine learning life cycle and implement a model to solve a machine learning problem of your choosing. You will select a data set and choose a predictive problem that the data set supports. You will then inspect the data with your problem in mind and begin to formulate a project plan. You will then implement the machine learning project plan.

You will complete the following tasks:

1. Build Your DataFrame
2. Define Your ML Problem
3. Perform exploratory data analysis to understand your data.
4. Define Your Project Plan
5. Implement Your Project Plan:
    - Prepare your data for your model.
    - Fit your model to the training data and evaluate your model.
    - Improve your model's performance.

Part 1: Build Your DataFrame

You will have the option to choose one of four data sets that you have worked with in this program:

- The "census" data set that contains Census information from 1994: `censusData.csv`
- Airbnb NYC "listings" data set: `airbnbListingsData.csv`
- World Happiness Report (WHR) data set: `WHR2018Chapter2OnlineData.csv`
- Book Review data set: `bookReviewsData.csv`

Note that these are variations of the data sets that you have worked with in this program. For example, some do not include some of the preprocessing necessary for specific models.

Load a Data Set and Save it as a Pandas DataFrame

The code cell below contains filenames (path + filename) for each of the four data sets available to you.

Task: In the code cell below, use the same method you have been using to load the data using `pd.read_csv()` and save it to DataFrame `df`.

You can load each file as a new DataFrame to inspect the data before choosing your data set.

In [2]:
```
# File names of the four data sets
adultDataSet_filename = os.path.join(os.getcwd(), "data", "censusData.csv")
airbnbDataSet_filename = os.path.join(os.getcwd(), "data", "airbnbListingsData.csv"
WHRDataSet_filename = os.path.join(os.getcwd(), "data", "WHR2018Chapter2OnlineData.
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsData.csv


df = pd.read_csv(airbnbDataSet_filename)
# print(df.head())
print(df.columns)
# print(df['review_scores_rating'])
# print(df.head(2))
```

```
Index(['name', 'description', 'neighborhood_overview', 'host_name',
       'host_location', 'host_about', 'host_response_rate',
       'host_acceptance_rate', 'host_is_superhost', 'host_listings_count',
       'host_total_listings_count', 'host_has_profile_pic',
       'host_identity_verified', 'neighbourhood_group_cleansed', 'room_type',
       'accommodates', 'bathrooms', 'bedrooms', 'beds', 'amenities', 'price',
       'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
       'maximum_minimum_nights', 'minimum_maximum_nights',
       'maximum_maximum_nights', 'minimum_nights_avg_ntm',
       'maximum_nights_avg_ntm', 'has_availability', 'availability_30',
       'availability_60', 'availability_90', 'availability_365',
       'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d',
       'review_scores_rating', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value', 'instant_bookable',
       'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'n_host_verifications'],
      dtype='object')
```

In [3]:
```
df.head()
```

Out[3]:

	name	description	neighborhood_overview	host_name	host_location	host_abo
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	Jennifer	New York, New York, United States	A New York since 2000! N passion creatin
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	LisaRoxanne	New York, New York, United States	Laid-ba Native Ne York (formerly b coas
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Rebecca	Brooklyn, New York, United States	Rebecca is artist/design and Henoch
3	Large Furnished Room Near B'way	Please don't expect the luxury here just a bas...	Theater district, many restaurants around here.	Shunichi	New York, New York, United States	I used to wo for a financ industry b no
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	MaryEllen	New York, New York, United States	Welcome family life wi my oldest tw away

5 rows × 50 columns

Part 2: Define Your ML Problem

Next you will formulate your ML Problem. In the markdown cell below, answer the following questions:

1. List the data set you have chosen.
2. What will you be predicting? What is the label?
3. Is this a supervised or unsupervised learning problem? Is this a clustering, classification or regression problem? Is it a binary classificaiton or multi-class classifiction problem?
4. What are your features? (note: this list may change after your explore your data)
5. Explain why this is an important problem. In other words, how would a company create value with a model that predicts this label?

My answers:

1. The data set that I have chosen is "airbnbListingsData.csv".
2. I want to predict whether the review scores will be high or low based on multiple features. I will convert the column "review_scores_rating" into "review_scores_rating_high" and "review_scores_rating_low". Every scores that is >= 3.0 is considered high, and it is considered low if it is < 3.0.
3. This is a supervised learning problem. This is a binary classification. I will use binary indicators--I would transform data to binary based on meeting a True/False condition.
4. Current Feature: ['name', 'description', 'neighborhood_overview', 'host_location', 'host_about', 'host_response_rate', 'host_acceptance_rate', 'host_is_superhost', 'host_listings_count', 'host_total_listings_count', 'host_has_profile_pic', 'host_identity_verified', 'neighbourhood_group_cleansed', 'room_type', 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'amenities', 'price', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights', 'maximum_minimum_nights', 'minimum_maximum_nights', 'maximum_maximum_nights', 'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'has_availability', 'availability_30', 'availability_60', 'availability_90', 'availability_365', 'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d', 'review_scores_rating', 'review_scores_cleanliness', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location', 'review_scores_value', 'instant_bookable', 'calculated_host_listings_count', 'calculated_host_listings_count_entire_homes', 'calculated_host_listings_count_private_rooms', 'calculated_host_listings_count_shared_rooms', 'reviews_per_month',]
5. In my opinion, it is an important problem, as if companies want to invest in and host multiple Airbnbs, they would love to predict which Airbnb can potentially receive high scores before buying back that Airbnb and earn profit in the future. The rating is based on different features, such as host_location, host_response_rate, availability_30. Besides that, they can use the model to stimulate a scenario--they would use an example house as a data point to check if it would become a good Airbnb before they actually buy it/ build it in that specific location. The prediction will also give companies incentives to take good care of the property, which are reflected through some features, such as "review_scores_cleanliness", "review_scores_communication", "instant_bookable".

In [4]:
```
df['instant_bookable']
```

```
Out[4]: 0        False
        1        False
        2        False
        3        False
        4        False
                 ...  
        28017     True
        28018    False
        28019     True
        28020    False
        28021     True
        Name: instant_bookable, Length: 28022, dtype: bool
```

```
In [5]: df[28021:28022]
```

Out[5]:

	name	description	neighborhood_overview	host_name	host_location	host_abou
28021	Large, modern, private 1 bedroom in beach condo	Private bedroom on its own floor with very lar...	Beach, surf shop, stop and shop, Dunkin' Donut...	Justine	US	Na

1 rows × 50 columns

Part 3: Understand Your Data

The next step is to perform exploratory data analysis. Inspect and analyze your data set with your machine learning problem in mind. Consider the following as you inspect your data:

1. What data preparation techniques would you like to use? These data preparation techniques may include:
   - addressing missingness, such as replacing missing values with means
   - finding and replacing outliers
   - renaming features and labels
   - finding and replacing outliers
   - performing feature engineering techniques such as one-hot encoding on categorical features
   - selecting appropriate features and removing irrelevant features
   - performing specific data cleaning and preprocessing techniques for an NLP problem
   - addressing class imbalance in your data sample to promote fair AI
2. What machine learning model (or models) you would like to use that is suitable for your predictive problem and data?

- Are there other data preparation techniques that you will need to apply to build a balanced modeling data set for your problem and model? For example, will you need to scale your data?

3. How will you evaluate and improve the model's performance?

    - Are there specific evaluation metrics and methods that are appropriate for your model?

Think of the different techniques you have used to inspect and analyze your data in this course. These include using Pandas to apply data filters, using the Pandas `describe()` method to get insight into key statistics for each column, using the Pandas `dtypes` property to inspect the data type of each column, and using Matplotlib and Seaborn to detect outliers and visualize relationships between features and labels. If you are working on a classification problem, use techniques you have learned to determine if there is class imbalance.

Task: Use the techniques you have learned in this course to inspect and analyze your data. You can import additional packages that you have used in this course that you will need to perform this task.

Note: You can add code cells if needed by going to the **Insert** menu and clicking on **Insert Cell Below** in the drop-drown menu.

Display Summary Statistics by Column

In [6]:
```
df.describe(include = 'all')
```

Out[6]:

	name	description	neighborhood_overview	host_name	host_location	host_about
count	28017	27452	18206	28022	27962	17077
unique	27386	25952	15800	7566	1364	11962
top	Water View King Bed Hotel Room	Welcome to UNTITLED (Adj.) at 3 Freeman Alley!...	We're located in a safe and quiet residential ...	Karen	New York, New York, United States	I’m a New York native that loves to eat & enjo...
freq	27	61	34	246	16059	191
mean	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN

11 rows × 50 columns

The data types of each column

In [7]:
```
df.dtypes
```

```
Out[7]: name                                              object
        description                                       object
        neighborhood_overview                             object
        host_name                                         object
        host_location                                     object
        host_about                                        object
        host_response_rate                               float64
        host_acceptance_rate                             float64
        host_is_superhost                                   bool
        host_listings_count                              float64
        host_total_listings_count                        float64
        host_has_profile_pic                                bool
        host_identity_verified                              bool
        neighbourhood_group_cleansed                      object
        room_type                                         object
        accommodates                                       int64
        bathrooms                                        float64
        bedrooms                                         float64
        beds                                             float64
        amenities                                         object
        price                                            float64
        minimum_nights                                     int64
        maximum_nights                                     int64
        minimum_minimum_nights                           float64
        maximum_minimum_nights                           float64
        minimum_maximum_nights                           float64
        maximum_maximum_nights                           float64
        minimum_nights_avg_ntm                           float64
        maximum_nights_avg_ntm                           float64
        has_availability                                    bool
        availability_30                                    int64
        availability_60                                    int64
        availability_90                                    int64
        availability_365                                   int64
        number_of_reviews                                  int64
        number_of_reviews_ltm                              int64
        number_of_reviews_l30d                             int64
        review_scores_rating                             float64
        review_scores_cleanliness                        float64
        review_scores_checkin                            float64
        review_scores_communication                      float64
        review_scores_location                           float64
        review_scores_value                              float64
        instant_bookable                                    bool
        calculated_host_listings_count                     int64
        calculated_host_listings_count_entire_homes        int64
        calculated_host_listings_count_private_rooms       int64
        calculated_host_listings_count_shared_rooms        int64
        reviews_per_month                                float64
        n_host_verifications                               int64
        dtype: object
```

```
In [8]: df.head(5)
```

Out[8]:

	name	description	neighborhood_overview	host_name	host_location	host_abo
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	Jennifer	New York, New York, United States	A New York since 2000! M passion creati
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	LisaRoxanne	New York, New York, United States	Laid-ba Native Ne York (formerly b coas
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Rebecca	Brooklyn, New York, United States	Rebecca is artist/design and Henoch
3	Large Furnished Room Near B'way	Please don’t expect the luxury here just a bas...	Theater district, many restaurants around here.	Shunichi	New York, New York, United States	I used to wo for a financ industry b nc
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	MaryEllen	New York, New York, United States	Welcome family life wi my oldest tv away

5 rows × 50 columns

Display shape of df

In [9]:
```
df.shape
```

Out[9]:
```
(28022, 50)
```

Define the label

My goal is to train a machine learning model that predicts the review_scores_rating whether it is high (>=3) or low (<3). This is an example of supervised learning and is a binary classification problem. In our data set, our label will be review_scores_rating column.

In [10]:
```
df['review_scores_rating']
```

```
Out[10]: 0        4.70
         1        4.45
         2        5.00
         3        4.21
         4        4.91
                  ...
         28017    5.00
         28018    5.00
         28019    1.00
         28020    5.00
         28021    5.00
         Name: review_scores_rating, Length: 28022, dtype: float64
```

Obtain the data type of the values on this column:

```
In [11]: df['review_scores_rating'].dtype
```

```
Out[11]: dtype('float64')
```

Display the first 15 uniques values of the "review_scores_rating" column

```
In [12]: df['review_scores_rating'].unique()[:15]
```

```
Out[12]: array([4.7 , 4.45, 5.  , 4.21, 4.91, 4.56, 4.88, 4.86, 4.87, 4.76, 4.52,
                4.89, 4.66, 4.74, 4.39])
```

Identify Features

Review: Simply by inspecting the data, let us identify some columns that should not serve as features--those that will not help us solve our predicive ML problems.

```
In [13]: colnames = [x for x in list(df.columns)if '_name' in x]
         print(colnames)
```

```
['host_name']
```

```
In [14]: df['host_name']
```

```
Out[14]: 0           Jennifer
         1        LisaRoxanne
         2            Rebecca
         3           Shunichi
         4          MaryEllen
                     ...
         28017          Vicky
         28018         Samuel
         28019         Carlos
         28020          Lexia
         28021        Justine
         Name: host_name, Length: 28022, dtype: object
```

Drop column(s) that have names/IDs.

```
In [15]: df.drop(colnames, axis = 1, inplace=True)
In [16]: df.drop(columns=['n_host_verifications'], axis = 1, inplace = True)
```

Check again columns

```
In [17]: df.columns
```

```
Out[17]: Index(['name', 'description', 'neighborhood_overview', 'host_location',
               'host_about', 'host_response_rate', 'host_acceptance_rate',
               'host_is_superhost', 'host_listings_count', 'host_total_listings_count',
               'host_has_profile_pic', 'host_identity_verified',
               'neighbourhood_group_cleansed', 'room_type', 'accommodates',
               'bathrooms', 'bedrooms', 'beds', 'amenities', 'price', 'minimum_nights',
               'maximum_nights', 'minimum_minimum_nights', 'maximum_minimum_nights',
               'minimum_maximum_nights', 'maximum_maximum_nights',
               'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'has_availability',
               'availability_30', 'availability_60', 'availability_90',
               'availability_365', 'number_of_reviews', 'number_of_reviews_ltm',
               'number_of_reviews_l30d', 'review_scores_rating',
               'review_scores_cleanliness', 'review_scores_checkin',
               'review_scores_communication', 'review_scores_location',
               'review_scores_value', 'instant_bookable',
               'calculated_host_listings_count',
               'calculated_host_listings_count_entire_homes',
               'calculated_host_listings_count_private_rooms',
               'calculated_host_listings_count_shared_rooms', 'reviews_per_month'],
              dtype='object')
```

```
In [18]: df.dtypes
```

```
Out[18]: name                                              object
         description                                       object
         neighborhood_overview                             object
         host_location                                     object
         host_about                                        object
         host_response_rate                               float64
         host_acceptance_rate                             float64
         host_is_superhost                                   bool
         host_listings_count                              float64
         host_total_listings_count                        float64
         host_has_profile_pic                                bool
         host_identity_verified                              bool
         neighbourhood_group_cleansed                      object
         room_type                                         object
         accommodates                                       int64
         bathrooms                                        float64
         bedrooms                                         float64
         beds                                             float64
         amenities                                         object
         price                                            float64
         minimum_nights                                     int64
         maximum_nights                                     int64
         minimum_minimum_nights                           float64
         maximum_minimum_nights                           float64
         minimum_maximum_nights                           float64
         maximum_maximum_nights                           float64
         minimum_nights_avg_ntm                           float64
         maximum_nights_avg_ntm                           float64
         has_availability                                    bool
         availability_30                                    int64
         availability_60                                    int64
         availability_90                                    int64
         availability_365                                   int64
         number_of_reviews                                  int64
         number_of_reviews_ltm                              int64
         number_of_reviews_l30d                             int64
         review_scores_rating                             float64
         review_scores_cleanliness                        float64
         review_scores_checkin                            float64
         review_scores_communication                      float64
         review_scores_location                           float64
         review_scores_value                              float64
         instant_bookable                                    bool
         calculated_host_listings_count                     int64
         calculated_host_listings_count_entire_homes        int64
         calculated_host_listings_count_private_rooms       int64
         calculated_host_listings_count_shared_rooms        int64
         reviews_per_month                                float64
         dtype: object
```

```
In [19]: df['neighbourhood_group_cleansed']
```

```
Out[19]: 0        Manhattan
         1         Brooklyn
         2         Brooklyn
         3        Manhattan
         4        Manhattan
                    ...
         28017       Queens
         28018     Brooklyn
         28019     Brooklyn
         28020     Brooklyn
         28021       Queens
         Name: neighbourhood_group_cleansed, Length: 28022, dtype: object
```

```
In [20]: df['name']
```

```
Out[20]: 0                                 Skylit Midtown Castle
         1        Whole flr w/private bdrm, bath & kitchen(pls r...
         2                 Spacious Brooklyn Duplex, Patio + Garden
         3                         Large Furnished Room Near B'way 
         4                        Cozy Clean Guest Room - Family Apt
                                        ...
         28017                              Astoria Luxury suite 2A
         28018    Newly renovated suite in the heart of Williams...
         28019        Perfect Room to Stay in Brooklyn! Near Metro!
         28020         New Beautiful Modern One Bedroom in Brooklyn
         28021      Large, modern, private 1 bedroom in beach condo
         Name: name, Length: 28022, dtype: object
```

Check which columns have missing data

```
In [21]: nan_count = np.sum(df.isnull(), axis = 0)
         nan_count
```

```
Out[21]: name                                               5
         description                                      570
         neighborhood_overview                           9816
         host_location                                     60
         host_about                                     10945
         host_response_rate                             11843
         host_acceptance_rate                           11113
         host_is_superhost                                  0
         host_listings_count                                0
         host_total_listings_count                          0
         host_has_profile_pic                               0
         host_identity_verified                             0
         neighbourhood_group_cleansed                       0
         room_type                                          0
         accommodates                                       0
         bathrooms                                          0
         bedrooms                                        2918
         beds                                            1354
         amenities                                          0
         price                                              0
         minimum_nights                                     0
         maximum_nights                                     0
         minimum_minimum_nights                             0
         maximum_minimum_nights                             0
         minimum_maximum_nights                             0
         maximum_maximum_nights                             0
         minimum_nights_avg_ntm                             0
         maximum_nights_avg_ntm                             0
         has_availability                                   0
         availability_30                                    0
         availability_60                                    0
         availability_90                                    0
         availability_365                                   0
         number_of_reviews                                  0
         number_of_reviews_ltm                              0
         number_of_reviews_l30d                             0
         review_scores_rating                               0
         review_scores_cleanliness                          0
         review_scores_checkin                              0
         review_scores_communication                        0
         review_scores_location                             0
         review_scores_value                                0
         instant_bookable                                   0
         calculated_host_listings_count                     0
         calculated_host_listings_count_entire_homes        0
         calculated_host_listings_count_private_rooms       0
         calculated_host_listings_count_shared_rooms        0
         reviews_per_month                                  0
         dtype: int64
```

```
In [22]: nan_detected = nan_count!=0
         nan_detected
```

```
Out[22]: name                                             True
         description                                      True
         neighborhood_overview                            True
         host_location                                    True
         host_about                                       True
         host_response_rate                               True
         host_acceptance_rate                             True
         host_is_superhost                               False
         host_listings_count                             False
         host_total_listings_count                       False
         host_has_profile_pic                            False
         host_identity_verified                          False
         neighbourhood_group_cleansed                    False
         room_type                                       False
         accommodates                                    False
         bathrooms                                       False
         bedrooms                                         True
         beds                                             True
         amenities                                       False
         price                                           False
         minimum_nights                                  False
         maximum_nights                                  False
         minimum_minimum_nights                          False
         maximum_minimum_nights                          False
         minimum_maximum_nights                          False
         maximum_maximum_nights                          False
         minimum_nights_avg_ntm                          False
         maximum_nights_avg_ntm                          False
         has_availability                                False
         availability_30                                 False
         availability_60                                 False
         availability_90                                 False
         availability_365                                False
         number_of_reviews                               False
         number_of_reviews_ltm                           False
         number_of_reviews_l30d                          False
         review_scores_rating                            False
         review_scores_cleanliness                       False
         review_scores_checkin                           False
         review_scores_communication                     False
         review_scores_location                          False
         review_scores_value                             False
         instant_bookable                                False
         calculated_host_listings_count                  False
         calculated_host_listings_count_entire_homes     False
         calculated_host_listings_count_private_rooms    False
         calculated_host_listings_count_shared_rooms     False
         reviews_per_month                               False
         dtype: bool
```

Review:

Since replacing the missing values with the mean only makes sense for the columns that conatin numerical values (and not for strings), let's create another condition: the type of the column must be int or float.

I will create a series that conatins "True" if the type of the columns is either int64 or float64.

I will combine the two binary series (nan_detected and is_int_or_float) into a new series named to_impute. It will contain the value "True" if a column contains missing values and is of type "int" or "float".

In [23]:
```
is_int_or_float = (df.dtypes == 'int64') | (df.dtypes == 'float64')
is_int_or_float
```

```
Out[23]: name                                             False
         description                                      False
         neighborhood_overview                            False
         host_location                                    False
         host_about                                       False
         host_response_rate                                True
         host_acceptance_rate                              True
         host_is_superhost                                False
         host_listings_count                               True
         host_total_listings_count                         True
         host_has_profile_pic                             False
         host_identity_verified                           False
         neighbourhood_group_cleansed                     False
         room_type                                        False
         accommodates                                      True
         bathrooms                                         True
         bedrooms                                          True
         beds                                              True
         amenities                                        False
         price                                             True
         minimum_nights                                    True
         maximum_nights                                    True
         minimum_minimum_nights                            True
         maximum_minimum_nights                            True
         minimum_maximum_nights                            True
         maximum_maximum_nights                            True
         minimum_nights_avg_ntm                            True
         maximum_nights_avg_ntm                            True
         has_availability                                 False
         availability_30                                   True
         availability_60                                   True
         availability_90                                   True
         availability_365                                  True
         number_of_reviews                                 True
         number_of_reviews_ltm                             True
         number_of_reviews_l30d                            True
         review_scores_rating                              True
         review_scores_cleanliness                         True
         review_scores_checkin                             True
         review_scores_communication                       True
         review_scores_location                            True
         review_scores_value                               True
         instant_bookable                                 False
         calculated_host_listings_count                    True
         calculated_host_listings_count_entire_homes       True
         calculated_host_listings_count_private_rooms      True
         calculated_host_listings_count_shared_rooms       True
         reviews_per_month                                 True
         dtype: bool
```

```
In [24]: to_impute = nan_detected & is_int_or_float
         to_impute
```

```
Out[24]: name                                            False
         description                                     False
         neighborhood_overview                           False
         host_location                                   False
         host_about                                      False
         host_response_rate                               True
         host_acceptance_rate                             True
         host_is_superhost                               False
         host_listings_count                             False
         host_total_listings_count                       False
         host_has_profile_pic                            False
         host_identity_verified                          False
         neighbourhood_group_cleansed                    False
         room_type                                       False
         accommodates                                    False
         bathrooms                                       False
         bedrooms                                         True
         beds                                             True
         amenities                                       False
         price                                           False
         minimum_nights                                  False
         maximum_nights                                  False
         minimum_minimum_nights                          False
         maximum_minimum_nights                          False
         minimum_maximum_nights                          False
         maximum_maximum_nights                          False
         minimum_nights_avg_ntm                          False
         maximum_nights_avg_ntm                          False
         has_availability                                False
         availability_30                                 False
         availability_60                                 False
         availability_90                                 False
         availability_365                                False
         number_of_reviews                               False
         number_of_reviews_ltm                           False
         number_of_reviews_l30d                          False
         review_scores_rating                            False
         review_scores_cleanliness                       False
         review_scores_checkin                           False
         review_scores_communication                     False
         review_scores_location                          False
         review_scores_value                             False
         instant_bookable                                False
         calculated_host_listings_count                  False
         calculated_host_listings_count_entire_homes     False
         calculated_host_listings_count_private_rooms    False
         calculated_host_listings_count_shared_rooms     False
         reviews_per_month                               False
         dtype: bool
```

Let's display a list that contains just the selected column names contained in to_impute

```
In [25]: df.columns[to_impute]
```

```
Out[25]: Index(['host_response_rate', 'host_acceptance_rate', 'bedrooms', 'beds'], dtype='o
         bject')
```

```
In [26]: to_impute_selected = ['host_response_rate', 'host_acceptance_rate', 'bedrooms', 'be
```

Keeping record of the missingness: creating dummy variables

For every column listed in to_impute_selected, create a new corresponding column with the name _na. These columns will contain a "True" or "False" value in place of NaN.

```
In [27]: for column_name in to_impute_selected:
             df[column_name + '_na'] = df[column_name].isnull()
```

Check that the DataFrame contains the new variables

```
In [28]: df.head(5)
```

Out[28]:

	name	description	neighborhood_overview	host_location	host_about	host_resp
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	New York, New York, United States	A New Yorker since 2000! My passion is creatin...	
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	New York, New York, United States	Laid-back Native New Yorker (formerly bi-coast...	
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Brooklyn, New York, United States	Rebecca is an artist/designer, and Henoch is i...	
3	Large Furnished Room Near B'way	Please don't expect the luxury here just a bas...	Theater district, many restaurants around here.	New York, New York, United States	I used to work for a financial industry but no...	
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	New York, New York, United States	Welcome to family life with my oldest two away...	

5 rows × 52 columns

Replacing the missing values with mean values of the column

For every column listed in to_impute_selected, I would fill the missing values with the corresponding mean of all values in the column. I will not create new columns. Just do replacement.

```
In [29]: for column_name in to_impute_selected:
             df[column_name].fillna(np.mean(df[column_name]), inplace = True)
```

```
In [30]: df.head()
```

Out[30]:

	name	description	neighborhood_overview	host_location	host_about	host_resp
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	New York, New York, United States	A New Yorker since 2000! My passion is creatin...	
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	New York, New York, United States	Laid-back Native New Yorker (formerly bi-coast...	
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Brooklyn, New York, United States	Rebecca is an artist/designer, and Henoch is i...	
3	Large Furnished Room Near B'way	Please don’t expect the luxury here just a bas...	Theater district, many restaurants around here.	New York, New York, United States	I used to work for a financial industry but no...	
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	New York, New York, United States	Welcome to family life with my oldest two away...	

5 rows × 52 columns

I check my results below. The code displays the count of missing values for each of the selected columns.

```
In [31]: for column_name in to_impute_selected:
             print("{} missing values count :{}".format(column_name, np.sum(df[column_name].
host_response_rate missing values count :0
host_acceptance_rate missing values count :0
bedrooms missing values count :0
beds missing values count :0
```

Handle Outliers: I will detect and replace outliers in the data using winsorization

In [32]:
```
df['price']
```

Out[32]:
```
0         150.0
1          75.0
2         275.0
3          68.0
4          75.0
          ...
28017      89.0
28018    1000.0
28019      64.0
28020      84.0
28021      70.0
Name: price, Length: 28022, dtype: float64
```

Take care of outliers: I will create a new version of the price column, named "label_price", in which I will replace the top and bottom 1% outlier values with the corresponding percentile value. I will add this new column to my data frame.

In [33]:
```
import scipy.stats as stats
df['label_price'] = stats.mstats.winsorize(df['price'], limits=[0.01, 0.01])
```

Verify Let's verify that the new column label_price was added to my data frame.

In [34]:
```
df.head()
```

Out[34]:

	name	description	neighborhood_overview	host_location	host_about	host_resp
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	New York, New York, United States	A New Yorker since 2000! My passion is creatin...	
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	New York, New York, United States	Laid-back Native New Yorker (formerly bi-coast...	
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Brooklyn, New York, United States	Rebecca is an artist/designer, and Henoch is i...	
3	Large Furnished Room Near B'way	Please don't expect the luxury here just a bas...	Theater district, many restaurants around here.	New York, New York, United States	I used to work for a financial industry but no...	
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	New York, New York, United States	Welcome to family life with my oldest two away...	

5 rows × 53 columns

In [35]:

```
print(df['price'])
print(df['label_price'])
```

```
0         150.0
1          75.0
2         275.0
3          68.0
4          75.0
          ...
28017      89.0
28018    1000.0
28019      64.0
28020      84.0
28021      70.0
Name: price, Length: 28022, dtype: float64
0        150.0
1         75.0
2        275.0
3         68.0
4         75.0
         ...
28017     89.0
28018    899.0
28019     64.0
28020     84.0
28021     70.0
Name: label_price, Length: 28022, dtype: float64
```

Check if Winsorization works I will check to make sure that the values of "price" column and "label_price" column are *not identical*. I will do this by subtracting the two columns and finding the resulting *unique values* of the resulting differences. Of note, if all values are identical, the difference would not contain unique values. If it is the case, I know that the outlier removal step did not work. Let's see.

```
In [36]: (df['price']-df['label_price']).unique()
```

```
Out[36]: array([  0.,   1., 101.,  51.,  -1., 100.,  58.,  81.,  26.,  96.,  15.,
         25.,  41.,   6.,   7.,  46.,  83.,  99.,  44.,  43.,  93.,  78.,
         71.,   2.,  87.,  86.,  50.,  12.])
```

```
In [37]: print(df['label_price'])
```

```
0        150.0
1         75.0
2        275.0
3         68.0
4         75.0
         ...
28017     89.0
28018    899.0
28019     64.0
28020     84.0
28021     70.0
Name: label_price, Length: 28022, dtype: float64
```

Drop the price column.

```
In [38]: df.drop(columns = 'price', axis = 1, inplace = True)
```

```
In [39]: df['host_location']
```

```
Out[39]: 0        New York, New York, United States
         1        New York, New York, United States
         2        Brooklyn, New York, United States
         3        New York, New York, United States
         4        New York, New York, United States
                                ...
         28017      Queens, New York, United States
         28018    New York, New York, United States
         28019                                   US
         28020    New York, New York, United States
         28021                                   US
         Name: host_location, Length: 28022, dtype: object
```

```
In [40]: df['neighbourhood_group_cleansed']
```

```
Out[40]: 0        Manhattan
         1         Brooklyn
         2         Brooklyn
         3        Manhattan
         4        Manhattan
                    ...
         28017       Queens
         28018     Brooklyn
         28019     Brooklyn
         28020     Brooklyn
         28021       Queens
         Name: neighbourhood_group_cleansed, Length: 28022, dtype: object
```

```
In [41]: df.drop(columns = 'host_location', axis = 1, inplace = True)
```

```
In [42]: df.columns
```

```
Out[42]: Index(['name', 'description', 'neighborhood_overview', 'host_about',
               'host_response_rate', 'host_acceptance_rate', 'host_is_superhost',
               'host_listings_count', 'host_total_listings_count',
               'host_has_profile_pic', 'host_identity_verified',
               'neighbourhood_group_cleansed', 'room_type', 'accommodates',
               'bathrooms', 'bedrooms', 'beds', 'amenities', 'minimum_nights',
               'maximum_nights', 'minimum_minimum_nights', 'maximum_minimum_nights',
               'minimum_maximum_nights', 'maximum_maximum_nights',
               'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'has_availability',
               'availability_30', 'availability_60', 'availability_90',
               'availability_365', 'number_of_reviews', 'number_of_reviews_ltm',
               'number_of_reviews_l30d', 'review_scores_rating',
               'review_scores_cleanliness', 'review_scores_checkin',
               'review_scores_communication', 'review_scores_location',
               'review_scores_value', 'instant_bookable',
               'calculated_host_listings_count',
               'calculated_host_listings_count_entire_homes',
               'calculated_host_listings_count_private_rooms',
               'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
               'host_response_rate_na', 'host_acceptance_rate_na', 'bedrooms_na',
               'beds_na', 'label_price'],
              dtype='object')
```

```
In [43]: df['host_about'][0]
```

```
Out[43]: "A New Yorker since 2000! My passion is creating beautiful, unique spaces where un
forgettable memories are made. It's my pleasure to host people from around the wor
ld and meet new faces. Welcome travelers! \r\n\r\nI am a Sound Therapy Practitione
r and Kundalini Yoga & Meditation teacher. I work with energy and sound for relaxa
tion and healing, using Symphonic gong, singing bowls, tuning forks, drums, voice
and other instruments."
```

```
In [44]: df['bedrooms_na']
```

```
Out[44]: 0         True
1        False
2        False
3        False
4        False
         ...
28017    False
28018    False
28019    False
28020    False
28021    False
Name: bedrooms_na, Length: 28022, dtype: bool
```

```
In [45]: print(df.iloc[0])
```

```
name                                                                        Skylit M
idtown Castle
description                                     Beautiful, spacious skylit studio in
the heart...
neighborhood_overview                           Centrally located in the heart of Ma
nhattan ju...
host_about                                      A New Yorker since 2000! My passion 
is creatin...
host_response_rate                                                                  
0.8
host_acceptance_rate                                                                
0.17
host_is_superhost                                                                   
True
host_listings_count                                                                 
8.0
host_total_listings_count                                                           
8.0
host_has_profile_pic                                                                
True
host_identity_verified                                                              
True
neighbourhood_group_cleansed                                                        
Manhattan
room_type                                                                         En
tire home/apt
accommodates                                                                        
1
bathrooms                                                                           
1.0
bedrooms                                                                            
1.329708
beds                                                                                
1.0
amenities                                       ["Extra pillows and blankets", "Baki
ng sheet",...
minimum_nights                                                                      
30
maximum_nights                                                                      
1125
minimum_minimum_nights                                                              
30.0
maximum_minimum_nights                                                              
30.0
minimum_maximum_nights                                                              
1125.0
maximum_maximum_nights                                                              
1125.0
minimum_nights_avg_ntm                                                              
30.0
maximum_nights_avg_ntm                                                              
1125.0
has_availability                                                                    
True
availability_30                                                                     
3
```

```
availability_60
33
availability_90
63
availability_365
338
number_of_reviews
48
number_of_reviews_ltm
0
number_of_reviews_l30d
0
review_scores_rating
4.7
review_scores_cleanliness
4.62
review_scores_checkin
4.76
review_scores_communication
4.79
review_scores_location
4.86
review_scores_value
4.41
instant_bookable
False
calculated_host_listings_count
3
calculated_host_listings_count_entire_homes
3
calculated_host_listings_count_private_rooms
0
calculated_host_listings_count_shared_rooms
0
reviews_per_month
0.33
host_response_rate_na
False
host_acceptance_rate_na
False
bedrooms_na
True
beds_na
False
label_price
150.0
Name: 0, dtype: object
```

Drop the column_na after filling them up with the mean values

```
In [46]: df.drop(columns = 'host_response_rate_na', axis = 1, inplace = True)
         df.drop(columns = 'host_acceptance_rate_na', axis = 1, inplace = True)
         df.drop(columns = 'bedrooms_na', axis = 1, inplace = True)
         df.drop(columns = 'beds_na', axis = 1, inplace = True)
```

Check the data types of each column again

```
In [47]: df.dtypes
```

```
Out[47]: name                                              object
description                                       object
neighborhood_overview                             object
host_about                                        object
host_response_rate                               float64
host_acceptance_rate                             float64
host_is_superhost                                   bool
host_listings_count                              float64
host_total_listings_count                        float64
host_has_profile_pic                                bool
host_identity_verified                              bool
neighbourhood_group_cleansed                      object
room_type                                         object
accommodates                                       int64
bathrooms                                        float64
bedrooms                                         float64
beds                                             float64
amenities                                         object
minimum_nights                                     int64
maximum_nights                                     int64
minimum_minimum_nights                           float64
maximum_minimum_nights                           float64
minimum_maximum_nights                           float64
maximum_maximum_nights                           float64
minimum_nights_avg_ntm                           float64
maximum_nights_avg_ntm                           float64
has_availability                                    bool
availability_30                                    int64
availability_60                                    int64
availability_90                                    int64
availability_365                                   int64
number_of_reviews                                  int64
number_of_reviews_ltm                              int64
number_of_reviews_l30d                             int64
review_scores_rating                             float64
review_scores_cleanliness                        float64
review_scores_checkin                            float64
review_scores_communication                      float64
review_scores_location                           float64
review_scores_value                              float64
instant_bookable                                    bool
calculated_host_listings_count                     int64
calculated_host_listings_count_entire_homes        int64
calculated_host_listings_count_private_rooms       int64
calculated_host_listings_count_shared_rooms        int64
reviews_per_month                                float64
label_price                                      float64
dtype: object
```

About object type I will take care of name, description, neighborhood_overview, host_about, neighbourhood_group_cleansed, room_type, amenities later on.

In [48]:
```
object_data_type_columns = list(df.select_dtypes(include=['object']).columns)
```

In [49]:
```
print(object_data_type_columns)
```

```
['name', 'description', 'neighborhood_overview', 'host_about', 'neighbourhood_group_
cleansed', 'room_type', 'amenities']
```

In [50]:
```
boolean_data_type_columns = list(df.select_dtypes(include=['bool']).columns)
```

In [51]:
```
print(boolean_data_type_columns)
```

```
['host_is_superhost', 'host_has_profile_pic', 'host_identity_verified', 'has_availab
ility', 'instant_bookable']
```

About boolean type I will take care of them now

In [52]:
```
df['host_is_superhost']
```

Out[52]:
```
0        True
1        True
2        True
3        True
4        True
         ...
28017    True
28018    True
28019    True
28020    True
28021    True
Name: host_is_superhost, Length: 28022, dtype: bool
```

Review: Each entry in the "host_is_super_host" column contains one of two values: True or False. Therefore, I will replace the "host_is_super_host" column with two new columns (one column per value). I will use the function pd.get_dummies() as it will reuturn a new DataFrame with the new one-hot encoded values.

In [53]:
```
df_Superhost = pd.get_dummies(df['host_is_superhost'], prefix = 'Host_is_super_host
df_Superhost
```

Out[53]:

	Host_is_super_host__True
0	1
1	1
2	1
3	1
4	1
...	...
28017	1
28018	1
28019	1
28020	1
28021	1

28022 rows × 1 columns

Observe: I observe that all of my host is super host?! I will verify that nunique() function

In [54]:
```
df['host_is_superhost'].nunique()
```

Out[54]:
```
1
```

Observe: Okie. I have all super host in my data. Since the pd.get_dummies() function returned a new DataFrame rather than making the changes to the original DataFram df, I will add the new df_Superhost to my df, and delete the original "host_is_superhost" column from my dataframe.

In [55]:
```
df = df.join(df_Superhost)
df.drop(columns = "host_is_superhost", inplace = True)
```

Verify: I will inspect my data frame to see the changes that have been made. My data frame now conatins columns "Host_is_super_host__True" and no longer contains the "host_is_superhost".

In [56]:
```
df.columns
```

```
Out[56]: Index(['name', 'description', 'neighborhood_overview', 'host_about',
                'host_response_rate', 'host_acceptance_rate', 'host_listings_count',
                'host_total_listings_count', 'host_has_profile_pic',
                'host_identity_verified', 'neighbourhood_group_cleansed', 'room_type',
                'accommodates', 'bathrooms', 'bedrooms', 'beds', 'amenities',
                'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
                'maximum_minimum_nights', 'minimum_maximum_nights',
                'maximum_maximum_nights', 'minimum_nights_avg_ntm',
                'maximum_nights_avg_ntm', 'has_availability', 'availability_30',
                'availability_60', 'availability_90', 'availability_365',
                'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d',
                'review_scores_rating', 'review_scores_cleanliness',
                'review_scores_checkin', 'review_scores_communication',
                'review_scores_location', 'review_scores_value', 'instant_bookable',
                'calculated_host_listings_count',
                'calculated_host_listings_count_entire_homes',
                'calculated_host_listings_count_private_rooms',
                'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
                'label_price', 'Host_is_super_host__True'],
               dtype='object')
```

Review boolean list: ['host_is_superhost', 'host_has_profile_pic', 'host_identity_verified', 'has_availability', 'instant_bookable']

Will do one-hot encoding now.

```
In [57]: df_HasProfilePicture = pd.get_dummies(df['host_has_profile_pic'], prefix='has_profi
         df_HasProfilePicture
```

Out[57]:

	has_profile_pic__True
0	1
1	1
2	1
3	1
4	1
...	...
28017	1
28018	1
28019	1
28020	1
28021	1

28022 rows × 1 columns

```
In [58]: df = df.join(df_HasProfilePicture)
         df.drop(columns = "host_has_profile_pic", inplace = True)
```

```
In [59]: # 'host_identity_verified'
         df_IdentityVerified = pd.get_dummies(df['host_identity_verified'], prefix='identity
         print(df_HasProfilePicture)
         df = df.join(df_IdentityVerified)
         df.drop(columns = "host_identity_verified", inplace = True)

         # 'has_availability'
         df_HasAvailability = pd.get_dummies(df['has_availability'], prefix='has_availabilit
         print(df_HasAvailability)
         df = df.join(df_HasAvailability)
         df.drop(columns = "has_availability", inplace = True)

         # 'instant_bookable'
         df_InstantBookable = pd.get_dummies(df['instant_bookable'], prefix='instant_bookabl
         print(df_InstantBookable)
         df = df.join(df_InstantBookable)
         df.drop(columns = "instant_bookable", inplace = True)
```

```
       has_profile_pic__True
0                          1
1                          1
2                          1
3                          1
4                          1
...                      ...
28017                      1
28018                      1
28019                      1
28020                      1
28021                      1

[28022 rows x 1 columns]
       has_availability__False  has_availability__True
0                            0                       1
1                            0                       1
2                            0                       1
3                            0                       1
4                            0                       1
...                        ...                     ...
28017                        0                       1
28018                        0                       1
28019                        0                       1
28020                        0                       1
28021                        0                       1

[28022 rows x 2 columns]
       instant_bookable__False  instant_bookable__True
0                            1                       0
1                            1                       0
2                            1                       0
3                            1                       0
4                            1                       0
...                        ...                     ...
28017                        0                       1
28018                        1                       0
28019                        0                       1
28020                        1                       0
28021                        0                       1

[28022 rows x 2 columns]
```

Check again on data types of each columns.

In [60]:
```
df.dtypes
```

```
Out[60]: name                                              object
         description                                       object
         neighborhood_overview                             object
         host_about                                        object
         host_response_rate                               float64
         host_acceptance_rate                             float64
         host_listings_count                              float64
         host_total_listings_count                        float64
         neighbourhood_group_cleansed                      object
         room_type                                         object
         accommodates                                       int64
         bathrooms                                        float64
         bedrooms                                         float64
         beds                                             float64
         amenities                                         object
         minimum_nights                                     int64
         maximum_nights                                     int64
         minimum_minimum_nights                           float64
         maximum_minimum_nights                           float64
         minimum_maximum_nights                           float64
         maximum_maximum_nights                           float64
         minimum_nights_avg_ntm                           float64
         maximum_nights_avg_ntm                           float64
         availability_30                                    int64
         availability_60                                    int64
         availability_90                                    int64
         availability_365                                   int64
         number_of_reviews                                  int64
         number_of_reviews_ltm                              int64
         number_of_reviews_l30d                             int64
         review_scores_rating                             float64
         review_scores_cleanliness                        float64
         review_scores_checkin                            float64
         review_scores_communication                      float64
         review_scores_location                           float64
         review_scores_value                              float64
         calculated_host_listings_count                     int64
         calculated_host_listings_count_entire_homes        int64
         calculated_host_listings_count_private_rooms       int64
         calculated_host_listings_count_shared_rooms        int64
         reviews_per_month                                float64
         label_price                                      float64
         Host_is_super_host__True                           uint8
         has_profile_pic__True                              uint8
         identity_verified__True                            uint8
         has_availability__False                            uint8
         has_availability__True                             uint8
         instant_bookable__False                            uint8
         instant_bookable__True                             uint8
         dtype: object
```

Review object type columns:

```
In [61]: print(object_data_type_columns)
```

```
['name', 'description', 'neighborhood_overview', 'host_about', 'neighbourhood_group_
cleansed', 'room_type', 'amenities']
```

In [62]:
```
df[object_data_type_columns].nunique()
```

Out[62]:
```
name                              27386
description                       25952
neighborhood_overview             15800
host_about                        11962
neighbourhood_group_cleansed          5
room_type                             4
amenities                         25020
dtype: int64
```

Do one-hot encoding for 'Neighbourhood_group_cleansed'

In [63]:
```
df_neighbourhood = pd.get_dummies(df['neighbourhood_group_cleansed'], prefix='neigh
print(df_neighbourhood)
df = df.join(df_neighbourhood)
df.drop(columns = "neighbourhood_group_cleansed", inplace = True)
```

```
       neighbourhood__Bronx  neighbourhood__Brooklyn  \
0                         0                        0
1                         0                        1
2                         0                        1
3                         0                        0
4                         0                        0
...                     ...                      ...
28017                     0                        0
28018                     0                        1
28019                     0                        1
28020                     0                        1
28021                     0                        0

       neighbourhood__Manhattan  neighbourhood__Queens  \
0                             1                      0
1                             0                      0
2                             0                      0
3                             1                      0
4                             1                      0
...                         ...                    ...
28017                         0                      1
28018                         0                      0
28019                         0                      0
28020                         0                      0
28021                         0                      1

       neighbourhood__Staten Island
0                                 0
1                                 0
2                                 0
3                                 0
4                                 0
...                             ...
28017                             0
28018                             0
28019                             0
28020                             0
28021                             0

[28022 rows x 5 columns]
```

Do one-hot encoding for 'Room type'

In [64]:
```
df_RoomType = pd.get_dummies(df['room_type'], prefix='room_type_')
print(df_RoomType)
df = df.join(df_RoomType)
df.drop(columns = "room_type", inplace = True)
```

```
       room_type__Entire home/apt  room_type__Hotel room  \
0                               1                      0
1                               1                      0
2                               1                      0
3                               0                      0
4                               0                      0
...                           ...                    ...
28017                           0                      0
28018                           1                      0
28019                           0                      0
28020                           1                      0
28021                           0                      0

       room_type__Private room  room_type__Shared room
0                            0                       0
1                            0                       0
2                            0                       0
3                            1                       0
4                            1                       0
...                        ...                     ...
28017                        1                       0
28018                        0                       0
28019                        1                       0
28020                        0                       0
28021                        1                       0

[28022 rows x 4 columns]
```

In [65]:
```
df['amenities'].nunique()
```

Out[65]:
```
25020
```

In [66]:
```
print(df['amenities'][0])
print(df['amenities'][1])
```

```
["Extra pillows and blankets", "Baking sheet", "Luggage dropoff allowed", "TV", "Han
gers", "Ethernet connection", "Long term stays allowed", "Carbon monoxide alarm", "W
ifi", "Heating", "Dishes and silverware", "Air conditioning", "Free street parking",
"Essentials", "Hot water", "Bathtub", "Kitchen", "Fire extinguisher", "Cooking basic
s", "Dedicated workspace", "Hair dryer", "Stove", "Smoke alarm", "Keypad", "Iron",
"Oven", "Paid parking off premises", "Refrigerator", "Bed linens", "Cleaning before
checkout", "Coffee maker"]
["Extra pillows and blankets", "Luggage dropoff allowed", "Free parking on premise
s", "Pack \u2019n play/Travel crib", "Microwave", "Hangers", "Lockbox", "Long term s
tays allowed", "Carbon monoxide alarm", "High chair", "Wifi", "Heating", "Shampoo",
"Dishes and silverware", "Air conditioning", "Free street parking", "Essentials", "H
ot water", "Bathtub", "Kitchen", "Cable TV", "Fire extinguisher", "Cooking basics",
"Dedicated workspace", "Hair dryer", "Stove", "Children\u2019s books and toys", "TV
with standard cable", "Smoke alarm", "Iron", "Oven", "Refrigerator", "Bed linens",
"Baby safety gates", "Coffee maker"]
```

Too many unique values for "amenities"! Transforming this many categorical values would slow down the computation down the line. Instead, I will convert the top 10 most frequent values in column "amenities."

In [67]:
```
top_10_Amenities = list(df['amenities'].value_counts().head(10).index)
print(top_10_Amenities)
```

```
['["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monoxide alarm", "Fir
e extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shampoo", "Smoke alar
m", "First aid kit", "Air conditioning", "Essentials"]', '["Fire extinguisher", "Loc
k on bedroom door", "Smoke alarm", "Shower gel", "Dedicated workspace", "Conditione
r", "Building staff", "Body soap", "Hot water", "TV", "Hangers", "First aid kit", "H
air dryer", "Bed linens", "Long term stays allowed", "Air conditioning", "Carbon mon
oxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowed", "Wifi", "Esse
ntials"]', '["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon monox
ide alarm", "Security cameras on property", "Dedicated workspace", "Hair dryer", "El
evator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air
conditioning", "Essentials"]', '["Long term stays allowed"]', '["Fire extinguisher",
"Lock on bedroom door", "Smoke alarm", "Dedicated workspace", "Conditioner", "Buildi
ng staff", "Body soap", "Hot water", "TV", "Hangers", "First aid kit", "Hair dryer",
"Long term stays allowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo",
"Iron", "Heating", "Luggage dropoff allowed", "Wifi", "Essentials"]', '["Luggage dro
poff allowed", "TV", "Keurig coffee machine", "Hot water kettle", "Microwave", "Firs
t aid kit", "Hangers", "Long term stays allowed", "Carbon monoxide alarm", "Mini fri
dge", "Building staff", "Wifi", "Laundromat nearby", "Heating", "Shampoo", "Dishes a
nd silverware", "Air conditioning", "Essentials", "Hot water", "Conditioner", "Saf
e", "Fire extinguisher", "Clothing storage", "Hair dryer", "Room-darkening shades",
"Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linens", "Coffee maker"]',
'["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Washer", "El
evator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Private entranc
e", "Dryer", "Air conditioning", "Essentials", "Hot water"]', '["Kitchen", "Long ter
m stays allowed", "Cable TV", "Carbon monoxide alarm", "Wifi", "TV with standard cab
le", "Heating", "Shampoo", "Smoke alarm", "Air conditioning", "Essentials"]', '["Air
conditioning", "Dedicated workspace", "Security cameras on property", "Carbon monoxi
de alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private entrance", "Hangers",
"Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essentials", "Long term stays
allowed"]', '["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air co
nditioning", "Essentials"]']
```

Next step:

Now that I have obtained the most ten frequent values of Amenities. I will transform my dataframe to represent these values numerically. I will create new columns to represent "amenities" by create ten one-hot encoded columns.

In [68]:
```
for value  in top_10_Amenities:
    df['Amenities_' + value] = np.where(df['amenities']==value, 1, 0)

#Remove the original column from my data frame
df.drop(columns = 'amenities', inplace = True)

#Inspect my data frame
df.head(5)
```

Out[68]:

	name	description	neighborhood_overview	host_about	host_response_rate	hos
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	A New Yorker since 2000! My passion is creatin...	0.800000	
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	Laid-back Native New Yorker (formerly bi-coast...	0.090000	
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Rebecca is an artist/designer, and Henoch is i...	1.000000	
3	Large Furnished Room	Please don't expect the luxury here just a bas...	Theater district, many restaurants around here.	I used to work for a financial industry but no...	1.000000	

	Near B'way				
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	Welcome to family life with my oldest two away...	0.906901

5 rows × 65 columns

```
In [69]: df[100:101][:]
```

Out[69]:

	name	description	neighborhood_overview	host_about	host_response_rate	ho
100	(F) Excellent/Pvt Rm	<b>The space</b> <br />Large Private room with ...	NaN	Registered Nurse who is very open and flexible...	0.9	

1 rows × 65 columns

In [70]:
```
df[100:200]['Amenities_["Long term stays allowed"]']
```

```
Out[70]: 100    0
         101    0
         102    0
         103    0
         104    0
               ..
         195    0
         196    0
         197    0
         198    0
         199    0
         Name: Amenities_["Long term stays allowed"], Length: 100, dtype: int64
```

```
In [71]: df['Amenities_["Long term stays allowed"]'].nunique()
```

```
Out[71]: 2
```

Check again data types of columns

```
In [72]: df.dtypes
```

```
Out[72]: name
         object
         description
         object
         neighborhood_overview
         object
         host_about
         object
         host_response_rate
         float64

         ...
         Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water ke
         ttle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon
         monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "He
         ating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot
         water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair drye
         r", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "B
         ed linens", "Coffee maker"]      int64
         Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "W
         asher", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Pr
         ivate entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
         int64
         Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
         m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air co
         nditioning", "Essentials"]
         int64
         Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
         y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private e
         ntrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essent
         ials", "Long term stays allowed"]
         int64
         Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air con
         ditioning", "Essentials"]
         int64
         Length: 65, dtype: object
```

```
In [73]: #Double check to see which columns still have object data type
         object_data_type_columns = list(df.select_dtypes(include=['object']).columns)
         print(object_data_type_columns)
```

```
['name', 'description', 'neighborhood_overview', 'host_about']
```

```
In [74]: df.shape
```

```
Out[74]: (28022, 65)
```

Check again on NaN values. Will randomly select 60% of rows that don't have NaN values.

```
In [75]: nan_count = np.sum(df.isnull(), axis = 0)
         print(nan_count)
         nan_detected = nan_count!=0
```

```
name
5
description
570
neighborhood_overview
9816
host_about
10945
host_response_rate
0

...
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water kett
le", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon mon
oxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heatin
g", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wate
r", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer", "R
oom-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linen
s", "Coffee maker"]        0
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Was
her", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Privat
e entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
0
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air cond
itioning", "Essentials"]
0
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private ent
rance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]
0
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air condi
tioning", "Essentials"]
0
Length: 65, dtype: int64
```

In [76]:
```
print(nan_detected)
```

```
name
True
description
True
neighborhood_overview
True
host_about
True
host_response_rate
False

...
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water kett
le", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon mon
oxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heatin
g", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wate
r", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer", "R
oom-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linen
s", "Coffee maker"]    False
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Was
her", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Privat
e entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
False
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air cond
itioning", "Essentials"]
False
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private ent
rance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]
False
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air condi
tioning", "Essentials"]
False
Length: 65, dtype: bool
```

In [77]:
```
print(nan_count[:10])
```

```
name                          5
description                 570
neighborhood_overview      9816
host_about                10945
host_response_rate            0
host_acceptance_rate          0
host_listings_count           0
host_total_listings_count     0
accommodates                  0
bathrooms                     0
dtype: int64
```

In [78]:
```
print(nan_count[10:20])
```

```
bedrooms                  0
beds                      0
minimum_nights            0
maximum_nights            0
minimum_minimum_nights    0
maximum_minimum_nights    0
minimum_maximum_nights    0
maximum_maximum_nights    0
minimum_nights_avg_ntm    0
maximum_nights_avg_ntm    0
dtype: int64
```

```
In [79]: print(nan_count[20:30])
```

```
availability_30              0
availability_60              0
availability_90              0
availability_365             0
number_of_reviews            0
number_of_reviews_ltm        0
number_of_reviews_l30d       0
review_scores_rating         0
review_scores_cleanliness    0
review_scores_checkin        0
dtype: int64
```

```
In [80]: print(nan_count[30:40])
```

```
review_scores_communication                     0
review_scores_location                          0
review_scores_value                             0
calculated_host_listings_count                  0
calculated_host_listings_count_entire_homes     0
calculated_host_listings_count_private_rooms    0
calculated_host_listings_count_shared_rooms     0
reviews_per_month                               0
label_price                                     0
Host_is_super_host__True                        0
dtype: int64
```

```
In [81]: print(nan_count[40:])
```

```
has_profile_pic__True
0
identity_verified__True
0
has_availability__False
0
has_availability__True
0
instant_bookable__False
0
instant_bookable__True
0
neighbourhood__Bronx
0
neighbourhood__Brooklyn
0
neighbourhood__Manhattan
0
neighbourhood__Queens
0
neighbourhood__Staten Island
0
room_type__Entire home/apt
0
room_type__Hotel room
0
room_type__Private room
0
room_type__Shared room
0
Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monoxide alar
m", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shampoo", "Sm
oke alarm", "First aid kit", "Air conditioning", "Essentials"]
0
Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Shower gel",
"Dedicated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "T
V", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long term stays allowe
d", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Lugg
age dropoff allowed", "Wifi", "Essentials"]
0
Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon monoxid
e alarm", "Security cameras on property", "Dedicated workspace", "Hair dryer", "Elev
ator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air c
onditioning", "Essentials"]
0
Amenities_["Long term stays allowed"]
0
Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Dedicated wo
rkspace", "Conditioner", "Building staff", "Body soap", "Hot water", "TV", "Hanger
s", "First aid kit", "Hair dryer", "Long term stays allowed", "Air conditioning", "C
arbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowed", "Wif
i", "Essentials"]
0
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water kett
le", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon mon
oxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heatin
g", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wate
r", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer", "R
oom-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linen
s", "Coffee maker"]    0
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Was
her", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Privat
e entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
0
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air cond
itioning", "Essentials"]
0
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private ent
rance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]
0
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air condi
tioning", "Essentials"]
0
dtype: int64
```

Friday, 08/02/2024 Drop "Host_About" -- I think for now, I will temporarily drop host_about column. If I have more time in the future, I want to look into if host's information can be used to boost the rating_score, and whether or not this information could be wrongfully used to discriminate against a certain host population.

In [82]:
```
new_df = df.drop(columns = "host_about", inplace = False)
print(new_df.head(5))
```

```
                                                name  \
0                              Skylit Midtown Castle
1  Whole flr w/private bdrm, bath & kitchen(pls r...
2           Spacious Brooklyn Duplex, Patio + Garden
3                   Large Furnished Room Near B'way 
4                 Cozy Clean Guest Room - Family Apt

                                         description  \
0  Beautiful, spacious skylit studio in the heart...
1  Enjoy 500 s.f. top floor in 1899 brownstone, w...
2  We welcome you to stay in our lovely 2 br dupl...
3  Please don’t expect the luxury here just a bas...
4  Our best guests are seeking a safe, clean, spa...

                              neighborhood_overview  host_response_rate  \
0  Centrally located in the heart of Manhattan ju...            0.800000
1  Just the right mix of urban center and local n...            0.090000
2                                                NaN            1.000000
3    Theater district, many restaurants around here.            1.000000
4  Our neighborhood is full of restaurants and ca...            0.906901

   host_acceptance_rate  host_listings_count  host_total_listings_count  \
0              0.170000                  8.0                        8.0
1              0.690000                  1.0                        1.0
2              0.250000                  1.0                        1.0
3              1.000000                  1.0                        1.0
4              0.791953                  1.0                        1.0

   accommodates  bathrooms  bedrooms  ...  \
0             1        1.0  1.329708  ...
1             3        1.0  1.000000  ...
2             4        1.5  2.000000  ...
3             2        1.0  1.000000  ...
4             1        1.0  1.000000  ...

   Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monoxide a
larm", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shampoo",
"Smoke alarm", "First aid kit", "Air conditioning", "Essentials"]  \
0                                                  0
1                                                  0
2                                                  0
3                                                  0
4                                                  0

   Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Shower ge
l", "Dedicated workspace", "Conditioner", "Building staff", "Body soap", "Hot wate
r", "TV", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long term stays a
llowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating",
"Luggage dropoff allowed", "Wifi", "Essentials"]  \
0                                                  0
1                                                  0
2                                                  0
3                                                  0
4                                                  0

   Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon mono
xide alarm", "Security cameras on property", "Dedicated workspace", "Hair dryer", "E
levator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Ai
r conditioning", "Essentials"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

   Amenities_["Long term stays allowed"]  \
0                                       0
1                                       0
2                                       0
3                                       0
4                                       0

   Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Dedicated
workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "TV", "Hanger
s", "First aid kit", "Hair dryer", "Long term stays allowed", "Air conditioning", "C
arbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowed", "Wif
i", "Essentials"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

   Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water k
ettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon
monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heat
ing", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wat
er", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer",
"Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed lin
ens", "Coffee maker"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

   Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV",
"Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Pr
ivate entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

   Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide ala
rm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air con
ditioning", "Essentials"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
```

```
4                                                   0

   Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on proper
ty", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private en
trance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

   Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air co
nditioning", "Essentials"]
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0

[5 rows x 64 columns]
```

In [83]:
```
print(new_df.columns)
```

```
Index(['name', 'description', 'neighborhood_overview', 'host_response_rate',
       'host_acceptance_rate', 'host_listings_count',
       'host_total_listings_count', 'accommodates', 'bathrooms', 'bedrooms',
       'beds', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
       'maximum_minimum_nights', 'minimum_maximum_nights',
       'maximum_maximum_nights', 'minimum_nights_avg_ntm',
       'maximum_nights_avg_ntm', 'availability_30', 'availability_60',
       'availability_90', 'availability_365', 'number_of_reviews',
       'number_of_reviews_ltm', 'number_of_reviews_l30d',
       'review_scores_rating', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value',
       'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
       'identity_verified__True', 'has_availability__False',
       'has_availability__True', 'instant_bookable__False',
       'instant_bookable__True', 'neighbourhood__Bronx',
       'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
       'neighbourhood__Queens', 'neighbourhood__Staten Island',
       'room_type__Entire home/apt', 'room_type__Hotel room',
       'room_type__Private room', 'room_type__Shared room',
       'Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monox
ide alarm", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shamp
oo", "Smoke alarm", "First aid kit", "Air conditioning", "Essentials"]',
       'Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Show
er gel", "Dedicated workspace", "Conditioner", "Building staff", "Body soap", "Hot w
ater", "TV", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long term stay
s allowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heatin
g", "Luggage dropoff allowed", "Wifi", "Essentials"]',
       'Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon
monoxide alarm", "Security cameras on property", "Dedicated workspace", "Hair drye
r", "Elevator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alar
m", "Air conditioning", "Essentials"]',
       'Amenities_["Long term stays allowed"]',
       'Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Dedi
cated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "TV",
"Hangers", "First aid kit", "Hair dryer", "Long term stays allowed", "Air conditioni
ng", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowe
d", "Wifi", "Essentials"]',
       'Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot wa
ter kettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Ca
rbon monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby",
"Heating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Ho
t water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair drye
r", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed
linens", "Coffee maker"]',
       'Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable T
V", "Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating",
"Private entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]',
       'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxid
e alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Ai
r conditioning", "Essentials"]',
       'Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on p
roperty", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Priva
te entrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Esse
ntials", "Long term stays allowed"]',
       'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "A
ir conditioning", "Essentials"]'],
      dtype='object')
```

In [84]:
```
# STILL HAVE NULL value
# name                              5
# description                     570
# neighborhood_overview          9816
```

In [85]:
```
new_df.shape
```

Out[85]:
```
(28022, 64)
```

In [86]:
```
#Obtain rows for which the name is available and ignores missing values
```

In [87]:
```
df_name_notnull = df[new_df['name'].notnull()]
#Obtain the number of rows in df_name_not_null
num_rows = df_name_notnull.shape[0]
```

In [88]:
```
print(num_rows)
```

```
28017
```

In [89]:
```
#Obtain a 80% random sample of rows from df_name_notnull and save the indices of th
#Because I want my model reproducible, I will use random seed.
my_percentage = 0.8
random_seed = 1234
df_subset = df_name_notnull.sample(int(my_percentage * num_rows), random_state = ra

print(df_subset.shape)
```

```
(22413, 65)
```

In [90]:
```
print(df_subset)
```

```
                                                   name  \
6803                   Private Bedroom & Bath- Brooklyn
8112              Bright UWS Studio with great location
8091               Private Suite in Brooklyn Townhouse.
1804                Large 2BR/2B next to Lincoln Center
9942        Sunny & cozy 1BR in the heart of Fort Greene
...                                                 ...
23645           Bright, Sunny, and Nice studio apartment
21809   Sun-Filled 2BR/2BA + Private Home Office! #10345
18964  Luxury, lower east side 1bedroom!! Super hip area
253                  Chelsea living, 2BR best location
11842  Central Studio with 12 foot ceilings - Not Shared

                                            description  \
6803   Located in the emerging Bklyn neighborhood, Cr...
8112   Charming Upper West side Brownstone building. ...
8091   COZY ARTIST'S TOWNHOUSE<br />Best neighborhood...
1804   <b>The space</b><br />Homey, clean and invitin...
9942   I'm new to airbnb and would love to host you! ...
...                                                 ...
23645  Bright and sunny studio apartment on the 5th f...
21809  This 2-bedroom, 2-bath, with home office (BR 3...
18964  Video intercom for easy entry and deliveries.<...
253    Private bedroom with twin bunk bed in spacious...
11842  DATES CURRENTLY AVAILABLE CAN BE EXTENDED:<br ...

                                  neighborhood_overview  \
6803   Awesome park view on a quiet, tree-lined block...
8112                                                NaN
8091   Best neighborhood in New York.  Cool and fun, ...
1804                                                NaN
9942                                                NaN
...                                                 ...
23645                                               NaN
21809                                               NaN
18964  Awesome places to visit in the area. <br /><br...
253    Chelsea is among the best neighborhoods in Man...
11842  Less than 5 minute walk to: <br />Bryant Park<...

                                             host_about  host_response_rate  \
6803                                                NaN            0.906901
8112                                                NaN            0.906901
8091   Filmmaker. Live between New York and Los Angel...            1.000000
1804   We live in NYC with our 3 years old daughter, ...            0.906901
9942                                                               0.906901
...                                                 ...                 ...
23645                                               NaN            0.960000
21809                                               NaN            0.906901
18964  My name is Sarah. I live between miami and Man...            0.900000
253    Native Manhattanite \r\nMom, DJ, Friend to man...            0.000000
11842                  Business professional, easy going            0.906901

       host_acceptance_rate  host_listings_count  host_total_listings_count  \
6803               0.791953                  1.0                        1.0
8112               0.791953                  1.0                        1.0
8091               0.930000                  4.0                        4.0
```

```
1804             0.791953                 1.0                        1.0
9942             0.791953                 1.0                        1.0
...                   ...                 ...                        ...
23645            0.750000                 9.0                        9.0
21809            0.830000                 1.0                        1.0
18964            0.500000                 2.0                        2.0
253              0.791953                 1.0                        1.0
11842            0.791953                 2.0                        2.0

       accommodates  bathrooms  ...  \
6803              2        1.0  ...
8112              2        1.0  ...
8091              1        1.0  ...
1804              4        2.0  ...
9942              2        1.0  ...
...             ...        ...  ...
23645             2        1.0  ...
21809             4        2.0  ...
18964             3        1.0  ...
253               2        1.0  ...
11842             4        1.0  ...

       Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monoxi
de alarm", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shampo
o", "Smoke alarm", "First aid kit", "Air conditioning", "Essentials"]  \
6803                                                   0
8112                                                   0
8091                                                   0
1804                                                   0
9942                                                   0
...                                                  ...
23645                                                  0
21809                                                  0
18964                                                  0
253                                                    0
11842                                                  0

       Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Showe
r gel", "Dedicated workspace", "Conditioner", "Building staff", "Body soap", "Hot wa
ter", "TV", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long term stays
allowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating",
"Luggage dropoff allowed", "Wifi", "Essentials"]  \
6803                                                   0
8112                                                   0
8091                                                   0
1804                                                   0
9942                                                   0
...                                                  ...
23645                                                  0
21809                                                  0
18964                                                  0
253                                                    0
11842                                                  0

       Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon
monoxide alarm", "Security cameras on property", "Dedicated workspace", "Hair drye
r", "Elevator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alar
m", "Air conditioning", "Essentials"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Long term stays allowed"]  \
6803                                       0
8112                                       0
8091                                       0
1804                                       0
9942                                       0
...                                      ...
23645                                      0
21809                                      0
18964                                      0
253                                        0
11842                                      0

       Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Dedic
ated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "TV", "H
angers", "First aid kit", "Hair dryer", "Long term stays allowed", "Air conditionin
g", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowe
d", "Wifi", "Essentials"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot wat
er kettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Car
bon monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby",
"Heating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Ho
t water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair drye
r", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed
linens", "Coffee maker"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
```

```
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable T
V", "Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating",
"Private entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide
alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air
conditioning", "Essentials"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on pr
operty", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Privat
e entrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essen
tials", "Long term stays allowed"]  \
6803                                                     0
8112                                                     0
8091                                                     0
1804                                                     0
9942                                                     0
...                                                    ...
23645                                                    0
21809                                                    0
18964                                                    0
253                                                      0
11842                                                    0

       Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Ai
r conditioning", "Essentials"]
6803                                                     0
```

```
8112                                                       0
8091                                                       0
1804                                                       0
9942                                                       0
...                                                      ...
23645                                                      0
21809                                                      0
18964                                                      0
253                                                        0
11842                                                      0

[22413 rows x 65 columns]
```

In [91]:
```
df_description_notnull = df_subset[df_subset['description'].notnull()]
#Obtain the number of rows in df_description_notnull
num_rows = df_description_notnull.shape[0]
print(num_rows)
```

```
21956
```

In [92]:
```
#Obtain a 80% random sample of rows
df_subset = df_description_notnull.sample(int(my_percentage * num_rows), random_sta
```

In [93]:
```
df_subset.head()
```

```
Out[93]:
```

	name	description	neighborhood_overview	host_about	host_response_rate
4741	Bright 1 br apt in Greenpoint Brooklyn	Unique, bright, Sunny 1 bdrm in Greenpoint Bro...	At the North end of Greenpoint Brooklyn, an am...	I've been in Brooklyn for 10 years, San Franci...	0.860000
17143	Classic Park Slope charm with private deck	Cozy home on the parlor floor of a classic bro...	We fell in love with the charm of the apartmen...	Currently based in the Bay Area, with my husba...	0.906901
26230	Nicky's Place... Clean, Comfortable, and Cozy	This newly renovated one bdrm apt blends moder...	If you want it you can find it in this bustlin...	I'm a native New Yorker and CUNY student. I us...	0.906901
21702	Beautiful Bushwick Apartment Steps to NYC Subway	Conveniently located in the heart of Bushwick,...	Edgy and increasingly hip, Bushwick is surroun...	NaN	1.000000

18316	Home sweet home in Astoria- NY	Room located in the Queens neighborhood of Ast...	Astoria is a neighborhood famous for patricity...	I am very friendly	0.906901

5 rows × 65 columns

In [94]:
```
df_neighborhood_overview_notnull = df_subset[df_subset['neighborhood_overview'].not
num_rows = df_neighborhood_overview_notnull.shape[0]
print(num_rows)
```

```
11633
```

In [95]:
```
#Obtain a 80% random sample of rows
df_subset = df_neighborhood_overview_notnull.sample(int(my_percentage * num_rows),
```

In [96]:
```
print(df_subset.head(5))
```

```
                                                name  \
12978                           Beautiful NYC Apartment
12398                                   Heaven On Earth
1568                           Sunny, Comfortable Space
18391  Sunny spacious private room w/ separate entrance
1986                           PRIME 1br in Williamsburg

                                             description  \
12978  A Gorgeous HUGE Furnished Room in a very nice ...
12398  This is a beautiful colonial house located in ...
1568   Apt with two private bedrooms, a kitchen, LR/D...
18391  Sunny spacious private room in a one story hou...
1986   PRIME Fully furnished 1b garden level apt in a...

                                  neighborhood_overview  \
12978  Lots of cafes, restaurants (cuisine from any c...
12398  Our neighborhood is quiet but friendly with pl...
1568   Crown Heights is a slice of local life in Broo...
18391  It's is a very quiet block in a lovely safe cl...
1986   Our neighborhood is great, has fantastic ameni...

                                             host_about  host_response_rate  \
12978                                               NaN            0.906901
12398                                               NaN            0.906901
1568   I love to travel! As a result I have visited a...            1.000000
18391  Sharing my home with both travelers and local ...            1.000000
1986                                                NaN            1.000000

       host_acceptance_rate  host_listings_count  host_total_listings_count  \
12978              1.000000                  1.0                        1.0
12398              0.791953                  4.0                        4.0
1568               1.000000                  1.0                        1.0
18391              0.840000                  2.0                        2.0
1986               0.930000                  1.0                        1.0

       accommodates  bathrooms  ...  \
12978             4        1.0  ...
12398             2        1.0  ...
1568              2        1.0  ...
18391             1        1.0  ...
1986              2        1.0  ...

       Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon monoxi
de alarm", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "Shampo
o", "Smoke alarm", "First aid kit", "Air conditioning", "Essentials"]  \
12978                                                  0
12398                                                  0
1568                                                   0
18391                                                  0
1986                                                   0

       Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Showe
r gel", "Dedicated workspace", "Conditioner", "Building staff", "Body soap", "Hot wa
ter", "TV", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long term stays
allowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating",
"Luggage dropoff allowed", "Wifi", "Essentials"]  \
12978                                                   0
12398                                                   0
1568                                                    0
18391                                                   0
1986                                                    0

       Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carbon
monoxide alarm", "Security cameras on property", "Dedicated workspace", "Hair drye
r", "Elevator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alar
m", "Air conditioning", "Essentials"]  \
12978                                                   0
12398                                                   0
1568                                                    0
18391                                                   0
1986                                                    0

       Amenities_["Long term stays allowed"]  \
12978                                      0
12398                                      0
1568                                       0
18391                                      0
1986                                       0

       Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Dedic
ated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "TV", "H
angers", "First aid kit", "Hair dryer", "Long term stays allowed", "Air conditionin
g", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff allowe
d", "Wifi", "Essentials"]  \
12978                                                   0
12398                                                   0
1568                                                    0
18391                                                   0
1986                                                    0

       Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot wat
er kettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Car
bon monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby",
"Heating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Ho
t water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair drye
r", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed
linens", "Coffee maker"]  \
12978                                                   0
12398                                                   0
1568                                                    0
18391                                                   0
1986                                                    0

       Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable T
V", "Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating",
"Private entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]  \
12978                                                   0
12398                                                   0
1568                                                    0
18391                                                   0
1986                                                    0
```

```
       Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide
alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air
conditioning", "Essentials"]  \
12978                                                        0
12398                                                        0
1568                                                         0
18391                                                        0
1986                                                         0

       Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on pr
operty", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Privat
e entrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essen
tials", "Long term stays allowed"]  \
12978                                                        0
12398                                                        0
1568                                                         0
18391                                                        0
1986                                                         0

       Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Ai
r conditioning", "Essentials"]
12978                                                        0
12398                                                        0
1568                                                         0
18391                                                        0
1986                                                         0

[5 rows x 65 columns]
```

In [97]:
```
print(df_subset.shape)
```

```
(9306, 65)
```

In [98]:
```
nan_count = np.sum(df_subset.isnull(), axis = 0)
print(nan_count)
# nan_detected = nan_count!=0
# nan_detected
```

```
name
0
description
0
neighborhood_overview
0
host_about
3049
host_response_rate
0

...
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water kett
le", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon mon
oxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heatin
g", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wate
r", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer", "R
oom-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linen
s", "Coffee maker"]       0
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Was
her", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Privat
e entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
0
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air cond
itioning", "Essentials"]
0
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private ent
rance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]
0
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air condi
tioning", "Essentials"]
0
Length: 65, dtype: int64
```

In [ ]:

```
```

In [99]:

```
df_subset.drop(columns = "host_about", inplace = True)
```

In [100…

```
nan_count = np.sum(df_subset.isnull(), axis = 0)
nan_detected = nan_count!=0
print(nan_detected)
```

```
name
False
description
False
neighborhood_overview
False
host_response_rate
False
host_acceptance_rate
False

...
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water kett
le", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon mon
oxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "Heatin
g", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot wate
r", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair dryer", "R
oom-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "Bed linen
s", "Coffee maker"]    False
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "Was
her", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Privat
e entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
False
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air cond
itioning", "Essentials"]
False
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private ent
rance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essential
s", "Long term stays allowed"]
False
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air condi
tioning", "Essentials"]
False
Length: 64, dtype: bool
```

In [101…
```
df_subset.shape
```

Out[101…
```
(9306, 64)
```

In [102…
```
df_subset.columns
```

```
Out[102…   Index(['name', 'description', 'neighborhood_overview', 'host_response_rate',
                  'host_acceptance_rate', 'host_listings_count',
                  'host_total_listings_count', 'accommodates', 'bathrooms', 'bedrooms',
                  'beds', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
                  'maximum_minimum_nights', 'minimum_maximum_nights',
                  'maximum_maximum_nights', 'minimum_nights_avg_ntm',
                  'maximum_nights_avg_ntm', 'availability_30', 'availability_60',
                  'availability_90', 'availability_365', 'number_of_reviews',
                  'number_of_reviews_ltm', 'number_of_reviews_l30d',
                  'review_scores_rating', 'review_scores_cleanliness',
                  'review_scores_checkin', 'review_scores_communication',
                  'review_scores_location', 'review_scores_value',
                  'calculated_host_listings_count',
                  'calculated_host_listings_count_entire_homes',
                  'calculated_host_listings_count_private_rooms',
                  'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
                  'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
                  'identity_verified__True', 'has_availability__False',
                  'has_availability__True', 'instant_bookable__False',
                  'instant_bookable__True', 'neighbourhood__Bronx',
                  'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
                  'neighbourhood__Queens', 'neighbourhood__Staten Island',
                  'room_type__Entire home/apt', 'room_type__Hotel room',
                  'room_type__Private room', 'room_type__Shared room',
                  'Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Carbon mon
           oxide alarm", "Fire extinguisher", "Elevator", "Hair dryer", "Wifi", "Heating", "S
           hampoo", "Smoke alarm", "First aid kit", "Air conditioning", "Essentials"]',
                  'Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "Sh
           ower gel", "Dedicated workspace", "Conditioner", "Building staff", "Body soap", "H
           ot water", "TV", "Hangers", "First aid kit", "Hair dryer", "Bed linens", "Long ter
           m stays allowed", "Air conditioning", "Carbon monoxide alarm", "Shampoo", "Iron",
           "Heating", "Luggage dropoff allowed", "Wifi", "Essentials"]',
                  'Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV", "Carb
           on monoxide alarm", "Security cameras on property", "Dedicated workspace", "Hair d
           ryer", "Elevator", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke
           alarm", "Air conditioning", "Essentials"]',
                  'Amenities_["Long term stays allowed"]',
                  'Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "De
           dicated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "T
           V", "Hangers", "First aid kit", "Hair dryer", "Long term stays allowed", "Air cond
           itioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff
           allowed", "Wifi", "Essentials"]',
                  'Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot
           water kettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed",
           "Carbon monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat near
           by", "Heating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essential
           s", "Hot water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage",
           "Hair dryer", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Ele
           vator", "Bed linens", "Coffee maker"]',
                  'Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable
           TV", "Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heatin
           g", "Private entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]',
                  'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
           ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
           "Air conditioning", "Essentials"]',
                  'Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on
           property", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Pr
           ivate entrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm",
           "Essentials", "Long term stays allowed"]',
                  'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
           "Air conditioning", "Essentials"]'],
                 dtype='object')
```

In [103…
```
df_subset[:][:10].dtypes
```

Out[103…
```
name
object
description
object
neighborhood_overview
object
host_response_rate
float64
host_acceptance_rate
float64

...
Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot water ke
ttle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed", "Carbon
monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat nearby", "He
ating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essentials", "Hot
water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage", "Hair drye
r", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Elevator", "B
ed linens", "Coffee maker"]      int64
Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable TV", "W
asher", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heating", "Pr
ivate entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]
int64
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air co
nditioning", "Essentials"]
int64
Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on propert
y", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Private e
ntrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm", "Essent
ials", "Long term stays allowed"]
int64
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air con
ditioning", "Essentials"]
int64
Length: 64, dtype: object
```

Using filter to filter out which Airbnb has rating_score > 3.5

In [104…
```
print(df_subset['review_scores_value'])
```

```
12978    3.67
12398    5.00
1568     4.60
18391    4.95
1986     4.73
         ... 
1553     4.45
9723     4.71
14579    3.50
26099    3.75
22218    4.64
Name: review_scores_value, Length: 9306, dtype: float64
```

Align indices I will only use df_subset from now on, so I will realign indices before filtering.

In [105…

```
df_subset_reindexed = df_subset.loc[df_subset.index]
print(df_subset_reindexed.shape)
```

```
(9306, 64)
```

In [106…

```
df_subset_reindexed.head(5)
```

```
Out[106…
```

	name	description	neighborhood_overview	host_response_rate	host_accepta
12978	Beautiful NYC Apartment	A Gorgeous HUGE Furnished Room in a very nice ...	Lots of cafes, restaurants (cuisine from any c...	0.906901	1
12398	Heaven On Earth	This is a beautiful colonial house located in ...	Our neighborhood is quiet but friendly with pl...	0.906901	0
1568	Sunny, Comfortable Space	Apt with two private bedrooms, a kitchen, LR/D...	Crown Heights is a slice of local life in Broo...	1.000000	1
18391	Sunny spacious private room w/ separate entrance	Sunny spacious private room in a	It's is a very quiet block in a lovely safe cl...	1.000000	0

		one story hou...			
1986	PRIME 1br in Williamsburg	PRIME Fully furnished 1b garden level apt in a...	Our neighborhood is great, has fantastic ameni...	1.000000	(

5 rows × 64 columns

In [107…
```
# condition1 = df_subset['review_scores_value'] > 3.5
# df_rating_above_average = df_subset[condition1]
# print(condition1)
```

In [108…
```
print(df_subset['Amenities_["Hangers", "Long term stays allowed", "Iron", "TV", "Ca
```

```
1
```

In [109…
```
# Drop that column, because all data points that I sample has value 0 at that colum
# Not helpful to keep.
df_subset.drop(columns = 'Amenities_["Hangers", "Long term stays allowed", "Iron",
```

In [110…
```
print(df_subset['Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke ala
```

```
1
```

In [111…
```
# Drop as well. Not helpful.
df_subset.drop(columns = 'Amenities_["Fire extinguisher", "Lock on bedroom door", "
```

In [112…
```
print(df_subset['Amenities_["Hangers", "Long term stays allowed", "Iron", "Cable TV
```

```
1
```

In [113…
```
df_subset.drop(columns = 'Amenities_["Hangers", "Long term stays allowed", "Iron",
```

In [114…
```
df_subset.columns
```

```
Out[114…  Index(['name', 'description', 'neighborhood_overview', 'host_response_rate',
                 'host_acceptance_rate', 'host_listings_count',
                 'host_total_listings_count', 'accommodates', 'bathrooms', 'bedrooms',
                 'beds', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
                 'maximum_minimum_nights', 'minimum_maximum_nights',
                 'maximum_maximum_nights', 'minimum_nights_avg_ntm',
                 'maximum_nights_avg_ntm', 'availability_30', 'availability_60',
                 'availability_90', 'availability_365', 'number_of_reviews',
                 'number_of_reviews_ltm', 'number_of_reviews_l30d',
                 'review_scores_rating', 'review_scores_cleanliness',
                 'review_scores_checkin', 'review_scores_communication',
                 'review_scores_location', 'review_scores_value',
                 'calculated_host_listings_count',
                 'calculated_host_listings_count_entire_homes',
                 'calculated_host_listings_count_private_rooms',
                 'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
                 'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
                 'identity_verified__True', 'has_availability__False',
                 'has_availability__True', 'instant_bookable__False',
                 'instant_bookable__True', 'neighbourhood__Bronx',
                 'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
                 'neighbourhood__Queens', 'neighbourhood__Staten Island',
                 'room_type__Entire home/apt', 'room_type__Hotel room',
                 'room_type__Private room', 'room_type__Shared room',
                 'Amenities_["Long term stays allowed"]',
                 'Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke alarm", "De
          dicated workspace", "Conditioner", "Building staff", "Body soap", "Hot water", "T
          V", "Hangers", "First aid kit", "Hair dryer", "Long term stays allowed", "Air cond
          itioning", "Carbon monoxide alarm", "Shampoo", "Iron", "Heating", "Luggage dropoff
          allowed", "Wifi", "Essentials"]',
                 'Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine", "Hot
          water kettle", "Microwave", "First aid kit", "Hangers", "Long term stays allowed",
          "Carbon monoxide alarm", "Mini fridge", "Building staff", "Wifi", "Laundromat near
          by", "Heating", "Shampoo", "Dishes and silverware", "Air conditioning", "Essential
          s", "Hot water", "Conditioner", "Safe", "Fire extinguisher", "Clothing storage",
          "Hair dryer", "Room-darkening shades", "Sound system", "Smoke alarm", "Iron", "Ele
          vator", "Bed linens", "Coffee maker"]',
                 'Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron", "Cable
          TV", "Washer", "Elevator", "Bed linens", "Wifi", "TV with standard cable", "Heatin
          g", "Private entrance", "Dryer", "Air conditioning", "Essentials", "Hot water"]',
                 'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
          ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
          "Air conditioning", "Essentials"]',
                 'Amenities_["Air conditioning", "Dedicated workspace", "Security cameras on
          property", "Carbon monoxide alarm", "Lock on bedroom door", "Wifi", "Shampoo", "Pr
          ivate entrance", "Hangers", "Iron", "Heating", "TV", "Hair dryer", "Smoke alarm",
          "Essentials", "Long term stays allowed"]',
                 'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
          "Air conditioning", "Essentials"]'],
                dtype='object')
```

```
In [115…  print(df_subset['Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "He
```

```
          2
```

```
In [116…  # Keep the above column
```

```
In [117…  print(df_subset['Amenities_["Air conditioning", "Dedicated workspace", "Security ca
```

```
1
```

```
In [118…  df_subset.drop(columns = 'Amenities_["Air conditioning", "Dedicated workspace", "Se
```

```
In [119…  print(df_subset['Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carb
```

```
2
```

```
In [120…  #Keep the above column
```

```
In [121…  print(df_subset['Amenities_["Hangers", "Kitchen", "Long term stays allowed", "Iron"
```

```
1
```

```
In [122…  df_subset.drop(columns =  'Amenities_["Hangers", "Kitchen", "Long term stays allowe
```

```
In [123…  print(df_subset['Amenities_["Luggage dropoff allowed", "TV", "Keurig coffee machine
```

```
1
```

```
In [124…  df_subset.drop(columns = 'Amenities_["Luggage dropoff allowed", "TV", "Keurig coffe
```

```
In [125…  print(df_subset['Amenities_["Fire extinguisher", "Lock on bedroom door", "Smoke ala
```

```
1
```

```
In [126…  df_subset.drop(columns = 'Amenities_["Fire extinguisher", "Lock on bedroom door", "
```

```
In [127…  print(df_subset['Amenities_["Long term stays allowed"]'].nunique())
```

```
2
```

Drop 'host_listings_count' as we already have 'host_total_listings_count'

```
In [128…  df_subset.drop(columns = 'host_listings_count', inplace = True)
```

```
In [129…  df_subset.columns
```

```
Out[129…  Index(['name', 'description', 'neighborhood_overview', 'host_response_rate',
                 'host_acceptance_rate', 'host_total_listings_count', 'accommodates',
                 'bathrooms', 'bedrooms', 'beds', 'minimum_nights', 'maximum_nights',
                 'minimum_minimum_nights', 'maximum_minimum_nights',
                 'minimum_maximum_nights', 'maximum_maximum_nights',
                 'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'availability_30',
                 'availability_60', 'availability_90', 'availability_365',
                 'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d',
                 'review_scores_rating', 'review_scores_cleanliness',
                 'review_scores_checkin', 'review_scores_communication',
                 'review_scores_location', 'review_scores_value',
                 'calculated_host_listings_count',
                 'calculated_host_listings_count_entire_homes',
                 'calculated_host_listings_count_private_rooms',
                 'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
                 'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
                 'identity_verified__True', 'has_availability__False',
                 'has_availability__True', 'instant_bookable__False',
                 'instant_bookable__True', 'neighbourhood__Bronx',
                 'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
                 'neighbourhood__Queens', 'neighbourhood__Staten Island',
                 'room_type__Entire home/apt', 'room_type__Hotel room',
                 'room_type__Private room', 'room_type__Shared room',
                 'Amenities_["Long term stays allowed"]',
                 'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
          ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
          "Air conditioning", "Essentials"]',
                 'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
          "Air conditioning", "Essentials"]'],
                dtype='object')
```

Right now, for the sake of time, I couldn't do deep learning/ neural network to do sentiment analysis to see if the text in name, description, neighborhood_overview have impact on the 'review_scores_rating'. Will drop them for now.

```
In [130…  df_final = df_subset.drop(columns = 'name', inplace = False)
```

```
In [131…  # df_final = df_final.drop(columns = 'description', inplace = True)
```

```
In [132…  # df_final.drop(columns = 'neighborhood_overview', inplace = True)
```

```
In [133…  # df_final.columns
```

```
In [134…  df_final.columns
```

Out[134…

```
Index(['description', 'neighborhood_overview', 'host_response_rate',
       'host_acceptance_rate', 'host_total_listings_count', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'minimum_nights', 'maximum_nights',
       'minimum_minimum_nights', 'maximum_minimum_nights',
       'minimum_maximum_nights', 'maximum_maximum_nights',
       'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'availability_30',
       'availability_60', 'availability_90', 'availability_365',
       'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d',
       'review_scores_rating', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value',
       'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
       'identity_verified__True', 'has_availability__False',
       'has_availability__True', 'instant_bookable__False',
       'instant_bookable__True', 'neighbourhood__Bronx',
       'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
       'neighbourhood__Queens', 'neighbourhood__Staten Island',
       'room_type__Entire home/apt', 'room_type__Hotel room',
       'room_type__Private room', 'room_type__Shared room',
       'Amenities_["Long term stays allowed"]',
       'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
"Air conditioning", "Essentials"]',
       'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
"Air conditioning", "Essentials"]'],
      dtype='object')
```

In [135…

```
df_final.drop(columns ='description', inplace = True)
```

In [136…

```
df_final.drop(columns = 'neighborhood_overview', inplace = True)
```

In [137…

```
df_final.columns
```

```
Out[137… Index(['host_response_rate', 'host_acceptance_rate',
                'host_total_listings_count', 'accommodates', 'bathrooms', 'bedrooms',
                'beds', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
                'maximum_minimum_nights', 'minimum_maximum_nights',
                'maximum_maximum_nights', 'minimum_nights_avg_ntm',
                'maximum_nights_avg_ntm', 'availability_30', 'availability_60',
                'availability_90', 'availability_365', 'number_of_reviews',
                'number_of_reviews_ltm', 'number_of_reviews_l30d',
                'review_scores_rating', 'review_scores_cleanliness',
                'review_scores_checkin', 'review_scores_communication',
                'review_scores_location', 'review_scores_value',
                'calculated_host_listings_count',
                'calculated_host_listings_count_entire_homes',
                'calculated_host_listings_count_private_rooms',
                'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
                'label_price', 'Host_is_super_host__True', 'has_profile_pic__True',
                'identity_verified__True', 'has_availability__False',
                'has_availability__True', 'instant_bookable__False',
                'instant_bookable__True', 'neighbourhood__Bronx',
                'neighbourhood__Brooklyn', 'neighbourhood__Manhattan',
                'neighbourhood__Queens', 'neighbourhood__Staten Island',
                'room_type__Entire home/apt', 'room_type__Hotel room',
                'room_type__Private room', 'room_type__Shared room',
                'Amenities_["Long term stays allowed"]',
                'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
          ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
          "Air conditioning", "Essentials"]',
                'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
          "Air conditioning", "Essentials"]'],
               dtype='object')
```

```
In [138… df_final.dtypes
```

Out[138…

```
host_response_rate
float64
host_acceptance_rate
float64
host_total_listings_count
float64
accommodates
int64
bathrooms
float64
bedrooms
float64
beds
float64
minimum_nights
int64
maximum_nights
int64
minimum_minimum_nights
float64
maximum_minimum_nights
float64
minimum_maximum_nights
float64
maximum_maximum_nights
float64
minimum_nights_avg_ntm
float64
maximum_nights_avg_ntm
float64
availability_30
int64
availability_60
int64
availability_90
int64
availability_365
int64
number_of_reviews
int64
number_of_reviews_ltm
int64
number_of_reviews_l30d
int64
review_scores_rating
float64
review_scores_cleanliness
float64
review_scores_checkin
float64
review_scores_communication
float64
review_scores_location
float64
review_scores_value
float64
```

```
calculated_host_listings_count
int64
calculated_host_listings_count_entire_homes
int64
calculated_host_listings_count_private_rooms
int64
calculated_host_listings_count_shared_rooms
int64
reviews_per_month
float64
label_price
float64
Host_is_super_host__True
uint8
has_profile_pic__True
uint8
identity_verified__True
uint8
has_availability__False
uint8
has_availability__True
uint8
instant_bookable__False
uint8
instant_bookable__True
uint8
neighbourhood__Bronx
uint8
neighbourhood__Brooklyn
uint8
neighbourhood__Manhattan
uint8
neighbourhood__Queens
uint8
neighbourhood__Staten Island
uint8
room_type__Entire home/apt
uint8
room_type__Hotel room
uint8
room_type__Private room
uint8
room_type__Shared room
uint8
Amenities_["Long term stays allowed"]
int64
Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide alar
m", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air co
nditioning", "Essentials"]      int64
Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Air con
ditioning", "Essentials"]
int64
dtype: object
```

Using Filter to filter out which Airbnb has "review_scores_values" above 3.5

```
In [139…  condition1 = df['review_scores_rating'] > 3.5
          print(condition1)
```

```
0         True
1         True
2         True
3         True
4         True
         ...
28017     True
28018     True
28019    False
28020     True
28021     True
Name: review_scores_rating, Length: 28022, dtype: bool
```

Double check what is the difference between review_scores_rating and review_scores_value.

```
In [140…  df_final['review_scores_rating']
```

```
Out[140…
12978    3.67
12398    5.00
1568     4.70
18391    4.95
1986     4.72
         ...
1553     4.51
9723     4.86
14579    3.50
26099    3.75
22218    4.64
Name: review_scores_rating, Length: 9306, dtype: float64
```

```
In [141…  df_final['review_scores_value']
```

```
Out[141…
12978    3.67
12398    5.00
1568     4.60
18391    4.95
1986     4.73
         ...
1553     4.45
9723     4.71
14579    3.50
26099    3.75
22218    4.64
Name: review_scores_value, Length: 9306, dtype: float64
```

```
In [142…  (df['review_scores_rating']-df['review_scores_value']).unique()
```

Out[142…

```
array([ 0.29, -0.19,  0.  , -0.15, -0.01, -0.03, -0.01,  0.04, -0.14,
        0.09,  0.06,  0.01, -0.06,  0.36,  0.03, -0.16,  0.11,  0.14,
       -0.04,  0.1 ,  0.21,  0.2 ,  0.02,  0.27,  0.07,  0.25,  0.02,
        0.08, -0.06, -0.02, -0.02, -0.18, -0.11, -0.07,  0.67,  0.06,
        0.01, -0.13, -0.03, -0.1 ,  0.2 ,  0.05,  0.17,  0.14, -0.08,
        0.12,  0.07,  0.15,  0.19,  0.11, -0.09,  0.15, -0.07, -0.25,
        0.27, -0.16, -0.1 ,  0.16,  0.05,  0.33,  0.42, -0.12,  0.1 ,
        0.22,  0.12, -0.18, -0.2 ,  0.24,  0.13,  0.44, -0.26, -0.5 ,
        0.18,  0.31,  0.32,  0.24, -0.4 ,  0.09, -0.66,  0.19,  0.5 ,
       -1.  ,  0.23,  0.03, -0.04,  0.08, -0.05, -0.37,  0.25,  0.18,
        0.28,  0.13,  0.3 , -0.75,  0.4 , -0.15, -0.23, -0.11, -1.33,
        1.  ,  0.22,  0.26,  0.31,  0.37, -0.05, -0.17, -0.29,  0.6 ,
        0.23,  0.28, -0.12,  0.83,  0.16,  5.  , -0.33,  0.34,  0.58,
        0.26, -1.5 ,  0.48, -5.  ,  0.55,  0.4 ,  0.43, -0.38,  0.47,
        0.43, -0.21, -0.67, -0.08,  0.17, -0.23,  0.38, -0.22,  0.49,
       -0.22, -0.24, -0.27,  0.3 , -0.44,  0.27,  0.87,  2.  , -0.14,
       -0.09,  0.38, -0.17, -0.27,  0.45,  0.62,  0.56,  0.43, -0.32,
       -2.  , -0.42,  0.8 , -0.34, -0.31, -3.  , -0.53,  0.72,  0.21,
       -0.2 , -0.31,  0.57, -0.3 , -0.19, -0.4 ,  0.04,  0.4 ,  0.14,
        0.75, -0.43, -0.28,  0.45, -0.14, -0.71,  4.  ,  0.35,  0.32,
        1.33,  0.39,  0.64, -0.43, -0.57,  0.52,  0.47,  0.59,  0.11,
        0.56,  0.72, -0.35, -0.56, -0.13, -0.43,  0.39, -0.36,  0.41,
       -0.55,  0.64, -0.28,  0.46, -0.4 , -0.83,  0.6 ,  0.2 , -0.58,
        0.28,  0.62,  1.5 ,  0.34, -0.28, -0.32, -0.24, -0.45,  0.44,
       -0.33, -0.6 ,  0.21,  0.38,  0.61, -0.41, -0.44, -0.55,  0.66,
       -0.68, -0.62, -0.86,  0.35, -0.47, -0.48,  0.73,  0.15, -0.84,
       -0.6 , -0.54,  0.6 ,  0.09, -0.8 , -1.2 , -0.3 ,  0.85, -1.34,
        1.67, -0.6 ,  0.36, -0.36,  0.67, -0.38,  3.  ,  0.7 , -0.45,
       -4.  ,  0.04, -0.15, -0.11,  0.72, -0.37,  0.22, -0.3 ,  0.54,
       -0.59, -0.06, -0.17,  0.92, -0.2 ,  1.17, -0.8 , -0.1 ,  0.51,
       -0.87,  0.5 , -1.25, -0.07,  0.77,  1.25, -0.26, -0.19, -0.72,
        0.44, -0.37, -0.12, -0.35, -0.25,  0.8 ,  0.5 , -0.66, -0.22,
        0.55,  0.82,  0.34])
```

Okie, I look online, and it seems like review_scores_value reflects guests' opinion on the value they received for the price they paid. It answers the question: "Was the listing worth the price?"

About review_scores_rating, this scores represents the overall rating given by guests for their stay. It is a general evaluation of their entire experience at the listing.

Analyze the range of these two columns

In [143…

```
print(df_final['review_scores_rating'].describe())
```

```
count    9306.000000
mean        4.701919
std         0.477517
min         0.000000
25%         4.630000
50%         4.830000
75%         5.000000
max         5.000000
Name: review_scores_rating, dtype: float64
```

```
In [144…  print(df_final['review_scores_value'].describe())
```

```
count    9306.000000
mean        4.668107
std         0.472672
min         0.000000
25%         4.590000
50%         4.790000
75%         4.940000
max         5.000000
Name: review_scores_value, dtype: float64
```

Okie. Both columns have the range from min: 0.000000 to max: 5.000000.

```
In [ ]:
```

FILTERING

```
In [145…  condition1 = (df['review_scores_rating'] > 3.5)
          condition1
```

```
Out[145…  0         True
          1         True
          2         True
          3         True
          4         True
                    ...
          28017     True
          28018     True
          28019    False
          28020     True
          28021     True
          Name: review_scores_rating, Length: 28022, dtype: bool
```

```
In [146…  df_final['Rating_Above_Average'] = condition1
```

Do one-hot-encoding for Rating Above Average

```
In [147…  df_final_Above = pd.get_dummies(df_final['Rating_Above_Average'], prefix = 'Above_3
          print(df_final_Above)
```

```
       Above_3.5__False  Above_3.5__True
12978                 0                1
12398                 0                1
1568                  0                1
18391                 0                1
1986                  0                1
...                 ...              ...
1553                  0                1
9723                  0                1
14579                 1                0
26099                 0                1
22218                 0                1

[9306 rows x 2 columns]
```

In [148…
```
df_final = df_final.join(df_final_Above)
```

In [149…
```
df_final['Host_is_super_host__True']
```

Out[149…
```
12978    1
12398    1
1568     1
18391    1
1986     1
        ..
1553     1
9723     1
14579    1
26099    1
22218    1
Name: Host_is_super_host__True, Length: 9306, dtype: uint8
```

In [150…
```
print(df_final['Host_is_super_host__True'].nunique())
```

```
1
```

In [151…
```
df_final.drop(columns = 'Host_is_super_host__True', inplace = True)
```

I will also drop the review_scores_rating column because I no longer need it.

In [ ]:
```

```

In [152…
```
df_final['Rating_Above_Average']
```

Out[152…
```
12978     True
12398     True
1568      True
18391     True
1986      True
         ...
1553      True
9723      True
14579    False
26099     True
22218     True
Name: Rating_Above_Average, Length: 9306, dtype: bool
```

In [153…
```
df_final.drop(columns = 'review_scores_rating', inplace = True)
```

Part 4: Define Your Project Plan

Now that you understand your data, in the markdown cell below, define your plan to implement the remaining phases of the machine learning life cycle (data preparation, modeling, evaluation) to solve your ML problem. Answer the following questions:

- Do you have a new feature list? If so, what are the features that you chose to keep and remove after inspecting the data?
- Explain different data preparation techniques that you will use to prepare your data for modeling.
- What is your model (or models)?
- Describe your plan to train your model, analyze its performance and then improve the model. That is, describe your model building, validation and selection plan to produce a model that generalizes well to new data.

Answer

1. Yes, I do have a new feature list. The below are my new features. For Amenities, I chose out the top 10, do one hot encoding, and then take subset of the sample to move on. I also dropped features that have text data. I think if I have time, I would do neuron network for it.

In [154…
```
df_final.columns
```

Out[154…
```
Index(['host_response_rate', 'host_acceptance_rate',
       'host_total_listings_count', 'accommodates', 'bathrooms', 'bedrooms',
       'beds', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights',
       'maximum_minimum_nights', 'minimum_maximum_nights',
       'maximum_maximum_nights', 'minimum_nights_avg_ntm',
       'maximum_nights_avg_ntm', 'availability_30', 'availability_60',
       'availability_90', 'availability_365', 'number_of_reviews',
       'number_of_reviews_ltm', 'number_of_reviews_l30d',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'label_price', 'has_profile_pic__True', 'identity_verified__True',
       'has_availability__False', 'has_availability__True',
       'instant_bookable__False', 'instant_bookable__True',
       'neighbourhood__Bronx', 'neighbourhood__Brooklyn',
       'neighbourhood__Manhattan', 'neighbourhood__Queens',
       'neighbourhood__Staten Island', 'room_type__Entire home/apt',
       'room_type__Hotel room', 'room_type__Private room',
       'room_type__Shared room', 'Amenities_["Long term stays allowed"]',
       'Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monox
ide alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm",
"Air conditioning", "Essentials"]',
       'Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating",
"Air conditioning", "Essentials"]',
       'Rating_Above_Average', 'Above_3.5__False', 'Above_3.5__True'],
      dtype='object')
```

2. I used one-hot-encoding to translate boolean data types to integer. I also used winsorization to handle the price of Airbnb.

3. I plan to use KNN Classifier and use K-Fold Cross Validation when training my KNN classifer. I will use built-in cross-validation tools from scikit-learn.

Part 5: Implement Your Project Plan

Task: In the code cell below, import additional packages that you have used in this course that you will need to implement your project plan.

Step 0: I will import the scikit-learn KNeighborsClassifier, the train_test_split() function for splitting the data into training and test sets, and the metric accuracy_score to evaluate my model.

In [155…
```
# YOUR CODE HERE
# Import additional packages
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix
```

Step 1: (Already built my data frame). Now, define the label and identitfy features.

Definde the label: This is a binary classification problem in which we will predict if a specifc Airbnb has review_rating_score is above 3.5 or not. The label is the 'Rating_Above_Average' column.

Identify Features My features will be all of the remaining columns in the dataset.

Step 2: Create Labeled Examples from the Data Set.

- I will get the 'Rating_Above_Average' column from DataFrame df and assigns it to the variable y. This is our label.
- Gets all other columns from DataFrame df and assigns them to the variable X. These are my features

In [156…
```
y = df_final['Rating_Above_Average']
X = df_final.drop(columns = 'Rating_Above_Average', axis=1)
```

In [157…
```
print(y)
```

```
12978     True
12398     True
1568      True
18391     True
1986      True
         ...
1553      True
9723      True
14579    False
26099     True
22218     True
Name: Rating_Above_Average, Length: 9306, dtype: bool
```

In [158…
```
print(X)
```

```
       host_response_rate  host_acceptance_rate  host_total_listings_count  \
12978            0.906901              1.000000                        1.0
12398            0.906901              0.791953                        4.0
1568             1.000000              1.000000                        1.0
18391            1.000000              0.840000                        2.0
1986             1.000000              0.930000                        1.0
...                   ...                   ...                        ...
1553             1.000000              0.990000                        2.0
9723             1.000000              0.690000                        3.0
14579            1.000000              0.670000                      114.0
26099            1.000000              0.830000                        6.0
22218            1.000000              0.980000                        1.0

       accommodates  bathrooms  bedrooms  beds  minimum_nights  \
12978             4        1.0       1.0   2.0              30
12398             2        1.0       1.0   2.0              30
1568              2        1.0       2.0   2.0              30
18391             1        1.0       1.0   1.0               3
1986              2        1.0       1.0   1.0               3
...             ...        ...       ...   ...             ...
1553              2        1.0       1.0   1.0               1
9723              4        2.0       2.0   2.0               2
14579             1        1.0       1.0   1.0              30
26099             2        2.0       1.0   1.0               3
22218             6        1.0       2.0   4.0               5

       maximum_nights  minimum_minimum_nights  ...  \
12978            1125                    30.0  ...
12398              30                    30.0  ...
1568             1125                    30.0  ...
18391             120                     3.0  ...
1986               30                     3.0  ...
...               ...                     ...  ...
1553               11                     1.0  ...
9723             1125                     2.0  ...
14579            1125                    30.0  ...
26099              25                     3.0  ...
22218             200                     5.0  ...

       neighbourhood__Staten Island  room_type__Entire home/apt  \
12978                             0                           0
12398                             0                           0
1568                              0                           1
18391                             0                           0
1986                              0                           1
...                             ...                         ...
1553                              0                           0
9723                              0                           1
14579                             0                           0
26099                             0                           0
22218                             0                           1

       room_type__Hotel room  room_type__Private room  room_type__Shared room  \
12978                      0                        1                       0
12398                      0                        1                       0
1568                       0                        0                       0
```

```
18391                    0                        1                       0
1986                     0                        0                       0
...                    ...                      ...                     ...
1553                     0                        1                       0
9723                     0                        0                       0
14579                    0                        1                       0
26099                    0                        1                       0
22218                    0                        0                       0

       Amenities_["Long term stays allowed"]  \
12978                                      0
12398                                      0
1568                                       0
18391                                      0
1986                                       0
...                                      ...
1553                                       0
9723                                       0
14579                                      0
26099                                      0
22218                                      0

       Amenities_["Kitchen", "Long term stays allowed", "Cable TV", "Carbon monoxide
alarm", "Wifi", "TV with standard cable", "Heating", "Shampoo", "Smoke alarm", "Air
conditioning", "Essentials"]  \
12978                                                  0
12398                                                  0
1568                                                   0
18391                                                  0
1986                                                   0
...                                                  ...
1553                                                   0
9723                                                   0
14579                                                  0
26099                                                  0
22218                                                  0

       Amenities_["Kitchen", "Long term stays allowed", "TV", "Wifi", "Heating", "Ai
r conditioning", "Essentials"]  \
12978                                                  0
12398                                                  0
1568                                                   0
18391                                                  0
1986                                                   0
...                                                  ...
1553                                                   0
9723                                                   0
14579                                                  0
26099                                                  0
22218                                                  0

       Above_3.5__False  Above_3.5__True
12978                 0                1
12398                 0                1
1568                  0                1
18391                 0                1
```

```
1986                  0                 1
...                 ...               ...
1553                  0                 1
9723                  0                 1
14579                 1                 0
26099                 0                 1
22218                 0                 1

[9306 rows x 53 columns]
```

Step 3a: Create Training and Test Data Sets

I will split 10% of my data and setting it aside as a test set to be used for a final evaluation.

In [159…
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.10, random_st
```

I inspect the training and test data sets

In [160…
```
print(X_train.shape)
```

```
(8375, 53)
```

In [161…
```
print(X_test.shape)
```

```
(931, 53)
```

In [162…
```
X_train.head()
```

Out[162…

	host_response_rate	host_acceptance_rate	host_total_listings_count	accommodates
12562	0.906901	0.791953	1.0	2
17379	1.000000	1.000000	2.0	2
18912	0.950000	0.500000	2.0	2
3254	0.906901	0.791953	1.0	2
15062	0.906901	0.791953	1.0	2

5 rows × 53 columns

Step 3b: Perform a grid search to identify the optimal value of K for my KNN classifier.

3b.1. Setp up a Parameter Grid

Review: I will create a dictionary called "param_grid" that contains 10 possible hyperparameter values for K. The dictionary should contain the following key/value pair:

- A key called "n_neighbors"
- A value which is a list consisting of 10 values for the hyperparameter K.
- The values for K will be in a range that starts at 2 and ends with square root of number of examples in my training set X_train.

In [163...

```
num_examples = X_train.shape[0]
param_grid = dict(n_neighbors = [int(x) for x in np.linspace(2, np.sqrt(num_example
print(param_grid)
```

```
{'n_neighbors': [2, 11, 21, 31, 41, 51, 61, 71, 81, 91]}
```

3b.2. Perform Grid Search Cross-Validation: I will use GridSearchCV to search over the different values of hyperparameter K to find *the one* that results in the best cross_validation (CV) score.

In [164...

```
print("Running Grid Search...")
# 1. I will create a KNeighborsClassifier model object without supplying arguments.
my_model = KNeighborsClassifier()
```

```
Running Grid Search...
```

In [165...

```
#2. Run a grid search with 5-fold cross-validation and assign the output to the obj
my_grid = GridSearchCV(my_model, param_grid, cv=5)
print(my_grid)
```

```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [2, 11, 21, 31, 41, 51, 61, 71, 81,
                                         91]})
```

3. I will fit the model (use the "my_grid" variable on the training data and assign the fitted model to the variable "my_grid_search"

In [166...

```
grid_search = my_grid.fit(X_train, y_train)
print('Done')
```

```
Done
```

3b.3. I will retrieve the value of the hyperparameter K for which the best score was attained.

In [167…
```
best_k = grid_search.best_params_['n_neighbors']
print(best_k)
```

```
31
```

Step 4: Train the optimal KNN Model and Make Predictions

I will initialize a KNeighborsClassifier model object with the best value of hyperparameter K and fit the model to the training data.

In [168…
```
my_model_best = KNeighborsClassifier(best_k)
my_model_best.fit(X_train, y_train)
```

Out[168…
```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=31)
```

About prediction

1. I will use the "predict_proba" function to predict CLASS PROBABILITIES for the test set. This function returns two columns, one column per class label. The first column contains the probability (prob.) that an unlabeled example belongs to class False (aka Above Average is False). The second column belongs to class True.
2. I will use the predict() function with my_model_best to predict the class labels for the test set.

In [171…
```
#1. Make predictions on the test data using the predict_proba() method
pp = my_model_best.predict_proba(X_test)
probability_predictions = []
for i in pp:
    probability_predictions.append(i[1])
#2. Make predictions on the test data using the predict() metho
class_label_predictions = my_model_best.predict(X_test)
```

In [172…
```
print(probability_predictions)
```

```
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8064516129032258, 0.9032258064516129, 0.8
709677419354839, 0.8064516129032258, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 0.96774
1935483871, 0.9032258064516129, 0.967741935483871, 0.967741935483871, 1.0, 0.9677419
35483871, 1.0, 0.9032258064516129, 0.967741935483871, 1.0, 0.967741935483871, 1.0,
1.0, 0.8387096774193549, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 0.967741935483871,
1.0, 1.0, 0.967741935483871, 0.9032258064516129, 1.0, 1.0, 0.9354838709677419, 1.0,
1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.96774193548387
1, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.
0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 1.0, 0.967741935483871,
0.967741935483871, 0.9354838709677419, 1.0, 0.967741935483871, 0.8709677419354839,
1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 0.967
741935483871, 1.0, 1.0, 0.9032258064516129, 1.0, 0.9354838709677419, 0.8709677419354
839, 0.9032258064516129, 1.0, 1.0, 0.9354838709677419, 1.0, 0.9354838709677419, 1.0,
0.967741935483871, 1.0, 0.9354838709677419, 0.9032258064516129, 1.0, 1.0, 1.0, 1.0,
0.8709677419354839, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.967741935483871,
0.967741935483871, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 1.0, 1.0, 0.9677419354838
71, 0.7741935483870968, 0.9032258064516129, 0.967741935483871, 1.0, 1.0, 0.935483870
9677419, 1.0, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.967741935483871, 1.
0, 1.0, 0.7096774193548387, 1.0, 0.967741935483871, 1.0, 0.967741935483871, 1.0, 1.
0, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871,
0.967741935483871, 1.0, 0.967741935483871, 1.0, 1.0, 0.9032258064516129, 1.0, 0.8709
677419354839, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 1.
0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 0.967741935483871, 0.9354838709
677419, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.
0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 0.
9354838709677419, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 0.96774193548
3871, 0.9354838709677419, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9677419
35483871, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.967741935483871, 0.93548
38709677419, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 1.
0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.870967741935483
9, 1.0, 0.967741935483871, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 1.0, 0.96
7741935483871, 0.8387096774193549, 1.0, 1.0, 1.0, 0.967741935483871, 0.9032258064516
129, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0,
1.0, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 0.9677419354
83871, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.9354838709677419, 1.0, 1.0,
0.967741935483871, 1.0, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.9677419354
83871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.
0, 1.0, 0.967741935483871, 1.0, 0.9354838709677419, 0.9354838709677419, 1.0, 1.0, 0.
967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.8709677419354839, 1.
0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 0.8709677419354839, 1.0, 0.9677
41935483871, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 0.967741935483871, 0.96774193548
3871, 0.9354838709677419, 0.967741935483871, 1.0, 0.9354838709677419, 0.967741935483
871, 1.0, 1.0, 0.967741935483871, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0,
0.967741935483871, 1.0, 1.0, 0.8387096774193549, 0.967741935483871, 0.93548387096774
19, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.935483870967741
9, 1.0, 0.9354838709677419, 1.0, 1.0, 0.9354838709677419, 1.0, 1.0, 0.96774193548387
1, 1.0, 0.967741935483871, 0.967741935483871, 1.0, 0.967741935483871, 1.0, 0.8064516
129032258, 1.0, 1.0, 0.967741935483871, 0.9354838709677419, 0.967741935483871, 0.935
4838709677419, 0.7419354838709677, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0,
0.967741935483871, 1.0, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 0.9677419354
83871, 0.967741935483871, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.9354838709677419, 1.0, 0.93548387
09677419, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.9032258064516129, 0.96774193
5483871, 0.7741935483870968, 1.0, 0.967741935483871, 1.0, 0.9354838709677419, 1.0,
1.0, 1.0, 0.8387096774193549, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.903
2258064516129, 0.967741935483871, 0.967741935483871, 1.0, 0.9032258064516129, 1.0,
1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.6774193548387096, 0.93548387096774
19, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 0.967741935
483871, 0.967741935483871, 1.0, 0.5806451612903226, 1.0, 1.0, 0.9354838709677419, 0.
967741935483871, 1.0, 1.0, 0.9354838709677419, 1.0, 1.0, 1.0, 1.0, 0.96774193548387
1, 1.0, 0.9354838709677419, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9354838709
677419, 0.967741935483871, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 0.9677419
35483871, 0.6774193548387096, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.9354838709677
419, 0.967741935483871, 0.967741935483871, 0.967741935483871, 1.0, 0.903225806451612
9, 0.9032258064516129, 1.0, 1.0, 0.9032258064516129, 0.9032258064516129, 1.0, 1.0,
1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419,
0.9354838709677419, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 0.93548387
09677419, 1.0, 1.0, 1.0, 0.8387096774193549, 0.6774193548387096, 1.0, 0.967741935483
871, 1.0, 1.0, 0.967741935483871, 0.9354838709677419, 0.967741935483871, 0.967741935
483871, 0.9032258064516129, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 0.903225806451612
9, 1.0, 0.9354838709677419, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 0.645161
2903225806, 0.9032258064516129, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 0.90322580645
16129, 1.0, 0.967741935483871, 0.967741935483871, 1.0, 0.9354838709677419, 0.9677419
35483871, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 0.5806451612903226, 0.9677419
35483871, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871,
0.967741935483871, 1.0, 0.967741935483871, 0.9032258064516129, 0.967741935483871, 1.
0, 0.9354838709677419, 1.0, 0.9354838709677419, 1.0, 1.0, 0.6774193548387096, 1.0,
1.0, 1.0, 0.6451612903225806, 0.5806451612903226, 0.967741935483871, 0.9677419354838
71, 1.0, 0.967741935483871, 0.9032258064516129, 1.0, 0.9354838709677419, 0.967741935
483871, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.96774193548387
1, 1.0, 1.0, 1.0, 1.0, 0.7741935483870968, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8709
677419354839, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 0.96774193548387
1, 0.9354838709677419, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.93548387096774
19, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9677
41935483871, 0.8709677419354839, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 0.870967741
9354839, 1.0, 1.0, 1.0, 0.9032258064516129, 1.0, 1.0, 1.0, 1.0, 0.6451612903225806,
0.967741935483871, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 1.0, 0.9032258064516129, 0.967741935483871, 0.9354838709677419, 1.0, 1.0,
0.967741935483871, 0.9032258064516129, 1.0, 1.0, 0.967741935483871, 1.0, 0.967741935
483871, 1.0, 0.7419354838709677, 0.9354838709677419, 1.0, 0.9032258064516129, 0.9354
838709677419, 0.967741935483871, 1.0, 1.0, 1.0, 0.9032258064516129, 1.0, 1.0, 1.0,
1.0, 1.0, 0.9354838709677419, 1.0, 1.0, 1.0, 1.0, 0.8709677419354839, 1.0, 0.9354838
709677419, 0.9032258064516129, 0.9354838709677419, 1.0, 1.0, 0.9354838709677419, 0.9
67741935483871, 0.8709677419354839, 1.0, 1.0, 1.0, 0.967741935483871, 1.0, 0.6451612
903225806, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 0.9354838709677419, 1.0,
1.0, 0.9354838709677419, 1.0, 0.9354838709677419, 0.967741935483871, 1.0, 0.83870967
74193549, 1.0, 0.9354838709677419, 1.0, 0.7741935483870968, 1.0, 0.967741935483871,
1.0, 1.0, 1.0, 0.967741935483871, 1.0, 0.967741935483871, 0.8709677419354839, 0.9677
41935483871, 0.967741935483871, 0.967741935483871, 1.0, 0.967741935483871, 0.9677419
35483871, 0.6451612903225806, 0.9354838709677419, 1.0, 1.0, 1.0, 0.967741935483871,
1.0, 1.0, 0.9354838709677419, 0.967741935483871, 0.9032258064516129, 1.0, 1.0, 0.967
741935483871, 0.9354838709677419, 1.0, 0.9354838709677419, 0.967741935483871, 1.0,
0.967741935483871, 1.0, 1.0, 0.967741935483871, 0.9354838709677419, 1.0, 1.0, 0.9677
41935483871, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.96774
1935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.
967741935483871, 1.0, 1.0, 1.0, 0.9032258064516129, 0.967741935483871, 1.0, 1.0, 1.
0, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.
0, 0.9354838709677419, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 0.9354
838709677419, 0.967741935483871, 0.9354838709677419, 1.0, 0.967741935483871, 0.96774
1935483871, 0.967741935483871, 1.0, 0.967741935483871, 0.9354838709677419, 1.0, 1.0,
1.0, 0.967741935483871, 1.0, 1.0, 1.0, 0.9354838709677419, 1.0, 1.0, 1.0, 0.87096774
19354839, 1.0, 0.967741935483871, 1.0, 1.0, 1.0, 1.0, 1.0, 0.967741935483871, 1.0,
1.0, 0.967741935483871, 1.0, 1.0, 0.9032258064516129, 1.0, 1.0, 0.9354838709677419,
0.967741935483871, 1.0, 0.9354838709677419, 0.9032258064516129, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 1.0, 1.0, 0.967741935483871, 0.967741935483871, 1.0, 1.0, 0.83870967741935
49, 0.967741935483871, 1.0, 0.967741935483871, 1.0, 0.8709677419354839, 1.0, 0.96774
1935483871]
```

In [173…

```
print(class_label_predictions)
```

```
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
```

```
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True]
```

Step 4: Evaluate the Accuracy of the Model

I will compute and print the model's accuracy score using accuracy_score()

In [175…
```
acc_score = accuracy_score(y_test, class_label_predictions)
print('Accuracy score: ' + str(acc_score))
```

```
Accuracy score: 0.9817400644468314
```

I will create a confustion matrix to evaluate my model.

In [176…
```
my_confustion_matrix = confusion_matrix(y_test, class_label_predictions, labels=[Tr
pd.DataFrame(my_confustion_matrix, columns=['Predicted: Rating Above Average', 'Pre
index=['Actual: Rating Above Average', 'Actual: Rating Not Above Average'])
```

Out[176…

	Predicted: Rating Above Average	Predicted: Rating Not Above Average
Actual: Rating Above Average	914	0
Actual: Rating Not Above Average	17	0

Step 5: Plot the Precision-Recall Curve

Review: Scikit-learn uses 0.5 as the default for classification threshold. I will use the precision-recall curve to show the trade-off between precision and recall for different classification thresholds.

```
In [177…  from sklearn.metrics import precision_recall_curve
```

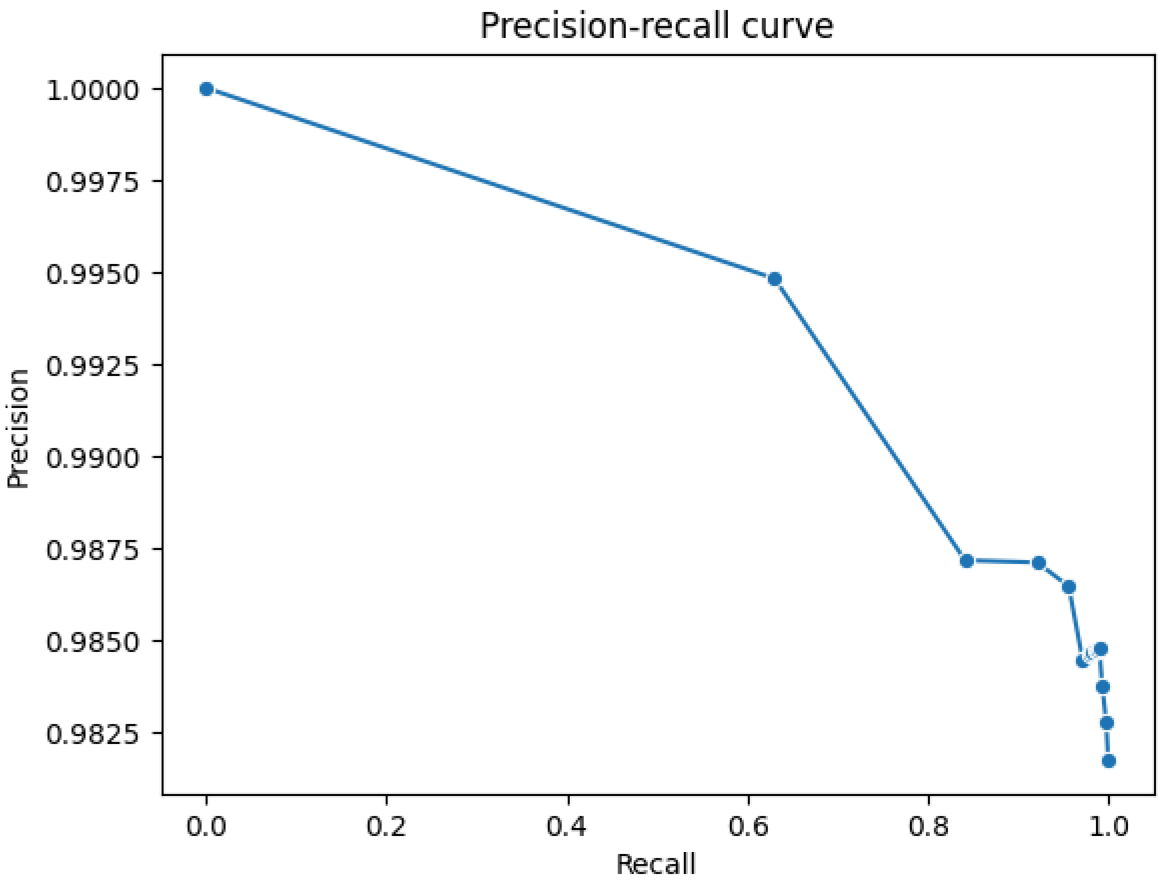
```
In [178…  precision, recall, thresholds = precision_recall_curve(y_test, probability_predicti
```

I will use seaborn's lineplot() method to visualize the precision-recall curve. The variable "recall" will be in on the x-axis and "precision" will be on the y-axis.

```
In [179…  fig = plt.figure()
          ax = fig.add_subplot(111)

          sns.lineplot(x=recall, y=precision, marker = 'o')

          plt.title("Precision-recall curve")
          plt.xlabel("Recall")
          plt.ylabel("Precision")
          plt.show()
```



Review:

- Precision = (True Positives) / (True Positives + False Positives)
- Recall = (True Positives) / (True Positives + False Negatives)

My Observation:

1. Half of the graph is kind of upper left corner (high precision but low recall).

2. The rest of the graph is kind of lower right corner (low precision but high recall). It means my model identifies many positive instances but also makes many false positive predictions.

PROVIDED TEMPLATE

Task: Use the rest of this notebook to carry out your project plan.

You will:

1. Prepare your data for your model.
2. Fit your model to the training data and evaluate your model.
3. Improve your model's performance by performing model selection and/or feature selection techniques to find best model for your problem.

Add code cells below and populate the notebook with commentary, code, analyses, results, and figures as you see fit.

In [169…

```
# YOUR CODE HERE
```