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

```
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.

```
# 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.csv")
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data",
"bookReviewsData.csv")

# YOUR CODE HERE
df = pd.read_csv(airbnbDataSet_filename)

df.head()
```

```
                                                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_name  \
0  Centrally located in the heart of Manhattan ju...     Jennifer
1  Just the right mix of urban center and local n...  LisaRoxanne
2                                                NaN      Rebecca
3    Theater district, many restaurants around here.     Shunichi
4  Our neighborhood is full of restaurants and ca...    MaryEllen

                        host_location  \
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

                                          host_about
host_response_rate  \
0  A New Yorker since 2000! My passion is creatin...
0.80
1  Laid-back Native New Yorker (formerly bi-coast...
```

```
0.09
2  Rebecca is an artist/designer, and Henoch is i...
1.00
3  I used to work for a financial industry but no...
1.00
4  Welcome to family life with my oldest two away...
NaN

   host_acceptance_rate  host_is_superhost
host_listings_count  ...  \
0                 0.17               True                  8.0  ...

1                 0.69               True                  1.0  ...

2                 0.25               True                  1.0  ...

3                 1.00               True                  1.0  ...

4                  NaN               True                  1.0  ...


   review_scores_communication  review_scores_location
review_scores_value  \
0                         4.79                    4.86
4.41
1                         4.80                    4.71
4.64
2                         5.00                    4.50
5.00
3                         4.42                    4.87
4.36
4                         4.95                    4.94
4.92

  instant_bookable calculated_host_listings_count  \
0            False                               3
1            False                               1
2            False                               1
3            False                               1
4            False                               1

   calculated_host_listings_count_entire_homes  \
0                                            3
1                                            1
2                                            1
3                                            0
4                                            0

   calculated_host_listings_count_private_rooms  \
0                                             0
```

```
1                                                   0
2                                                   0
3                                                   1
4                                                   1

   calculated_host_listings_count_shared_rooms  reviews_per_month  \
0                                            0               0.33
1                                            0               4.86
2                                            0               0.02
3                                            0               3.68
4                                            0               0.87

  n_host_verifications
0                    9
1                    6
2                    3
3                    4
4                    7

[5 rows x 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?

<Double click this Markdown cell to make it editable, and record your answers here.> I have chosen the data set 'airbnbListingsData.csv' for this analysis. The objective is to predict the price of Airbnb listings, with the label being 'price'. This is a supervised learning problem, specifically a regression problem, as the goal is to forecast a continuous numeric value. The dataset includes features such as host_is_superhost, host_has_profile_pic, host_identity_verified, has_availability, instant_bookable, 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, n_host_verifications,

neighbourhood_group_cleansed_Bronx, neighbourhood_group_cleansed_Brooklyn, neighbourhood_group_cleansed_Manhattan, neighbourhood_group_cleansed_Queens, neighbourhood_group_cleansed_Staten Island, room_type_Entire home/apt, room_type_Hotel room, room_type_Private room, and room_type_Shared room. This problem is important as dynamic pricing allows hosts to set prices that align with market demand, maximizing their revenue. Additionally, it provides Airbnb with insights into pricing trends across neighborhoods. It supports accurate pricing recommendations for hosts and guests, ultimately improving customer satisfaction by setting realistic expectations for accommodation costs.

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 the first few rows of the dataset
print("First few rows of the dataset:")
print(df.head())

# Display summary statistics
print("\nSummary statistics:")
print(df.describe())

# Display data types
print("\nData types:")
print(df.dtypes)
```

```
First few rows of the dataset:
                                                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_name  \
0  Centrally located in the heart of Manhattan ju...     Jennifer
1  Just the right mix of urban center and local n...  LisaRoxanne
2                                                NaN      Rebecca
3    Theater district, many restaurants around here.     Shunichi
4  Our neighborhood is full of restaurants and ca...    MaryEllen

                      host_location  \
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

                                          host_about
host_response_rate  \
0  A New Yorker since 2000! My passion is creatin...
0.80
1  Laid-back Native New Yorker (formerly bi-coast...
```

```
0.09
2  Rebecca is an artist/designer, and Henoch is i...
1.00
3  I used to work for a financial industry but no...
1.00
4  Welcome to family life with my oldest two away...
NaN

   host_acceptance_rate  host_is_superhost
host_listings_count  ...  \
0                 0.17               True                  8.0  ...

1                 0.69               True                  1.0  ...

2                 0.25               True                  1.0  ...

3                 1.00               True                  1.0  ...

4                  NaN               True                  1.0  ...


   review_scores_communication  review_scores_location
review_scores_value  \
0                         4.79                    4.86
4.41
1                         4.80                    4.71
4.64
2                         5.00                    4.50
5.00
3                         4.42                    4.87
4.36
4                         4.95                    4.94
4.92

  instant_bookable calculated_host_listings_count  \
0            False                               3
1            False                               1
2            False                               1
3            False                               1
4            False                               1

   calculated_host_listings_count_entire_homes  \
0                                            3
1                                            1
2                                            1
3                                            0
4                                            0

   calculated_host_listings_count_private_rooms  \
0                                             0
```

```
1                                                   0
2                                                   0
3                                                   1
4                                                   1

   calculated_host_listings_count_shared_rooms  reviews_per_month  \
0                                            0               0.33
1                                            0               4.86
2                                            0               0.02
3                                            0               3.68
4                                            0               0.87

  n_host_verifications
0                    9
1                    6
2                    3
3                    4
4                    7

[5 rows x 50 columns]

Summary statistics:
       host_response_rate  host_acceptance_rate
host_listings_count  \
count        16179.000000          16909.000000         28022.000000

mean             0.906901              0.791953            14.554778

std              0.227282              0.276732           120.721287

min              0.000000              0.000000             0.000000

25%              0.940000              0.680000             1.000000

50%              1.000000              0.910000             1.000000

75%              1.000000              1.000000             3.000000

max              1.000000              1.000000          3387.000000


       host_total_listings_count  accommodates     bathrooms
bedrooms  \
count               28022.000000  28022.000000  28022.000000
25104.000000
mean                   14.554778      2.874491      1.142174
1.329708
std                   120.721287      1.860251      0.421132
0.700726
min                     0.000000      1.000000      0.000000
```

```
1.000000
25%                     1.000000      2.000000      1.000000
1.000000
50%                     1.000000      2.000000      1.000000
1.000000
75%                     3.000000      4.000000      1.000000
1.000000
max                  3387.000000     16.000000      8.000000
12.000000

                beds         price  minimum_nights  ...
review_scores_checkin  \
count  26668.000000  28022.000000    28022.000000  ...
28022.000000
mean       1.629556    154.228749       18.689387  ...
4.814300
std        1.097104    140.816605       25.569151  ...
0.438603
min        1.000000     29.000000        1.000000  ...
0.000000
25%        1.000000     70.000000        2.000000  ...
4.810000
50%        1.000000    115.000000       30.000000  ...
4.960000
75%        2.000000    180.000000       30.000000  ...
5.000000
max       21.000000   1000.000000     1250.000000  ...
5.000000

       review_scores_communication  review_scores_location  \
count                  28022.000000            28022.000000
mean                       4.808041                4.750393
std                        0.464585                0.415717
min                        0.000000                0.000000
25%                        4.810000                4.670000
50%                        4.970000                4.880000
75%                        5.000000                5.000000
max                        5.000000                5.000000

       review_scores_value  calculated_host_listings_count  \
count         28022.000000                    28022.000000
mean              4.647670                        9.581900
std               0.518023                       32.227523
min               0.000000                        1.000000
25%               4.550000                        1.000000
50%               4.780000                        1.000000
75%               5.000000                        3.000000
max               5.000000                      421.000000

       calculated_host_listings_count_entire_homes  \
```

```
count                                    28022.000000
mean                                         5.562986
std                                         26.121426
min                                          0.000000
25%                                          0.000000
50%                                          1.000000
75%                                          1.000000
max                                        308.000000

       calculated_host_listings_count_private_rooms  \
count                                   28022.000000
mean                                        3.902077
std                                        17.972386
min                                         0.000000
25%                                         0.000000
50%                                         0.000000
75%                                         1.000000
max                                       359.000000

       calculated_host_listings_count_shared_rooms  reviews_per_month
\
count                                   28022.000000       28022.000000

mean                                        0.048283           1.758325

std                                         0.442459           4.446143

min                                         0.000000           0.010000

25%                                         0.000000           0.130000

50%                                         0.000000           0.510000

75%                                         0.000000           1.830000

max                                         8.000000         141.000000


       n_host_verifications
count          28022.000000
mean               5.169510
std                2.028497
min                1.000000
25%                4.000000
50%                5.000000
75%                7.000000
max               13.000000

[8 rows x 36 columns]
```

```
Data types:
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
```

```
# Check for missing values
print("\nMissing values count:")
print(df.isnull().sum())
```

```
Missing values count:
name                                            5
description                                   570
neighborhood_overview                        9816
host_name                                       0
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
n_host_verifications                              0
dtype: int64
```

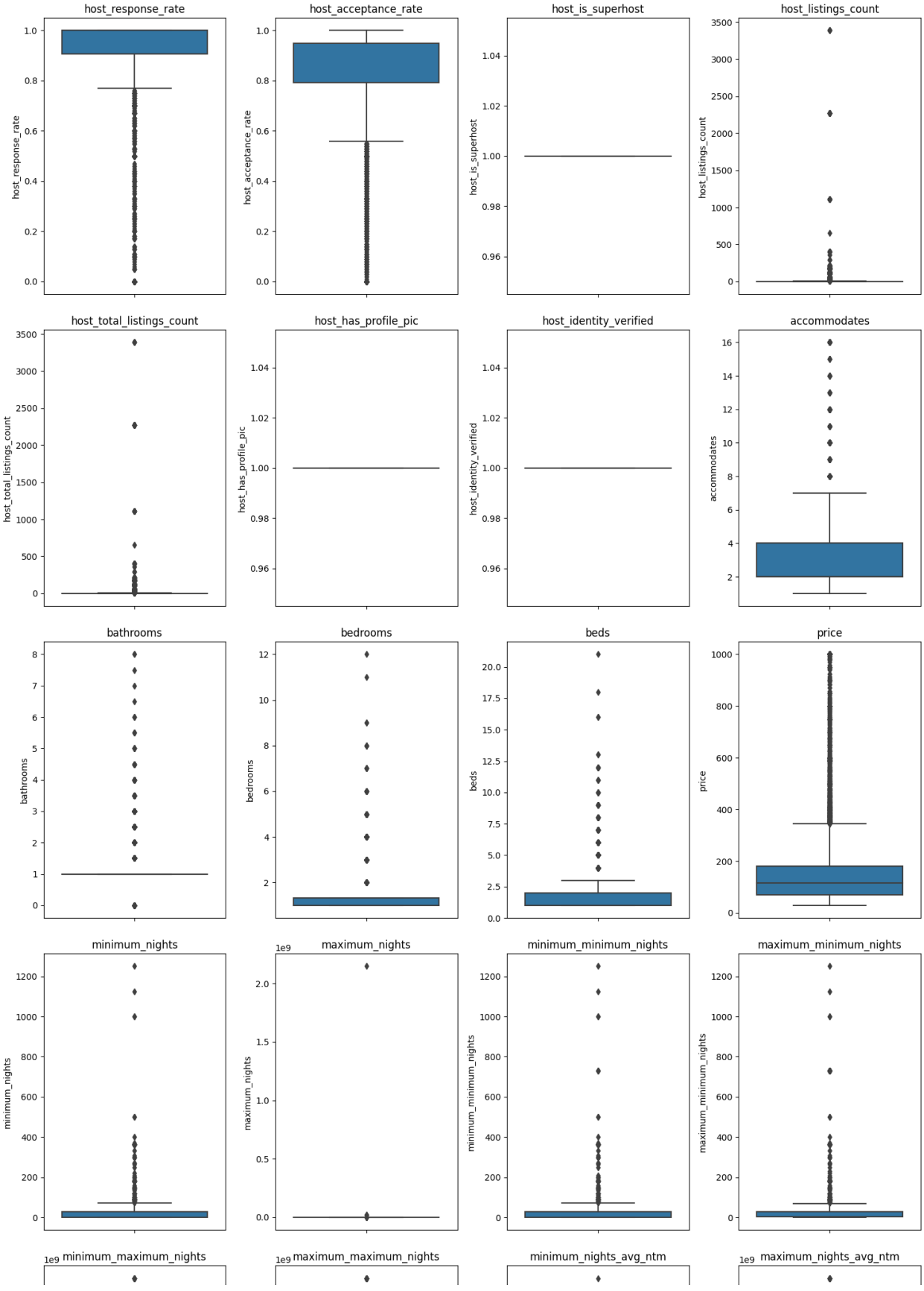
```
# Convert Boolean columns to integers if needed
boolean_columns = df.select_dtypes(include=[bool]).columns
for column in boolean_columns:
    df[column] = df[column].astype(int)

# Fill missing values with mean for numerical columns
for column in df.select_dtypes(include=[np.number]).columns:
    df[column].fillna(df[column].mean(), inplace=True)

# Verify that there are no missing values
print("\nMissing values count after filling:")
print(df.isnull().sum())
```

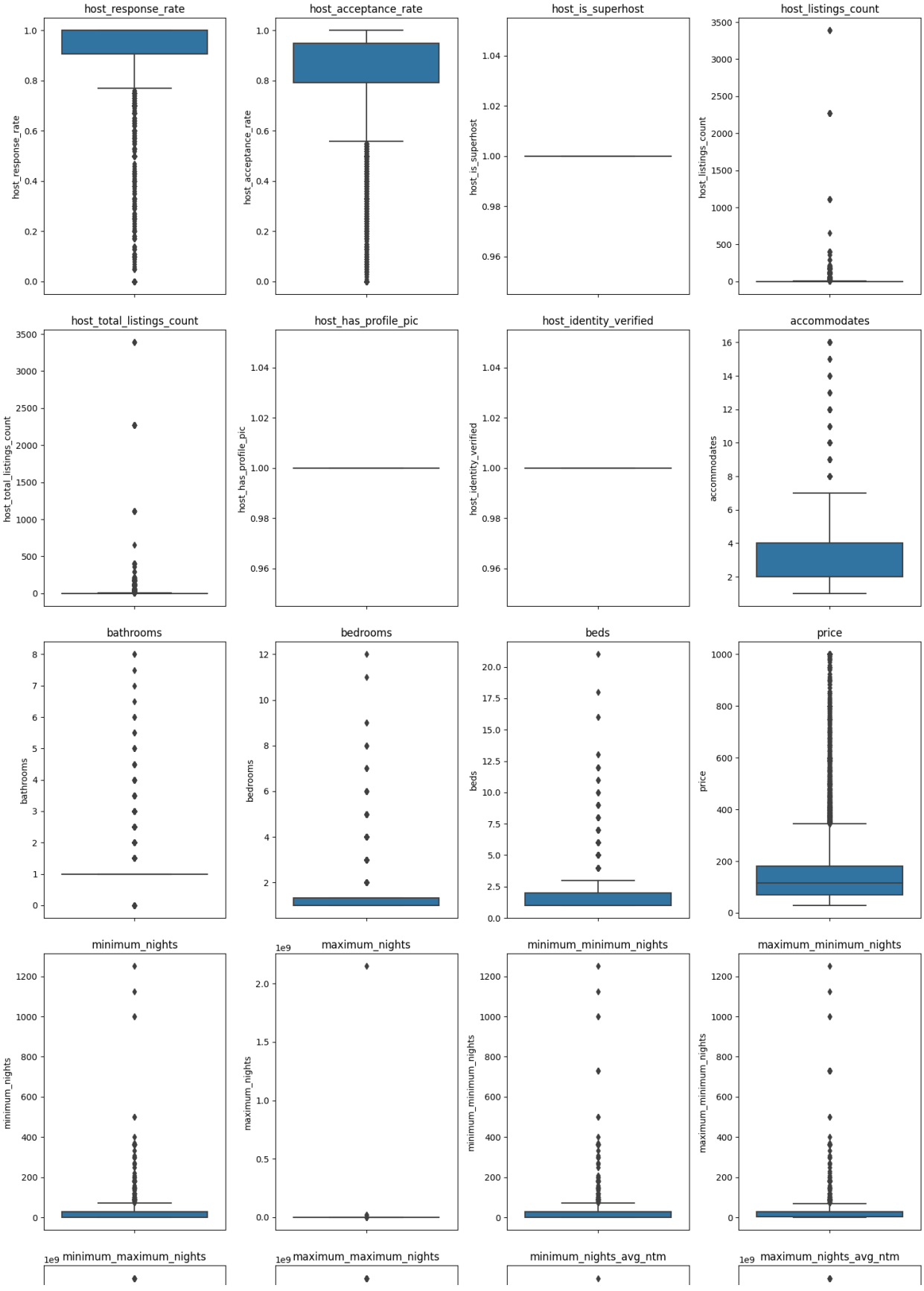
```
Missing values count after filling:
name                                            5
description                                   570
neighborhood_overview                        9816
host_name                                       0
host_location                                  60
host_about                                  10945
host_response_rate                              0
host_acceptance_rate                            0
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                                        0
beds                                            0
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
n_host_verifications                              0
dtype: int64
```

```
# Determine number of numerical columns
num_numerical_cols =
len(df.select_dtypes(include=[np.number]).columns)

# Calculate number of rows and columns for subplots
ncols = 4
nrows = int(np.ceil(num_numerical_cols / ncols))

# Plot boxplots to detect outliers for numerical columns
plt.figure(figsize=(15, 5 * nrows))
for i, column in
enumerate(df.select_dtypes(include=[np.number]).columns, 1):
    plt.subplot(nrows, ncols, i)
    sns.boxplot(y=df[column])
    plt.title(column)

plt.tight_layout()
plt.show()
# Plot boxplots to detect outliers for numerical columns
plt.figure(figsize=(15, 5 * nrows))
for i, column in
enumerate(df.select_dtypes(include=[np.number]).columns, 1):
```

```
    plt.subplot(nrows, ncols, i)
    sns.boxplot(y=df[column])
    plt.title(column)

plt.tight_layout()
plt.show()
```


host_response_rate
host_acceptance_rate
host_is_superhost
host_listings_count
host_total_listings_count
host_has_profile_pic
host_identity_verified
accommodates
bathrooms
bedrooms
beds
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


host_response_rate
host_acceptance_rate
host_is_superhost
host_listings_count
host_total_listings_count
host_has_profile_pic
host_identity_verified
accommodates
bathrooms
bedrooms
beds
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

```
# Remove outliers beyond 3 standard deviations
for column in df.select_dtypes(include=[np.number]).columns:
    mean = df[column].mean()
    std = df[column].std()
    df = df[(df[column] >= mean - 3 * std) & (df[column] <= mean + 3 *
std)]

# Check the columns present in the DataFrame
print("\nColumns in the DataFrame:")
print(df.columns)
```

```
Columns in the DataFrame:
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')
```

```
# Select appropriate features and remove irrelevant features
columns_to_drop = ['name', 'host_name', 'last_review',
'neighborhood_overview', 'host_about']
existing_columns_to_drop = [col for col in columns_to_drop if col in
df.columns]

# Drop columns that exist
df.drop(columns=existing_columns_to_drop, axis=1, inplace=True)
```

```python
# Renaming columns for better readability
df.rename(columns={'price': 'rental_price', 'minimum_nights':
'min_nights'}, inplace=True)

# Verify the changes
print("\nDataFrame after preprocessing:")
print(df.head())
```

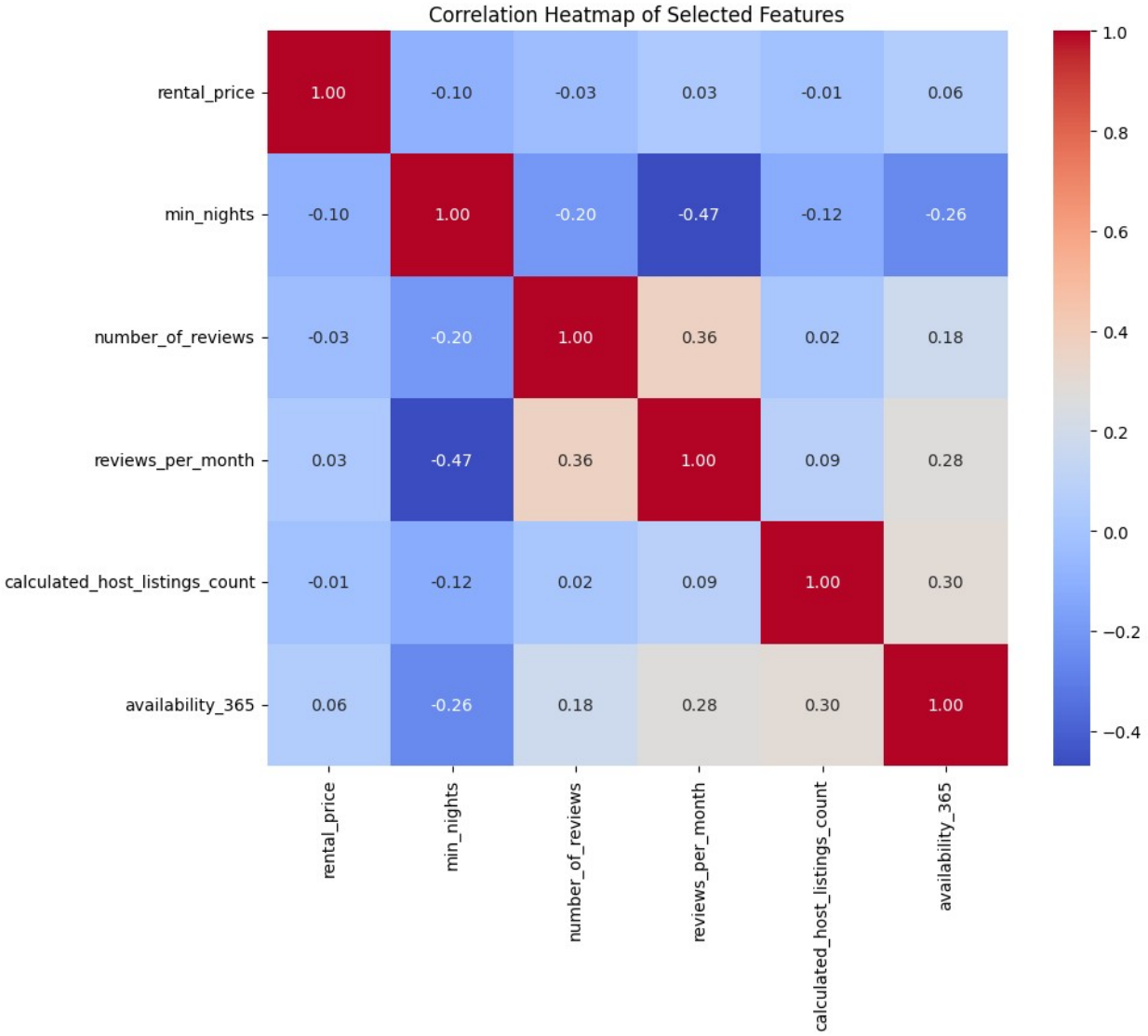
```
DataFrame after preprocessing:
                                          description  \
2   We welcome you to stay in our lovely 2 br dupl...
4   Our best guests are seeking a safe, clean, spa...
5   Beautiful house, gorgeous garden, patio, cozy ...
6   Comfortable studio apartment with super comfor...
10  A true open-plan loft in a repurposed factory ...

                        host_location  host_response_rate  \
2   Brooklyn, New York, United States            1.000000
4   New York, New York, United States            0.906901
5   New York, New York, United States            1.000000
6   New York, New York, United States            1.000000
10  New York, New York, United States            1.000000

    host_acceptance_rate  host_is_superhost  host_listings_count  \
2               0.250000                  1                  1.0
4               0.791953                  1                  1.0
5               1.000000                  1                  3.0
6               1.000000                  1                  1.0
10              0.610000                  1                  4.0

    host_total_listings_count  host_has_profile_pic
host_identity_verified  \
2                         1.0                     1
1
4                         1.0                     1
1
5                         3.0                     1
1
6                         1.0                     1
1
10                        4.0                     1
1

   neighbourhood_group_cleansed  ... review_scores_communication  \
2                      Brooklyn  ...                        5.00
4                     Manhattan  ...                        4.95
5                      Brooklyn  ...                        4.82
6                      Brooklyn  ...                        4.80
```

```
10                     Brooklyn  ...                          4.60

    review_scores_location  review_scores_value  instant_bookable  \
2                     4.50                 5.00                 0
4                     4.94                 4.92                 0
5                     4.87                 4.73                 0
6                     4.67                 4.57                 1
10                    5.00                 4.80                 0

    calculated_host_listings_count  \
2                                1
4                                1
5                                3
6                                1
10                               1

   calculated_host_listings_count_entire_homes  \
2                                            1
4                                            0
5                                            1
6                                            1
10                                           1

    calculated_host_listings_count_private_rooms  \
2                                              0
4                                              1
5                                              2
6                                              0
10                                             0

    calculated_host_listings_count_shared_rooms  reviews_per_month  \
2                                             0               0.02
4                                             0               0.87
5                                             0               1.48
6                                             0               1.24
10                                            0               0.06

    n_host_verifications
2                      3
4                      7
5                      7
6                      7
10                     4

[5 rows x 46 columns]
```

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

```
Index(['description', 'host_location', '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',
       'rental_price', 'min_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')

import seaborn as sns
import matplotlib.pyplot as plt

# Select a subset of features for the correlation heatmap
subset_features = ['rental_price', 'min_nights', 'number_of_reviews',
'reviews_per_month', 'calculated_host_listings_count',
'availability_365']
correlation_matrix = df[subset_features].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f',
cmap='coolwarm')
plt.title('Correlation Heatmap of Selected Features')
plt.show()
```



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.

<Double click this Markdown cell to make it editable, and record your answers here.>

Yes, I have refined my feature list after inspecting the data. I have chosen to retain features such as host_is_superhost, host_has_profile_pic, host_identity_verified, has_availability, instant_bookable, host_response_rate, host_acceptance_rate, host_listings_count, host_total_listings_count, accommodates, bathrooms, bedrooms, beds, 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, 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, n_host_verifications, and the encoded categorical features like neighbourhood_group_cleansed and room_type. Features such as name, host_name, last_review, neighborhood_overview, and host_about have been removed as they are deemed irrelevant for the regression task.

For data preparation, I will fill missing values in numerical columns with their mean and convert Boolean columns to integers. Outliers will be detected and removed using boxplots and by filtering values beyond three standard deviations from the mean. One-hot encoding will be applied to categorical variables, such as neighborhood and room type, to convert them into numerical format. Feature selection will prioritize those most relevant to predicting rental_price, and feature scaling may be applied if necessary, particularly for models sensitive to feature scaling.

The model selected for this task is K-Nearest Neighbors (KNN). My plan for model building involves fitting the KNN model to the training data and making predictions on the test set. I will evaluate the model's performance using Mean Squared Error (MSE) and R-squared (R^2) to assess its accuracy and generalizability. To enhance the model, I will use Grid Search to determine the optimal number of neighbors (k) that minimizes MSE. After identifying the best k, I will re-evaluate the model to confirm improvements. Additionally, I will visualize performance across different k values to ensure that the chosen model is robust and effective.

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.

```
# YOUR CODE HERE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
import numpy as np
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

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.

```
# Load your data
df = pd.read_csv('data_regressors/airbnbData_train.csv')

# Handle missing values (example: fill with mean)
df.fillna(df.mean(), inplace=True)

print(df.columns)
```

```
Index(['host_is_superhost', 'host_has_profile_pic',
'host_identity_verified',
       'has_availability', 'instant_bookable', 'host_response_rate',
       'host_acceptance_rate', 'host_listings_count',
       'host_total_listings_count', 'accommodates', 'bathrooms',
'bedrooms',
       'beds', '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',
'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',
       'n_host_verifications', 'neighbourhood_group_cleansed_Bronx',
```

```
       'neighbourhood_group_cleansed_Brooklyn',
       'neighbourhood_group_cleansed_Manhattan',
       'neighbourhood_group_cleansed_Queens',
       'neighbourhood_group_cleansed_Staten Island',
       'room_type_Entire home/apt', 'room_type_Hotel room',
       'room_type_Private room', 'room_type_Shared room'],
      dtype='object')
```

```
# Drop the target column
X = df.drop(columns=['price'])
y = df['price']

print(df.columns)
```

```
Index(['host_is_superhost', 'host_has_profile_pic',
'host_identity_verified',
       'has_availability', 'instant_bookable', 'host_response_rate',
       'host_acceptance_rate', 'host_listings_count',
       'host_total_listings_count', 'accommodates', 'bathrooms',
'bedrooms',
       'beds', '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',
'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',
       'n_host_verifications', 'neighbourhood_group_cleansed_Bronx',
       'neighbourhood_group_cleansed_Brooklyn',
       'neighbourhood_group_cleansed_Manhattan',
       'neighbourhood_group_cleansed_Queens',
       'neighbourhood_group_cleansed_Staten Island',
       'room_type_Entire home/apt', 'room_type_Hotel room',
       'room_type_Private room', 'room_type_Shared room'],
      dtype='object')
```

```
# Handle missing values
# For simplicity, we'll use mean imputation for numerical features and
mode imputation for categorical features
X.fillna(X.mean(), inplace=True)
X.fillna(X.mode().iloc[0], inplace=True)
```

```
# Convert categorical features to numerical using one-hot encoding
X = pd.get_dummies(X)

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize KNN regressor
knn = KNeighborsRegressor(n_neighbors=5)

# Fit the model
knn.fit(X_train, y_train)

# Make predictions
y_pred = knn.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 0.46347929625035084
R-squared: 0.518965494452688
```

```
# Define the parameter grid
param_grid = {'n_neighbors': list(range(1, 21))}

# Initialize GridSearchCV
grid_search = GridSearchCV(KNeighborsRegressor(), param_grid, cv=5,
scoring='neg_mean_squared_error')

# Fit GridSearchCV
grid_search.fit(X_train, y_train)
```

```
GridSearchCV(cv=5, estimator=KNeighborsRegressor(),
             param_grid={'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12,
                                         13, 14, 15, 16, 17, 18, 19,
20]},
             scoring='neg_mean_squared_error')
```

```
# Get the best parameters and the best model
best_k = grid_search.best_params_['n_neighbors']
best_knn = grid_search.best_estimator_
```

```python
# Evaluate the best model
y_pred_best = best_knn.predict(X_test)
mse_best = mean_squared_error(y_test, y_pred_best)
r2_best = r2_score(y_test, y_pred_best)

print(f"Best K: {best_k}")
print(f"Best Mean Squared Error: {mse_best}")
print(f"Best R-squared: {r2_best}")
```

```
Best K: 10
Best Mean Squared Error: 0.444458786820457
Best R-squared: 0.5387064438821201
```

```python
# Plot performance vs. k
results = grid_search.cv_results_
mean_test_scores = results['mean_test_score']

plt.figure(figsize=(10, 6))
plt.plot(range(1, 21), -mean_test_scores, marker='o', linestyle='--',
color='blue')
plt.title('Grid Search Performance for KNN')
plt.xlabel('Number of Neighbors (k)')
plt.ylabel('Mean Squared Error')
plt.xticks(range(1, 21))
plt.grid(True)
plt.show()
```