DefineAndSolveMLProblem

July 30, 2025

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

[1]:
```
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.

1.1 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.

[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.csv")
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsData.
 ↪csv")


df = pd.read_csv(airbnbDataSet_filename,header=0)

df.head()
```

[2]:
```
                                                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]
```

1.2 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?

1. airbnbDataSet
2. I will be analyzing which features contribute most to the price level (premium vs budget) of the airbnb. The label will be a ”is_premium” column newly created. I’ll be predicting whether an airbnb listing is ”premium” or not based on whether its above the threshold (median price).
3. This is a supervised learning problem. It is a classification problem.
4. My features are all the other columns except this new ”is_premium” column.
5. This is an important problem because a “premium” vs ”budget” classifier helps hosts and pricing platforms automatically tier listings, recommend competitive pricing strategies, and personalize promotion or search ranking. By understanding which characteristics affect/contribute to premium rates, Airbnb could boost host revenues, improve guest satisfaction with fair pricing, and optimize marketplace liquidity.

1.3 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.

[3]:
```
df.shape
```

[3]:
```
(28022, 50)
```

[4]:
```
df['price'].median()
```

[4]:
```
115.0
```

[17]:
```
#creating new is_premium column based on the threshold(median)
threshold = df['price'].median()
df['is_premium'] = (df['price'] > threshold).astype(int)
df.head()
```

```
      ␣
↪---------------------------------------------------------------------------

       KeyError                                  Traceback (most recent call␣
↪last)
```

```
       ~/.local/lib/python3.6/site-packages/pandas/core/indexes/base.py in␣
↪get_loc(self, key, method, tolerance)
      2897                 try:
   -> 2898                     return self._engine.get_loc(casted_key)
      2899                 except KeyError as err:


       pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()


       pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()


       pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()


       pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()


       KeyError: 'price'


   The above exception was the direct cause of the following exception:


       KeyError                                  Traceback (most recent call␣
↪last)

       <ipython-input-17-1ebf642a26c6> in <module>()
   ----> 1 threshold = df['price'].median()
         2 df['is_premium'] = (df['price'] > threshold).astype(int)
         3 df.head()
         4 df["host_response_rate"]


       ~/.local/lib/python3.6/site-packages/pandas/core/frame.py in␣
↪__getitem__(self, key)
      2904             if self.columns.nlevels > 1:
      2905                 return self._getitem_multilevel(key)
   -> 2906             indexer = self.columns.get_loc(key)
      2907             if is_integer(indexer):
      2908                 indexer = [indexer]
```

```
    ~/.local/lib/python3.6/site-packages/pandas/core/indexes/base.py in␣
↪get_loc(self, key, method, tolerance)
       2898                 return self._engine.get_loc(casted_key)
       2899             except KeyError as err:
    -> 2900                 raise KeyError(key) from err
       2901
       2902         if tolerance is not None:


    KeyError: 'price'
```

[6]:
```
#checking columns for null values
np.sum(df.isnull(), axis = 0)
```

[6]:
```
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
is_premium                                          0
dtype: int64
```

[7]:

```
df.dtypes
```

[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
is_premium                                        int64
dtype: object
```

[11]:

```
#dropping columns that aren't relevant/helpful
df.drop(columns = ['price','description',␣
 ↪'name','neighborhood_overview','host_about','host_name','host_location','amenities'],␣
 ↪inplace=True)
```

[19]:

```
df.drop(columns=['host_response_rate', 'host_acceptance_rate'], inplace=True)
```

[18]:

```
df['host_response_rate']
```

[18]:

```
0        0.80
1        0.09
2        1.00
3        1.00
4         NaN
         …
28017    1.00
28018    0.91
28019    0.99
```

```
28020    0.90
28021     NaN
Name: host_response_rate, Length: 28022, dtype: float64
```

[21]:

```
#replacing null values with the median
df['beds'].fillna(df['beds'].median(), inplace=True)
df['bedrooms'].fillna(df['beds'].median(), inplace=True)
```

[22]:

```
np.sum(df.isnull(), axis = 0)
```

[22]:

```
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
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
is_premium                                      0
dtype: int64
```

[23]:
```
#one hot encoding columns that have data type of object
to_encode = list(df.select_dtypes(include=['object']).columns)
to_encode
df[to_encode].nunique()
```

[23]:
```
neighbourhood_group_cleansed    5
room_type                       4
dtype: int64
```

[24]:
```
for colname in to_encode:
    df_encoded = pd.get_dummies(df[colname], prefix=colname +'_')
    df = df.join(df_encoded)
```

[25]:
```
df.head()
```

[25]:
```
   host_is_superhost  host_listings_count  host_total_listings_count  \
0               True                  8.0                        8.0
1               True                  1.0                        1.0
2               True                  1.0                        1.0
3               True                  1.0                        1.0
4               True                  1.0                        1.0

   host_has_profile_pic  host_identity_verified neighbourhood_group_cleansed  \
0                  True                    True                     Manhattan
1                  True                    True                      Brooklyn
2                  True                    True                      Brooklyn
3                  True                    True                     Manhattan
4                  True                    True                     Manhattan

         room_type  accommodates  bathrooms  bedrooms  …  is_premium  \
0  Entire home/apt             1        1.0       1.0  …           1
1  Entire home/apt             3        1.0       1.0  …           0
2  Entire home/apt             4        1.5       2.0  …           1
3     Private room             2        1.0       1.0  …           0
4     Private room             1        1.0       1.0  …           0

   neighbourhood_group_cleansed__Bronx  \
0                                    0
1                                    0
2                                    0
3                                    0
```

```
4                                    0

   neighbourhood_group_cleansed__Brooklyn  \
0                                       0
1                                       1
2                                       1
3                                       0
4                                       0

   neighbourhood_group_cleansed__Manhattan  \
0                                        1
1                                        0
2                                        0
3                                        1
4                                        1

   neighbourhood_group_cleansed__Queens  \
0                                     0
1                                     0
2                                     0
3                                     0
4                                     0

   neighbourhood_group_cleansed__Staten Island  room_type__Entire home/apt  \
0                                            0                           1
1                                            0                           1
2                                            0                           1
3                                            0                           0
4                                            0                           0

   room_type__Hotel room  room_type__Private room  room_type__Shared room
0                      0                        0                       0
1                      0                        0                       0
2                      0                        0                       0
3                      0                        1                       0
4                      0                        1                       0

[5 rows x 50 columns]
```

[26]:

```
df.drop(columns = to_encode ,axis=1, inplace=True)
```

[27]:

```
df.isnull().values.any()
```

[27]:

```
False
```

[30]:

```
features = list(df.loc[:, df.columns != 'is_premium'])
features
```

```
[30]: ['host_is_superhost',
       'host_listings_count',
       'host_total_listings_count',
       'host_has_profile_pic',
       'host_identity_verified',
       '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',
       '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',
       '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']
```

1.4 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.

Yes there is a new features list in the features variable shown in the cell above. I removed the ones that wouldn't have much significance (including host info), one hot encoded object columns, and for numerical columns I replaced the null values with the median/mean in order to obtain columns without any null values. I will use a DecisionTreeClassifier for this problem. I split the dataset with a 80/20 for the training and test sets. I will also go through different hyperparameters including max_depth to try to obtain the best/highest accuracy model. Finally, I will review feature importances to see which ones contributed most to the label.

1.5 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.

[31]:

```python
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, classification_report
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
```

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.

[34]:

```python
y = df['is_premium']
X = df[features]
```

[35]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=123)
```

[40]:

```
model = DecisionTreeClassifier(random_state=123)
model.fit(X_train, y_train)
preds = model.predict(X_test)
proba = model.predict_proba(X_test)[:, 1]
print("Accuracy:", accuracy_score(y_test, preds))
print("AUC:",      roc_auc_score(y_test, proba))
```

```
Accuracy: 0.7637823371989295
AUC: 0.7641754897758394
```

[41]:

```
param_grid = {
    'max_depth': [None, 5, 10, 20],
    'min_samples_leaf': [1, 5, 10],
    'criterion': ['gini', 'entropy']
}
```

[42]:

```
grid_dt = GridSearchCV(
    DecisionTreeClassifier(random_state=123),
    param_grid, cv=5, scoring='roc_auc', n_jobs=-1
)
```

[43]:

```
grid_dt.fit(X_train, y_train)
best_dt = grid_dt.best_estimator_
print("Best DT params:", grid_dt.best_params_)
print("Best DT CV AUC:", grid_dt.best_score_)
```

```
Best DT params: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf':
10}
Best DT CV AUC: 0.8873315123579278
```

[45]:

```
importances = pd.Series(best_dt.feature_importances_, index=X_train.columns)
print("\nTop 10 Features:\n", importances.sort_values(ascending=False).head(10))
```

```

Top 10 Features:
 room_type__Entire home/apt                      0.436553
neighbourhood_group_cleansed__Manhattan         0.100216
bedrooms                                        0.051191
accommodates                                    0.042843
calculated_host_listings_count                  0.041407
review_scores_location                          0.035859
availability_90                                 0.019425
availability_60                                 0.019111
calculated_host_listings_count_private_rooms    0.019012
```

```
availability_365                                0.018037
dtype: float64
```

[46]:

```
feature_imp = best_dt.feature_importances_
df_features = pd.DataFrame({
    'feature': X_train.columns,
    'importance': feature_imp
})
df_sorted = df_features.sort_values(by='importance', ascending=False).
 ↪reset_index(drop=True)
print("All feature importances:\n", df_sorted)
print("\nTop 10 features:\n", df_sorted.head(10))
```

```
All feature importances:
                                           feature  importance
0                       room_type__Entire home/apt    0.436553
1          neighbourhood_group_cleansed__Manhattan    0.100216
2                                         bedrooms    0.051191
3                                     accommodates    0.042843
4                   calculated_host_listings_count    0.041407
5                           review_scores_location    0.035859
6                                  availability_90    0.019425
7                                  availability_60    0.019111
8     calculated_host_listings_count_private_rooms    0.019012
9                                 availability_365    0.018037
10                          minimum_nights_avg_ntm    0.017454
11     calculated_host_listings_count_entire_homes    0.016025
12                          minimum_minimum_nights    0.014038
13                            review_scores_rating    0.013384
14                          maximum_minimum_nights    0.012950
15                       review_scores_cleanliness    0.012182
16                               number_of_reviews    0.012141
17                               reviews_per_month    0.011528
18                                       bathrooms    0.011253
19          neighbourhood_group_cleansed__Brooklyn    0.010705
20                                  maximum_nights    0.010237
21                            n_host_verifications    0.009739
22                             review_scores_value    0.008926
23                             host_listings_count    0.008383
24                                 availability_30    0.007475
25                           review_scores_checkin    0.005365
26                          maximum_maximum_nights    0.005075
27                       host_total_listings_count    0.004909
28                     review_scores_communication    0.004344
29                                            beds    0.003495
30                                  minimum_nights    0.003414
31                           number_of_reviews_ltm    0.003083
32            neighbourhood_group_cleansed__Queens    0.002910
```

```
33                        minimum_maximum_nights    0.002653
34                              instant_bookable    0.002087
35                         room_type__Hotel room    0.001566
36                        maximum_nights_avg_ntm    0.000734
37                        number_of_reviews_l30d    0.000292
38                       room_type__Private room    0.000000
39   neighbourhood_group_cleansed__Staten Island    0.000000
40                             host_is_superhost    0.000000
41           neighbourhood_group_cleansed__Bronx    0.000000
42   calculated_host_listings_count_shared_rooms    0.000000
43                              has_availability    0.000000
44                        host_identity_verified    0.000000
45                          host_has_profile_pic    0.000000
46                        room_type__Shared room    0.000000

Top 10 features:
                                         feature  importance
0                     room_type__Entire home/apt    0.436553
1        neighbourhood_group_cleansed__Manhattan    0.100216
2                                       bedrooms    0.051191
3                                   accommodates    0.042843
4                 calculated_host_listings_count    0.041407
5                         review_scores_location    0.035859
6                                availability_90    0.019425
7                                availability_60    0.019111
8  calculated_host_listings_count_private_rooms    0.019012
9                               availability_365    0.018037
```

From my DecisionTreeClassification modeling, it achieved a value of ROC-AUC of around 0.76. But after tuning my hyperparamters, I found that the best hyperparameters were ’criterion’: ’entropy’, ’max_depth’: 10, ’min_samples_leaf’: 10 after using a param_grid with different hyperparameter values to test out. The AUC value then increased to around 0.887. Then at the end I checked for the features that contributed most (most important) to the label (premium vs budget). The most influential features were listing type (Entire home/apt), location in Manhattan, and number of bedrooms and accommodates.