The Hong Kong University of Science and Technology

COMP 4211: Machine Learning

Group Project Report

Title: Airbnb Improvement Recommendation System

CHAN Yee Ki (20860858) - ykchanbq@connect.ust.hk

LOKESWARA Lovera (20798275) - llokeswara@connect.ust.hk

Lecture Section: L2

Division of Labour: 50% CHAN Yee Ki and 50% LOKESWARA Lovera

1. Overview and Proposal: LOKESWARA Lovera and CHAN Yee Ki
2. Data Exploration—Dataset1: CHAN Yee Ki
3. Data Exploration—Dataset2: LOKESWARA Lovera
4. Preprocessing: CHAN Yee Ki and LOKESWARA Lovera
5. Models: LOKESWARA Lovera
6. Regression: LOKESWARA Lovera
7. Classification: CHAN Yee Ki
8. Performance Enhancement: CHAN Yee Ki
9. Conclusion: LOKESWARA Lovera and CHAN Yee Ki
10. Report: LOKESWARA Lovera and Chan Yee Ki
11. PowerPoint Presentation: LOKESWARA Lovera and Chan Yee Ki
12. Video Presentation: LOKESWARA Lovera and Chan Yee Ki
13. Overall Discussion: We did all the discussion together for the code, report, and PowerPoint presentation together and review everything together.

1. Background and Overview

1.1 Project Background

We are creating a model to predict the score of an Airbnb listing. Using the data we got from the training data, we will give recommendations for Airbnb owners to increase their listing's review score and enhance their Airbnb business.

Since there are so many Airbnb listings and so many features that might affect the score of a listing, it creates a competitive market in the Airbnb industry. This raises some concerns for some Airbnb owners, so we would like to analyze how the listing's features affect the review scores because these scores also affect the Airbnb's rental rate (which also affects the owner's income rate). Thus, we are making a model to predict the listing's score based on its features.

The final purpose of this model is to find and give suggestions to the Airbnb owners on how they can improve their listing's review score. For example, if we find that there are two properties of the same type and one of them has a higher score, then we can analyze the difference in the features between these 2 properties and suggest the other owner to upgrade their property according to the first property listing. To improve our accuracy in giving suggestions, we are using 2 different datasets with several different models to see the effectiveness of our models.

We also want to note that there are some adjustments made to the target because of the nature of rating system. After consulting with the instructor, we believe that it is justifiable to remove the perfect ratings because it is not valuable for our interpretation as most people will give a rating of 5 even though that is not what they truly feel. Aside from that, we also try to remove some bad ratings because of imbalanced class. Thus, we made all these adjustments to increase our model's performance.

We are creating this project using the GPU, RAM, and disk space in Google Colab. We also used Google Colab Pro to create the project as it has less limitations regarding the GPU usage.

2. Data Exploration and Preparation

2.1 Dataset 1 (Kaggle Dataset Link)

2.1.1 Dataset 1 Overview

After we analyzed the dataset, we decided to drop some of the columns. The size of the used dataset: 28 columns with 6173 entries. There are 26 feature columns and 2 target columns (one regression target and one classification target).

Features Columns: ['Account_life', 'host_response_time', 'host_is_superhost', 'host_listings_count', 'host_identity_verified', 'neighbourhood_cleansed', 'property_type', 'room_type', 'accommodates', 'bathrooms_total', 'Bathroom_type', 'bedrooms', 'beds', 'amenities', 'price', 'minimum_nights', 'maximum_nights', 'has_availability', 'number_of_reviews', 'Latest_review', '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']

Target Columns: [‘review_scores_rating’, ‘classification_target’]

2.1.2 Missing Values

Here is a list of columns that have NaN values: ['host_response_time', 'bathrooms_total', 'bathroom_type', 'bedrooms', 'beds', 'latest_review', 'reviews_per_month', 'review_scores_rating', ‘classification_target’]. Here are the number of missing values in the whole dataset:

Column Name	Number of Null	Portion of Null
host_response_time	1481	0.23991576219018307
bathrooms_total	15	0.0024299368216426373
Bathroom_type	4748	0.7691560019439495
bedrooms	314	0.05086667746638587
beds	91	0.014741616717965332
Latest_review	590	0.0955775149846104
reviews_per_month	590	0.0955775149846104
review_scores_rating	590	0.0955775149846104
classification_target	590	0.0955775149846104

Note: After removing the NaN values in the review_scores_rating (regression target), we have a total of 5583 entries. This is important because this is not considered as valid data because we do not have the target values.

2.1.3 Feature Distribution

Numerical features:

Continuous Numerical Features: [‘bathrooms_total’, ’bedrooms’, ‘beds’, ‘price’, ‘latest_review’, ‘reviews_per_month’]

Discrete Numerical Features: [‘Account_life’, ‘ host_listings_count’, ’accommodates’, ‘minimum_nights’, ‘maximum_nights’, ‘number_of_reviews’, ‘calculated_host_listings_count’, ‘calculated_host_listings_count_entire_homes’, ‘calculated_host_listings_count_private_rooms‘, ‘calculated_host_listings_count_shared_rooms’]

The distribution of the first 3 numerical features:

	Account_life	host_listings_count	accommodates
count	3657	3657	3657
mean	2559.948045	2.441072	2.980859
std	884.920354	12.632899	1.547096
min	35	0	1
25%	2078	1	2
50%	2695	1	2
75%	3228	2	4
max	5002	372	16

The following visualizes the first 3 numerical features by using box plot:

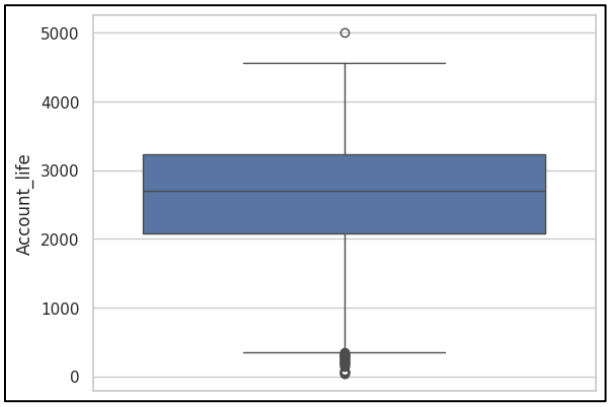

Figure 1. Boxplot Account_life

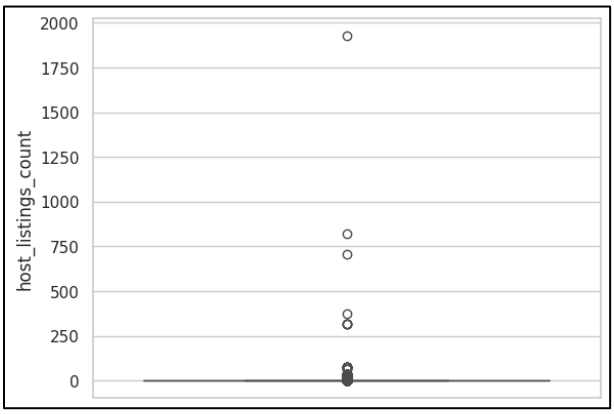

Figure 2. Boxplit host_listings_count

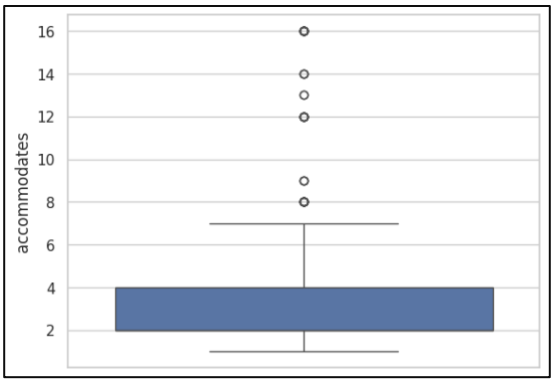

Figure 3. Boxplot Accommodates

Categorical Features:

By using the 'nunique' function in pandas to check the number of distinct elements of the feature, binary categorical features and other categorical features can be distinguished.

If the feature only has 2 distinct elements, then it will be identified as a binary categorical feature. Otherwise, the feature requires to identify it as ordinal or nominal manually by checking their distinct element.

Binary: ['host_is_superhost', 'host_identity_verified', 'has_availability', 'instant_bookable']
Nominal: ['neighbourhood_cleansed', 'property_type', 'room_type', 'bathroom_type', 'amenities']
Ordinal: ['host_response_time']

The distribution of the first 3 categorical features:

	host_response_time	host_is_superhost	host_identity_verified
count	2341	3132	3132
unique	4	2	2
top	within an hour	f	t
freq	1337	2415	2585

The following are the visualized the distribution of the first 3 categorical features by bar plot:

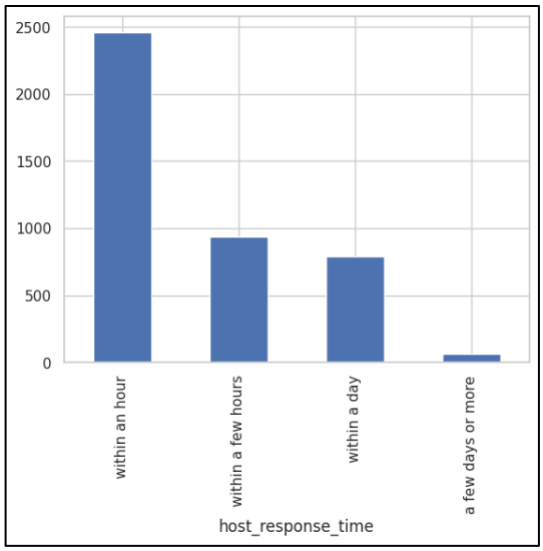

Figure 4. Bar plot host_response_time

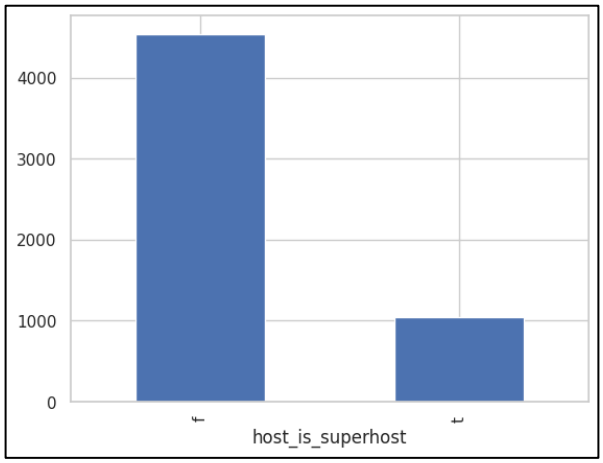

Figure 5. Bar plot host_is_superhost

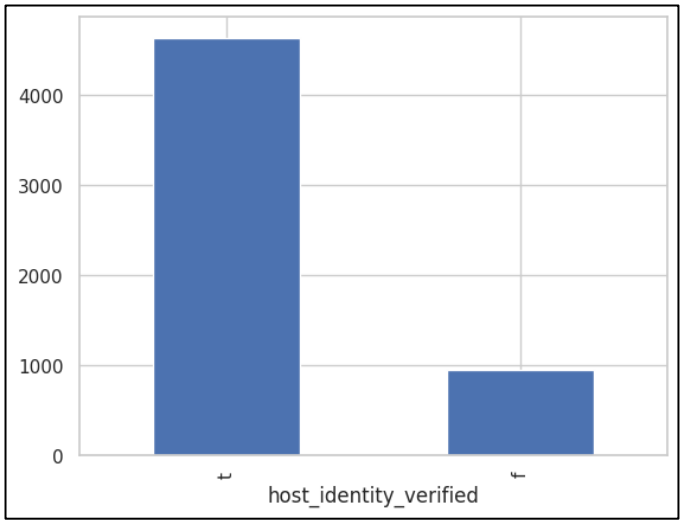

Figure 6. Bar plot host_identity_verified

2.1.4 Outliers

Here are the number of outliers in the numerical data columns.

Table 1. Outliers in Dataset 1

Column Name	Number of Outliers
host_listings_count	6
accommodates	59
price	105
minimum_nights	9
number_of_reviews	130
latest_review	86
calculated_host_listings_count	141
calculated_host_listings_count_entire_homes	102
calculated_host_listings_count_private_rooms	136
calculated_host_listings_count_shared_rooms	39
reviews_per_month	41

We removed the outliers in dataset 1 to increase our model's performance. The total outliers data removed is 525 data.

2.1.5 Correlation Analysis

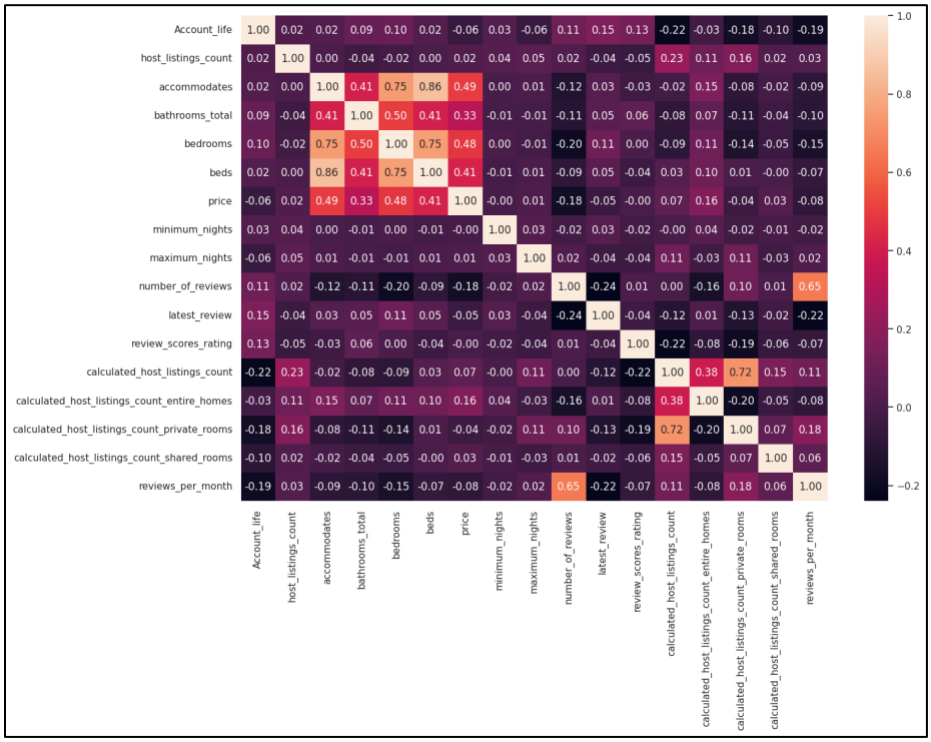

Figure 7. Correlation Matrix

2.1.6 Adjustments made to the Regression Target

Because our dataset does not have a balanced amount of data between the 0-5 ratings, we consulted with the instructor and decided to remove the data with ratings less than 4.5. Then, we scaled it using MinMaxScaler() to be in the range of (0, 5).

After that, we removed the data that still has the ratings = 5 because after consulting with the instructor, we decided to not focus too much on the "perfect" ratings because it does not have any valuable implications knowing that most people will give a rating = 5 if even if they don't think it's perfect.

As a result, our regression target has the following distribution, which is more balanced:

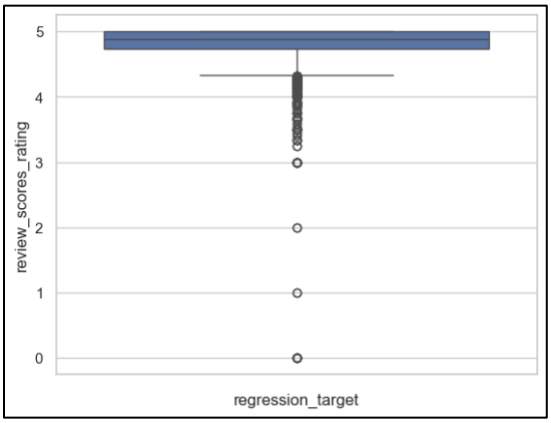

Figure 8. After adjustments

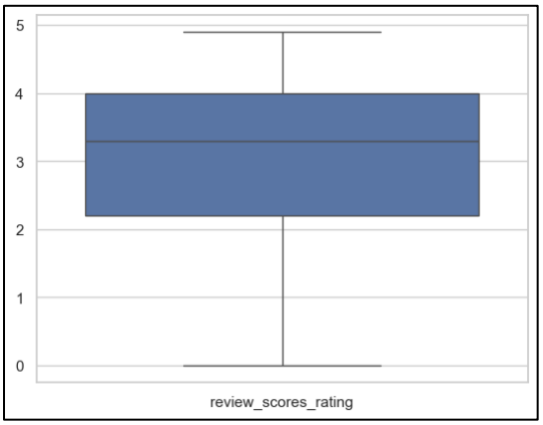

Figure 9. Before adjustments

2.2 Dataset 2 (Kaggle Dataset Link)

2.2.1 Dataset 2 Overview

This dataset is cleaned and there are 20 columns with 40078 entries. There are 18 feature columns and 2 target columns (one regression target and one classification target).

Feature columns: ['City', 'Price', 'Day', 'Room Type', 'Shared Room', 'Private Room', 'Person Capacity', 'Superhost', 'Multiple Rooms', 'Business', 'Cleanliness Rating', 'Bedrooms', 'City Center (km)', 'Metro Distance (km)', 'Attraction Index', 'Normalised Attraction Index', 'Restraunt Index', 'Normalised Restraunt Index']

Target columns: [‘regression_target’, ‘classification_target__2’]

2.2.2 Missing Values

There is no missing value in this dataset.

2.2.3 Feature Distribution

Numerical Features:

Discrete Numerical Feature: ['Multiple Rooms', 'Business', 'Bedrooms']

Continuous Numerical Feature: ['Price', 'Person Capacity', 'Cleanliness Rating', 'City Center (km)', 'Metro Distance (km)', 'Attraction Index', 'Normalised Attraction Index', 'Restraunt Index', 'Normalised Restraunt Index']

	Price	Person Capacity	Multiple Rooms
count	40078	40078	40078
mean	260.385731	3.241654	0.298318
std	282.724361	1.300120	0.457526
min	34.779339	2	0
25%	144.016085	2	0
50%	203.881325	3	0
75%	296.389566	4	1
max	18545.450285	6	1

The following visualizes the first 3 numerical features by using box plot:

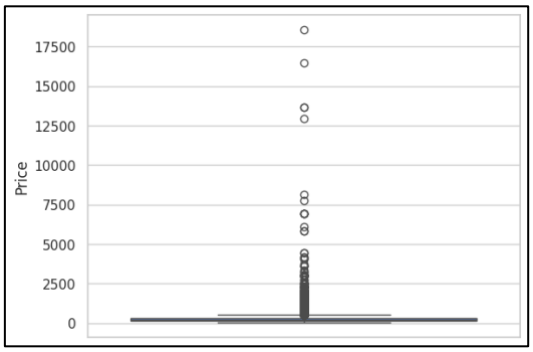

Figure 10. Boxplot Price

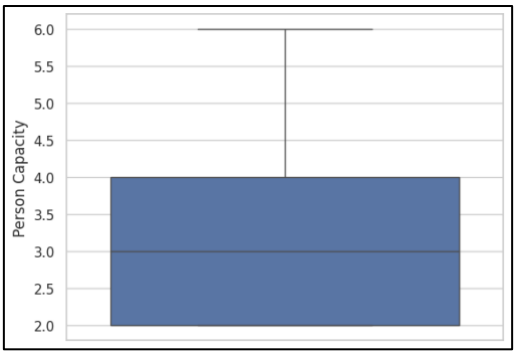

Figure 11. Boxplot Person Capacity

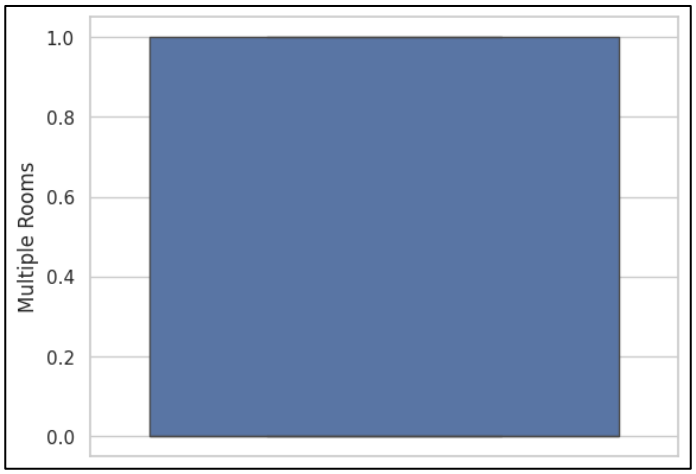

Figure 12. Boxplot Multiple Rooms

Categorical Features:

Nominal: ['City', 'Day', 'Room Type']

Binary: ['Shared Room', 'Private Room', 'Superhost']

	City	Day	Room Type
count	40078	40078	40078
unique	9	2	3
top	Rome	Weekday	Entire home/apt
freq	8713	20063	27237

The following visualized the distribution of the first 3 categorical features by bar plot:

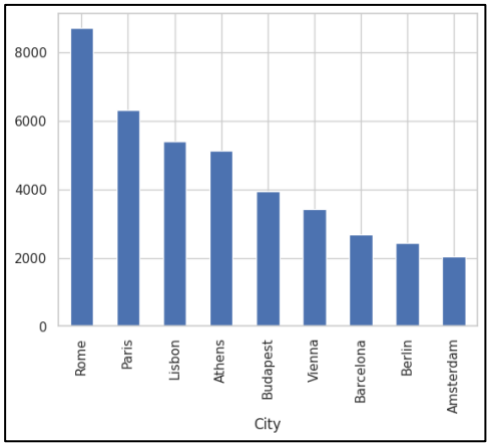

Figure 13. Bar plot City

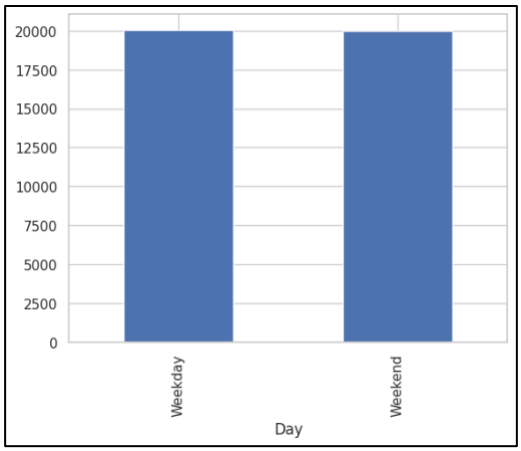

Figure 14. Bar plot Day

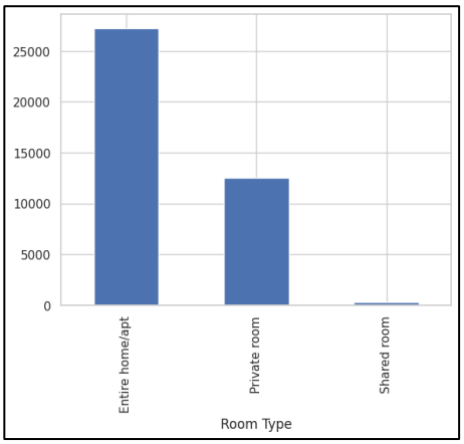

Figure 15. Bar plot Room Type

2.2.4 Outliers

Here are the number of outliers in the numerical data columns.

Table 2. Outliers in Dataset 2

Column Name	Number of Outliers
Price	356
Cleanliness Rating	520
Bedrooms	86
City Center (km)	598
Metro Distance (km)	833
Attraction Index	540
Normalised Attraction Index	502
Restraunt Index	610
Normalised Restraunt Index	236

We removed the outliers in dataset 2 to increase our model's performance. The total outliers data removed is 4134.

2.2.5 Correlation Analysis

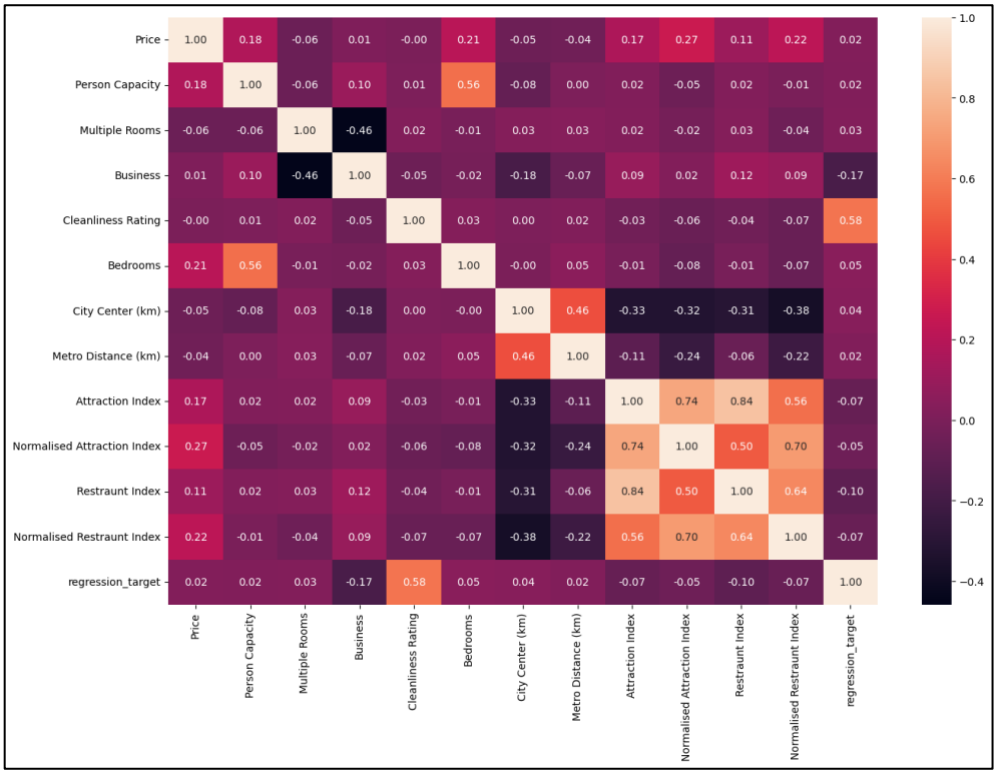

Figure 16. Correlation Matrix

2.2.6 Adjustments made to the Regression Target

Because our dataset does not have a balanced amount of data between the 0-5 ratings, we consulted with the instructor and decided to remove the data with ratings less than 4. Then, we scale it using MinMaxScaler() to be in the range (0, 5).

After that, we removed the data that still has the ratings = 5 because after consulting with the instructor, we decided to not focus too much on the "perfect" ratings because it does not have any valuable implications knowing that most people will give a rating = 5 if even if they don't think it's perfect.

As a result, our regression target has the following distribution, which is more balanced:

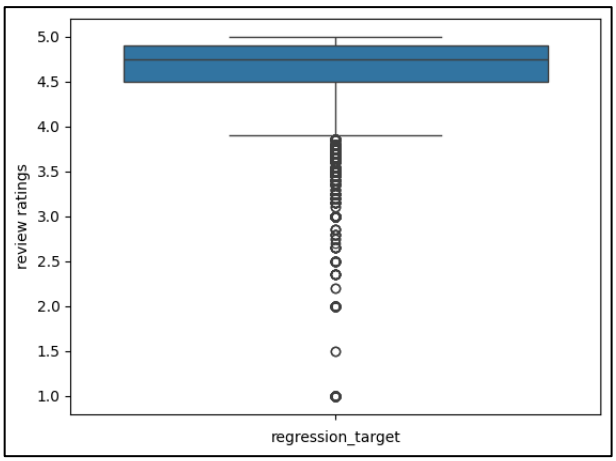

Figure 17. Before Adjustments

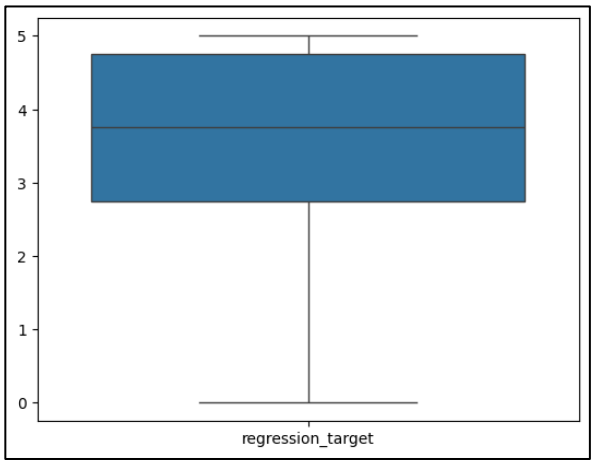

Figure 18. After Adjustments

3. Preprocessing Step

3.1 Preprocessing Dataset 1

3.1.1 Handle Missing Values

We are using different type of imputers for different columns.

Table 3. Imputers used

Column Name	Imputer Used
host_response_time	constant
bathrooms_total	mean
bathroom_type	most_frequent
bedrooms	mean
beds	mean

3.1.2 Normalization and Standardization

We normalize and standardize the numerical column using RobustScaler(). We chose to do this because it is not very sensitive to outliers. This scaler is based on percentiles and the resulting range is going to be larger than other scalers. We apply this scaler to all the numerical features.

3.1.3 Encoding Categorical Features

We are encoding the categorical features using different encoders. The binary and nominal columns are going to be encoded using the One-Hot Encoder and the Ordinal column is going to be encoded using the Ordinal Encoder.

Binary: [‘host_is_superhost’, ‘host_identity_verified’, ‘has_availability’, ‘instant_bookable’]
Nominal: [‘neighbourhood_cleansed’, ‘property_type’, ‘room_type’, ‘bathroom_type’]
Ordinal: [‘host_response_time’]

a. ‘Amenities’ Column

We did a special encoding process for this column. When we see the values of this column, we must get the One-Hot encoding for each of the items that exist in this column. After we did a One-Hot encoding, we see that there’s a total of 1287 unique amenities that exist in various Airbnb listings. So, to reduce the number of amenities, we check which amenities are most related to the target by using selectKBest (where k = 50) to find the best 50 amenities that are most related to the regression target.

3.1.4 Feature Selection

We did a feature selection for the ‘amenities’ column. We select the best 50 amenities to be concatenated with the other features. We also did another feature selection to select the best 50 features for model 1 and 40 features for model 5 among the features selected.

The feature selection method that we use is Select K Best features. We use this because we want to specify the number of features we will use for our models. We do not want to use too many features that are uncorrelated to the regression target.

3.2 Preprocessing Dataset 2

3.2.1 Handle Missing Values

There are no missing values in the whole dataset, so we do not use any imputer on this dataset.

3.2.2 Normalization and Standardization

After trying all the scalers, we decided to normalize and standardize the numerical column using RobustScaler() because it gives us the best performance result for our models. We chose to do this because it is not very sensitive to outliers. This scaler is based on percentiles and the resulting range is going to be larger than other scalers. We apply this scaler to all the numerical features.

3.2.3 Encoding Categorical Features

For this dataset, we analyzed the categorical features and decided to use One-Hot encoder because we do not have any ordinal categorical features. We apply this encoder to all the categorical features below:

Nominal: ['City', 'Day', 'Room Type']
Binary: ['Shared Room', 'Private Room', 'Superhost']

3.2.4 Feature Selection

In this dataset, we do not have any columns like amenities column in dataset 1, so we do not need to do special feature selection on one of the features, like what we did for dataset 1. However, we still did feature selection to select the best 20 features for model 1 and model 5.

The feature selection method that we use is Select K Best features. We use this because we want to specify the number of features we will use for our models. We do not want to use too many features that are uncorrelated to the regression target.

4. Models

The following models will be used in both regression and classification for all datasets.

4.1 Model 1 (Features: Best K out of all features)

Dataset 1: We select the best 50 amenities features and combine them with the other features. Among these new features, we are selecting the best 50 features to be the inputs and the regression and classification targets to be the target columns. This model has 50 features.

Dataset 2: We select the best 20 features out of all the features. This model has 20 features.

4.2 Model 2 (Features: Neighborhoods)

Dataset 1: We only use the neighborhood feature as the input. The neighborhood is represented as One-Hot encoded features. We do this to check whether the neighborhood affected the target variables or not. This model has 22 features.

Dataset 2: We only use the city feature as the input. The city feature is represented as One-Hot encoded features. We do this to check whether the neighborhood affected the target variables or not. This model has 9 features.

4.3 Model 3 (Features: Amenities; Only applied to Dataset 1)

We only use the amenities feature as the input. The amenities are represented as One-Hot encoded features. We do this to check which amenities affect the target variables.

4.3.1 Variation 1 (All Amenities)

We use the whole amenities features as input and the regression and classification targets as output. This model has 884 features.

4.3.2 Variation 2 (Best 50 Amenities)

We only use the best 50 amenities features as input and the target variables as output. We chose the best 50 amenities by using the Select K Best feature selection. This model has 50 features.

4.4 Model 4 (Features: All features, excluding the neighborhood and amenities)

Dataset 1: We are using all the features, excluding the neighborhood and the amenities. We do this to check the correlation between all the other features and the regression target. This model has 78 features.

Dataset 2: We are using all the features, excluding the city. We do this to check the correlation between all the other features and the regression target. This model has 22 features.

4.5 Model 5 (Features: The Best K out of the features in Model 4)

We chose this model to try and increase our model's accuracy by choosing the K most related features to the target variables.

Dataset 1: We choose the best 40 features out of the features selected in Model 4. We want to improve our model efficiency by excluding the least correlated features in Model 4. This model has 40 features.

Dataset 2: We choose the best 20 features out of the features selected in Model 4. We want to improve our model efficiency by excluding the least correlated features in Model 4. This model has 20 features.

4.6 Model 6 (Features: Selected features only)

We created this model by selecting only the features that we thought might have the highest correlation with the target variables.

Dataset 1: After analyzing the dataset, we chose the features that might be highly correlated with the target variables. The selected features are: [calculated_host_listings_count, calculated_host_listings_count_entire_homes, calculated_host_listings_count_private_rooms, calculated_host_listings_count_shared_rooms, price, bathrooms_total, bedrooms, latest_review, and the categorical features (excluding the neighborhood and amenities)]. This model has 10 features.

Dataset 2: After analyzing the dataset, we chose the features that might be highly correlated with the target variables. The selected features are: ['Price', 'Person Capacity', 'Superhost', 'Multiple Rooms', 'Business', 'Cleanliness Rating', 'Bedrooms', 'City Center (km)', 'Metro Distance (km)', 'Normalised Attraction Index', 'Normalised Restraunt Index', 'Day_Weekday', 'Day_Weekend', 'Room Type_Entire home/apt', 'Room Type_Private room', 'Room Type_Shared room', 'Shared Room_True', 'Private Room_True']. This model has 18 features.

4.7 Model 7 (Features: All Features; Only applied to Dataset 2)

Our dataset 2 has less features than dataset 1, so we decided to try the model with all the features for dataset 2. We used this model to find the overall correlation between all the features and the target variables. This model has a total of 31 features.

5. Regression

In the regression model, we are going to choose the model with the best R2 score because it is the metric that measures the proportion of the variance in the dependent variable that is explained by the independent variable in the regression model. A higher R2 score means that the model explains the variance better, thus is a better fit.

5.1 Feedforward Neural Networks

We tried using 4 layers for several hidden units (1, 8, 32, 64, 128, 300) and after analyzing, the model with 128 hidden performed the best. We set the other parameters as default. Thus, here are the results for the model with 4 **layers and 128 hidden units**:

Table 4. Feedforward Neural Networks (R2 Score, MSE)

	Dataset 1	Dataset 2
Model 1	-0.0661; 1.4660	0.3726; 1.0477
Model 2	-0.0243; 1.4086	0.0437; 1.5969
Model 3 (1)	0.0497; 1.3067	-
Model 3 (2)	0.0540; 1.3008	-
Model 4	-0.0345; 1.4226	0.3709; 1.0504
Model 5	-0.0971; 1.5087	0.3770; 1.0404
Model 6	0.1097; 1.2242	0.3618; 1.0657
Model 7	-	**0.3864; 1.0246**

The best model for dataset 1 when using Feedforward Neural Networks is model 6. It is the best in terms of R2 and MSE.

The best model for dataset 2 when using Feedforward Neural Networks is the model 7. It is also the best in terms of R2 and MSE.

5.2 Other Regression Methods Used

Table 5. Linear Regression (R2 Score, MSE)

	Dataset 1	Dataset 2
Model 1	**0.2152; 1.1894**	0.3663; 1.0581
Model 2	-0.0204; 1.5464	0.0440; 1.5964
Model 3 (1)	-3.8784; 5.8779	-
Model 3 (2)	0.0743; 1.4029	-
Model 4	0.1399; 1.3036	0.3619; 1.0655
Model 5	0.1606; 1.2722	0.3663; 1.0581
Model 6	0.1574; 1.2771	0.3574; 1.0730
Model 7	-	0.3747; 1.0441

We set the parameter settings in the Linear Regression function as default.

The best model for dataset 1 when using Linear Regression is model 1. It is the best in terms of R2 and MSE.

The best model for dataset 2 when using Linear Regression is the model 1 and model 5. It is also the best in terms of R2 and MSE.

Table 6. Polynomial (logarithmic) Regression (R2, MSE)

	Dataset 1	Dataset 2
Model 1	-0.0054; 1.5238	0.3584; 1.0713
Model 2	-0.0072; 1.5265	0.0435; 1.5972
Model 3 (1)	0.07437; 1.4029	-; -
Model 3 (2)	0.0653; 1.4167	-; -
Model 4	-0.0054; 1.5238	0.3545; 1.0778
Model 5	-0.0054; 1.5238	0.3584; 1.0713
Model 6	0.1120; 1.3458	0.3506; 1.0843
Model 7	-; -	0.3473; 1.0898

For the polynomial regression, we use the following settings: TweedieRegressor(power=1, alpha=0.5, link='log').

The best model for dataset 1 when using Logistic Regression is model 6. It is the best in terms of R2 and MSE.

The best model for dataset 2 when using Logistic Regression is the model 1 and model 5. It is also the best in terms of R2 and MSE.

Table 7. Stochastic Gradient Descent (SGD) Regression (R2 Score, MSE)

	Dataset 1	Dataset 2
Model 1	-1.2950; 1.9567	0.3654; 1.0596
Model 2	0.0046; 1.5351	0.0413; 1.6009
Model 3 (1)	0.1298; 1.4140	-; -
Model 3 (2)	0.0659; 1.4053	-; -
Model 4	-4.3218e+30; 6.5300e+30	0.3574; 1.0730
Model 5	-5.9373e+30; 8.9710e+30	0.3634; 1.0630
Model 6	0.1198; 1.2805	0.3557; 1.0758
Model 7	-; -	0.3747; 1.0441

For the SGD regression, we use the following settings: SGDRegressor(max_iter=1000, tol=1e-3).

The best model for dataset 1 when using SGD Regression is model 3 (1). It is the best in terms of R2. However, the best model in terms of MSE is model 6.

The best model for dataset 2 when using SGD Regression is the model 7. It is also the best in terms of R2 and MSE.

5.3 Deep Neural Networks

We built a deep neural network model by using dense and dropout layers in TensorFlow.keras.layers. In this deep neural network model, there are 17 layers. For the first 16 layers, there will be 8 dense layers and each dense layer connected with a dropout layer, where the dropout rate is 0.3. Each dense layer corresponds to the following different hidden units, including 300, 500, 300, 300, 300, 300, 500, 300. The dense layer used the exponential linear unit activation function. Such that the neural network can learn the features slower and deeper by this setting. The last layer only contains 1 hidden unit and uses a linear activation function.

Table 8. Using Deep Neural Networks with Dropout (R2 Score, MSE)

	Dataset 1	Dataset 2
Model 1	0.0470; 1.4533	-0.0025; 1.8066
Model 2	-0.0395; 1.5717	-0.0212; 1.7921
Model 3 (1)	0.0106; 1.4606	-; -
Model 3 (2)	0.0353; 1.5155	-; -
Model 4	0.0367; 1.5243	-0.2355; 2.2054
Model 5	0.0265; 1.5224	-0.0289; 1.8348
Model 6	0.1252; 1.3016	0.3108; 1.2314
Model 7	-; -	-0.04702; 1.7459

The best model for dataset 1 when using Deep Neural Network with Dropout is model 6. It is the best in terms of R2 and MSE.

The best model for dataset 2 when using Deep Neural Network with Dropout is the model 6. It is also the best in terms of R2 and MSE.

5.4 Conclusion for Regression

5.4.1 Dataset 1

After analyzing all the models with different regressions, we decided that the best model for dataset 1 is **Model 1 using Linear Regression**. The R2 score for this setting is: 0.2152 and the MSE is: 1.1894. So, we are using this model to predict the regression target of dataset 1.

5.4.2 Dataset 2

After analyzing all the models with different regressions, we decided that the best model for dataset 2 is **Model 7 using Feedforward Neural Network with 128 hidden units and 4 layers**. The R2 score for this setting is: 0.3864 and the MSE is: 1.0246. So, we are using this model to predict the regression target of dataset 2.

6. Classification

Since the class distribution is balanced, and the cost of misclassification is equal to all classes, we should decide the best model based on accuracy instead of the F1 score.

6.1 Logistic Regression (Accuracy, F1) ROC, AUC Curve

Table 9. Logistic Regression with 0.1 step size for different models (Accuracy; F1 Score)

	Dataset 1	Dataset 2
Model 1	0.4705; 0.6399	0.7888; 0.8363
Model 2	0.4896; 0.5137	0.6290; 0.7380
Model 3 (1)	0.5726; 0.3431	-; -
Model 3 (2)	0.5742; 0.6616	-; -
Model 4	0.5295; 0.0	0.7767; 0.8355
Model 5	0.4705; 0.6399	0.7887; 0.8332
Model 6	0.6061; 0.6308	0.7919; 0.8387
Model 7	-; -	0.8312; 0.2977

The following is the ROC graph for the best model in both datasets.

Dataset 1

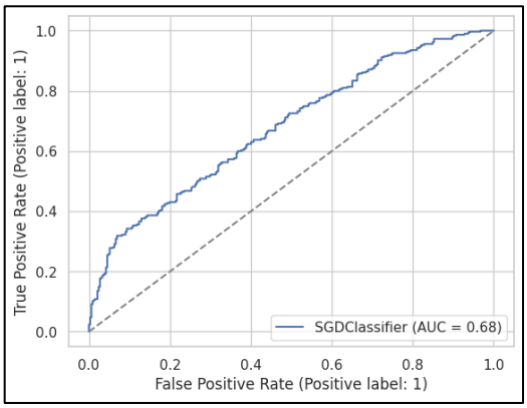

Figure 19. ROC Curve for Dataset 2

Dataset2

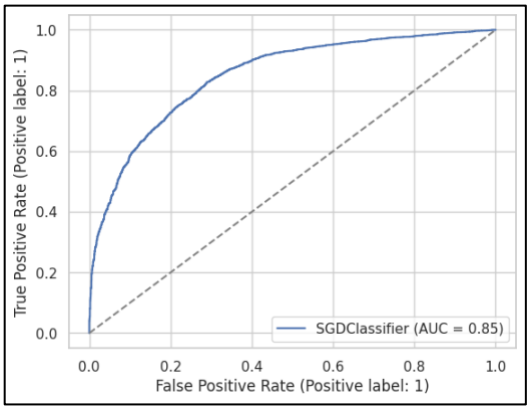

Figure 20. ROC Curve for Dataset 1

The model with the highest accuracy and F1 scores in dataset 1 is model 6 and model 3(2) respectively. Therefore, the best logistic regression model in dataset 1 is model 6.

The model with the highest accuracy and F1 scores in dataset 2 is model 7 and model 6 respectively. Therefore, the best logistic regression model in dataset 2 is model 7.

6.2 Feedforward Neural Networks

In training feedforward neural network for classification, we used 4 layers in MLPClassifier and tested with different hidden unit sizes, including 1, 8, 32, 64, 128 and 300.

We found out that all the models perform the best with the largest hidden unit size.

Table 10. Classification By Using Feedforward Neural Network by 300 Hidden Units (Mean Accuracy; Mean F1 Score)

	Dataset 1	Dataset 2
Model 1	0.5396; 0.5417	0.7921; 0.8378
Model 2	0.4805; 0.4577	0.6287; 0.7457
Model 3 (1)	0.5773; 0.6220	-; -
Model 3 (2)	0.5810; 0.5496	-; -
Model 4	0.5305; 0.5886	0.7916; 0.8401
Model 5	0.4795; 0.6418	0.7902; 0.8360
Model 6	0.6220; 0.5861	0.7919; 0.8387
Model 7	-; -	0.7938; 0.8396

In dataset 1, the 300 hidden unit feedforward neural network model with the highest mean accuracy and F1 score is **model 6** and **model 5** respectively. Thus, the best model in dataset 1 is **model 6**.

In dataset 2, the feedforward neural network models with the highest accuracy and F1 score is model 7 and model 4, respectively. Therefore, the best feedforward neural network model in dataset 2 is **model 7**.

6.3 Deep Neural Network

We built a deep neural network model by using dense and dropout layers in TensorFlow.keras.layers. In this deep neural network model, there are 17 layers. For the first 16 layers, there will be 8 dense layers and each dense layer connected with a dropout layer, where the dropout rate is 0.3. The dropout rate is 0.3 for all dropout layers. Each dense layer corresponds to the following different hidden units, including 300, 500, 1000, 300, 1000, 300, 500, 300. The dense layer used the exponential linear unit activation function. Such that the neural network can learn the features slower and deeper by this setting. The last layer only contains 1 hidden unit and uses sigmoid activation function.

Table 11. Using Deep Neural Network with Dropout (Accuracy, F1 score)

	Dataset 1	Dataset 2
Model 1	0.4508; 0.6215	0.6163; 0.7626
Model 2	0.5232; 0.0	0.6125; 0.7597
Model 3 (1)	0.5341; 0.0	-; -
Model 3 (2)	0.5396; 0.0	-; -
Model 4	0.4809, 0.6494	0.6286; 0.7719
Model 5	0.5382, 0.0	0.6202; 0.7656
Model 6	0.5368; 0.0.	0.6181; 0.7640
Model 7	-; -	0.7742; 0.6316

In dataset 1, model 3(2) and model 4 have the highest accuracy and F1 score respectively. For dataset 2, the model with the highest accuracy and F1 score is model 7 and model 4 respectively.

Thus, the best deep neural network model in dataset 1 and dataset 2 is model 3(2) and model 7 respectively.

6.4 Conclusion for Classification

Compared to all the classification methods, the best model in dataset 1 is **model 6 in feedforward neural network**. It has the highest accuracy score (0.6220) among all the models and classification methods.

For dataset 2, the best model is **model 7 in feedforward neural network with 300 hidden units**. It has the highest accuracy score (0.8312) among all the models and classification methods.

7. Performance Enhancement

We used Gridsearch to find the best parameter for our chosen model. So, we find the best model based on the regression and classification part (Part 5 and 6). Then, we are using the Gridsearch to find the best parameter to enhance our model's performance.

7.1 Dataset 1

7.1.1 Regression Model

The chosen model is Model 1 with Linear Regression. We tuned the linear regression model (LinearRegression from scikit-learn) with 4 folds of cross-validation, r-squared scoring, and the following parameters.

Table 12. Hyperparameter Tuning for Linear Regression

Tuned Parameters	Value
'fit_intercept'	True, False
'copy_X'	True, False
'n_jobs'	None, 3, 10

Here is the best parameter based on our Gridsearch:
{'copy_X': True, 'fit_intercept': False, 'n_jobs': None}
The resulting R2 score is: **0.2151**
The resulting MSE score is: **1.1894**

7.1.2 Classification model

Since the best classification model in dataset 1 is model 6 by using feedforward neural networks, we tuned the feedforward neural network (MLPClassifier from scikit-learn) with different parameters, early stopping, stochastic gradient descent as solver (solver = 'sgd'), 4 folds of cross validation and 4 layers where each layer contains 300 hidden units.

Table 13. Hyperparameter Tuning for Feedforward Neural Network

Trained Parameters	Value
'activation'	'logistic', 'relu'
'learning_rate'	'constant', 'invscaling'
'learning_rate_init'	0.001, 0.01, 0.1
'max_iter'	100, 300

The best parameter combination for the feedforward neural network in classification is:
{'activation': 'relu', 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'max_iter': 100}
The resulting Accuracy is: **0.5946**
The resulting F1 score is: **0.6184**

7.2 Dataset 2

7.2.1 Regression Model

The chosen model is the model with all the features using feedforward neural networks. We tuned the feedforward neural network model (MLPRegressor from scikit-learn) with 4 folds of cross-validation, r-squared scoring and the following parameters.

Table 14. Hyperparameter Tuning for Feedforward Neural Networks

Tuned Parameters	Value
'activation'	'relu','tanh'
' learning_rate'	'constant'
' learning_rate_init'	0.001, 0.01
'max_iter'	100, 200, 300

Here is the best parameter based on our Gridsearch:
{'activation': 'relu', 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'max_iter': 100}
The resulting R2 score is: **0.3892**
The resulting MSE score is: **1.0199**

7.2.2 Classification model

In dataset 2, the best classification model is the model 7 by using logistic regression.

Table 15. Hyperparameter Tuning for Logistic Regression

Tuned Parameters	Value
'C'	0.01, 0.1, 1, 10, 100
'solver'	'newton-cg', 'lbfgs', 'liblinear'
' penalty'	'l2'

The best parameter combination for the feedforward neural network in classification is:
{'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
The resulting accuracy score is: 0.80
The resulting F1 score is: 0.84

8. Conclusion

When we are doing the Airbnb improvement recommendation model, the rating of the host will not be given. We will only know the features and listings of the host. Based on the given data, we will use our trained classification model to identify whether the host performs good or badly.

If we identify the host performance as bad, then we will use the trained regression model to interpret the coefficients. For the features with positive coefficients, we can know that these features contribute positively to the host rating score. So, the hosts should improve (increase) those features to get a higher score and better performance. On the other hand, the host should avoid contributing (reduce) to the features with negative coefficients.

For example, when we are using model 1 for linear regression to do a regression model, we can get all the coefficients for the linear regression model. The following is the coefficient we get from the model 1 by using linear regression.

```
Feature: Account_life, Coefficient: 1.3325456798811183e-05
Feature: host_listings_count, Coefficient: -0.029987984852399424
Feature: accommodates, Coefficient: -0.23010026296535976
Feature: bathrooms_total, Coefficient: 0.04498057748661205
Feature: beds, Coefficient: -0.016591327996837376
Feature: price, Coefficient: 0.08301512718977647
Feature: maximum_nights, Coefficient: -0.008872502411751663
Feature: latest_review, Coefficient: 0.016269996716159342
Feature: calculated_host_listings_count, Coefficient: -0.18787118040177786
Feature: calculated_host_listings_count_private_rooms, Coefficient: -0.10644043912490776
Feature: host_is_superhost_f, Coefficient: -0.3655027892306457
Feature: host_is_superhost_t, Coefficient: 0.36550278923063195
Feature: neighbourhood_cleansed_Bijlmer-Oost, Coefficient: 0.3554390575122421
Feature: neighbourhood_cleansed_Centrum-West, Coefficient: -0.11926838324256384
Feature: neighbourhood_cleansed_Geuzenveld - Slotermeer, Coefficient: -0.3813008616408299
Feature: neighbourhood_cleansed_IJburg - Zeeburgereiland, Coefficient: 0.2879197961886938
Feature: neighbourhood_cleansed_Osdorp, Coefficient: -0.13063424293382753
Feature: property_type_Entire chalet, Coefficient: -0.9574507955982167
Feature: property_type_Entire guest suite, Coefficient: -0.07541323063269854
Feature: property_type_Entire loft, Coefficient: 0.03305941231437001
Feature: property_type_Entire rental unit, Coefficient: 0.06375280377753824
Feature: property_type_Entire serviced apartment, Coefficient: -1.4302026243111414
Feature: property_type_Private room in bed and breakfast, Coefficient: -0.2610147614375127
Feature: property_type_Private room in condo, Coefficient: -0.019227788024241545
Feature: property_type_Private room in guest suite, Coefficient: 0.09085308363720136
Feature: property_type_Private room in guesthouse, Coefficient: -1.513703804486282
Feature: property_type_Private room in hostel, Coefficient: -0.7909650542703559
Feature: property_type_Private room in houseboat, Coefficient: -0.07257154146809047
Feature: property_type_Private room in rental unit, Coefficient: -0.35777228894038626
Feature: property_type_Private room in villa, Coefficient: 0.3353773839659947
Feature: property_type_Room in boutique hotel, Coefficient: -0.7811679034725525
Feature: property_type_Room in hostel, Coefficient: -0.9703682777629085
Feature: property_type_Room in hotel, Coefficient: -2.292467478943429
Feature: room_type_Entire home/apt, Coefficient: -0.24188226343624444
Feature: room_type_Hotel room, Coefficient: -0.04785778807998716
Feature: bathroom_type_half-, Coefficient: 0.34881476713779125
Feature: bathroom_type_private, Coefficient: -0.10997979020484014
Feature: bathroom_type_shared, Coefficient: -0.23883497693293654
Feature: instant_bookable_f, Coefficient: 0.11670948083590098
Feature: instant_bookable_t, Coefficient: -0.11670948083590432
Feature: amenities_Coffee maker, Coefficient: 0.1657510968230372
Feature: amenities_Dishes and silverware, Coefficient: 0.03552133499281176
Feature: amenities_Essentials, Coefficient: 0.4426934544792231
Feature: amenities_Hangers, Coefficient: 0.08605609156655478
Feature: amenities_Host greets you, Coefficient: 0.12262267745148231
Feature: amenities_Hot water, Coefficient: 0.0016997167607951272
Feature: amenities_Iron, Coefficient: 0.16303781558433525
Feature: amenities_Luggage dropoff allowed, Coefficient: 0.18802912247944215
Feature: amenities_Paid parking off premises, Coefficient: 0.10334056060176659
Feature: amenities_Refrigerator, Coefficient: 0.015281343681569828
```

Figure 21. Coefficients from Linear Regression Model for model 1 dataset 1

We can see that ‘host_is_super_host_t’, ‘amenities_Coffee maker', ‘property_type_Private room in villa’ and others have positive coefficients. This means that if the hosts became super hosts, the listing has a coffee maker, and adding the listing with private room in villa, then the hosts can receive a higher rating score.

Aside from that, we can also use the regression model to predict the review score of a listing by passing the given features to the suitable model, based on their features. If it has the features of the dataset 1, then we pass it to the best-chosen model for dataset 1. Similarly, if it has the features of the dataset 2, then we pass it to the best-chosen model for dataset 2. After that, we will be able to get the predicted scores for the given listing. However, it is important to note that the scores passed when training the model are already processed and scaled. Thus, the resulting score is not going to be on the scale of 1 to 5, but rather from 4.5 to 5 (for dataset 1) and 4 to 5 (for dataset 2). So, to get the actual score, we should do some calculation and scale it to be in the original scale from 0-5. For example, given that we have a score of 2 for an instance of dataset 1, then to get the actual review score, we can divide that with 5 and add 4.5 to it. It will then be the actual score of the given listings. Here is the general formula to find the actual review score:

$$\frac{regression\ result}{5} + lower\ bound = actual\ review\ score$$

Note: The lower bound is 4.5 for data from dataset 1 and 4 for data from dataset 2.

Because the main purpose of our model is to help the host with bad listing’s score, we do not focus on those with high ratings score. Thus, by implementing this method, we will be able to help the hosts that want to increase or improve their Airbnb business.

We hope this project will be able to help solve real world situation and help the Airbnb owners to increase their revenue. We believe this project has never been implemented before as we never found such implementation before. Thus, this project is a new project that we figure can help solve real world problems.