HOTEL RATING PREDICTION

TEAM ID: SC_30
TEAM EMAIL: ahmedmohsen2492@gmail.com

TEAM MEMBERS

Ahmed Esmail Mohamed	20201700024
Ebrahim Tarek Mohamed	20201700009
Ahmed Mohsen Fathelbab	20201700067
Ebrahim Ahmed Abdelmaboud	20201700007
Ahmed Mohamed Kamal	20201700080
Yasmine Khaled Atta	20201701154

PREPROCESSING

1. SPLIT THE DATASET

20% for testing (58063 rows) 80% for training (232252 rows)

2. HOTEL_ADDRESS

It was divided into two columns **(hotel_country, hotel_city)** using regular expressions, pycountry and GeoText.

3. REVIEW_DATE

It was converted to DateTime first then extracted the year only to column **(Review_year)** then Review_Date column converted to ordinal.

4. TEXT CLEANING

Applied to prepare reviews for sentiment analysis

1. Make all text lower case.
2. We noticed that words like(wasn’t, weren’t …..) didn’t contain (’) so we replaced these words with their expanded form (was not, were not).
3. Lemmatization : to convert all words to it’s base form (running → run).
4. Tokenization : convert the sentence to a list of words.
5. Replace contractions : replace all words to it’s expanded form.

5. NEGATIVE_REVIEW, POSITIVE_REVIEW

Applied sentiment analysis to classify if the review positive or negative using vader.

Positive Review:

Negative Review:

6. REVIEW_TOTAL_NEGATIVE_WORD_COUNTS

If the review already negative so it filled by the word count but if the review is positive so it filled by zero.

7. **REVIEW_TOTAL_POSITIVE_WORD_COUNTS**

   If the review already positive so it filled by the word count but if the review is negative so it filled by zero.

8. **TAGS**

   It was divided into multiple columns **(type_of_trip, with_a_pet, people, Room_Type, nights, submitted_from_mobile).**

9. **DAYS_SINCE_REVIEW**

   it contains days or day word after the number and both have been removed then date was converted into numeric value using pd.to_numeric.

10. **LAT, LNG**

    Both contains null data and was filled with average.

11. **LABEL ENCODING (LabelEncoder model)**

    DEFINITION: is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

    Applied on **(Hotel_Name, Reviewer_Nationality, type_of_trip, people, Room_Type, hotel_country, hotel_city, Reviewer_Score).**

12. **FEATURE SELECTION**

    We used anova with k=4 and the top features were **(Average_Score, Review_Total_Negative_Word_Counts, Review_Total_Positive_Word_Counts, type_of_trip).**

12. **HANDLING OUTLIERS**

    We tried to do it but it reduces the accuracy so, we didn't apply it.

13. **FEATURE SCALING (MinMaxScaler)**

    DEFINITION: It involves transforming the feature values of a dataset so that they fall within a specified range, typically between 0 and 1.

    Applied on selected features except Reviewer_Score.

MODELS

Logistic Regression

DEFINITION: is the supervised Machine Learning model in which the model tries to predict the probability that an instance of belonging to a given class or not. For example email spam or not.

BEST SCORE: 0.6969326456928965

BEST PARAMETERS: {'C ': 10, 'multi_class ': 'multinomial', 'solver': 'newton-cg'}

ACCURACY: 70.10%

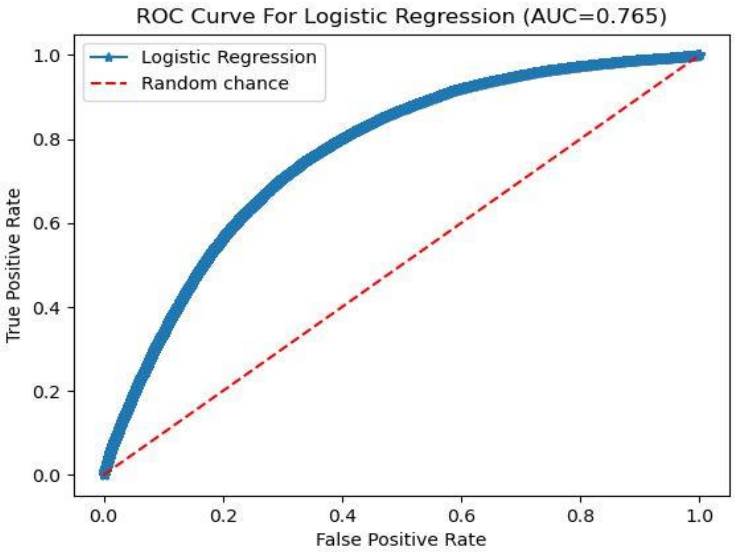
TRAINING TIME: 6.23 seconds

TESTING TIME: 0.00 seconds

```
Logistic Regression model:
Best parameters : {'C': 10, 'multi_class': 'multinomial', 'solver': 'newton-cg'}
Best Score : 0.6969326456928965
Training time :6.23 seconds
Testing time :0.00 seconds
Accuracy: 70.10%
                          precision    recall  f1-score   support

       High_Reviewer_Score       0.73      0.85      0.79     33042
Intermediate_Reviewer_Score       0.65      0.55      0.59     22549
        Low_Reviewer_Score       0.53      0.04      0.08      2472

                  accuracy                           0.70     58063
                 macro avg       0.63      0.48      0.49     58063
              weighted avg       0.69      0.70      0.68     58063
```



Decision Tree

DEFINITION: is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

BEST SCORE: 0.70378296182055

BEST PARAMETERS: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}

ACCURACY: 69.33%

TRAINING TIME: 0.14 seconds

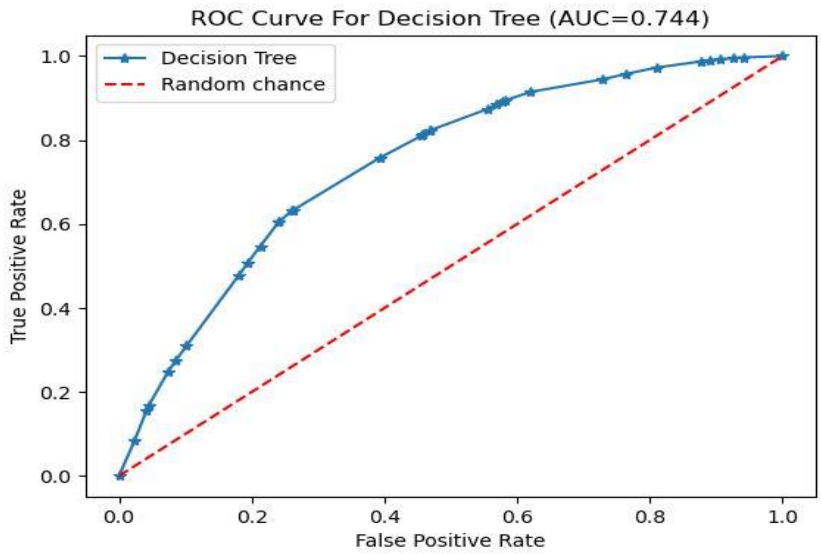
TESTING TIME: 0.01 seconds

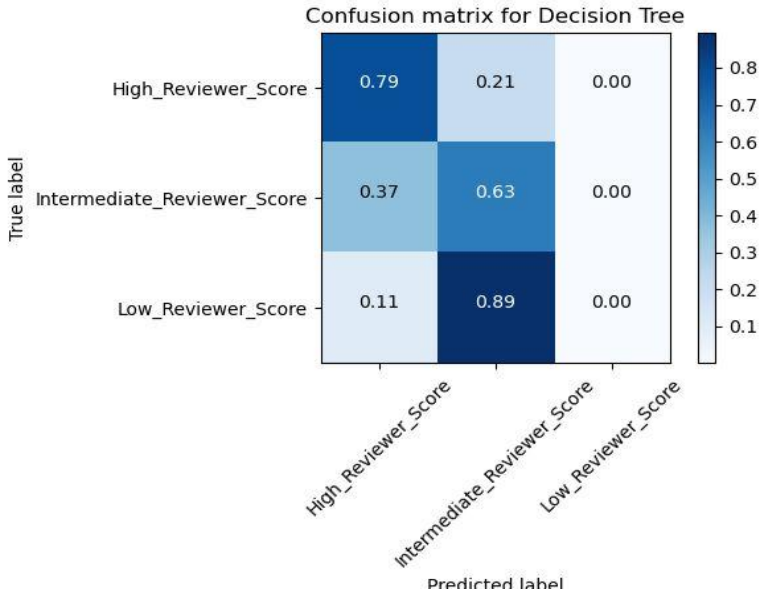
```
Decision Tree model:
Best parameters : {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best Score : 0.70378296182055
Training time :0.14 seconds
Testing time :0.01 seconds
Accuracy: 69.33%
                           precision    recall  f1-score   support

       High_Reviewer_Score       0.75      0.79      0.77     33042
Intermediate_Reviewer_Score       0.61      0.63      0.62     22549
        Low_Reviewer_Score       0.25      0.00      0.00      2472

                  accuracy                           0.69     58063
                 macro avg       0.54      0.47      0.46     58063
              weighted avg       0.67      0.69      0.68     58063


------------------------------------------------
```





Random Forest Regression

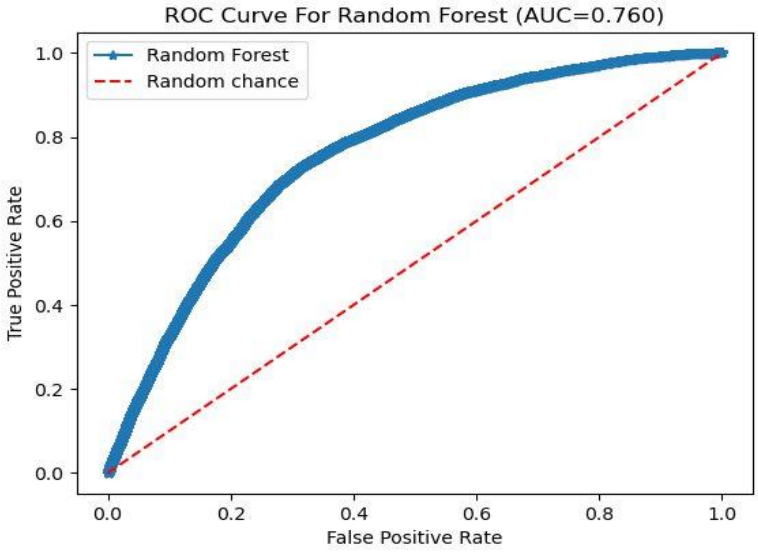
DEFINITION: is an ensemble learning method that combines multiple decision trees to make predictions.

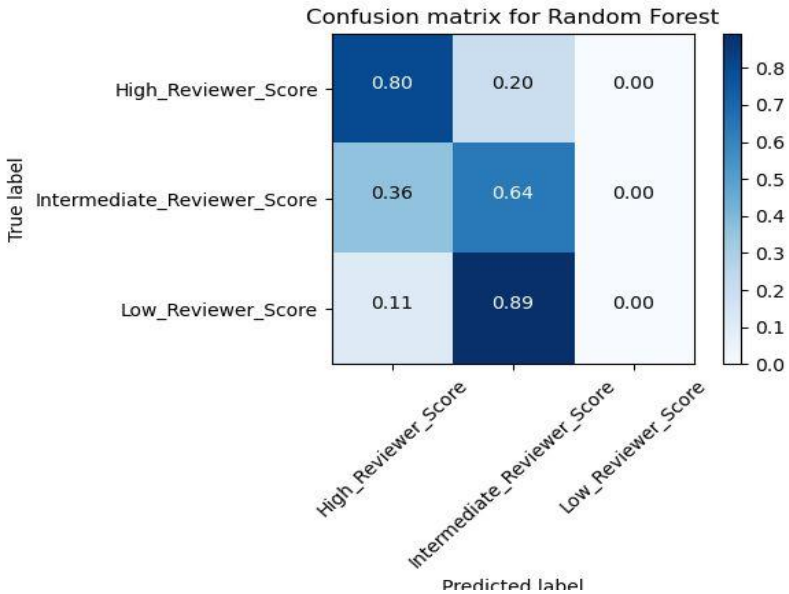
ACCURACY: 70.31%
BEST PARAMETERS: {'n_estimators': 1000, 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 3}
TRAINING TIME: 79.44 seconds
TESTING TIME: 5.60 seconds

```
Random Forest model:
Training time :79.44 seconds
Testing time :5.60 seconds
Accuracy: 70.31%
                          precision    recall  f1-score   support

       Low_Reviewer_Score       0.76      0.80      0.78     33042
Intermediate_Reviewer_Score       0.62      0.64      0.63     22549
      High_Reviewer_Score       0.00      0.00      0.00      2472

                 accuracy                           0.70     58063
                macro avg       0.46      0.48      0.47     58063
             weighted avg       0.67      0.70      0.69     58063
```





SVC

DEFINITION: Is a type of supervised machine learning algorithm that can be used for both classification and regression tasks. SVCs are a popular choice for many machine learning applications due to their ability to achieve high accuracy while still being computationally efficient.

ACCURACY: 70.33%

TRAINING TIME: 2988.96 seconds

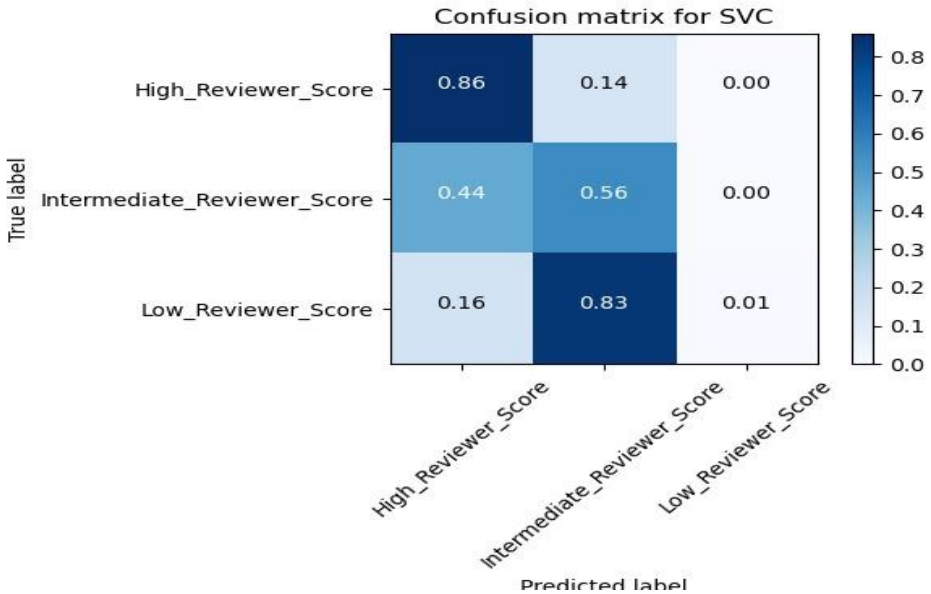
TESTING TIME: 853.94 seconds

```
SVC model:
Training time :2988.96 seconds
Testing time :853.94 seconds
Accuracy: 70.33%
                           precision    recall  f1-score   support

       High_Reviewer_Score       0.73      0.86      0.79     33042
Intermediate_Reviewer_Score       0.65      0.55      0.60     22549
        Low_Reviewer_Score       1.00      0.00      0.00      2472

                  accuracy                           0.70     58063
                 macro avg       0.79      0.47      0.46     58063
              weighted avg       0.71      0.70      0.68     58063

--------------------------------------
```



KNN

DEFINITION: is one of the simplest machine learning algorithms. Usually, k is a small, odd number - sometimes only 1. The larger k is, the more accurate the classification will be, but the longer it takes to perform the classification.

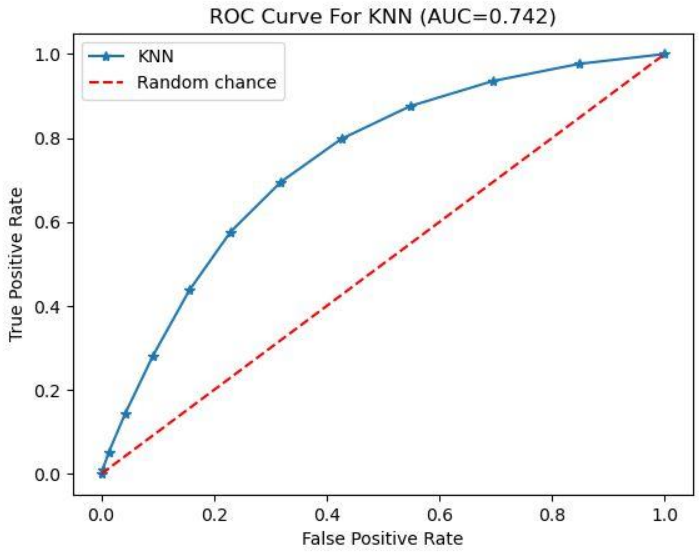
BEST SCORE: 0.6982243541517723
BEST PARAMETERS: {'n_neighbours': 11}
ACCURACY: 69.24%
TRAINING TIME: 0.25 seconds
TESTING TIME: 2.78 seconds

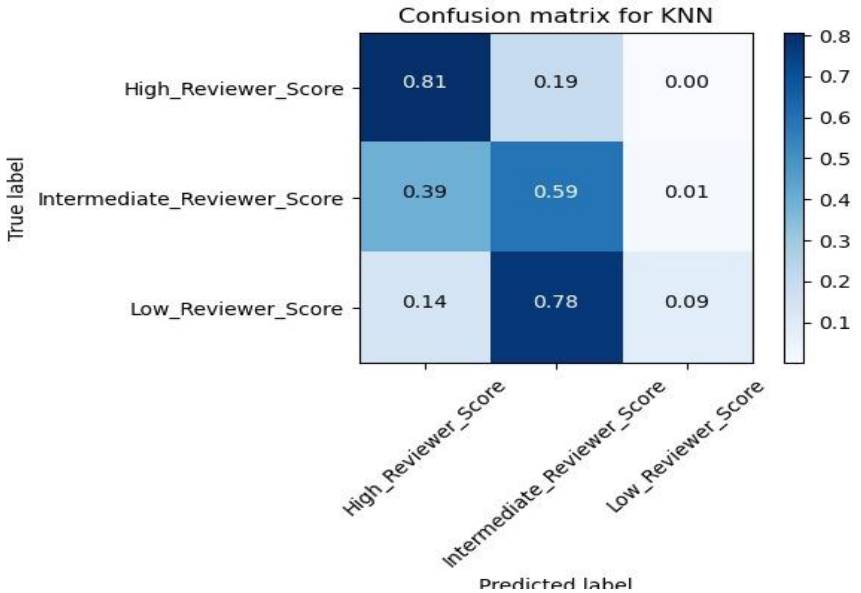
```
-----------------------------------------
KNN model:
Best parameters : {'n_neighbors': 11}
Best Score : 0.6982243541517723
Training time :0.25 seconds
Testing time :2.78 seconds
Accuracy: 69.24%
                          precision    recall  f1-score   support

       High_Reviewer_Score       0.74      0.80      0.77     33042
Intermediate_Reviewer_Score       0.61      0.60      0.61     22549
        Low_Reviewer_Score       0.44      0.09      0.14      2472

                  accuracy                           0.69     58063
                 macro avg       0.60      0.50      0.51     58063
              weighted avg       0.68      0.69      0.68     58063


-----------------------------------------
```





Hard Voting Classifier

DEFINITION: is a simple and effective ensemble method that can improve the accuracy and robustness of a machine learning model by reducing the risk of overfitting and increasing the diversity of the base classifiers. However, it works best when the individual classifiers have similar performance and are not highlycorrelated.

ACCURACY: 70.83%

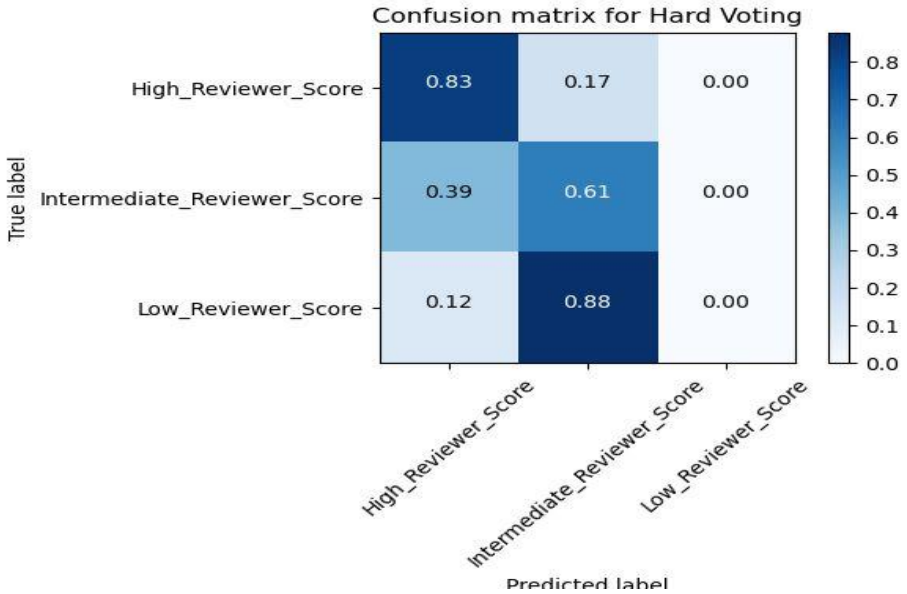
TRAINING TIME: 2440.52 seconds

TESTING TIME: 811.71 seconds

```
Hard Voting Classifier:
Training time :2440.52 seconds
Testing time :811.71 seconds
Accuracy: 70.83%
                           precision    recall  f1-score   support

       High_Reviewer_Score       0.75      0.83      0.79     33042
Intermediate_Reviewer_Score       0.64      0.61      0.62     22549
         Low_Reviewer_Score       1.00      0.00      0.00      2472

                   accuracy                           0.71     58063
                  macro avg       0.80      0.48      0.47     58063
               weighted avg       0.72      0.71      0.69     58063
```

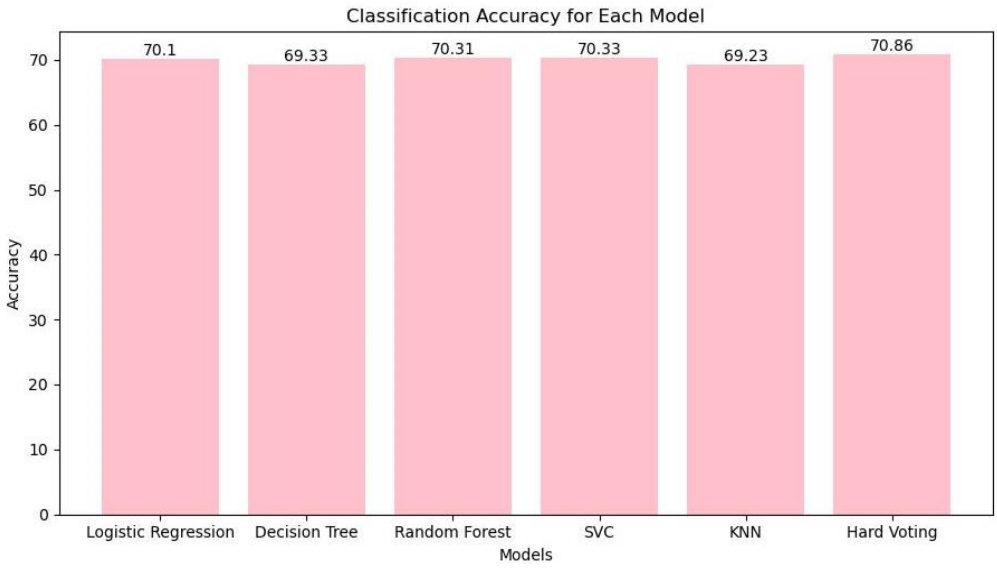


Gridsearchcv

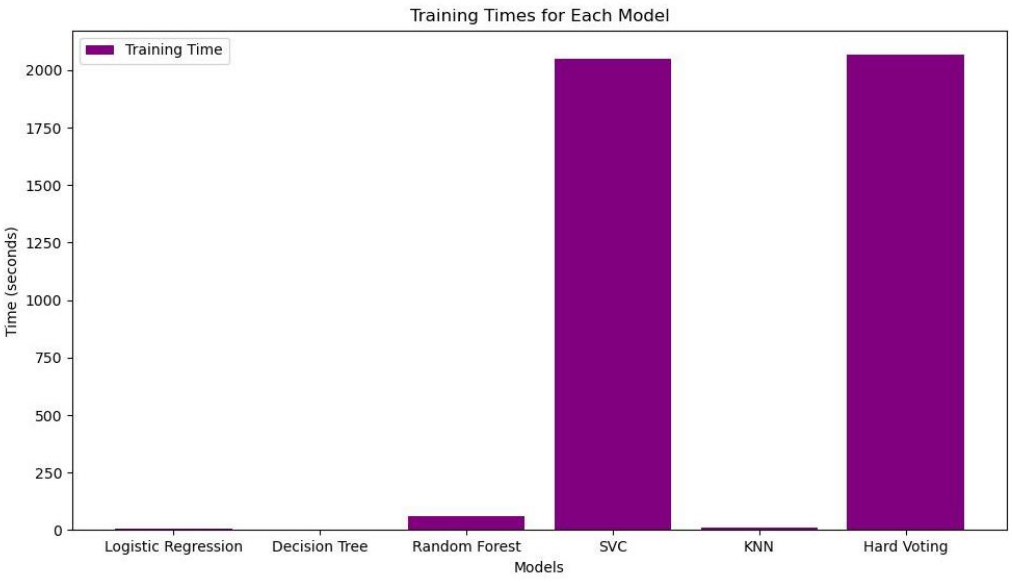
DEFINITION: a technique for finding the optimal parameter values from a given set of parameters in a grid

Applied to find best parameters for each model.

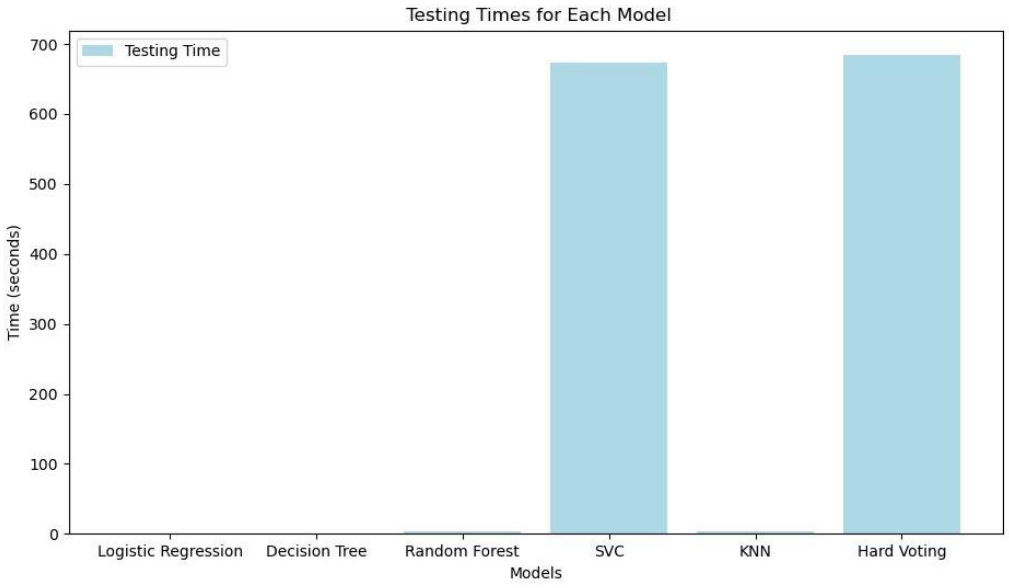
CLASSIFICATION ACCURACY GRAPH



TOTAL TRAINING TIME GRAPH



TOTAL TESTING TIME GRAPH

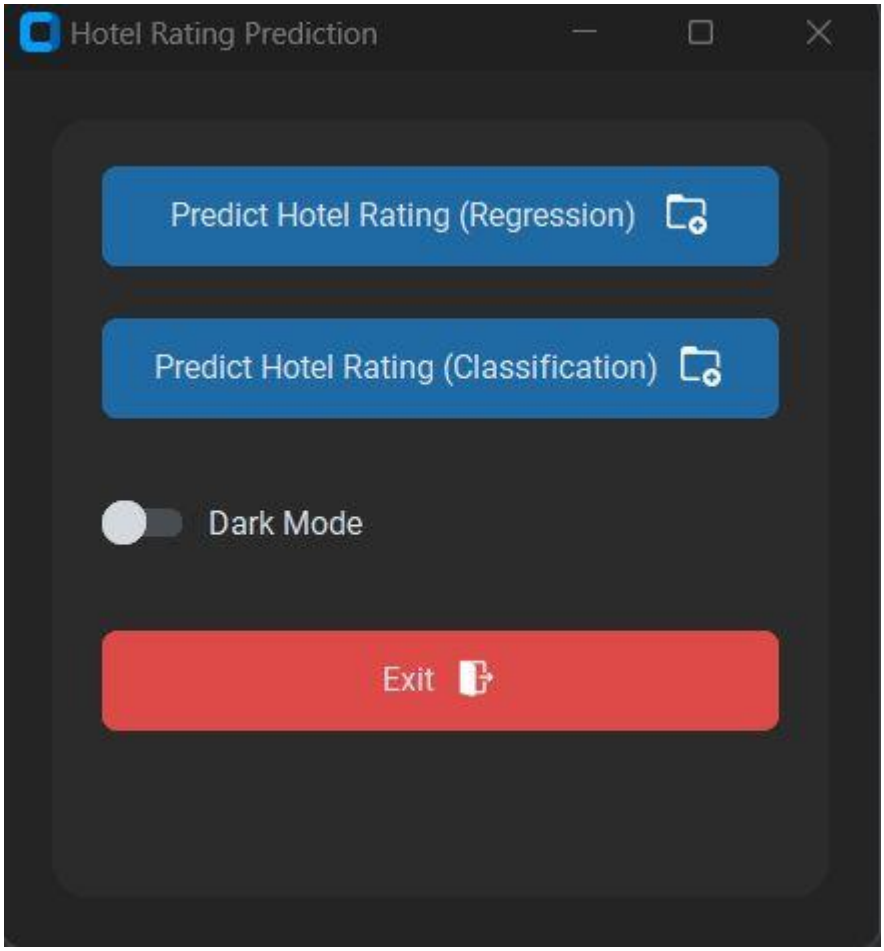


GUI

-We made a GUI that allows you predict Hotel Rating for Regression or Classification, After uploading the test file the program will save the prediction in a column for each model in a (.csv) file and will open this file.

-Also, we added Light and Dark mode theme in the GUI.

-We used tkinter library for this GUI:



CONCLUSION

In the Hotel Rating Prediction project , we aim to build a machine learning model that can predict the rating of a hotel based on various features such as address, Average Score, and customer reviews.

To accomplish this task, we first need to gather and preprocess the data. This involves collecting data from various sources such as hotel booking websites and customer review platforms, cleaning the data to remove any missing or incorrect values, and transforming the data into a format suitable for machine learning.

Next, we can use various machine learning algorithms such as logistic regression, Decision Tree, Random Forest regression, SVC, KNN, and Hard Voting Classifier to train and test our models. We can also use techniques such as feature selection, feature scaling and gridsearchcv to optimize the performance of our models.

Once we have trained and tested our models, we can use them to predict the ratings of new hotels based on their features. We can also evaluate the performance of our models using metrics such as accuracy and score.

Overall, the Hotel Rating Prediction project using Python requires a combination of data preprocessing, machine learning, and data analysis skills to successfully build an accurate and reliable model that can predict hotel ratings.