Date- 24 September 2023

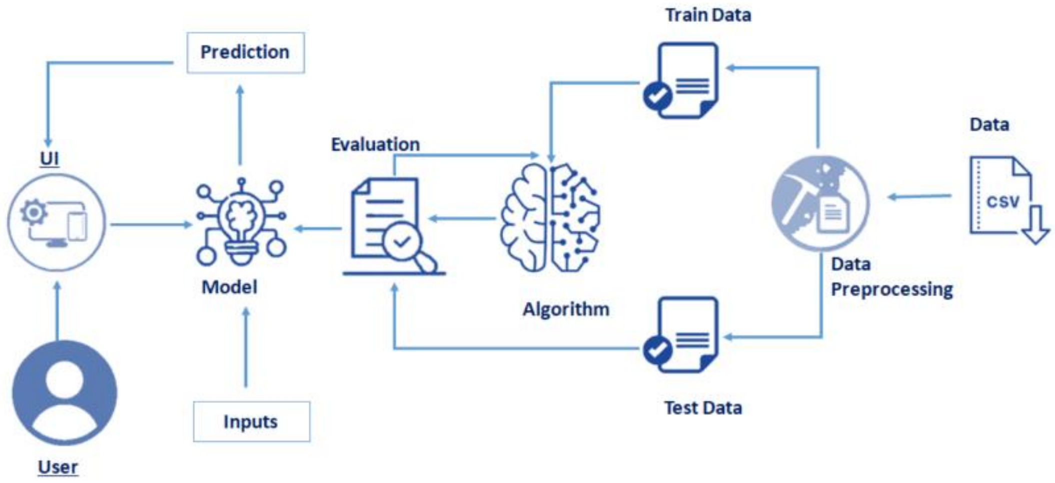
Team ID- 592337 (Satvik Marwah and Himanshu Rana)

Project Name Project – Airline review classification system

Introduction

In today's interconnected world, the airline industry serves as a critical catalyst for global travel and business. As air travel becomes increasingly accessible, the quality of service provided by airlines plays a pivotal role in shaping passenger experiences. This project focuses on the development of an airline review classification system using Classification models such as Decision Tree Classifier, Random Forest Classifier, XGBoost Classifier etc., The proliferation of social media platforms, travel websites, and online forums has given rise to a wealth of user-generated content, including airline reviews. Extracting actionable insights from this vast pool of unstructured text data has the potential to provide airlines with valuable information for refining their services and elevating passenger satisfaction. Throughout this report, we will delve into the methodology employed to preprocess the raw text data, the process of selecting pertinent features, the training and evaluation of the classification model, and the subsequent interpretation of the obtained results.

Technical Architecture:



Project Flow:

User interacts with the UI to enter the input.

● Entered input is analyzed by the model which is integrated.

● Once model analyses the input the prediction is showcased on the UI to accomplish this, we have to complete all

the activities listed below,

➢ Data Collection & Preparation

o Collect the dataset

o Data Preparation

o Exploratory Data Analysis

➢ Descriptive statistical

o Visual Analysis

➢ Model Building

o Training the model in multiple algorithms

o Testing the model

➢ Performance Testing

o Testing model with multiple evaluation metrics

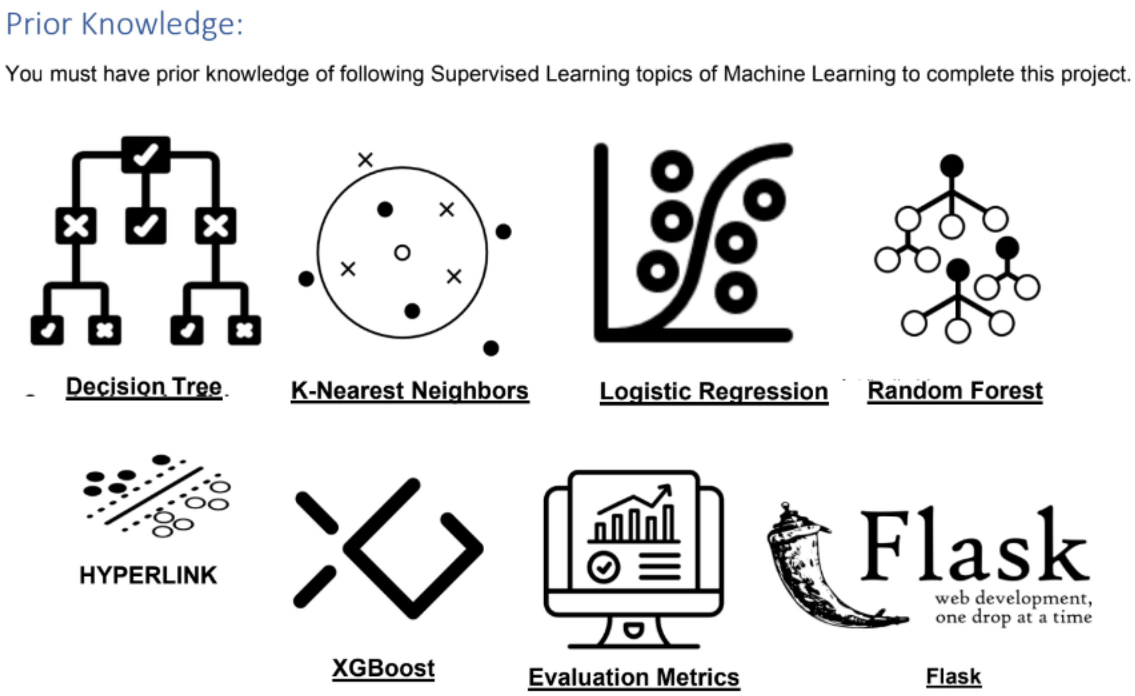
➢ Model Deployment

o Save the best model

o Integrate with Web Framework

➢ Project Demonstration & Documentation

o Record explanation Video for project end to end solution



Prior Knowledge:

You must have prior knowledge of following Supervised Learning topics of Machine Learning to complete this project.

Decision Tree

K-Nearest Neighbors

Logistic Regression

Random Forest

HYPERLINK

XGBoost

Evaluation Metrics

Flask

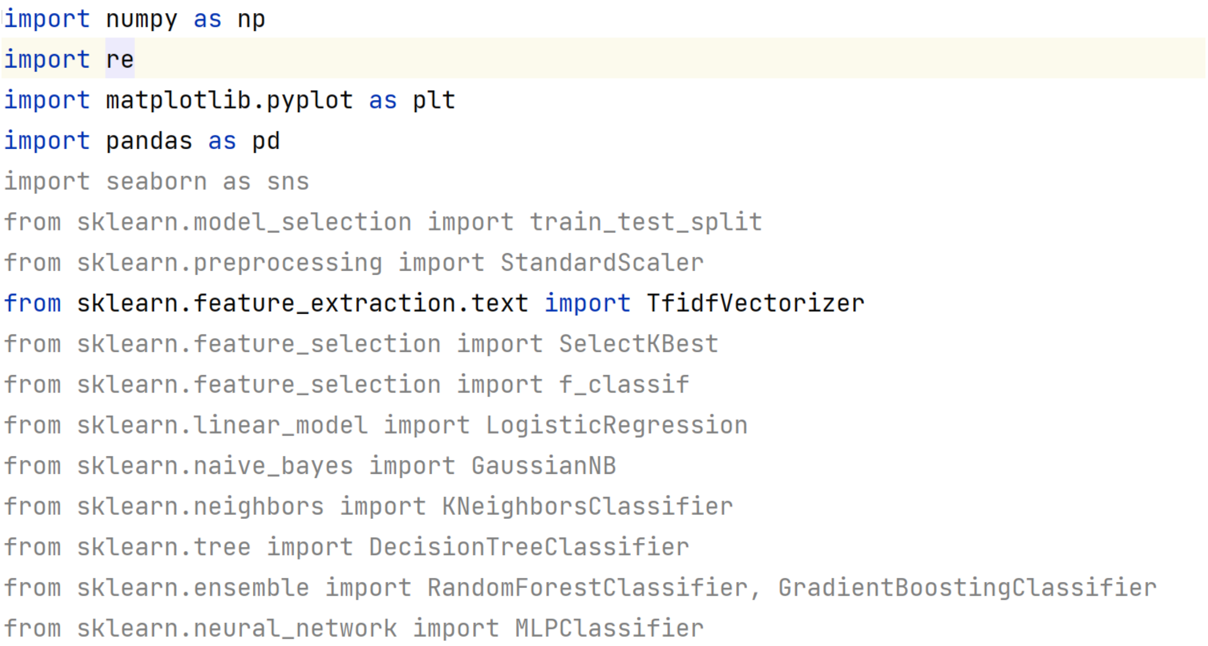
Milestone 1: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

Activity 1: Collect the dataset

There are many popular open sources for collecting the data. E.g.: kaggle.com, UCI repository, etc. In this project we have used .csv data. This data is downloaded from kaggle.com. After downloading we have to read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

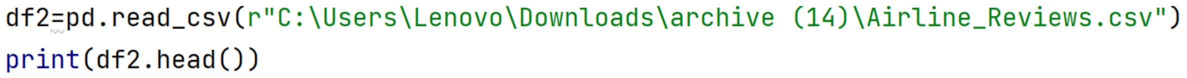
Activity 1.1: Importing the libraries



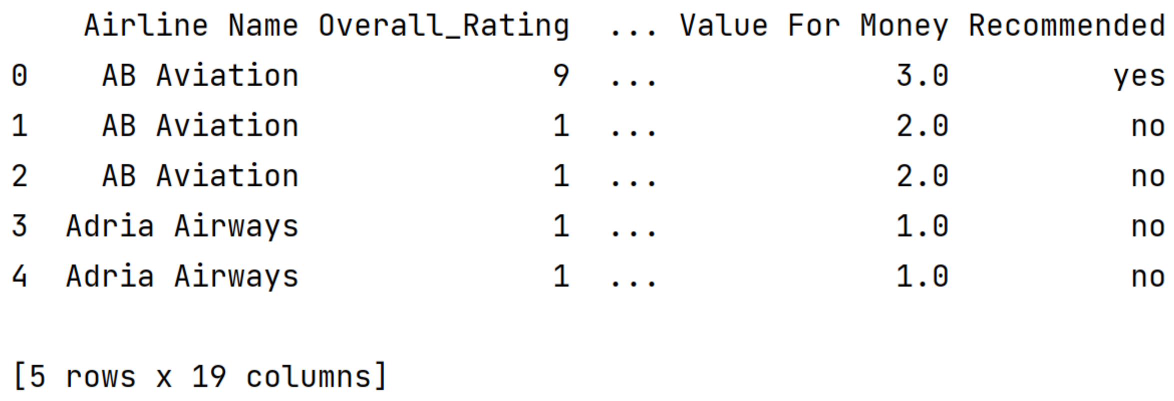
```
import numpy as np
import re
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier
```

Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas. In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



```
df2=pd.read_csv(r"C:\Users\Lenovo\Downloads\archive (14)\Airline_Reviews.csv")
print(df2.head())
```



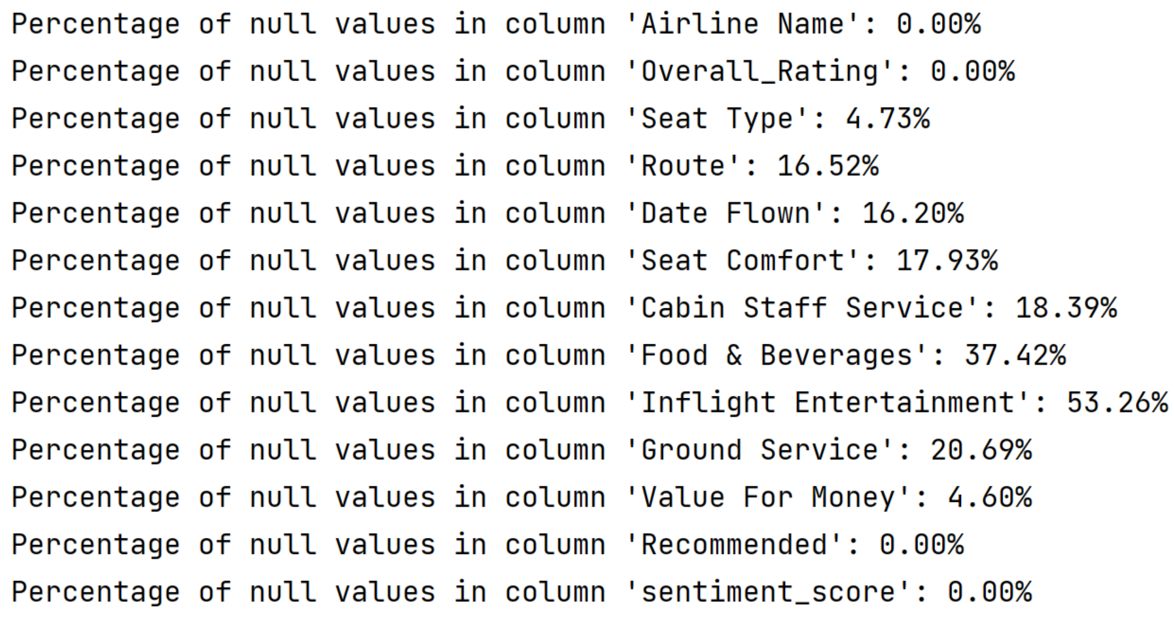
```
    Airline Name Overall_Rating  ... Value For Money Recommended
0     AB Aviation              9  ...             3.0         yes
1     AB Aviation              1  ...             2.0          no
2     AB Aviation              1  ...             2.0          no
3   Adria Airways              1  ...             1.0          no
4   Adria Airways              1  ...             1.0          no

[5 rows x 19 columns]
```

Activity 2: Data Preparation

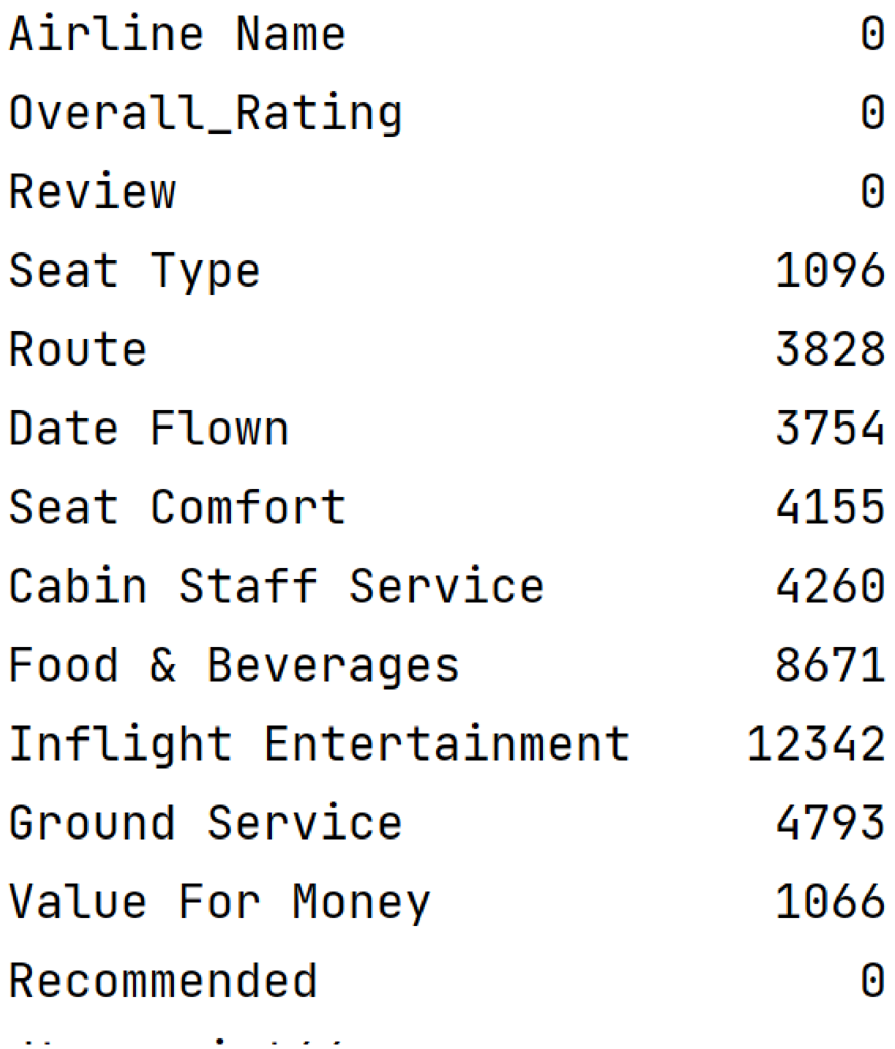
As we have understood how the data is, let's pre-process the collected data. The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps

● Handling missing values ● Handling categorical data

Activity 2.1: Handling missing values



```
Percentage of null values in column 'Airline Name': 0.00%
Percentage of null values in column 'Overall_Rating': 0.00%
Percentage of null values in column 'Seat Type': 4.73%
Percentage of null values in column 'Route': 16.52%
Percentage of null values in column 'Date Flown': 16.20%
Percentage of null values in column 'Seat Comfort': 17.93%
Percentage of null values in column 'Cabin Staff Service': 18.39%
Percentage of null values in column 'Food & Beverages': 37.42%
Percentage of null values in column 'Inflight Entertainment': 53.26%
Percentage of null values in column 'Ground Service': 20.69%
Percentage of null values in column 'Value For Money': 4.60%
Percentage of null values in column 'Recommended': 0.00%
Percentage of null values in column 'sentiment_score': 0.00%
```

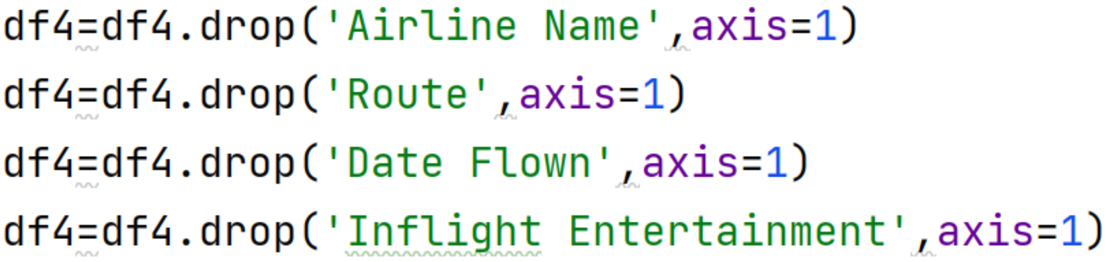


```
Airline Name                  0
Overall_Rating                0
Review                        0
Seat Type                  1096
Route                      3828
Date Flown                 3754
Seat Comfort               4155
Cabin Staff Service        4260
Food & Beverages           8671
Inflight Entertainment    12342
Ground Service             4793
Value For Money            1066
Recommended                   0
```

We fill all the null values with the median value of the column. For categorical columns the null values are filled with mode.

Handling categorical data

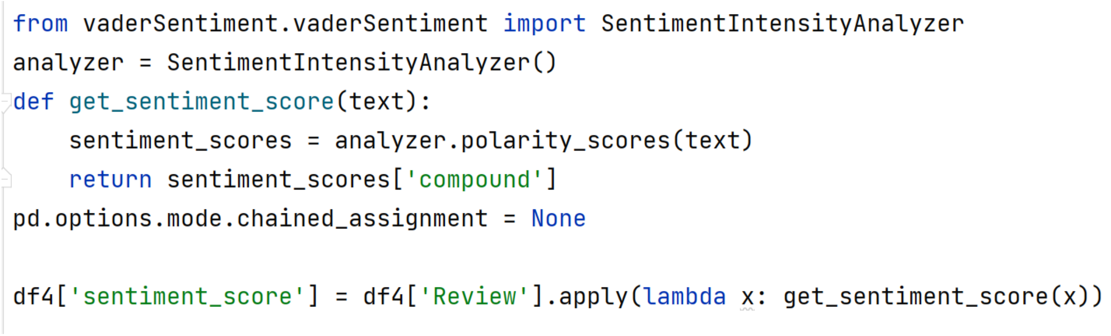
We handle categorial data by one hot encoding it. We drop the columns that that too many categories to handle.



```
df4=df4.drop('Airline Name',axis=1)
df4=df4.drop('Route',axis=1)
df4=df4.drop('Date Flown',axis=1)
df4=df4.drop('Inflight Entertainment',axis=1)
```

Handling 'Review' column

Review column is neither numerical nor categorical. It is rather just a piece of text. Hence we combine the review and review summary column into a single text column and then calculate its sentiment score thus converting it into a numerical value.

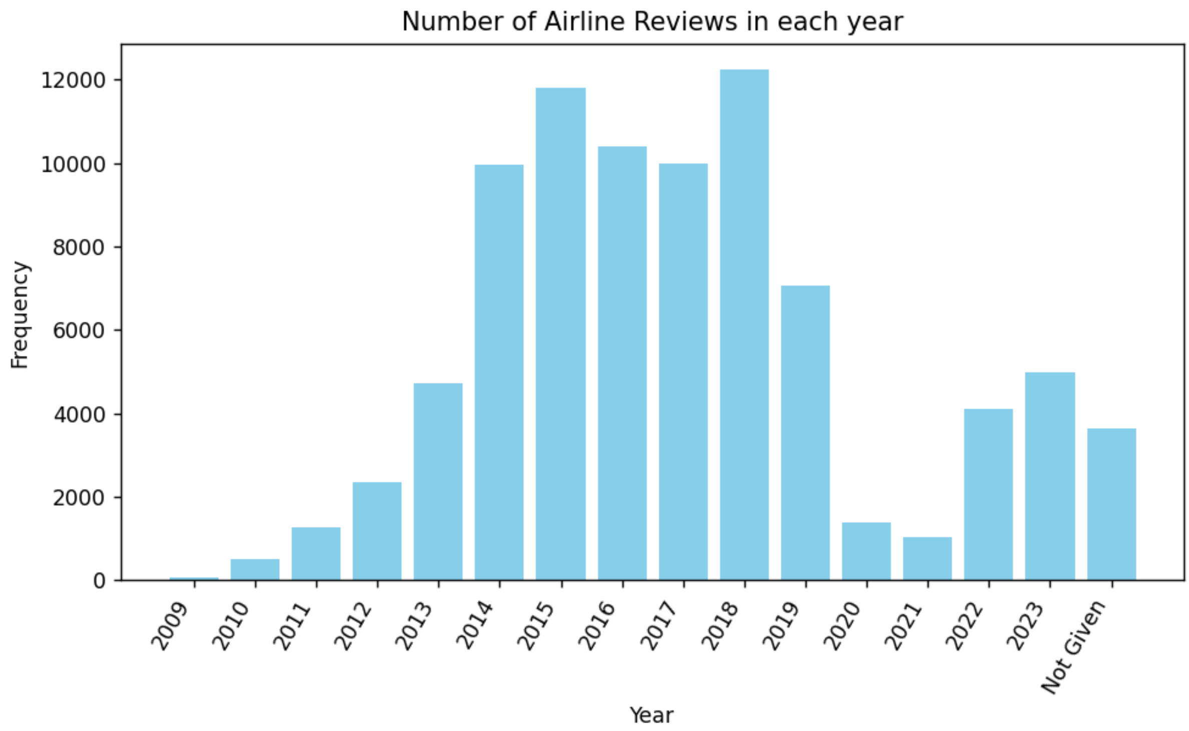


```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
def get_sentiment_score(text):
    sentiment_scores = analyzer.polarity_scores(text)
    return sentiment_scores['compound']
pd.options.mode.chained_assignment = None

df4['sentiment_score'] = df4['Review'].apply(lambda x: get_sentiment_score(x))
```

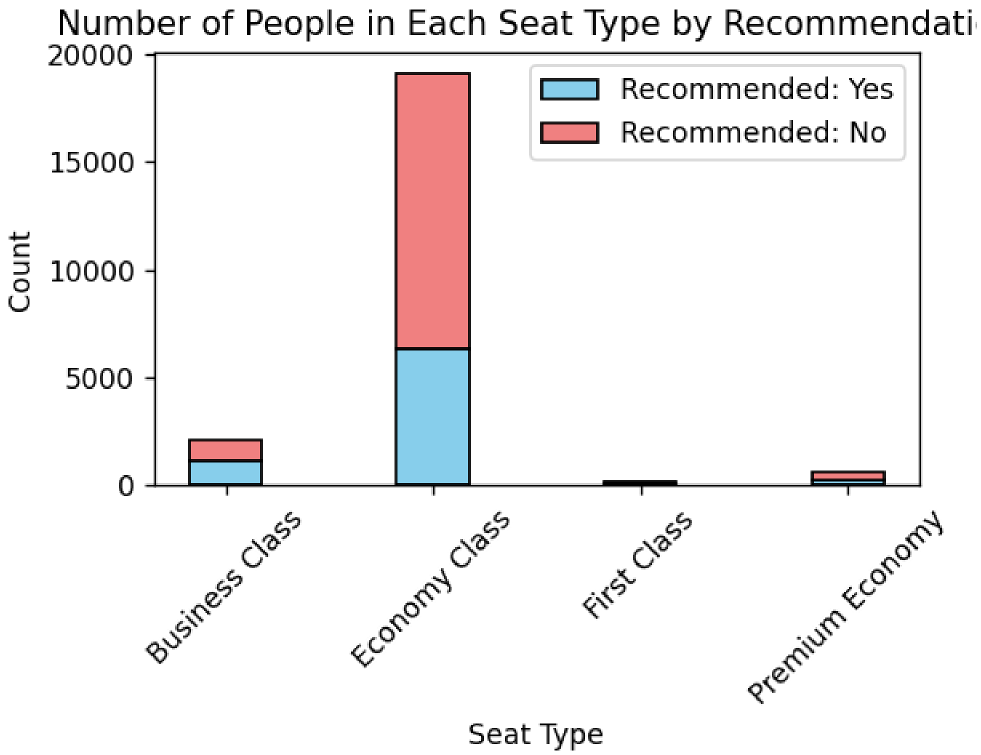
We have now fully preprocessed our data.

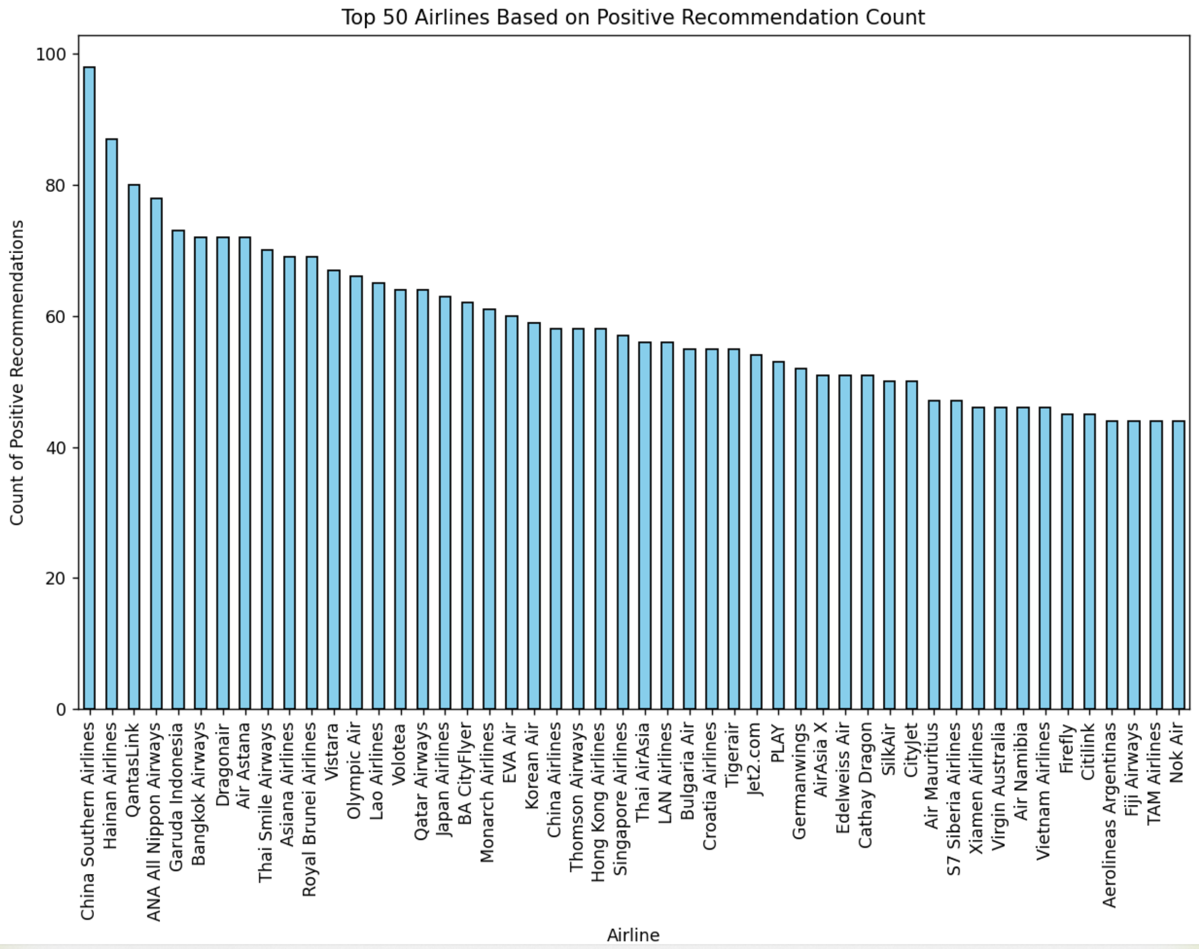
Milestone 3: Exploratory Data Analysis

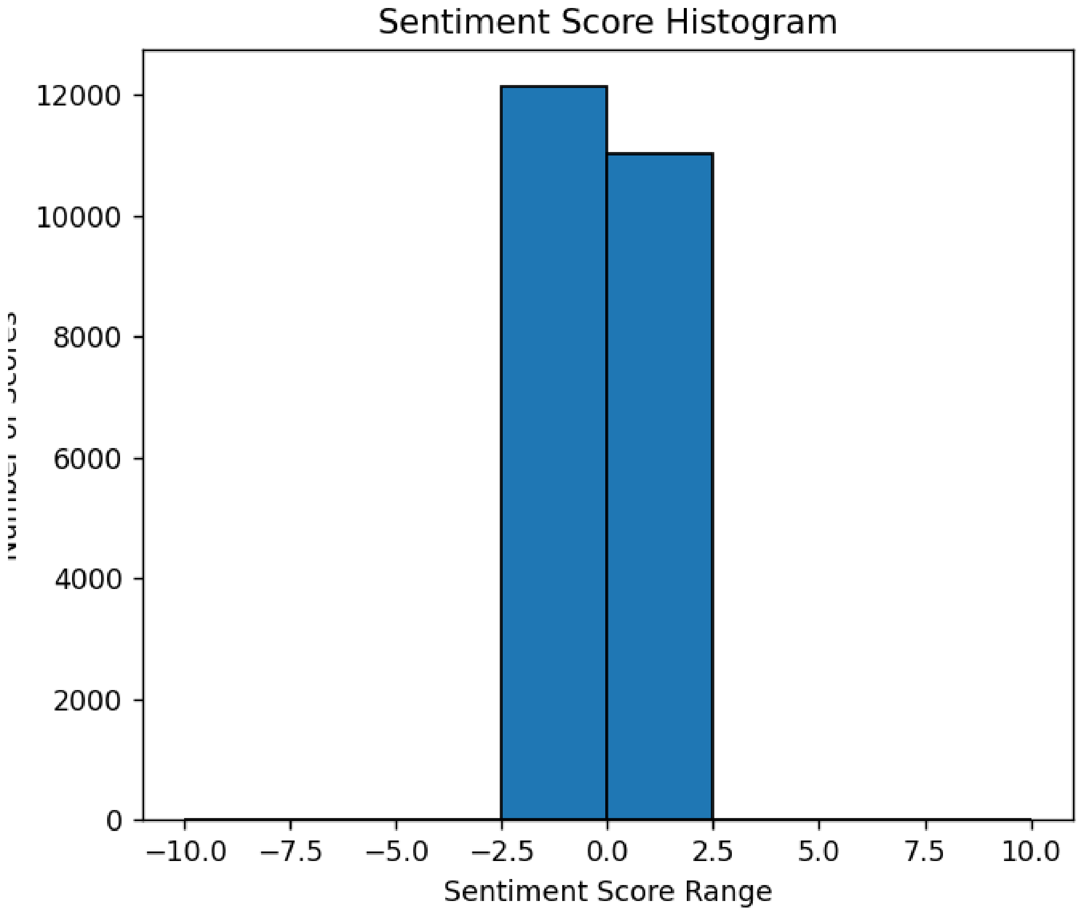
Here's a graph showing us the number of airline review each year.

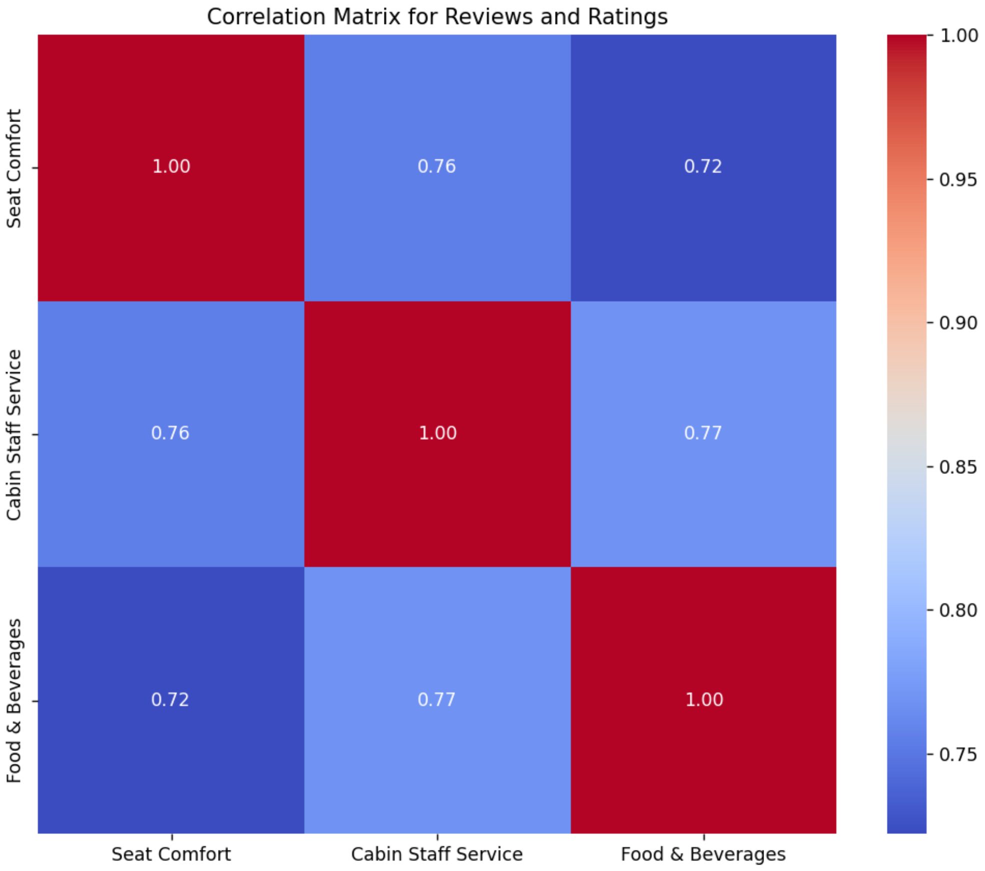


A graph showing the number of people recommding yes or no in each of the seat types: business class, economy class, first class and premium economy.



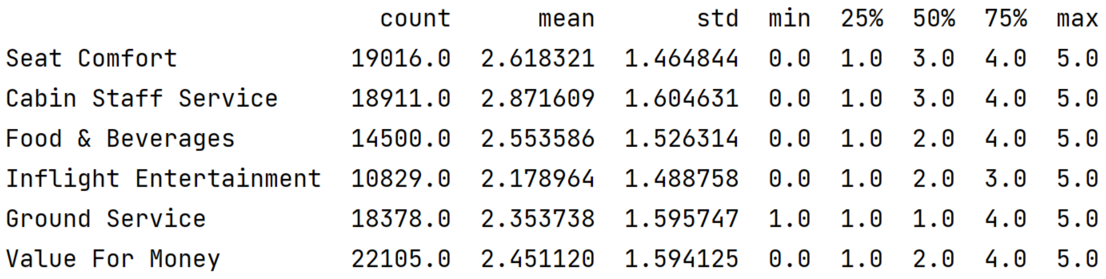

Top 50 Airlines Based on Positive Recommendation Count
Count of Positive Recommendations
0
20
40
60
80
100
Airline
China Southern Airlines
Hainan Airlines
QantasLink
ANA All Nippon Airways
Garuda Indonesia
Bangkok Airways
Dragonair
Air Astana
Thai Smile Airways
Asiana Airlines
Royal Brunei Airlines
Vistara
Olympic Air
Lao Airlines
Volotea
Qatar Airways
Japan Airlines
BA CityFlyer
Monarch Airlines
EVA Air
Korean Air
China Airlines
Thomson Airways
Hong Kong Airlines
Singapore Airlines
Thai AirAsia
LAN Airlines
Bulgaria Air
Croatia Airlines
Tigerair
Jet2.com
PLAY
Germanwings
AirAsia X
Edelweiss Air
Cathay Dragon
SilkAir
CityJet
Air Mauritius
S7 Siberia Airlines
Xiamen Airlines
Virgin Australia
Air Namibia
Vietnam Airlines
Firefly
Citilink
Aerolineas Argentinas
Fiji Airways
TAM Airlines
Nok Air


Sentiment Score Histogram
Number of Scores
0
2000
4000
6000
8000
10000
12000
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Sentiment Score Range



Descriptive statistical

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas have a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

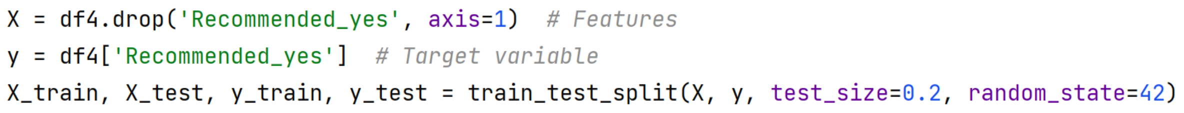


```
                        count      mean       std  min  25%  50%  75%  max
Seat Comfort          19016.0  2.618321  1.464844  0.0  1.0  3.0  4.0  5.0
Cabin Staff Service   18911.0  2.871609  1.604631  0.0  1.0  3.0  4.0  5.0
Food & Beverages      14500.0  2.553586  1.526314  0.0  1.0  2.0  4.0  5.0
Inflight Entertainment 10829.0  2.178964  1.488758  0.0  1.0  2.0  3.0  5.0
Ground Service        18378.0  2.353738  1.595747  1.0  1.0  1.0  4.0  5.0
Value For Money       22105.0  2.451120  1.594125  0.0  1.0  2.0  4.0  5.0
```

Milestone 4: Model Building

Activity 4.1: Splitting data into train and test

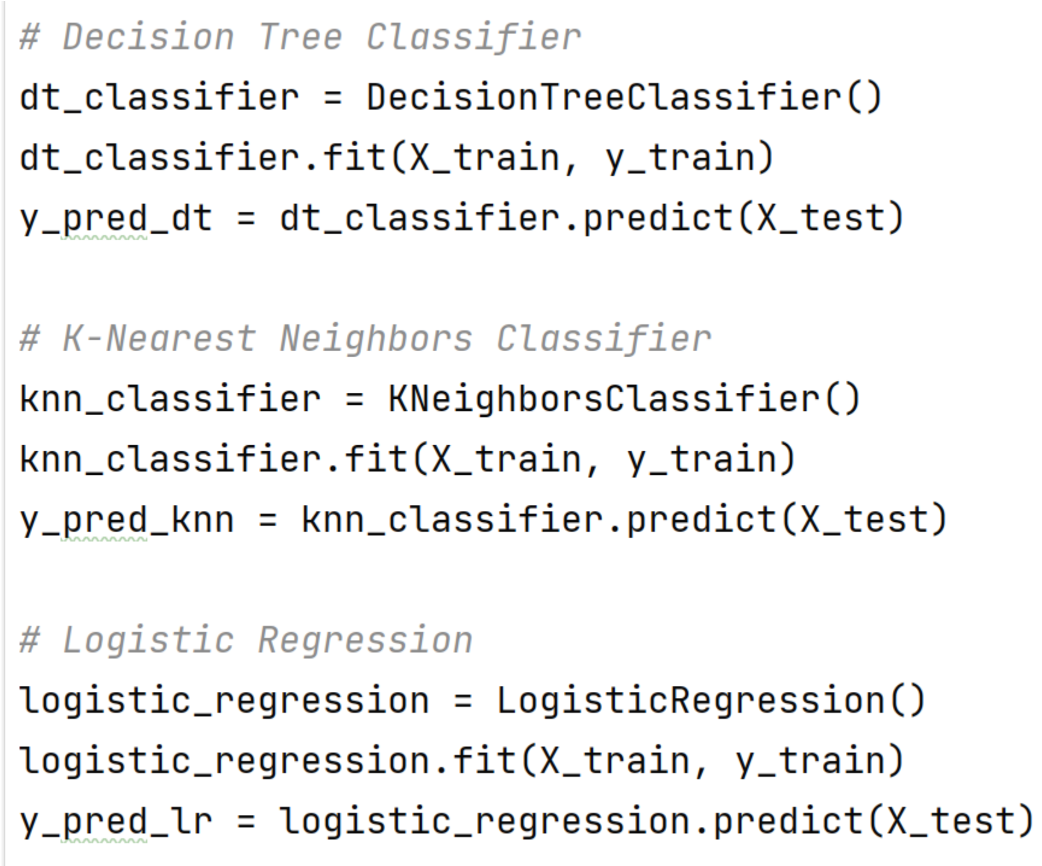
Now let’s split the Dataset into train and test sets. First split the dataset into X and y and then split the dataset. Here X and y variables are created. On X variable, nar is passed with dropping the target variable. And on y target variable is passed.



```
X = df4.drop('Recommended_yes', axis=1)  # Features
y = df4['Recommended_yes']  # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Activity 2: Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. We’re using 4 different classification algorithms: Decision tree, K-Nearesr neighbours, logistic regression and Random forest.



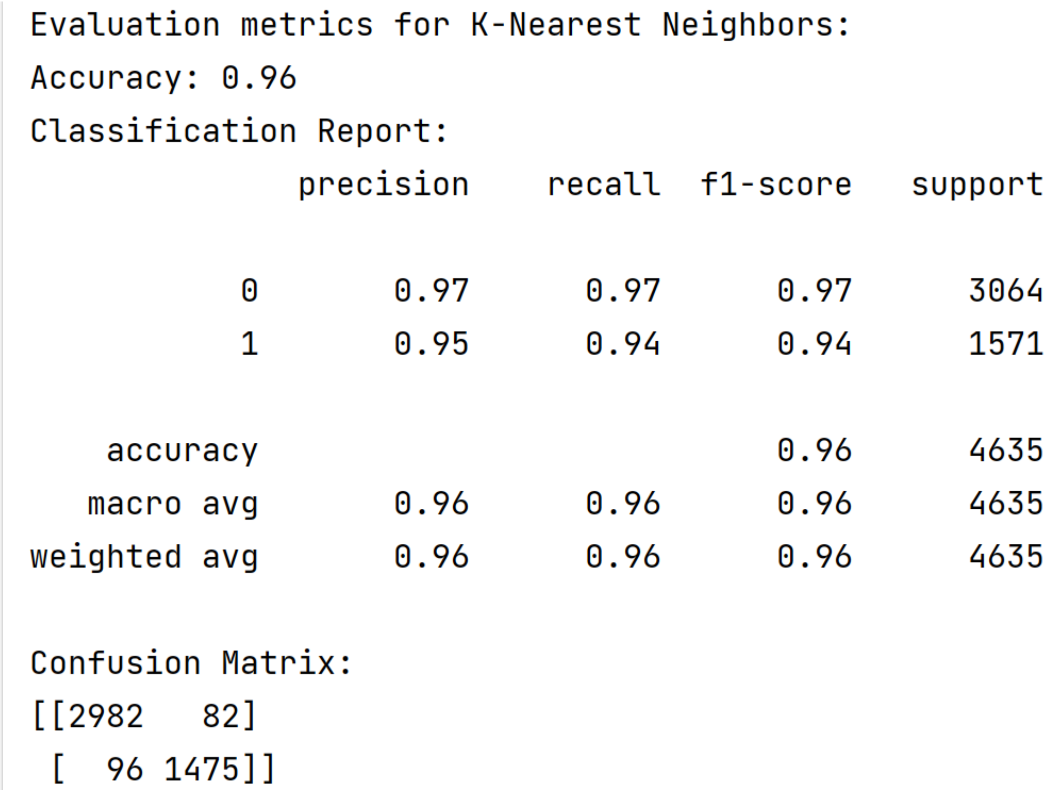
```
# Decision Tree Classifier
dt_classifier = DecisionTreeClassifier()
dt_classifier.fit(X_train, y_train)
y_pred_dt = dt_classifier.predict(X_test)

# K-Nearest Neighbors Classifier
knn_classifier = KNeighborsClassifier()
knn_classifier.fit(X_train, y_train)
y_pred_knn = knn_classifier.predict(X_test)

# Logistic Regression
logistic_regression = LogisticRegression()
logistic_regression.fit(X_train, y_train)
y_pred_lr = logistic_regression.predict(X_test)
```

```
# Random Forest Classifier
rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
```

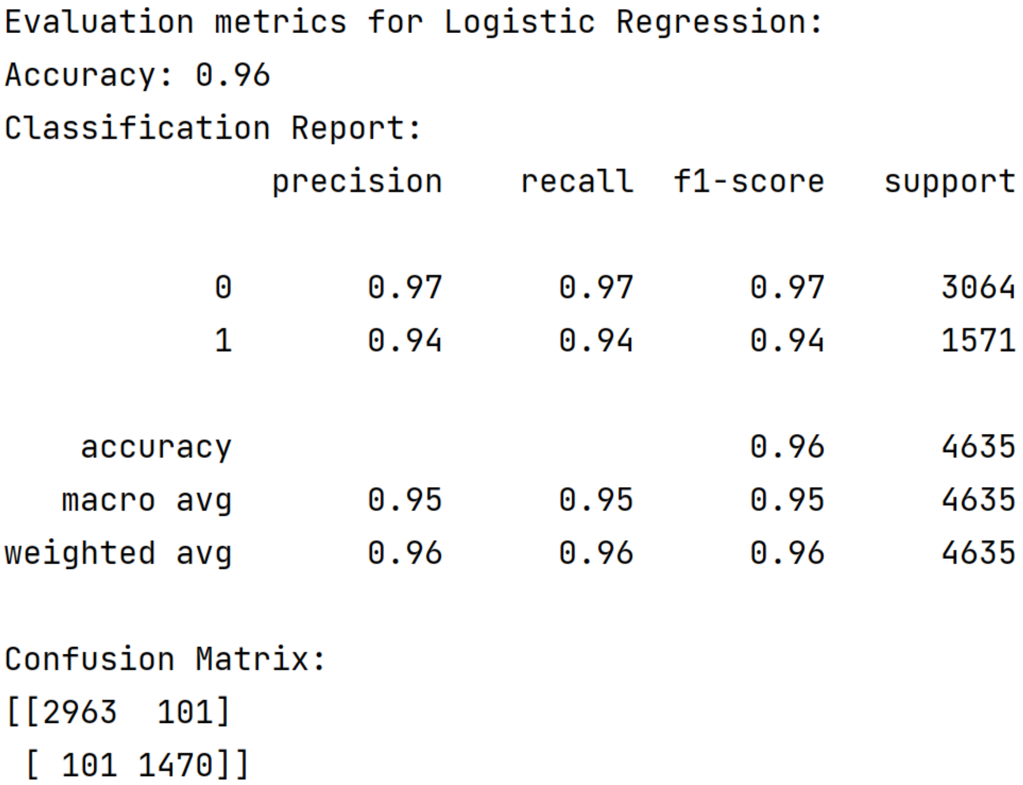
Evaluation metrics

Following are the results of all the 4 algorithms:

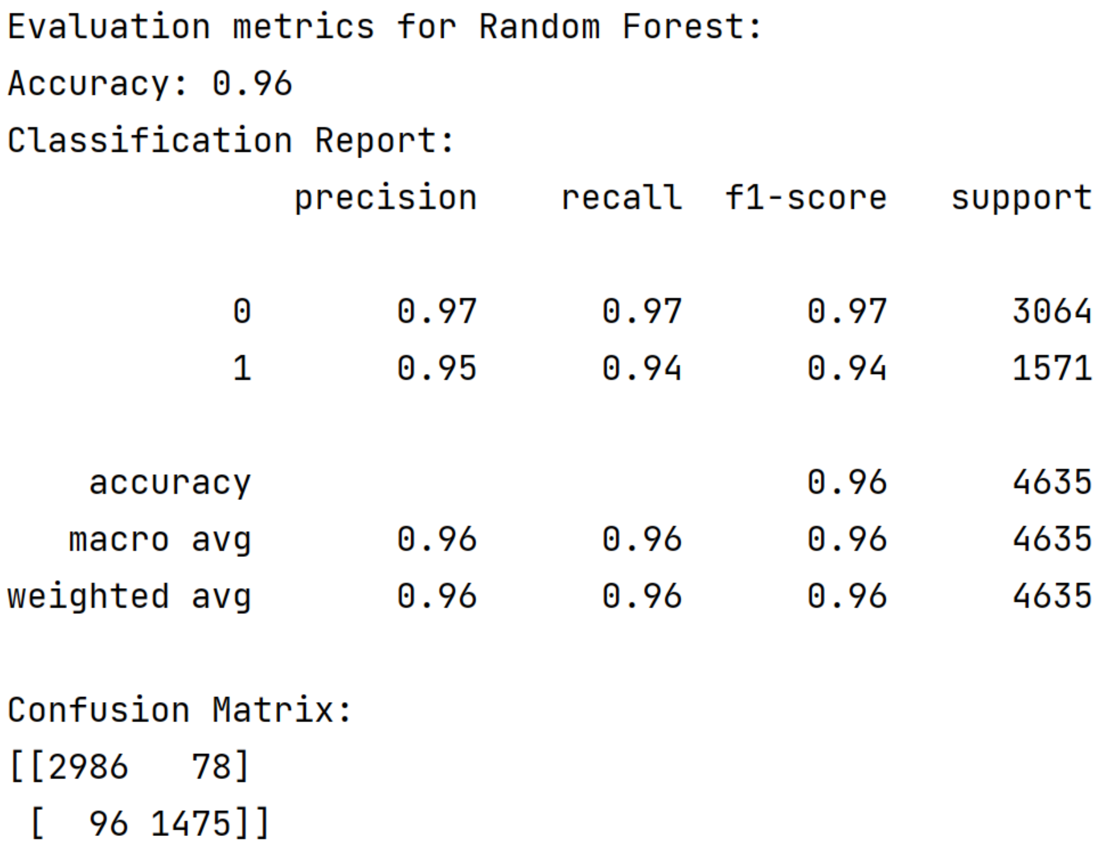
```
Evaluation metrics for Decision Tree:
Accuracy: 0.95
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      3064
           1       0.93      0.92      0.92      1571

    accuracy                           0.95      4635
   macro avg       0.94      0.94      0.94      4635
weighted avg       0.95      0.95      0.95      4635

Confusion Matrix:
[[2948  116]
 [ 126 1445]]
```



```
Evaluation metrics for K-Nearest Neighbors:
Accuracy: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      3064
           1       0.95      0.94      0.94      1571

    accuracy                           0.96      4635
   macro avg       0.96      0.96      0.96      4635
weighted avg       0.96      0.96      0.96      4635

Confusion Matrix:
[[2982   82]
 [  96 1475]]
```



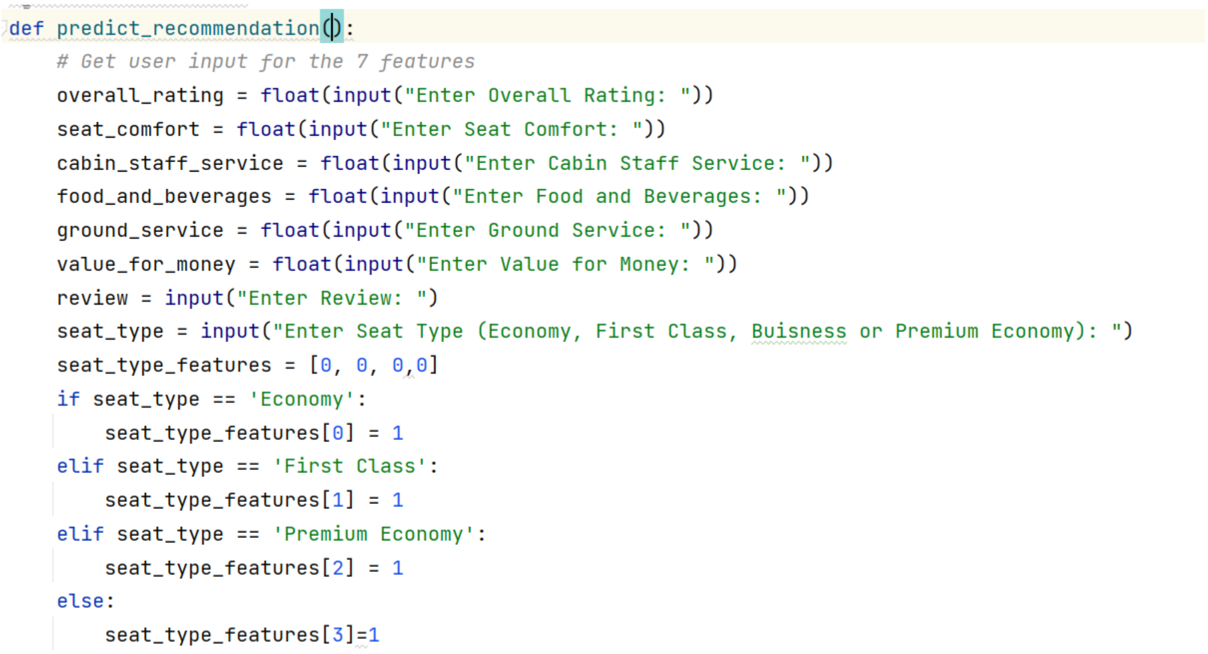
```
Evaluation metrics for Logistic Regression:
Accuracy: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      3064
           1       0.94      0.94      0.94      1571

    accuracy                           0.96      4635
   macro avg       0.95      0.95      0.95      4635
weighted avg       0.96      0.96      0.96      4635

Confusion Matrix:
[[2963  101]
 [ 101 1470]]
```



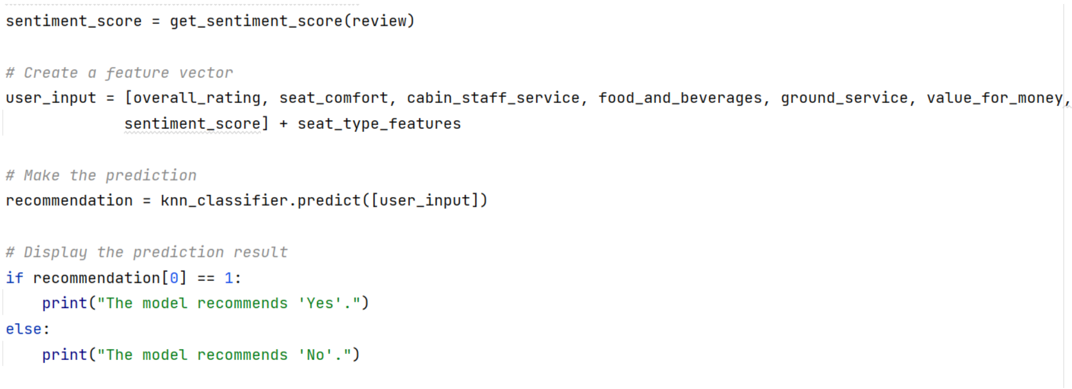
```
Evaluation metrics for Random Forest:
Accuracy: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      3064
           1       0.95      0.94      0.94      1571

    accuracy                           0.96      4635
   macro avg       0.96      0.96      0.96      4635
weighted avg       0.96      0.96      0.96      4635

Confusion Matrix:
[[2986   78]
 [  96 1475]]
```

Testing the model:

To test the model we create a function called predict_function. Using this function we can test the model using our own custom input.



```python
def predict_recommendation():
    # Get user input for the 7 features
    overall_rating = float(input("Enter Overall Rating: "))
    seat_comfort = float(input("Enter Seat Comfort: "))
    cabin_staff_service = float(input("Enter Cabin Staff Service: "))
    food_and_beverages = float(input("Enter Food and Beverages: "))
    ground_service = float(input("Enter Ground Service: "))
    value_for_money = float(input("Enter Value for Money: "))
    review = input("Enter Review: ")
    seat_type = input("Enter Seat Type (Economy, First Class, Buisness or Premium Economy): ")
    seat_type_features = [0, 0, 0,0]
    if seat_type == 'Economy':
        seat_type_features[0] = 1
    elif seat_type == 'First Class':
        seat_type_features[1] = 1
    elif seat_type == 'Premium Economy':
        seat_type_features[2] = 1
    else:
        seat_type_features[3]=1
```



```
sentiment_score = get_sentiment_score(review)

# Create a feature vector
user_input = [overall_rating, seat_comfort, cabin_staff_service, food_and_beverages, ground_service, value_for_money,
              sentiment_score] + seat_type_features

# Make the prediction
recommendation = knn_classifier.predict([user_input])

# Display the prediction result
if recommendation[0] == 1:
    print("The model recommends 'Yes'.")
else:
    print("The model recommends 'No'.")
```

.

Milestone 6: Model Deployment

Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

Activity 2: Integrate with Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server-side script
- Run the web application

Activity 2.1: Building Html Pages:

For this project we create an html file ‘airplane.html’ which is the template.

Activity 2.2: Build Python code

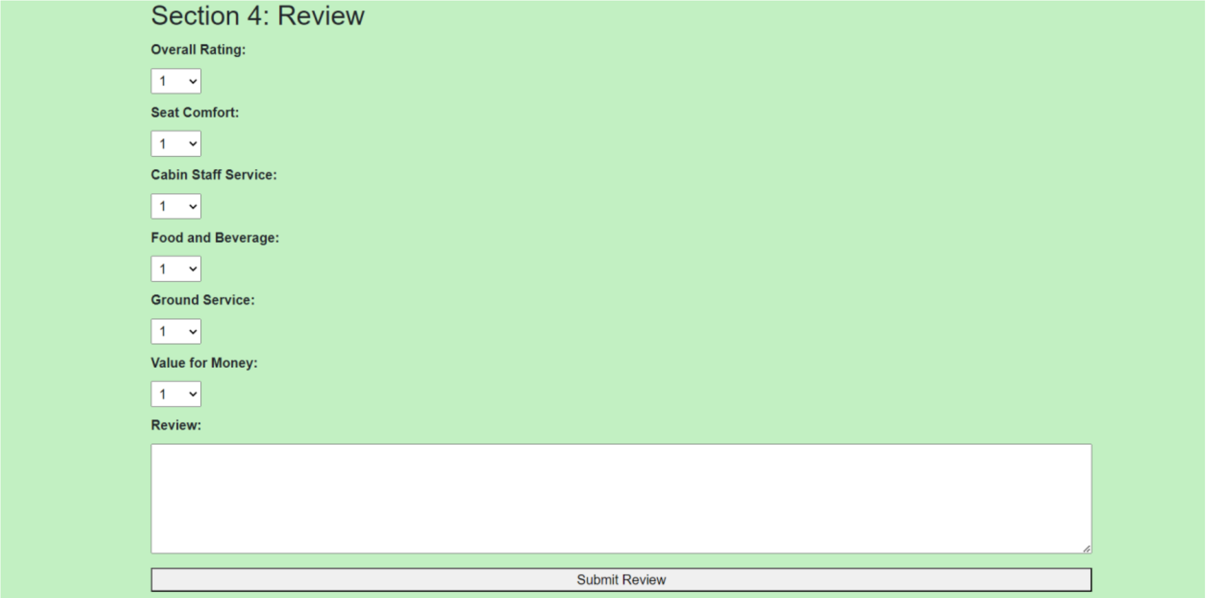
Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (name) as argument.

Render HTML page:

Here we will be using a declared constructor to route to the HTML page which we have created earlier. In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method. Retrieves the value from UI.

Activity 2.3: Run the web application

We run our code to see the results in the web application



Section 4: Review

Overall Rating: 1

Seat Comfort: 1

Cabin Staff Service: 1

Food and Beverage: 1

Ground Service: 1

Value for Money: 1

Review: