Airline Customer Satisfaction Analysis

CALIFORNIA STATE UNIVERSITY
LONG BEACH
College of Business

Manav Mangukiya (032178248)
Aryan Khunt (032178235)
Ramya Sree (032222760)

Machine Learning For Business Analytics
PROF. Mostafa Amini
Date:- 05/04/2024

Contents

1. Introduction

In the rapidly evolving airline industry, understanding passenger satisfaction is vital for an airline company in achieving a competitive edge and in return creating customer loyalty. The aviation sector is essential to the functioning of the world economy. It's about more than just travel; it's about bringing people together, facilitating trade, and promoting global economic expansion. The worldwide airline market was estimated by Yahoo Finance to be worth USD 553.9 billion in 2022. It is expected to grow to an incredible USD 735 billion by 2030. N-2This substantial growth presents both opportunities and challenges for airlines. This research project is going to apply in depth what factors influence airline passenger satisfaction using a comprehensive dataset sourced from Kaggle. By applying advanced statistical and machine learning techniques, I shall explore various dimensions of seat comfort, legroom, cleanliness, and onboard services to identify key drivers of satisfaction. The insights gained will not only help pinpoint areas for improvement but also in tailoring services that will meet or exceed passenger expectations. By rigorously carrying out data analysis, including the outlier detection, correlation analysis, and predictive modeling, this project will provide recommendations that might transform passengers' experiences and operational efficiencies in the airline sector. Our research focuses on the use of data analytics by airlines to improve passenger satisfaction, Which is a critical component of their success and competition in this fast-paced sector.

2. Objectives

- **Assess Overall Passenger Satisfaction**: To evaluate the general satisfaction levels among airline passengers based on various service aspects like seat comfort, cleanliness, in-flight services, and legroom.

- **Identify Key Satisfaction Drivers**: To determine which features most significantly influence passenger satisfaction, using statistical methods such as correlation and regression analysis.

- **Detect and Analyze Outliers**: To identify and examine outliers in the dataset to understand their impact on the analysis and ensure the accuracy of the predictive models.

- **Evaluate the Impact of Demographic Factors**: To analyze how different demographic factors like age, gender, and customer type influence satisfaction levels.

- **Predictive Modeling**: To develop a predictive model that accurately forecasts passenger satisfaction based on input variables, which can be utilized to improve service quality proactively.

- **Recommend Strategic Improvements**: Based on the analysis findings, to provide actionable recommendations for airlines to enhance passenger satisfaction and optimize service delivery.

- **Visualize Data Insights**: To create visual representations of the data that highlight significant patterns and trends, aiding in the comprehension and presentation of research findings.

3. Dataset Intro

This dataset consists of an overview of passenger satisfaction across various services in airline travel and is good enough for analysis and pattern recognition in customer service in the airline industry. The details of the dataset are:

Source:kaggle(https://www.kaggle.com/datasets/mysarahmadbhat/airline-passenger-satisfaction)

Dataset Title: Airline Passenger Satisfaction

Number of Entries: 129,880

Number of Attributes: 24

No.	Attribute Name	Description
1	ID	ID
2	Gender	Gender of the passengers (Female, Male)
3	Age	The actual age of the passengers
4	Customer Type	The customer type (Loyal customer, disloyal customer)
5	Type of Travel	Purpose of the flight of the passengers (Personal Travel, Business Travel)
6	Class	Travel class in the plane of the passengers (Business, Eco, Eco Plus)
7	Flight Distance	The flight distance of this journey
8	Departure Delay	Minutes delayed when arrival
9	Arrival Delay	Arrival Delay
10	Departure and Arrival Time Convenience	Satisfaction level of Departure/Arrival
11	Ease of Online Booking	booking platform score
12	Check-in Service	check-in service score
13	Online Boarding	online boarding conveniency
14	Gate Location	How Gate is accessible

15	On-board Service	Satisfaction level of On-board service
16	Seat Comfort	Seat comfort level
17	Leg Room Service	Leg Room service score
18	Cleanliness	cleanliness or hygiene in flight
19	Food and Drink	food and drink service
20	In-flight Service	Inflight service satisfaction
21	In-flight Wi-Fi Service	Wi-Fi availability
22	In-flight Entertainment	OTT or On Demand entertainment
23	Baggage Handling	Baggage or luggage handling service
24	Satisfaction	Target variable with level satisfied or neutral/dissatisfied

Table 1: description of dataset

4. Data Understanding

First of all, all the basic necessary libraries of python like Pandas, NumPy, Scikit-learn, Seaborn etc., are imported into Jupyter Notebook.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn import metrics
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score, accuracy_score, classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
import xgboost as xgb
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing
```

Figure 1: Importing all the necessary libraries

The data samples of the CSV file loaded into dataframe by using the read_csv() function are: This function returns the data of the CSV file as a two-dimensional data structure with encoded axes, called data, as shown below:

	ID	Gender	Age	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	...	On-board Service	Seat Comfort	Leg Room Service	Cleanliness	Food and Drink	In-flight Service	In-flight Wifi Service	In-flight Entertainment	Baggage Handling	Satisfaction
0	1	Male	48	First-time	Business	Business	821	2	5.0	3	...	3	5	2	5	5	5	3	5	5	Neutral or Dissatisfied
1	2	Female	35	Returning	Business	Business	821	26	39.0	2	...	5	4	5	5	3	5	2	5	5	Satisfied
2	3	Male	41	Returning	Business	Business	853	0	0.0	4	...	3	5	3	5	5	3	4	3	3	Satisfied
3	4	Male	50	Returning	Business	Business	1905	0	0.0	2	...	5	5	5	4	4	5	2	5	5	Satisfied
4	5	Female	49	Returning	Business	Business	3470	0	1.0	3	...	3	4	4	5	4	3	3	3	3	Satisfied

5 rows × 24 columns

Figure 2: Reading the dataset

```
data.shape
```

```
(129880, 24)
```

Figure 3: Shape of the dataset

```
1  data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 24 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   ID                                      129880 non-null  int64
 1   Gender                                  129880 non-null  object
 2   Age                                     129880 non-null  int64
 3   Customer Type                           129880 non-null  object
 4   Type of Travel                          129880 non-null  object
 5   Class                                   129880 non-null  object
 6   Flight Distance                         129880 non-null  int64
 7   Departure Delay                         129880 non-null  int64
 8   Arrival Delay                           129880 non-null  float64
 9   Departure and Arrival Time Convenience  129880 non-null  int64
 10  Ease of Online Booking                  129880 non-null  int64
 11  Check-in Service                        129880 non-null  int64
 12  Online Boarding                         129880 non-null  int64
 13  Gate Location                           129880 non-null  int64
 14  On-board Service                        129880 non-null  int64
 15  Seat Comfort                            129880 non-null  int64
 16  Leg Room Service                        129880 non-null  int64
 17  Cleanliness                             129880 non-null  int64
 18  Food and Drink                          129880 non-null  int64
 19  In-flight Service                       129880 non-null  int64
 20  In-flight Wifi Service                  129880 non-null  int64
 21  In-flight Entertainment                 129880 non-null  int64
 22  Baggage Handling                        129880 non-null  int64
 23  Satisfaction                            129880 non-null  object
dtypes: float64(1), int64(18), object(5)
memory usage: 23.8+ MB
```

Figure 4: Dataset Information

After successful storage of data, one can review the data in the tabular format and can access each part of data easily. The dataframe's shape is (12*9880,24)*, which indicates the number of samples and features from the dataset.

After Studying the data, we can see that there are two types of features is available.

1. Numerical features
2. Categorical features

The numerical features are:

1. ID
2. Age
3. Flight Distance
4. Departure Delay
5. Arrival Delay
6. Departure and Arrival Time Convenience
7. Ease of Online Booking
8. Check-in Service
9. Online Boarding
10. Gate Location
11. On-board Service
12. Seat Comfort
13. Leg Room Service
14. Cleanliness
15. Food and Drink
16. In-flight Service
17. In-flight Wifi Service
18. In-flight Entertainment
19. Baggage Handling

The Categorical features are:

1. Gender
2. customer type
3. type of travel
4. class
5. satisfaction

For a better understanding of the data we apply data.isnull().sum() for null values in in feature. After applying that we can see there is null value in arrival delay.

```
1  data.isnull().sum()
```

```
ID                                          0
Gender                                      0
Age                                         0
Customer Type                               0
Type of Travel                              0
Class                                       0
Flight Distance                             0
Departure Delay                             0
Arrival Delay                             393
Departure and Arrival Time Convenience      0
Ease of Online Booking                      0
Check-in Service                            0
Online Boarding                             0
Gate Location                               0
On-board Service                            0
Seat Comfort                                0
Leg Room Service                            0
Cleanliness                                 0
Food and Drink                              0
In-flight Service                           0
In-flight Wifi Service                      0
In-flight Entertainment                     0
Baggage Handling                            0
Satisfaction                                0
dtype: int64
```

Figure 5 : total number of Null values in each feature of dataset

After Detecting the null values in arrival delay, we replace the null values with the mean.

```
data['Arrival Delay'].fillna(data['Arrival Delay'].mean(),inplace=True)
```

Figure 6: Replacing null values with mean

5. Data visualization

We make some graphs and plots using the famous libraries matplotlib and seaborn to represent the dataset. Plots are also represented below. Individual bar graphs are created in order to understand the distribution of all attributes within the provided dataset.

(1) First, we start with the pie chart. This pie chart Pie chart shows that 56.55% individuals are Neutral or Dissatisfied. while 43.45% of passengers are satisfied.

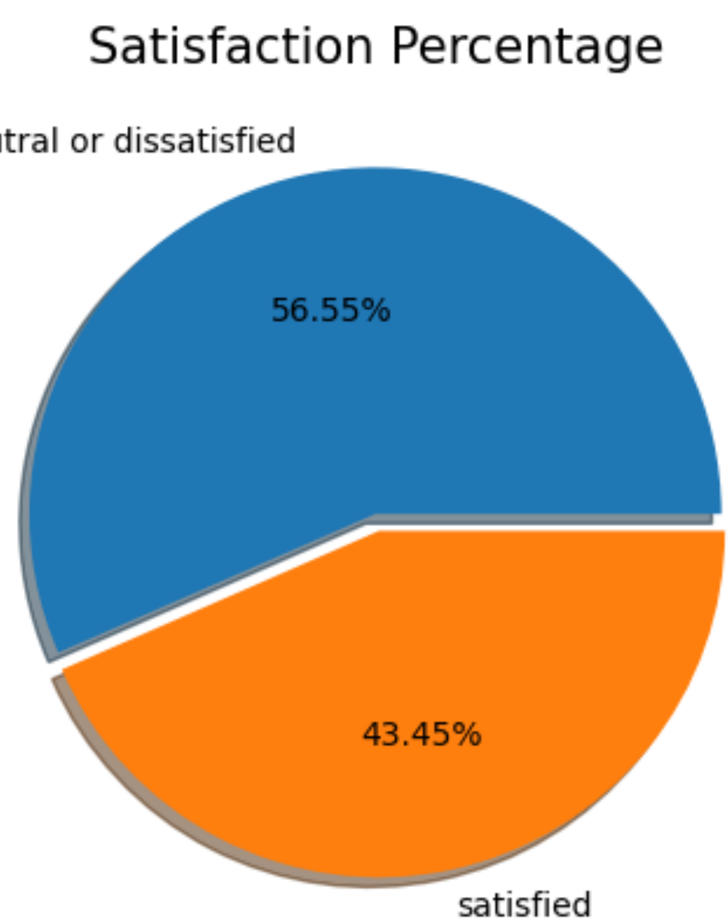


Figure 7: Pie chart of passenger satisfaction

(2) This bar chart representing the distribution of airline passengers by gender. The chart indicates that

- Approximately 49.26% of the passengers are male.
- Approximately 50.74% of the passengers are female.

This visual suggests that there is a nearly equal distribution between male and female passengers in the dataset.

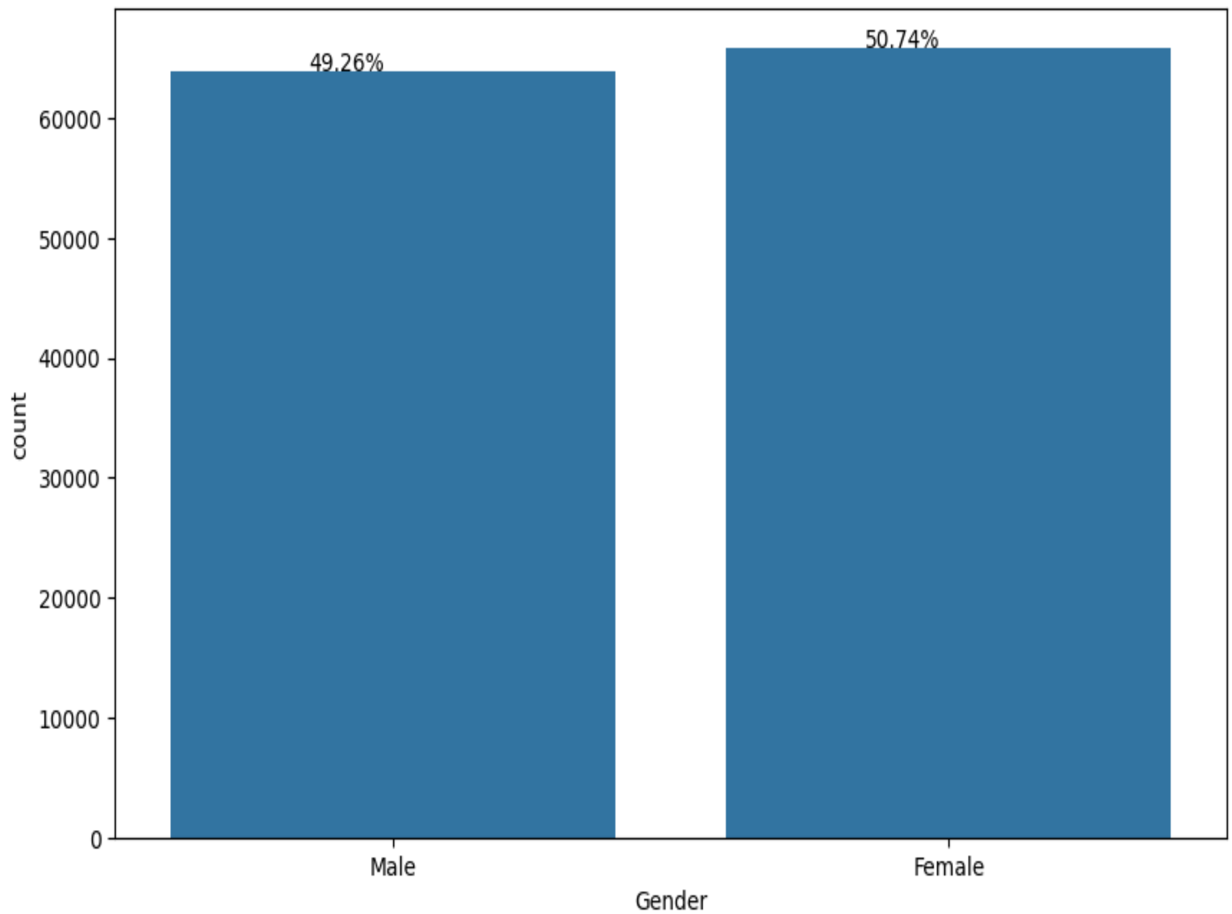


Figure 8: Bar plot for comparison between male and female

(3) This grouped bar chart showing the satisfaction levels of airline passengers categorized by gender. the chart indicates Both genders have a similar number of individuals who are satisfied, but there are slightly more females than males who are neutral or dissatisfied with the service or product. Overall, satisfaction levels appear quite balanced between the two genders.

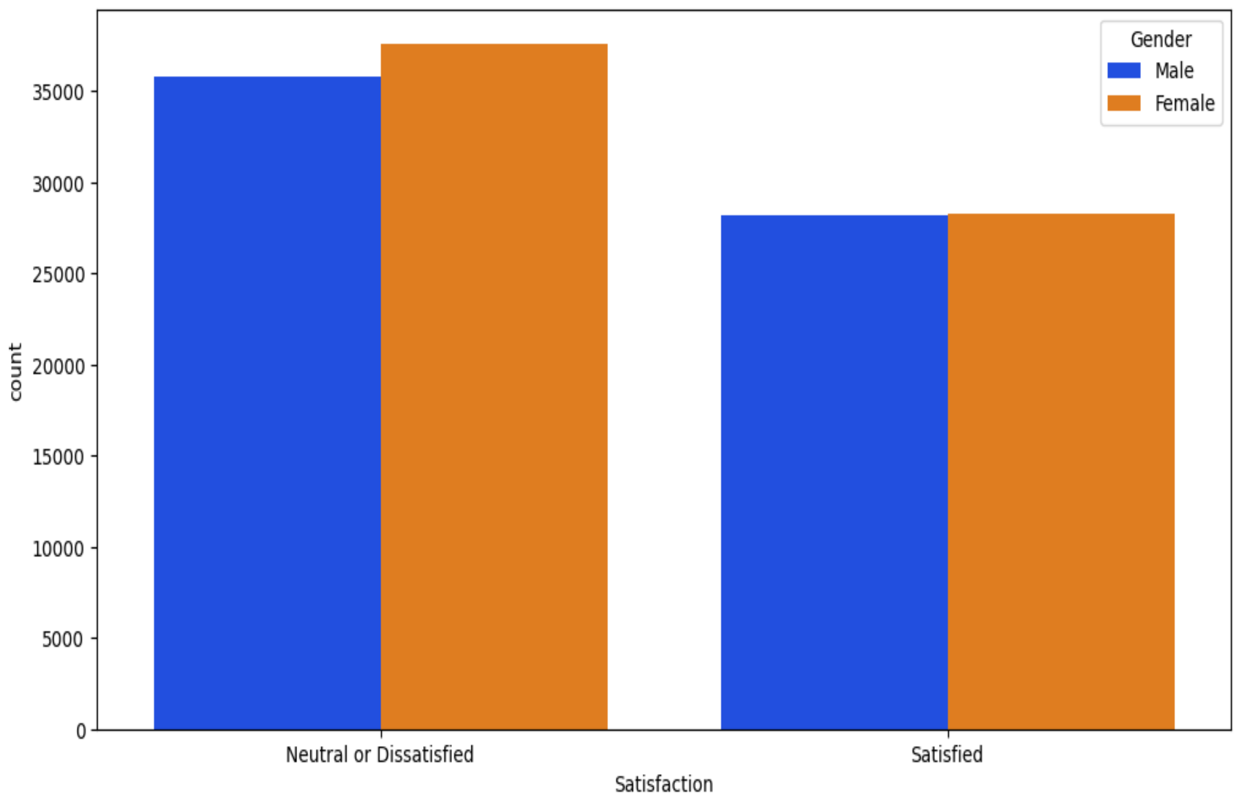


Figure 9: Airline Passenger Satisfaction by Gender

(4) The image you've uploaded is a bar chart titled "Age Distribution," which represents the age-wise distribution of airline passenger satisfaction. The chart is color-coded with two categories:

Green Bars: Represent passengers who are either neutral or dissatisfied.
Orange Bars: Represent passengers who are satisfied.

The x-axis lists passenger ages from approximately 7 to 85 years, and the y-axis shows the count of passengers in each age group. This visualization helps in analyzing how passenger satisfaction varies across different age groups, potentially indicating trends such as particular age groups being more likely to be satisfied or dissatisfied with airline services.

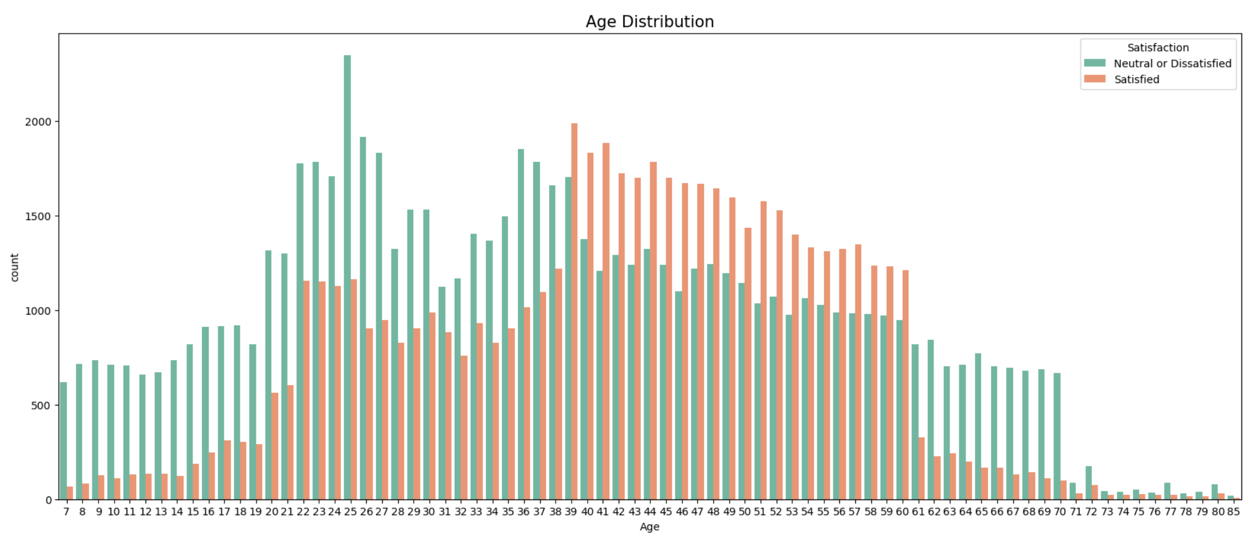


Figure 10: Age-Wise Satisfaction Trends Among Airline Passengers

After analyzing the chart we can say that people in minor age and youngers tend to be more dissatisfied while people aged from middle aged to sexagenarian tend to be more satisfied.

(5) This chart visually compares passenger satisfaction across different travel classes—Business, Economy, and Economy Plus—based on their satisfaction.

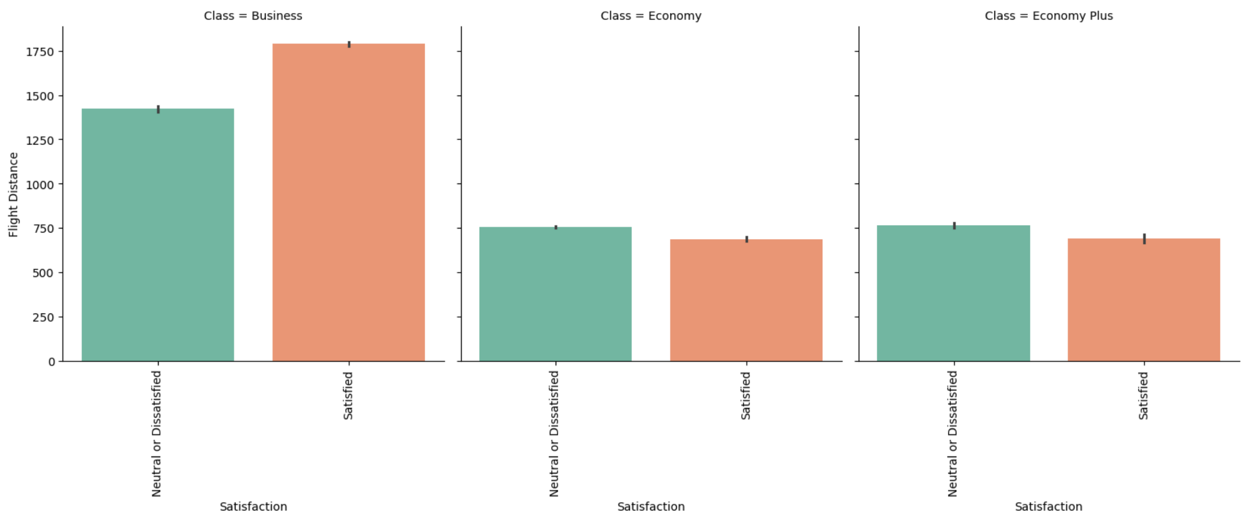


Figure 11 : Airline Passenger Satisfaction by travel class

In business class customers tends to be more satisfied. While in economy and eco plus class, passengers are having the same ratio of satisfied and dissatisfied.

(6) The chart you've provided depicts the relationship between seat comfort and passenger satisfaction in an airline context. It categorizes passengers into two groups—those who are neutral or dissatisfied and those who are satisfied. (rated from 0 to 5).

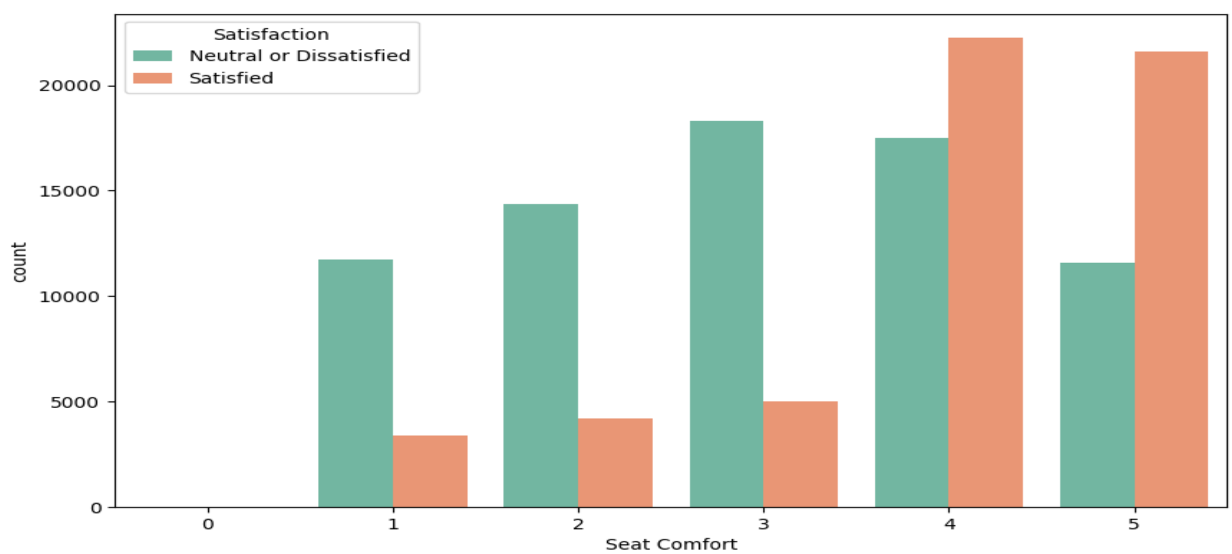
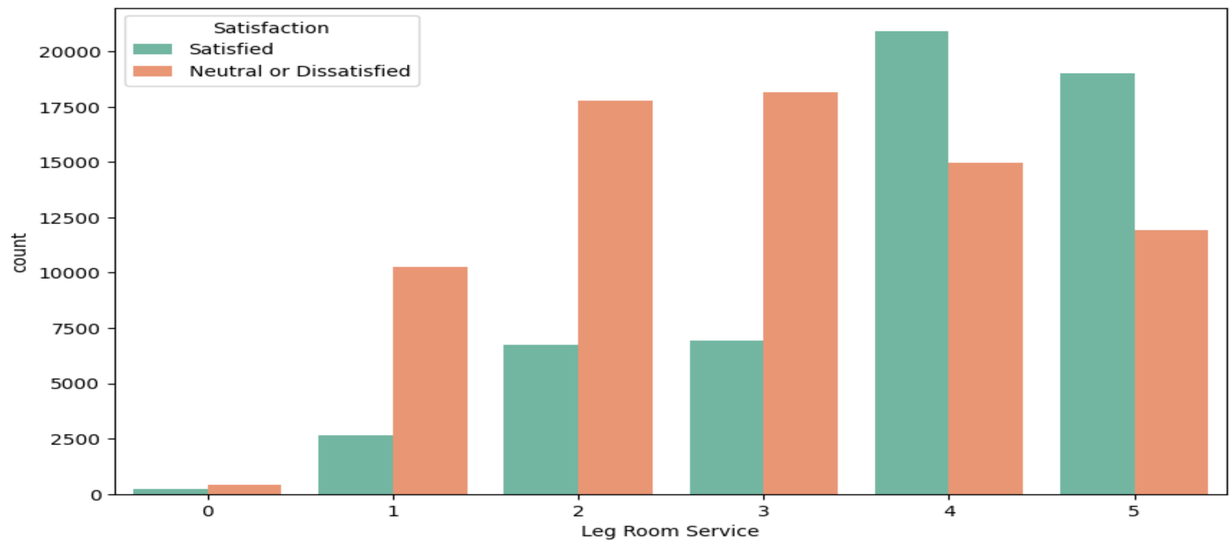


Figure 12 : Airline Passenger Satisfaction by seat comfort

(7) The chart shows the relationship between legroom service and passenger satisfaction. It categorizes passengers into two groups—those who are "Satisfied" and those who are "Neutral or



Dissatisfied"—across different levels of leg room service (rated from 0 to 5).

Figure 13 : Airline Passenger Satisfaction by Legroom

(8) The chart illustrates the impact of cleanliness on passenger satisfaction in an airline context. It categorizes passengers into two groups: "Neutral or Dissatisfied" and "Satisfied," across different levels of cleanliness (rated from 0 to 5).

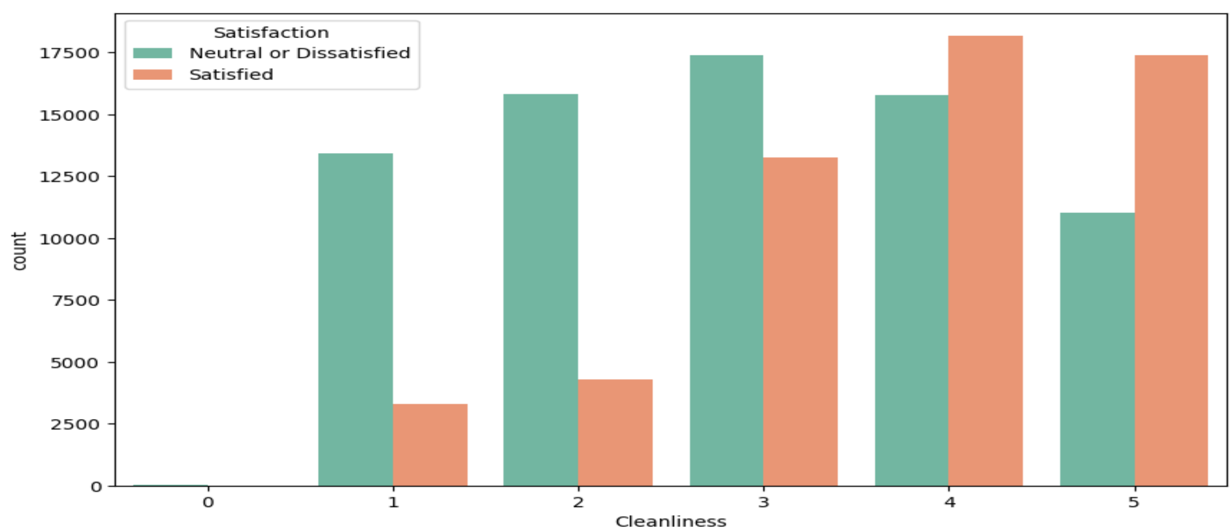


Figure 14 : Airline Passenger Satisfaction by Cleanliness

(9) The chart illustrates the impact of Food and drink on passenger satisfaction in an airline context. It categorizes passengers into two groups: "Neutral or Dissatisfied" and "Satisfied," across different levels of cleanliness (rated from 0 to 5).

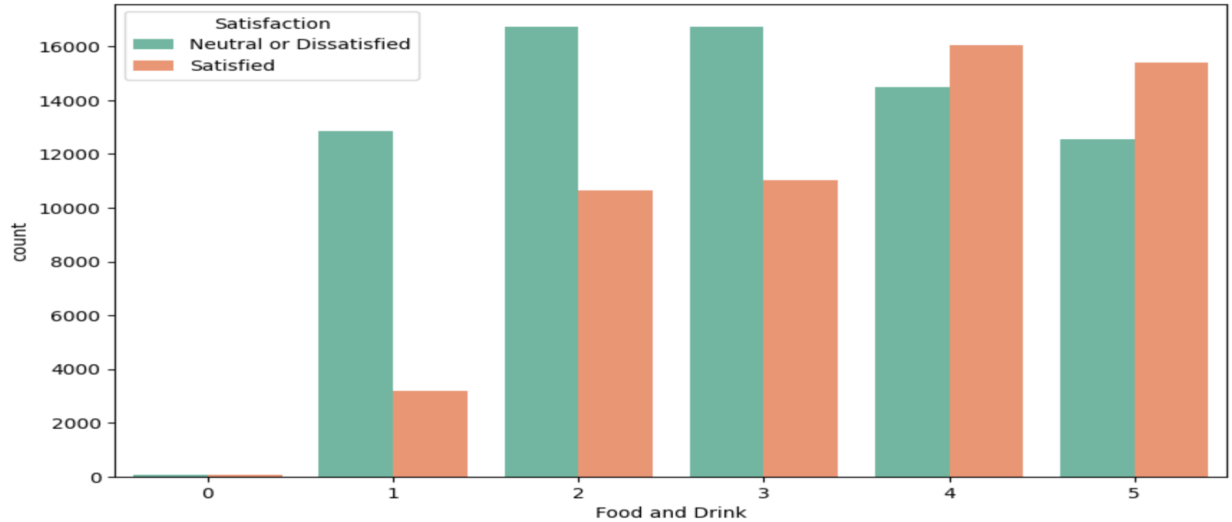


Figure 15 : Airline Passenger Satisfaction by Food and Drink

Food and drinks are not playing a major role to swing the customer satisfaction, because with 4 and 5 score of food/drinks not showing a big gap between satisfied and dissatisfied.

(10) The chart illustrates the impact of Bagge handling on passenger satisfaction in an airline context. It categorizes passengers into two groups: "Neutral or Dissatisfied" and "Satisfied" across different levels of cleanliness (rated from 0 to 5).

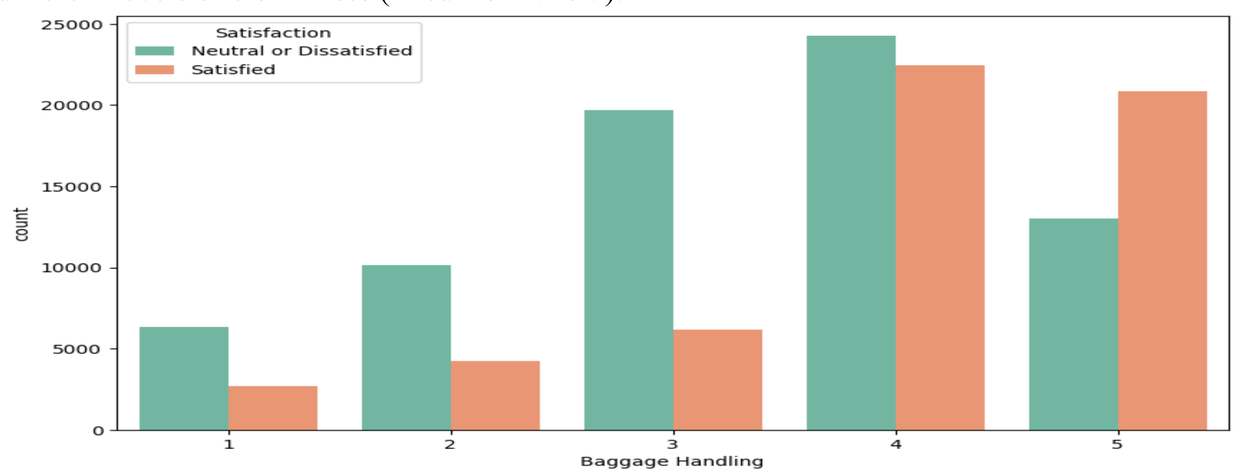


Figure 16 : Airline Passenger Satisfaction by Food and Drink

6. Data preprocessing

The detection of outliers is key to accurate and quality data by identifying data entry errors, experimental errors, or anomalies that distort statistical measures and models. Outliers can have a strong impact on the creation of strong machine learning models and making informed decisions, by maintaining the integrity and reliability of outcomes from analysis.

For detecting the outliers, we used seaborn library to create a boxplot for visualizing the distribution of variables in a dataframe.

```
data.plot(kind="box", subplots=True, layout=(6,4), figsize=(13,12))
```

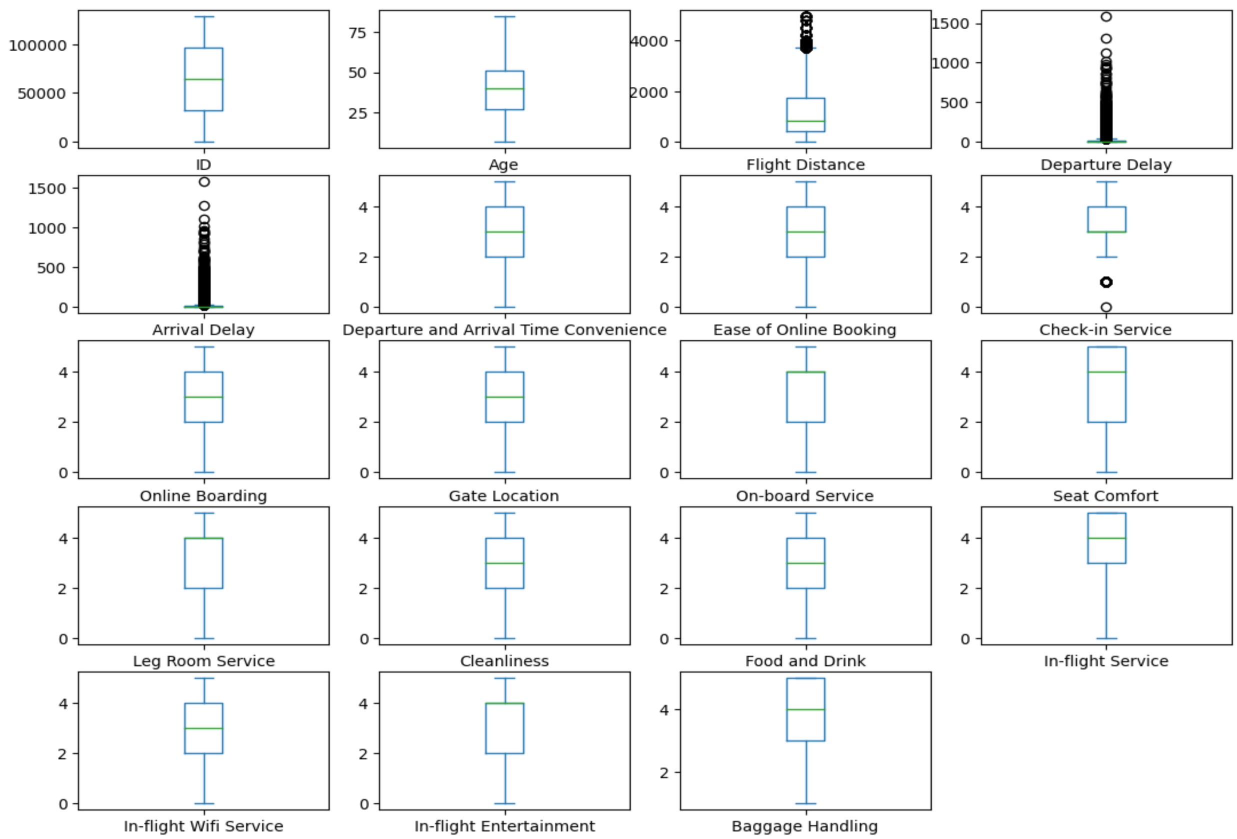


Figure 17 : Detecting Outliers

We Detected the outliers in Arrival delay and departure delay. We are replacing outliers with the Median.

This correlation matrix provides a visual representation of the relationships between various factors related to airline passenger experience and satisfaction. Each cell shows the correlation coefficient between pairs of variables, with color intensities ranging from dark blue (strong positive correlation) to white (no correlation). Notably, there are significant correlations between class and type of travel, satisfaction and in-flight entertainment, and several service factors like cleanliness and seat comfort. This matrix is useful for identifying key drivers of passenger satisfaction and areas where improvements can lead to enhanced customer experiences.

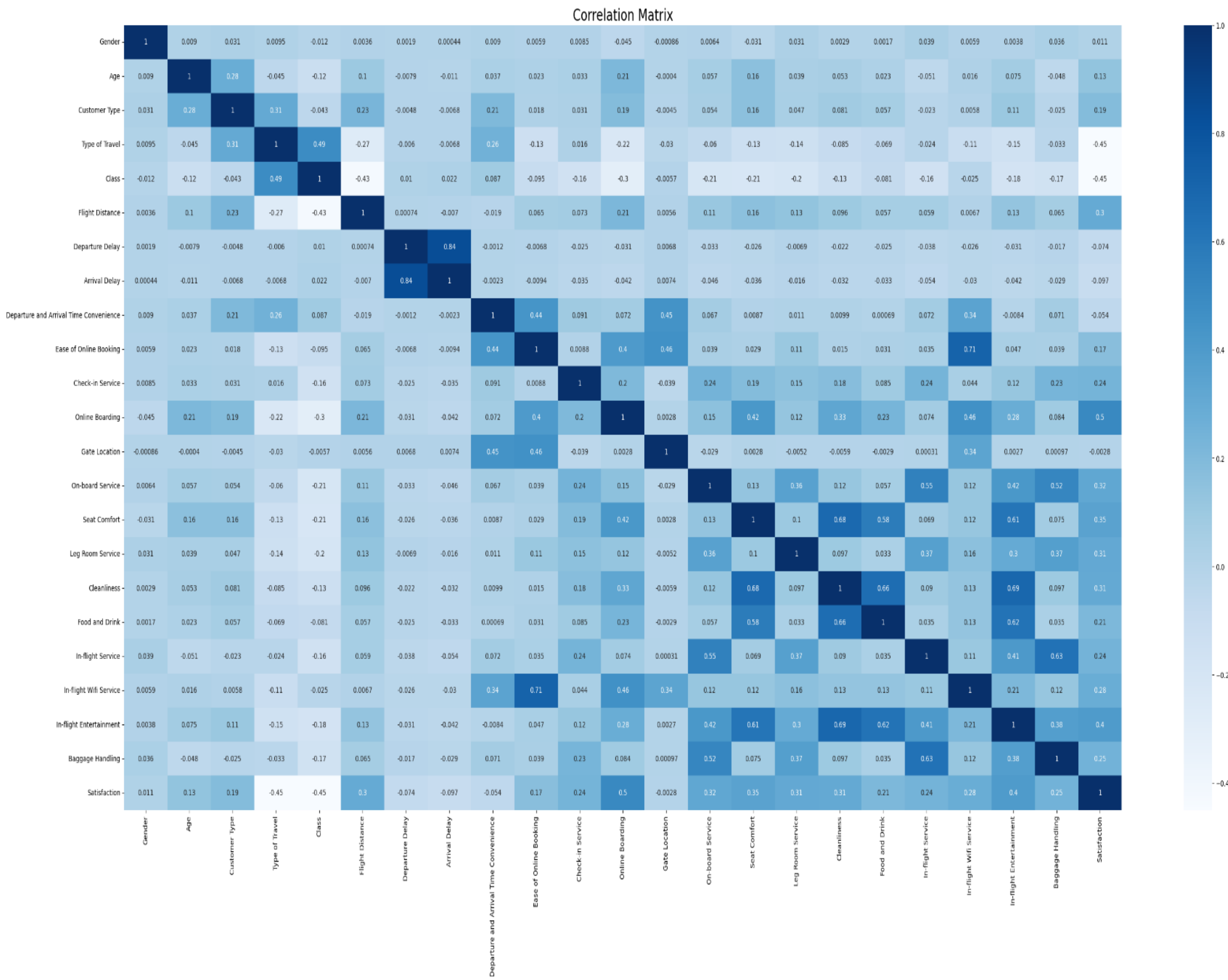


Figure 18 : correlation among variables

The correlation values indicate how different factors related to a flight experience are associated with overall passenger satisfaction.

Feature	Correlation Value
Online Boarding	0.50
In-flight Entertainment	0.40
Seat Comfort	0.35
On-board Service	0.32
Leg Room Service	0.31
Cleanliness	0.31
Flight Distance	0.30
In-flight Wi-Fi Service	0.28
Baggage Handling	0.25
Check-in Service	0.24
In-flight Service	0.24
Food and Drink	0.21
Customer Type	0.19
Ease of Online Booking	0.17
Age	0.13
Gender	0.011
Gate Location	-0.0028
Departure and Arrival Time Convenience	-0.054
Departure Delay	-0.074
Arrival Delay	-0.097
Type of Travel	-0.45
Class	-0.45

Table 2: Correlation values related with passenger satisfaction

This table reflects the relationship between each listed attribute and overall passenger satisfaction, with positive values indicating a positive relationship and negative values a negative relationship.

Correlation using VIF

The Variance Inflation Factor (VIF) is a method commonly used to identify multicollinearity in the independent variables of a regression model.

	variables	VIF Factor
0	Gender	1.960837
1	Age	6.177706
2	Customer Type	8.678068
3	Type of Travel	3.080949
4	Class	2.822857
5	Flight Distance	2.901078
6	Departure Delay	4.866379
7	Arrival Delay	4.913466
8	Departure and Arrival Time Convenience	8.368208
9	Ease of Online Booking	13.245150
10	Check-in Service	9.329659
11	Online Boarding	13.964220
12	Gate Location	8.893576
13	On-board Service	13.811314
14	Seat Comfort	18.485297
15	Leg Room Service	9.501520
16	Cleanliness	20.565990
17	Food and Drink	14.063519
18	In-flight Service	20.079215
19	In-flight Wifi Service	13.106106
20	In-flight Entertainment	27.294312
21	Baggage Handling	11.398089
22	Satisfaction	3.764256

Figure 19 : correlation using VIF

We removed the In-flight Entertainment feature to remove multi collinearity. Arrival Delay and Departure delay are having a high correlation, so we are removing one of them. Also, we Dropped ID feature as it is not contributing towards target prediction.

Label encoding is a technique used to convert categorical text data into a numerical format that can be interpreted by machine learning algorithms. It involves assigning a unique integer to each category of the variable. This process is essential when dealing with categorical data in machine learning, as most algorithms require input to be numeric. Here we are converting object features to numeric values.

```python
for column in data.columns:
    if data[column].dtype == np.number: continue

    data[column] = LabelEncoder().fit_transform(data[column])
```

Figure 20: converting object features to numeric values

7. Splitting the data

Before going into data splitting to train and test data, we will create feature (X) and target (y) variables from the dataset. In the dataset, we considered Satisfaction of passengers as the target variable and the remaining features considered as the independent variables.

```
[ ]  x = df1.drop(['Satisfaction'] , axis = 1).values
     y= df1['Satisfaction' ].values
     x_train , x_test , y_train , y_test = train_test_split(x,y , test_size= 0.2 , random_state=42)
```

Figure 21: Creating X & y

After forming the input and target variables, we will split into train and test datasets. We will train our model on the train dataset and observe the accuracy of built model prediction or classification on the test data. By splitting the dataset, we can obtain unbiased and real accuracy of model on unseen data.

The dataset is split into 80% of train data and 20% of test data, basically an 8-2 split.

8. Model building & training

To address the problem of predicting customer satisfaction based on the features in the airline customer satisfaction dataset, we used several machine learning models. This section details the setup, execution, and initial evaluation of these models. Mainly we used Classification models and ensemble models.

Classification models

1. Logistic Regression
2. Decision tree
3. Naïve bayes
4. KNN
5. SVC
6. MLP Classifier

Ensemble models

1. Random Forest
2. Bagging
3. Boosting
4. Voting
5. Gradient Boosting
6. XGBOOST

8.1 Logistic Regression

First starting with Logistic Regression, we trained model with a higher maximum iteration to ensure convergence given the complexity of the dataset. After predicting the target values on the test set, a confusion matrix was generated to visually assess the model's performance.

The Logistic Regression model achieved an accuracy of 87%. It was noted for its straightforward and effective approach to basic classification problems. The execution time was moderate at about 5 seconds, making it a balanced choice in terms of speed and performance.

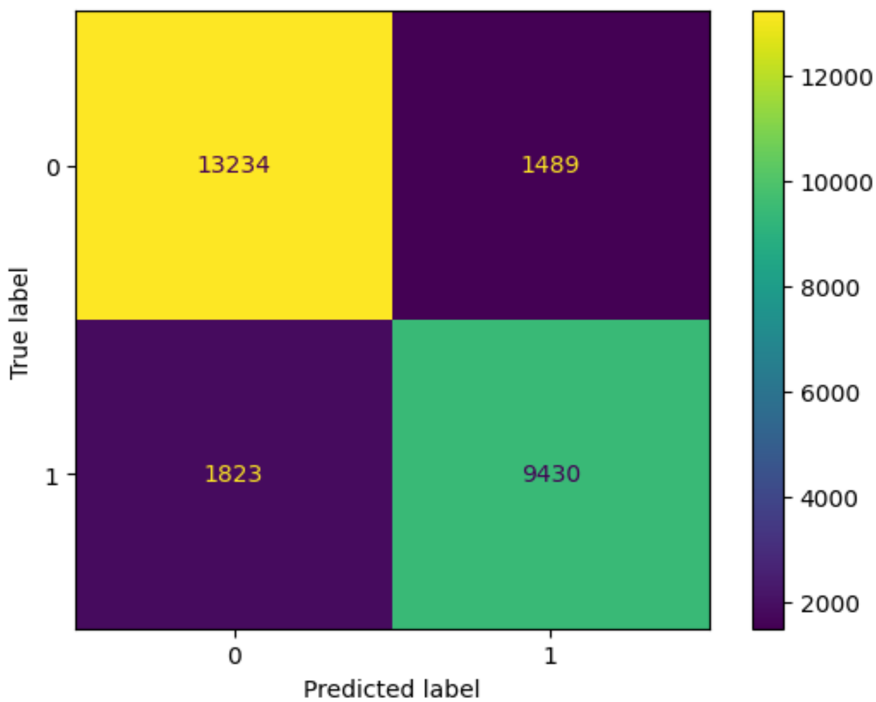


Figure 22: Confusion matrix of Logistic Regression

	precision	recall	f1-score	support
0	0.88	0.90	0.89	14723
1	0.86	0.84	0.85	11253
accuracy			0.87	25976
macro avg	0.87	0.87	0.87	25976
weighted avg	0.87	0.87	0.87	25976

The ROC curve for the Logistic Regression model demonstrates a strong discriminative performance, rapidly ascending towards a high True Positive Rate while maintaining a low False Positive Rate, indicative of a model with excellent sensitivity and specificity. The curve plateaus near the top left corner, reflecting minimal increase in false positives with additional gains in sensitivity.

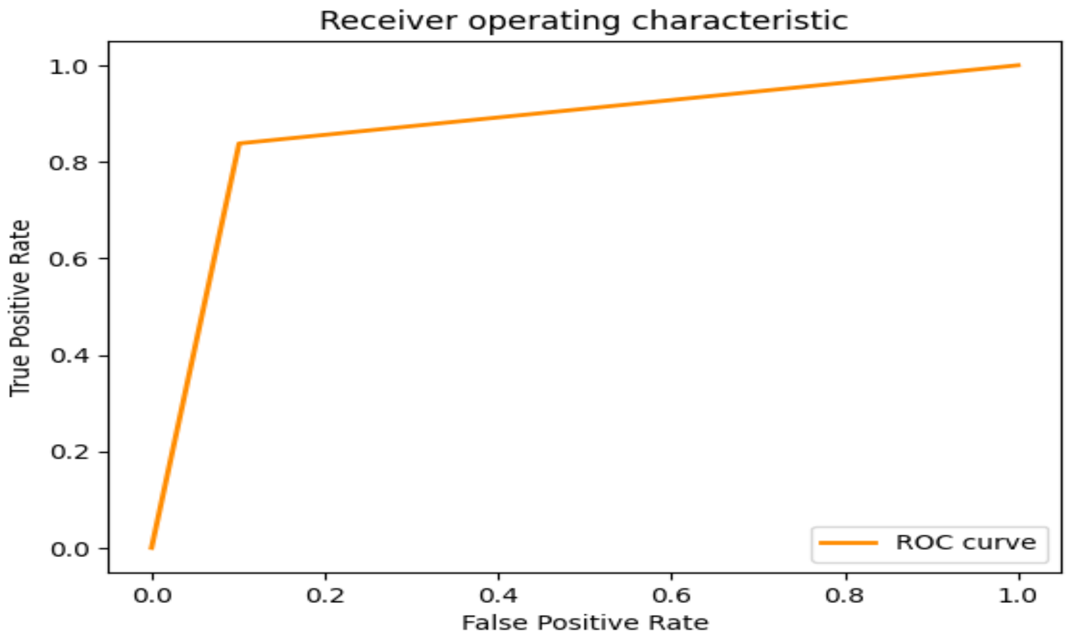


Figure 23: ROC curve for Logistic Regression

8.2 Decision Tree

A Decision Tree Classifier was initially set up with a max_depth of 5 to fit and predict the model. This was followed by generating and displaying confusion matrix and classification report to analyze the model's performance.

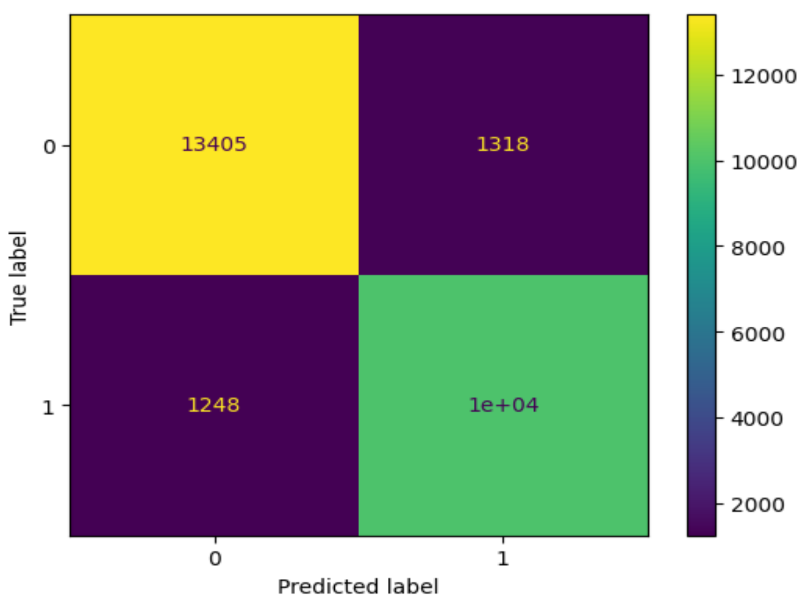


Figure 24: Confusion matrix of Decision Tree

	precision	recall	f1-score	support
0	0.91	0.91	0.91	14723
1	0.88	0.89	0.89	11253
accuracy			0.90	25976
macro avg	0.90	0.90	0.90	25976
weighted avg	0.90	0.90	0.90	25976

Further implementations experimented with increasing the max_depth to 8 and 10 to observe changes in performance, which was evaluated using the classification report. For max_depth=8 we got 94% accuracy.

	precision	recall	f1-score	support
0	0.94	0.96	0.95	14723
1	0.94	0.92	0.93	11253
accuracy			0.94	25976
macro avg	0.94	0.94	0.94	25976
weighted avg	0.94	0.94	0.94	25976

For max_depth=10, we achieved 95% accuracy.

	precision	recall	f1-score	support
0	0.94	0.97	0.95	14723
1	0.96	0.92	0.94	11253
accuracy			0.95	25976
macro avg	0.95	0.94	0.95	25976
weighted avg	0.95	0.95	0.95	25976

In our analysis, we computed the Receiver Operating Characteristic (ROC) curve for the Decision Tree model. The ROC curve, as shown, rises sharply to the top-left corner, maintaining a high True Positive Rate (TPR) while keeping the False Positive Rate (FPR) exceptionally low. This characteristic indicates strong model performance with high sensitivity and an excellent ability to classify the positive class accurately without increasing the false positives.

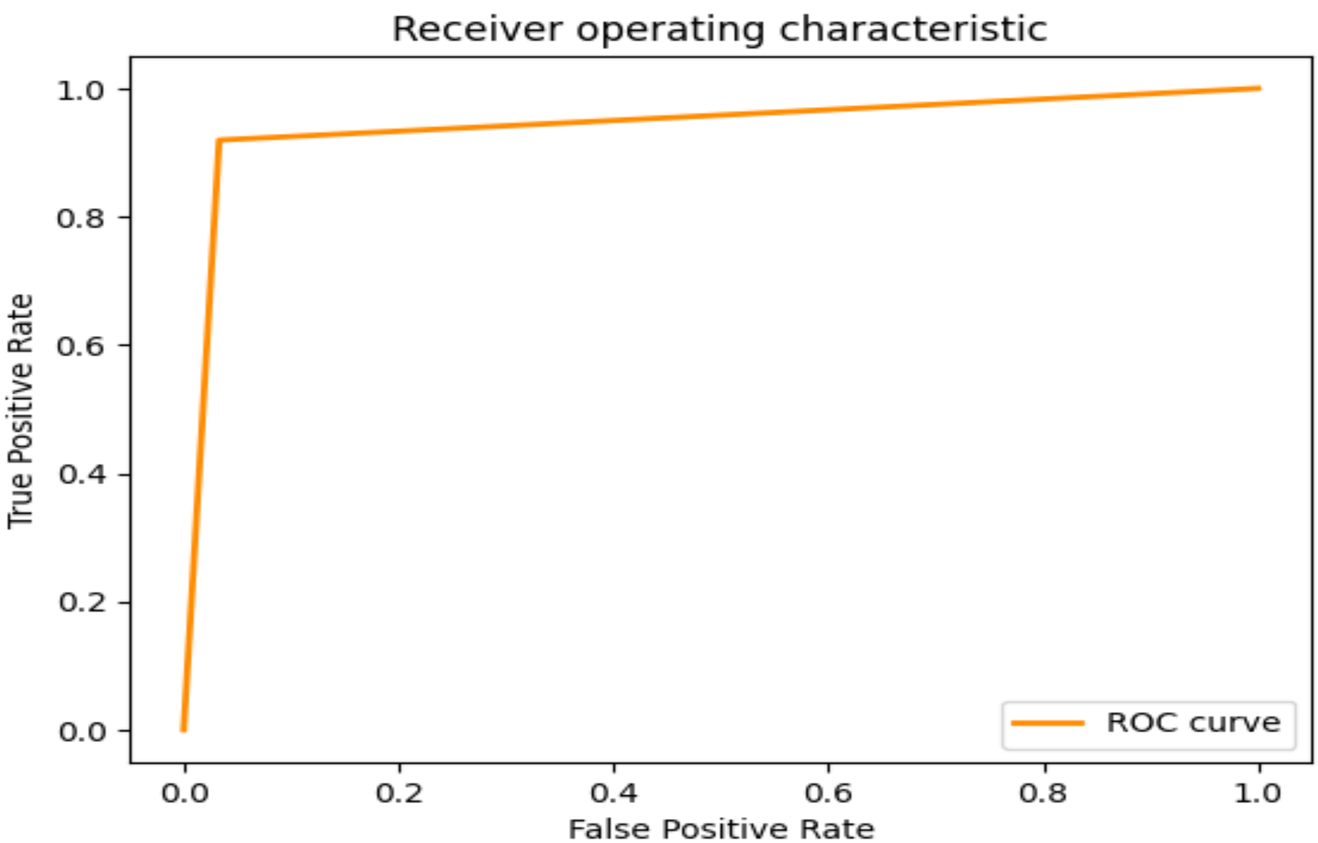


Figure 25: ROC curve of Decision Tree

8.3 Naïve bayes

We used Naïve bayes because showcasing its simplicity and effectiveness in classification tasks. By evaluating the model with a confusion matrix, we gain insights into its predictive capabilities and areas where it might be confusing one class for another.

	precision	recall	f1-score	support
0	0.87	0.90	0.89	14723
1	0.87	0.82	0.84	11253
accuracy			0.87	25976
macro avg	0.87	0.86	0.87	25976
weighted avg	0.87	0.87	0.87	25976

The Naive Bayes model exhibits robust performance in classifying customer satisfaction, achieving an overall accuracy of 87%. The F1-scores, 0.89 for satisfied and 0.84 for unsatisfied customers, further highlight its effectiveness. These metrics indicate that the Naive Bayes classifier performs uniformly well across varying conditions.

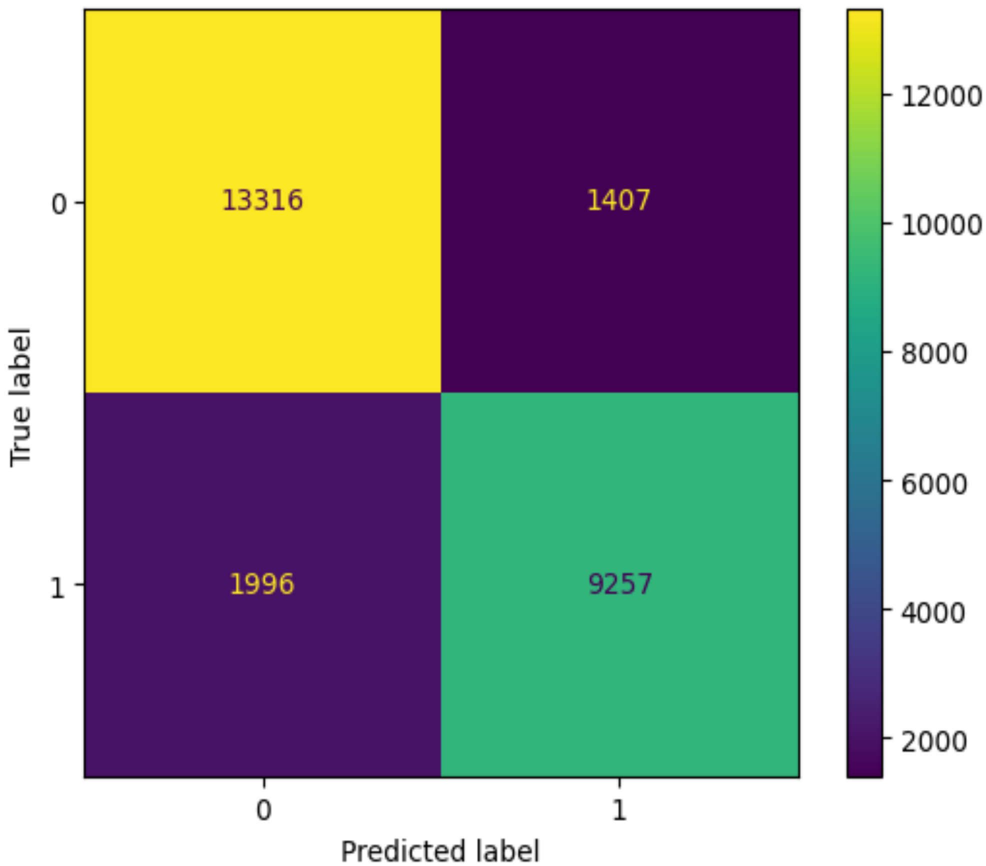
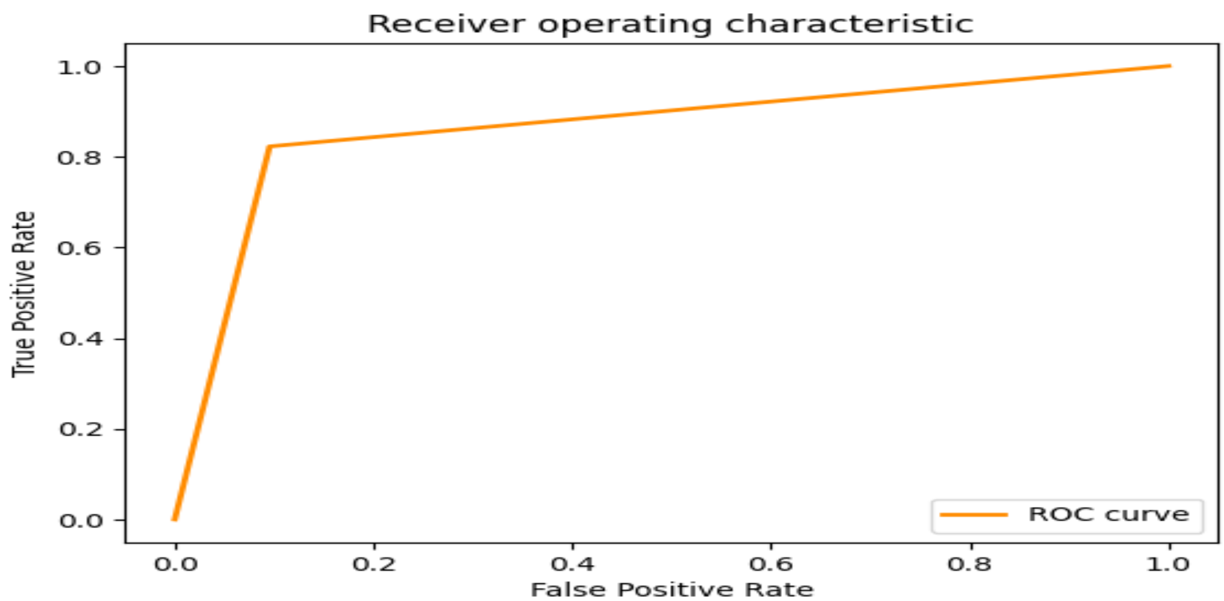


Figure 26: Confusion matrix of Naïve Bayes

The ROC curve for the Naive Bayes model shows curve rises sharply towards a high True Positive Rate (TPR) with a minimal increase in the False Positive Rate (FPR), indicative of strong sensitivity and specificity. The substantial area under the curve (AUC) highlights the model's effectiveness in classification, with its ability to correctly distinguish between the classes.



. **Figure 27: ROC curve of Naïve Bayes**

8.4 k-Nearest Neighbors (KNN)

In KNN, we have tuned its neighbors to get different results and high accuracy. Using n=3. This algorithm uses a distance rule to predict the nearest neighbor and give the results. Taken n= 3 and 8. We had the same accuracy for both.

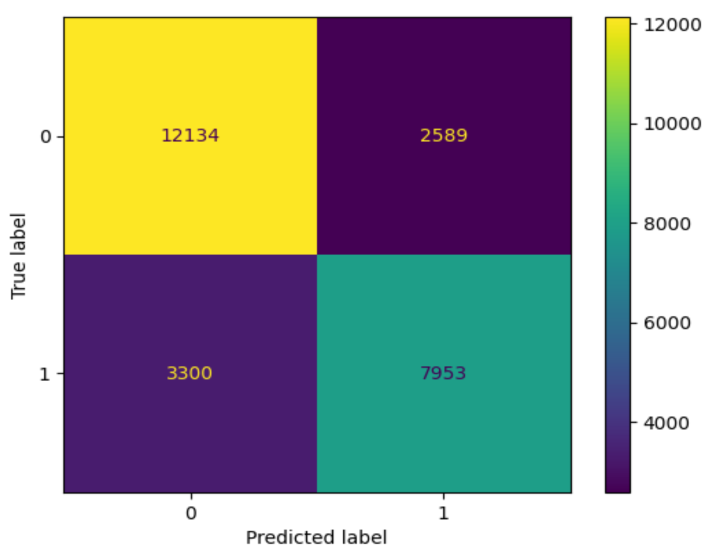


Figure 28: Confusion matrix of KNN

The model achieved an overall accuracy of 77%, indicating a moderate level of precision in distinguishing between the two classes. The precision for class 0 (satisfied customers) is 0.79, with a recall of 0.82, resulting in an F1-score of 0.80.

```
              precision    recall  f1-score   support

           0       0.79      0.82      0.80     14723
           1       0.75      0.71      0.73     11253

    accuracy                           0.77     25976
   macro avg       0.77      0.77      0.77     25976
weighted avg       0.77      0.77      0.77     25976
```

changing n_neighbors to 8

```
              precision    recall  f1-score   support

           0       0.75      0.87      0.81     14723
           1       0.79      0.63      0.70     11253

    accuracy                           0.77     25976
   macro avg       0.77      0.75      0.75     25976
weighted avg       0.77      0.77      0.76     25976
```

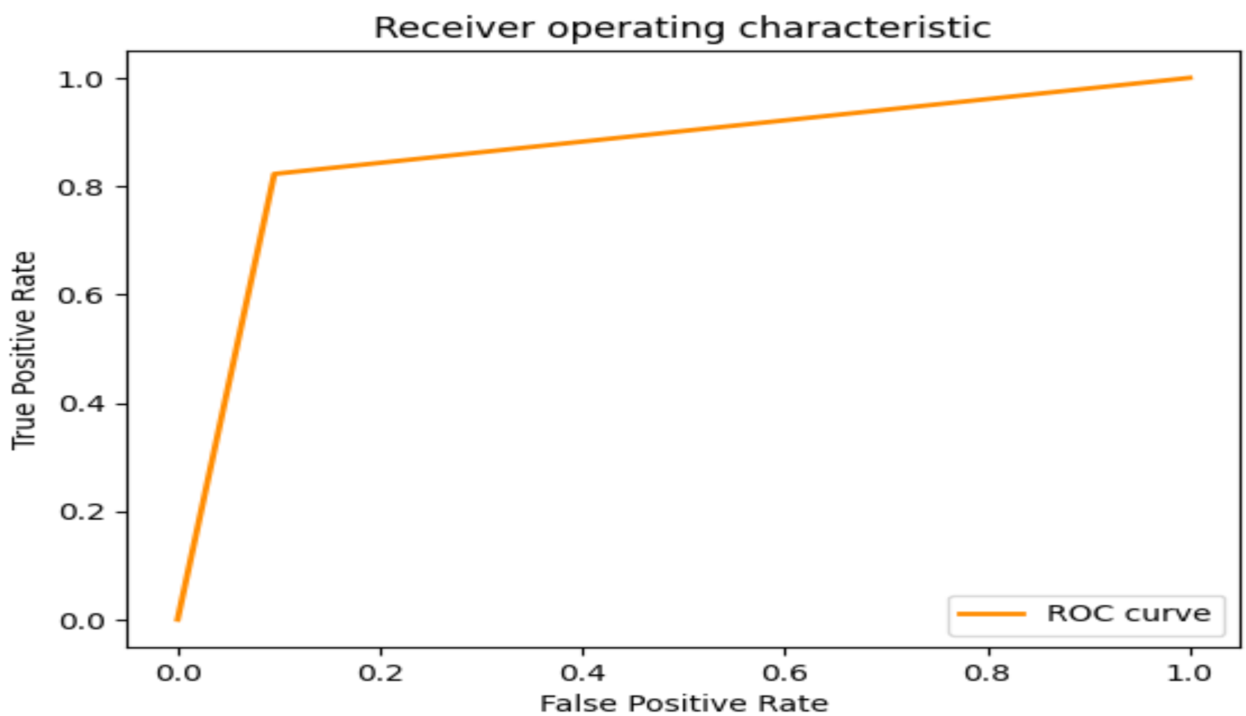


Figure 29: ROC curve of KNN

This particular ROC curve you've shown has a sharp increase around the FPR of 0.2, then another sharp rise near FPR 0.4, and then gradually approaches the top-right corner. This suggests that the classifier performs better than random guessing, as indicated by the area under the curve (AUC) being greater than 0.5.

8.5 SVC

SVM can be extended to handle non-linear boundaries using kernel tricks, which transform data into a higher-dimensional space where a linear hyperplane can effectively perform the separation. By setting c=50 we got the 86% accuracy and by setting c=100 we got 87% accuracy.

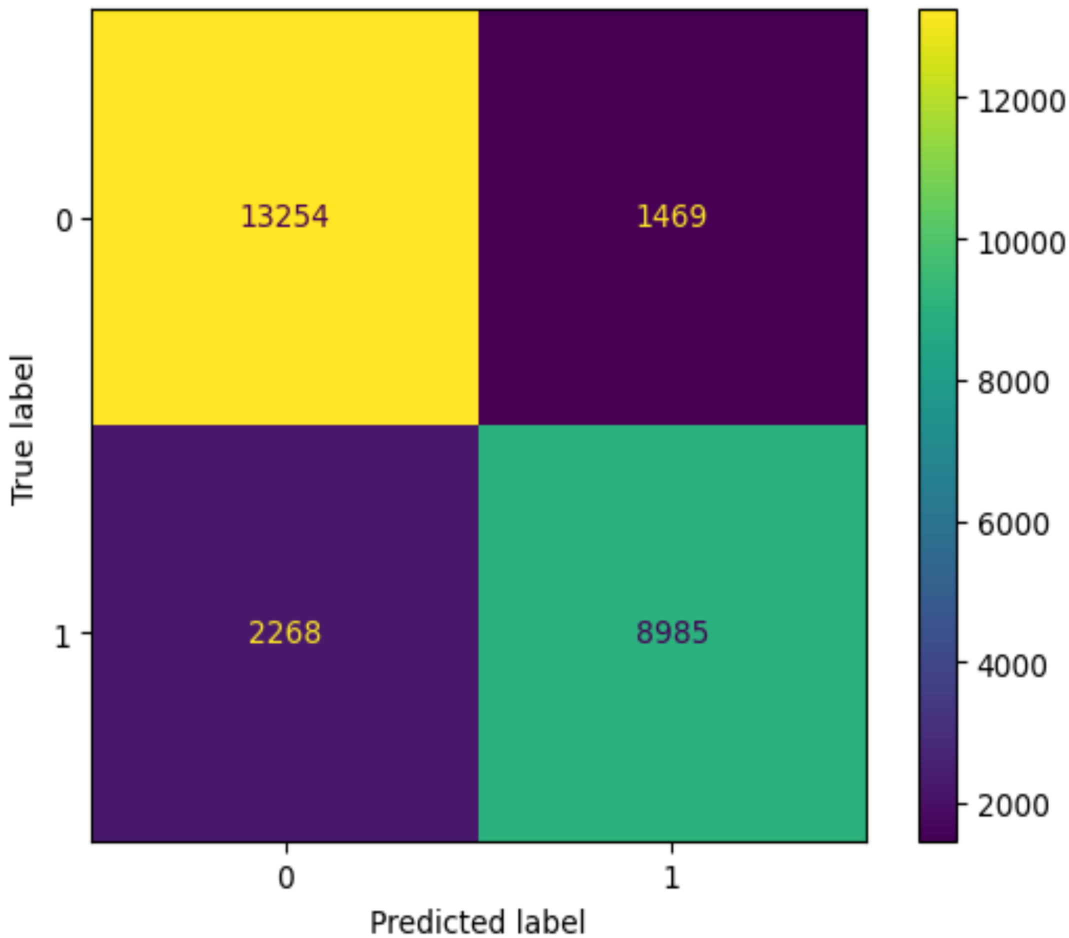


Figure 30: Confusion matrix of SVC

This confusion matrix shows the performance of a model with c=50. It shows that the model has correctly predicted 13,254 instances as class 0 (true negatives) and 8,985 instances as class 1 (true positives), demonstrating a strong ability to identify each class correctly. However, the model also made errors, misclassifying 1,469 instances as class 1 (false positives) and 2,268 instances as class 0 (false negatives). We got 87% accuracy with c=100.

	precision	recall	f1-score	support
0	0.85	0.90	0.88	14723
1	0.86	0.80	0.83	11253
accuracy			0.86	25976
macro avg	0.86	0.85	0.85	25976
weighted avg	0.86	0.86	0.86	25976

With c=100,

	precision	recall	f1-score	support
0	0.86	0.91	0.88	14723
1	0.87	0.81	0.84	11253
accuracy			0.87	25976
macro avg	0.87	0.86	0.86	25976
weighted avg	0.87	0.87	0.86	25976

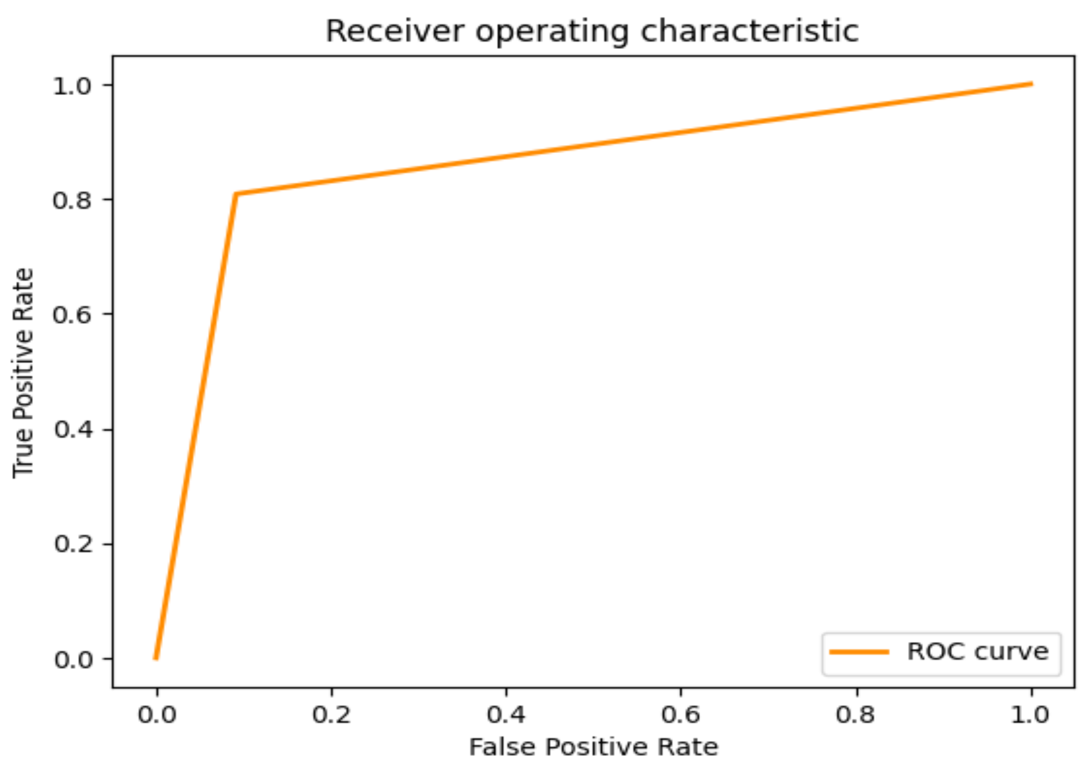


Figure 31: ROC curve of SVC

A model's performance in a binary classification task was evaluated, revealing that a precision of 0.86 and a recall of 0.91 were achieved for class 0, resulting in an F1-score of 0.88, indicative of a well-balanced prediction accuracy. For class 1, a precision of 0.87 is observed.

We can see a sharp increase in the True Positive Rate (TPR) is observed as the False Positive Rate (FPR) increases from 0 to approximately 0.2, where a plateau begins to form, indicating a high TPR for the majority of the FPR range. The ROC curve progresses towards the upper left corner, which is typically sought after in ROC analysis.

8.6 MLP Classifier

In neural network model we used MLP classifier for classification purpose. MLP is called as a Multilayer Perceptron (MLP).

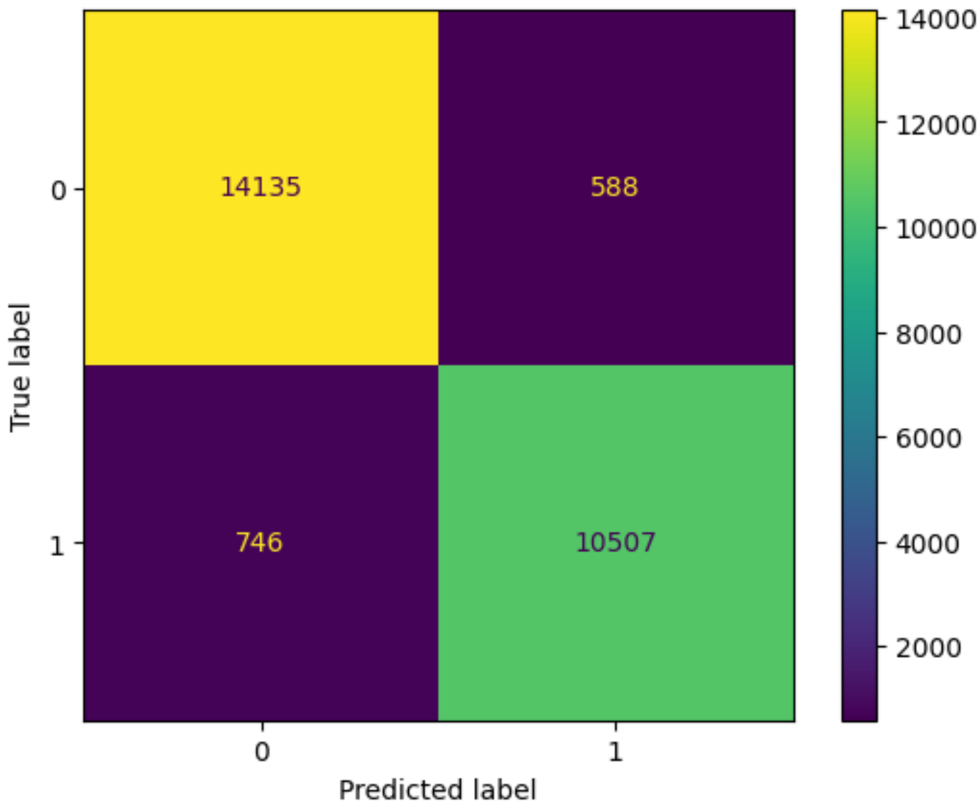


Figure 32: Confusion matrix of MLP

	precision	recall	f1-score	support
0	0.95	0.96	0.95	14723
1	0.95	0.93	0.94	11253
accuracy			0.95	25976
macro avg	0.95	0.95	0.95	25976
weighted avg	0.95	0.95	0.95	25976

The confusion matrix provided details of the classifier's performance with respect to actual and predicted classifications. It shows 14,135 true negatives, where class 0 is correctly identified, and 10,507 true positives, where class 1 is accurately predicted. The model boasts an accuracy of 0.95 and for class 0, the model achieves a precision of 0.95, a recall of 0.96, and an F1-score of 0.95 indicating a strong ability to correctly identify and classify true negatives, with a total support of 14,723 instances.

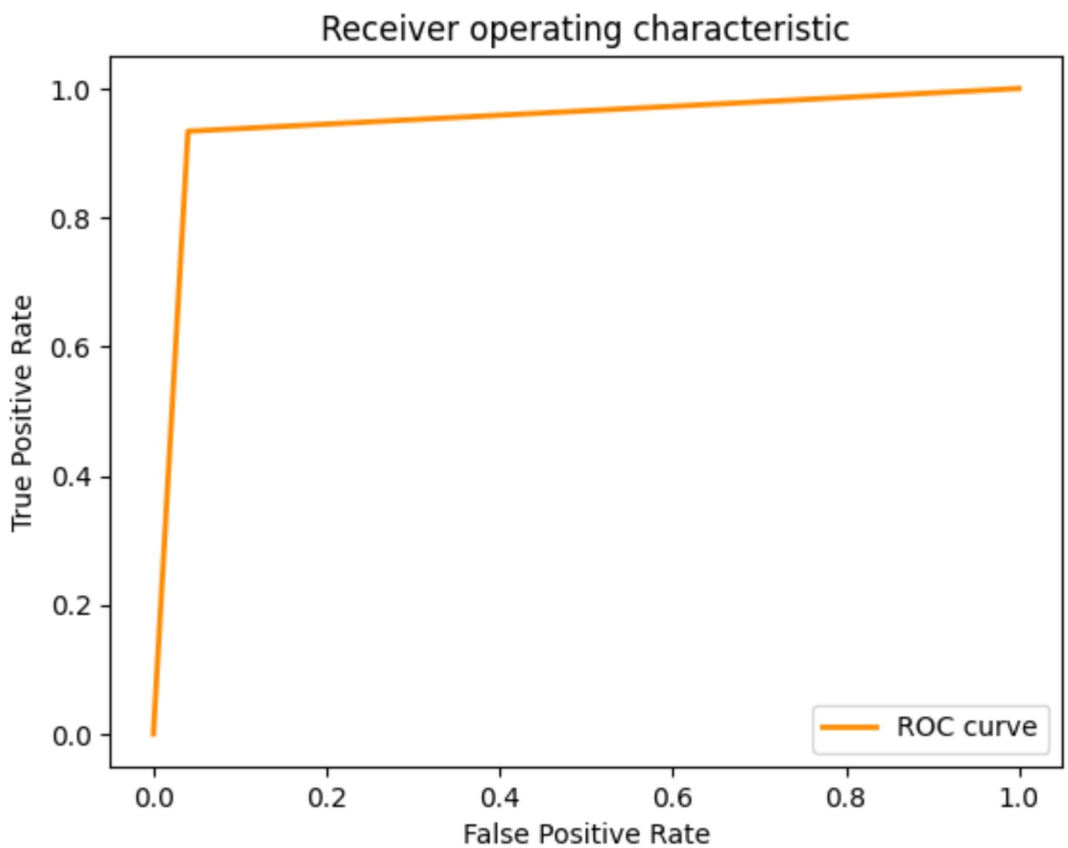


Figure 33: ROC curve of MLP

The curve rises sharply to the upper left corner, reaching close to the maximum true positive rate (TPR) of 1.0 while maintaining a low false positive rate (FPR), nearly 0, until approximately 0.2 on the FPR scale.

8.7 Random Forest

We use Random Forest Classifier is an ensemble learning method for classification . Setting the n_estimators parameter to 100 in a bagging model means that the ensemble will include 100 individual models or estimators.

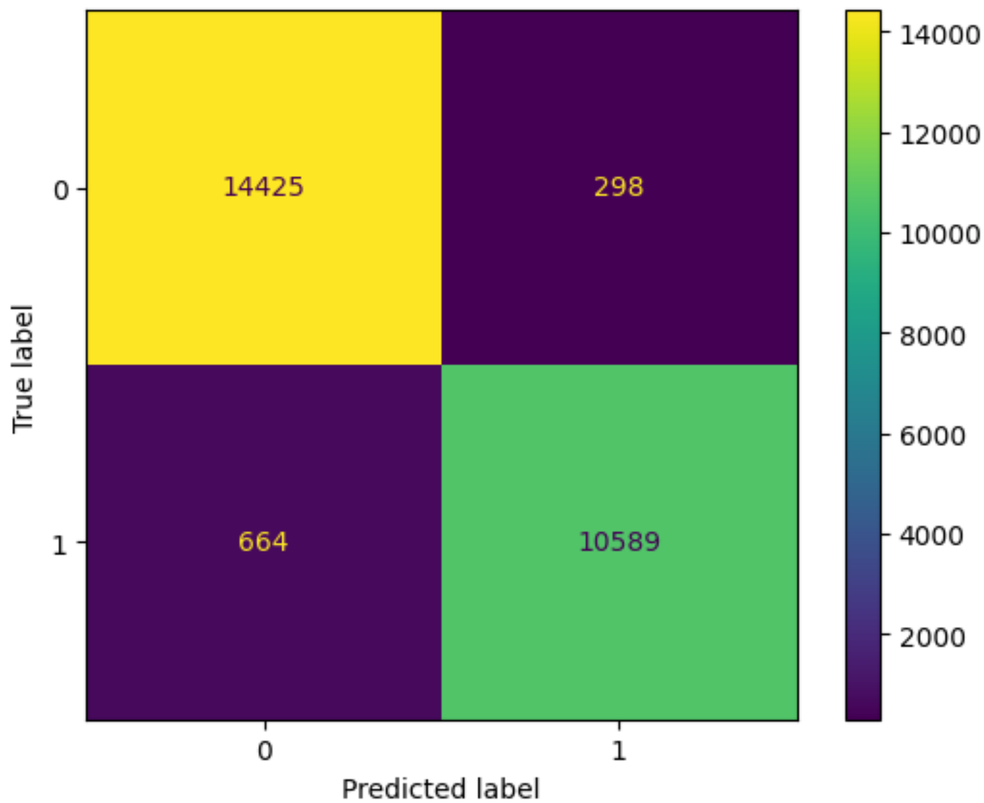


Figure 34: Confusion matrix of random Forest

The confusion matrix provided details of the classifier's performance with respect to actual and predicted classifications. It shows 14,425 true negatives, where class 0 is correctly identified, and 10,589 true positives, where class 1 is accurately predicted.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	14723
1	0.97	0.94	0.96	11253
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

The model boasts an accuracy of 0.96 and for class 0, the model achieves a precision of 0.96, a recall of 0.98, and an F1-score of 0.97, indicating a strong ability to correctly identify and classify true negatives, with a total support of 14,723 instances.

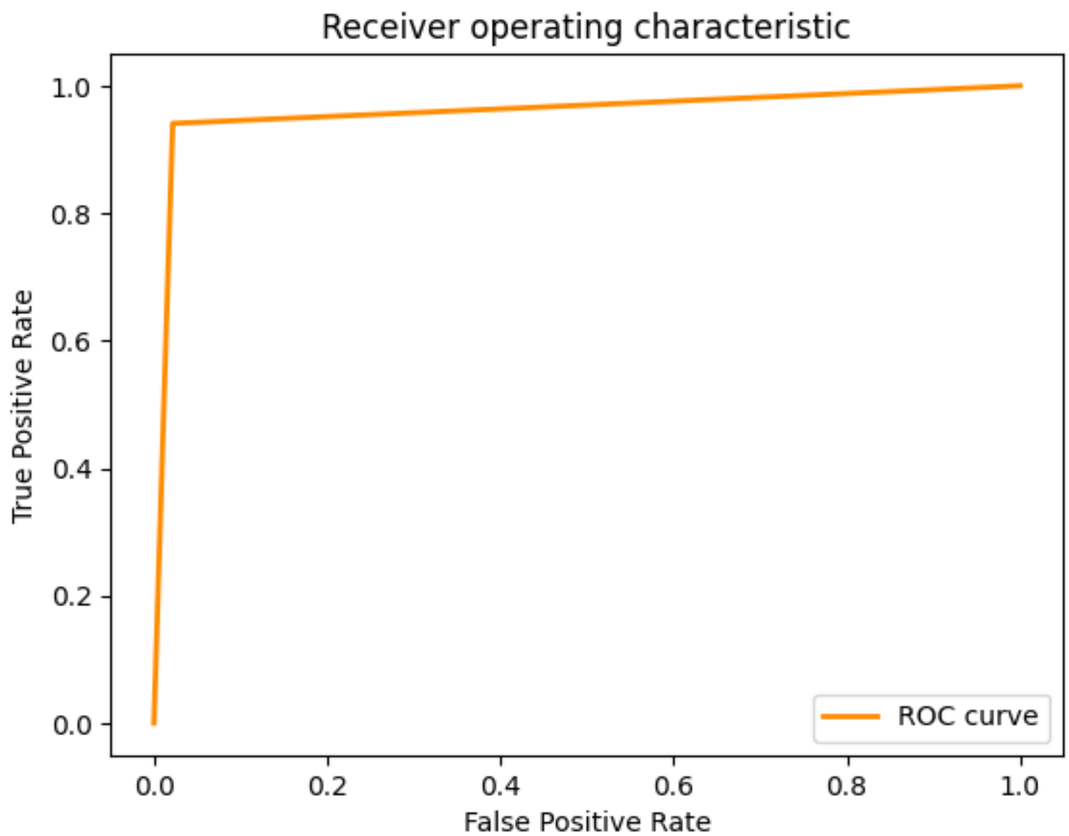


Figure 35: ROC of random Forest

The curve rises sharply to the upper left corner, reaching close to the maximum true positive rate (TPR) of 1.0 while maintaining a low false positive rate (FPR), nearly 0, until approximately 0.2 on the FPR scale.

8.8 Bagging

Bagging is an ensemble learning technique in which it uses models for each subset of training dataset, particularly decision tree methods. The bagging ensemble, composed of 20 decision trees and depth of tree is 3.

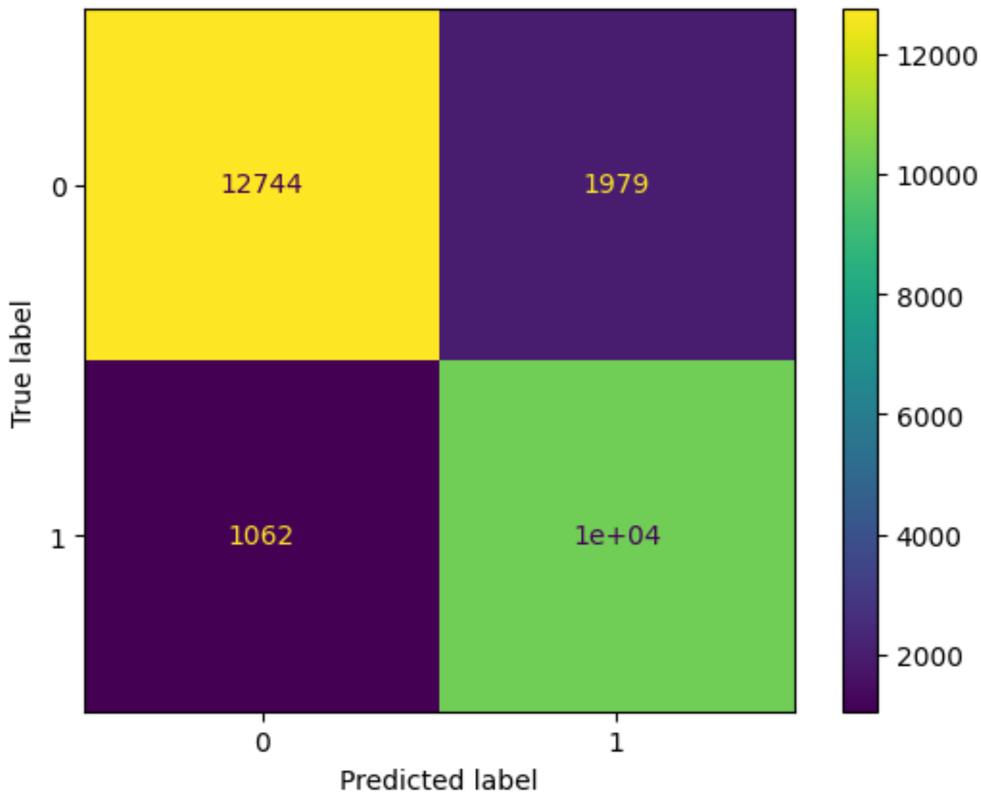


Figure 36: Confusion matrix of Bagging

For class 0, the model correctly predicted 12,744 instances as true negatives but erroneously labeled 1,979 instances as class 1 (false positives). Conversely, for class 1, the model successfully identified 10,000 instances as true positives.

	precision	recall	f1-score	support
0	0.92	0.87	0.89	14723
1	0.84	0.91	0.87	11253
accuracy			0.88	25976
macro avg	0.88	0.89	0.88	25976
weighted avg	0.89	0.88	0.88	25976

For class 0, the precision is 0.92, which indicates that 92% of the predictions made as class 0 are correct, and the recall is 0.87, showing that 87% of the actual class 0 instances were correctly identified by the model, resulting in an F1-score of 0.89. Class 1 shows a precision of 0.84 and a higher recall of 0.91, reflecting that 91% of actual class 1 instances were correctly predicted, with an F1-score of 0.8

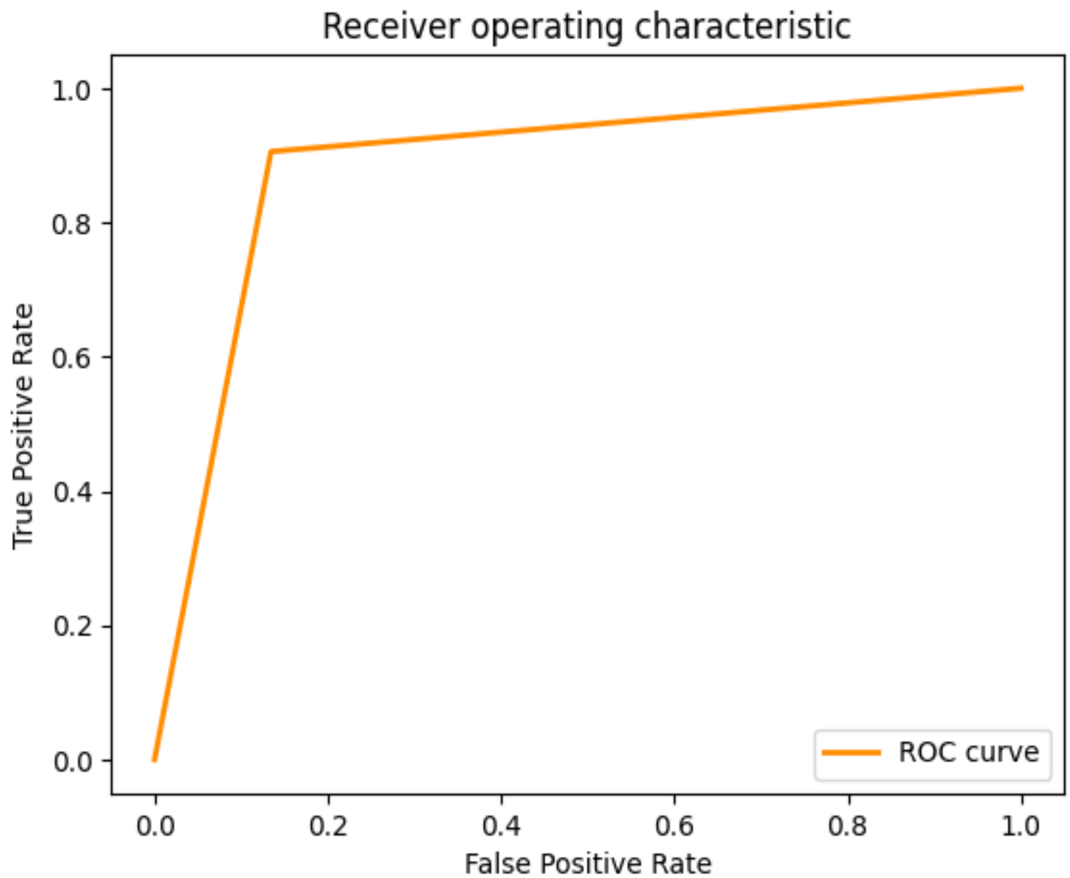


Figure 37: ROC curve of Bagging

For class 0, the precision is 0.92, which indicates that 92% of the predictions made as class 0 are correct, and the recall is 0.87, showing that 87% of the actual class 0 instances were correctly identified by the model, resulting in an F1-score of 0.89. Class 1 shows a precision of 0.84 and a higher recall of 0.91, reflecting that 91% of actual class 1 instances were correctly predicted, with an F1-score of 0.87.

8.9 Boosting

The bagging ensemble, composed of 200 decision trees and depth of tree is 3. It works by sequentially training a series of weak learners, typically decision trees, where each subsequent model attempts to improve the errors of its predecessors. Improving score on every turn.

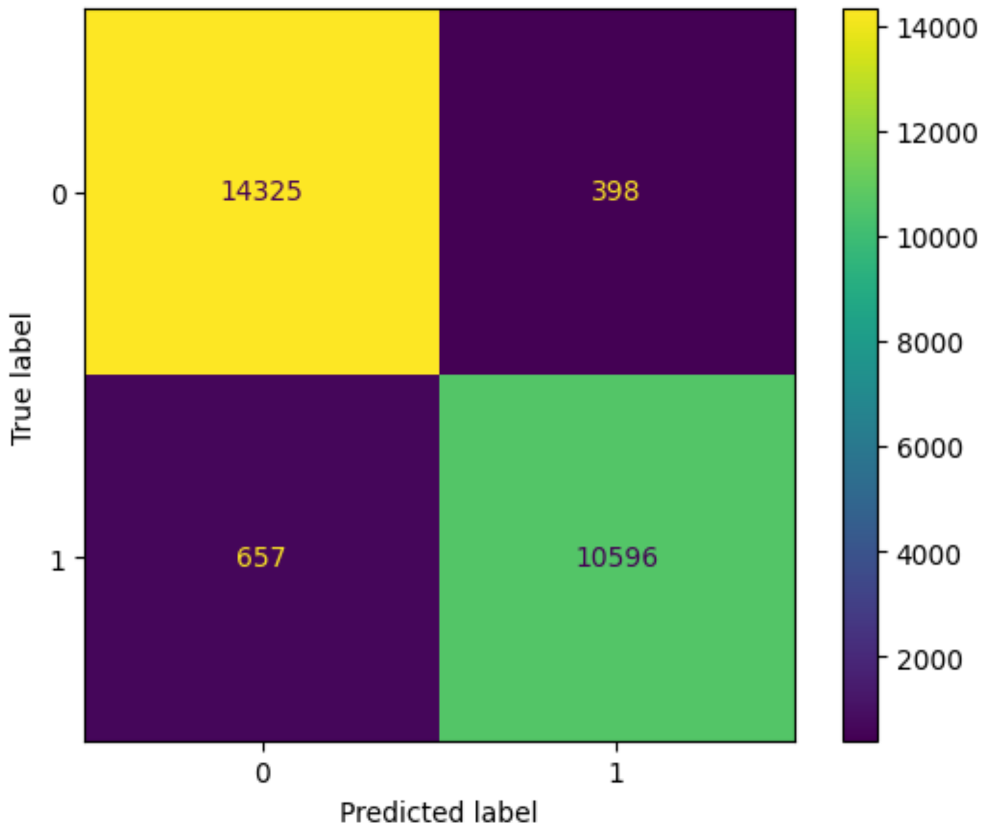


Figure 38: Confusion matrix of Boosting

The model successfully identified 14,325 instances as class 0 (true negatives) and 10,596 instances as class 1 (true positives), indicating a strong capability in correctly predicting both classes. There are relatively few errors, with 398 false positives (class 0 incorrectly predicted as class 1) and 657 false negatives (class 1 incorrectly predicted as class 0).

	precision	recall	f1-score	support
0	0.96	0.97	0.96	14723
1	0.96	0.94	0.95	11253
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

The precision, recall, and F1-score metrics for class 0 are all very high, standing at 0.96, 0.97, and 0.96 respectively, indicating that the model is extremely effective at identifying and classifying instances of class 0 correctly. For class 1, the precision is equally high at 0.96, with a slightly lower recall of 0.94, resulting in an F1-score of 0.95.

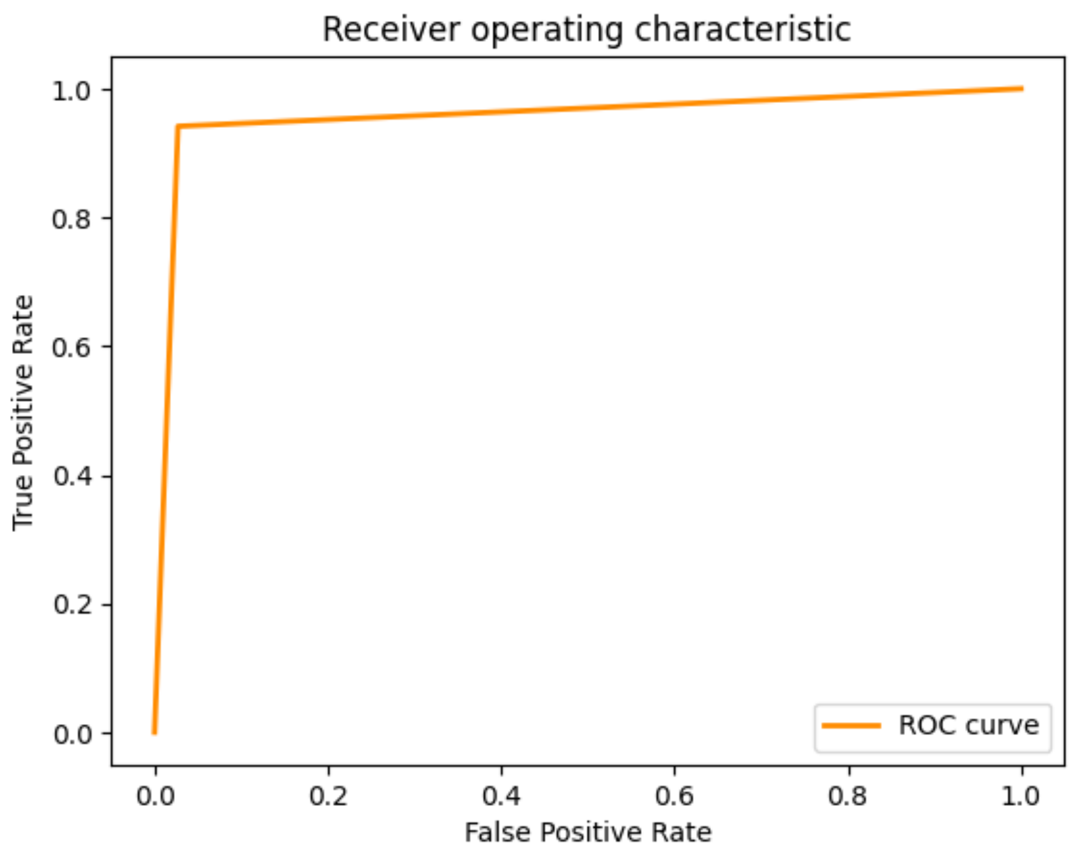


Figure 39: ROC curve of Boosting

The curve rises steeply towards the top-left corner, achieving a near-maximum True Positive Rate (TPR) while maintaining a very low False Positive Rate (FPR) across most of the threshold range. This shape suggests that the classifier has a strong discriminative ability, capable of effectively distinguishing between the positive and negative classes with the high accuracy.

8.10 Voting

We implemented a voting classifier in scikit-learn which integrates four distinct ML models-Logistic Regression, Gaussian Naive Bayes, Decision Tree Classifier, and K-Nearest Neighbors—into a unified ensemble classifier. The Voting Classifier we're using employs hard voting, which means it bases its predictions on the mode of the labels predicted by the individual models.

The confusion matrix illustrates the performance of the classification model, with 14,407 true negatives (correctly identified class 0) and 10,616 true positives (correctly identified class 1). There are 316 false positives (class 0 incorrectly identified as class 1) and 637 false negatives (class 1 incorrectly identified as class 0).

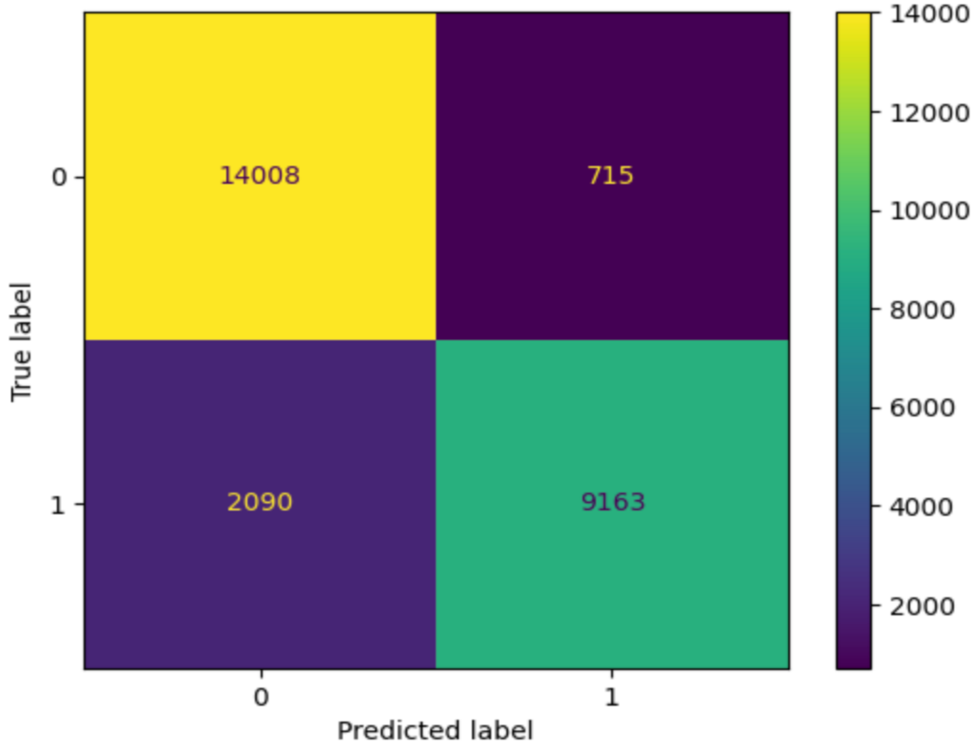


Figure 40: confusion matrix of voting

	precision	recall	f1-score	support
0	0.87	0.95	0.91	14723
1	0.93	0.82	0.87	11253
accuracy			0.89	25976
macro avg	0.90	0.88	0.89	25976
weighted avg	0.90	0.89	0.89	25976

The classifier achieves a precision of 0.87 for class 0 and 0.93 for class 1, indicating a relatively higher accuracy in correctly predicting class 1 instances as positive. The recall scores are 0.95 for class 0 and 0.82 for class 1, showing that the model is better at capturing the majority of actual class 0 instances. The F1-scores, which balance precision and recall, stand at 0.91 for class 0 and 0.87 for class 1, suggesting a slightly more balanced performance for class 0. Overall, the model attains an accuracy of 0.89.

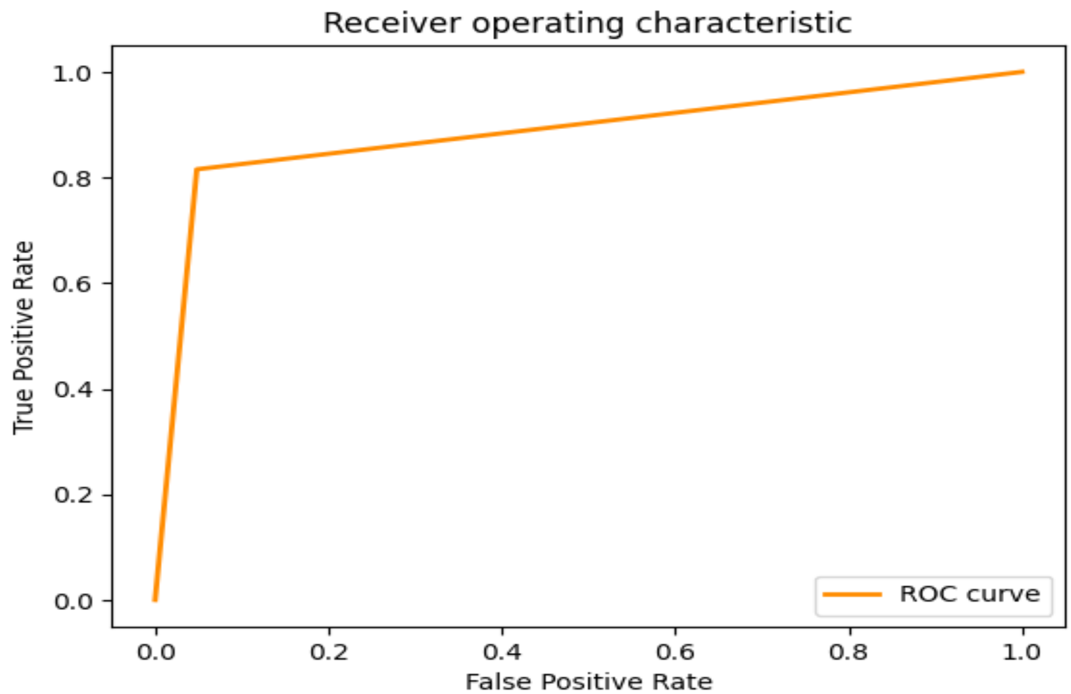


Figure 41: ROC curve of Voting classifier

8.11 Gradient Boosting using Gridsearch

We used Gradient Boosting Classifier using GridSearch to identify the best hyperparameters like estimators, learning rate and depth for improving model accuracy. A range of values for n_estimators, learning_rate, max_depth was tested. The best parameters identified were n_estimators at 250, learning_rate at 0.1, and max_depth at 4.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	14723
1	0.97	0.94	0.96	11253
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

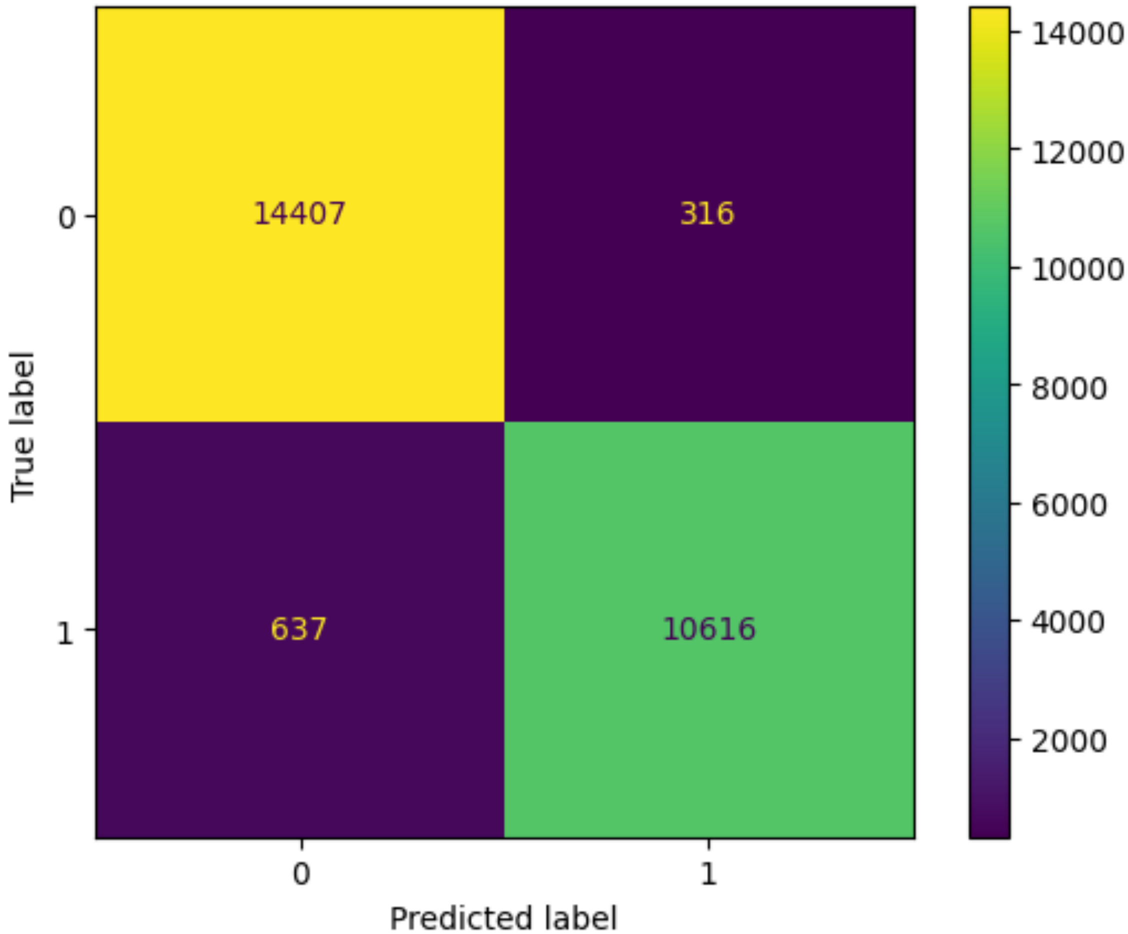


Figure 42: Confusion matrix of Gradient Boosting

The confusion matrix illustrates the performance of the classification model, with 14,407 true negatives (correctly identified class 0) and 10,616 true positives (correctly identified class 1). There are 316 false positives (class 0 incorrectly identified as class 1) and 637 false negatives (class 1 incorrectly identified as class 0).

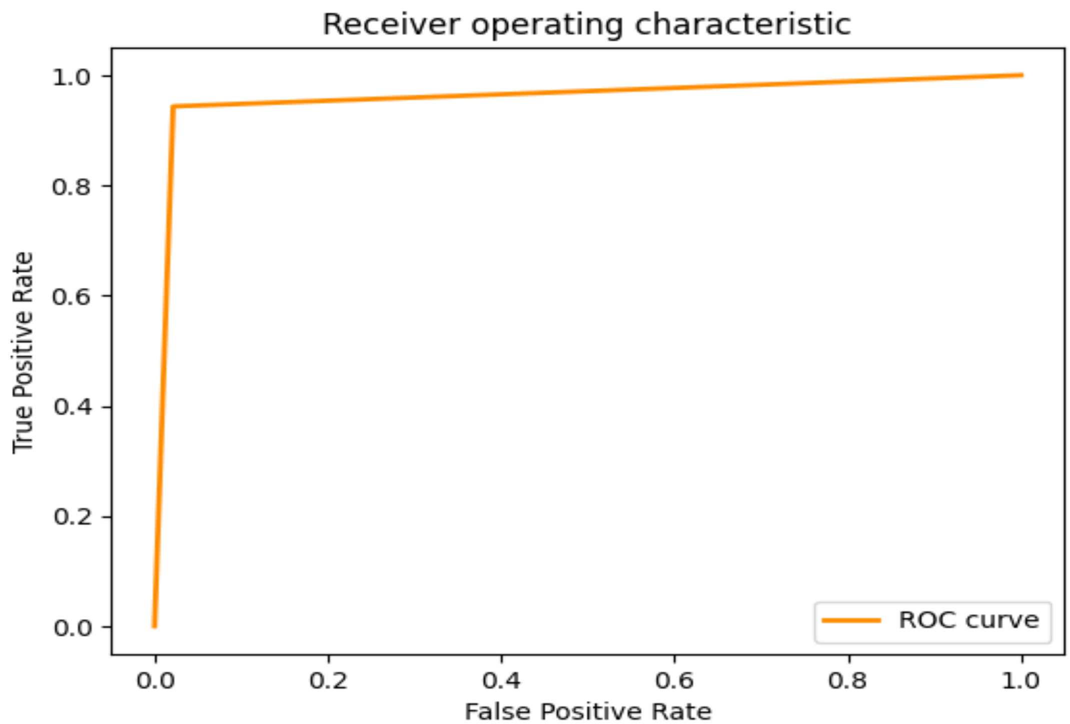


Figure 43: ROC curve of Gradient Boosting

The curve quickly approaches the top-left corner of the curve, showing that the model effectively distinguishes between the two classes with minimal errors. This sharp ascent suggests that the model achieves a high sensitivity (true positive rate) while maintaining a low false positive rate, reinforcing its robustness and reliability in making accurate prediction.

8.12 XGBOOST

We used XGBClassifier in XGBOOST model for purpose of classification. Basically, this model built on the regularized gradient boosting framework, it sequentially constructs an ensemble of weak learners, typically decision trees, to iteratively correct errors and enhance predictive performance.

For class 0, the precision is 0.96, recall is 0.98, and F1-score is 0.97. For class 1, the precision is 0.97, recall is 0.95, and F1-score is 0.96. These metrics indicate that the model performs well in distinguishing both classes, with high precision, recall, and F1-scores.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	14723
1	0.97	0.95	0.96	11253
accuracy			0.96	25976
macro avg	0.97	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

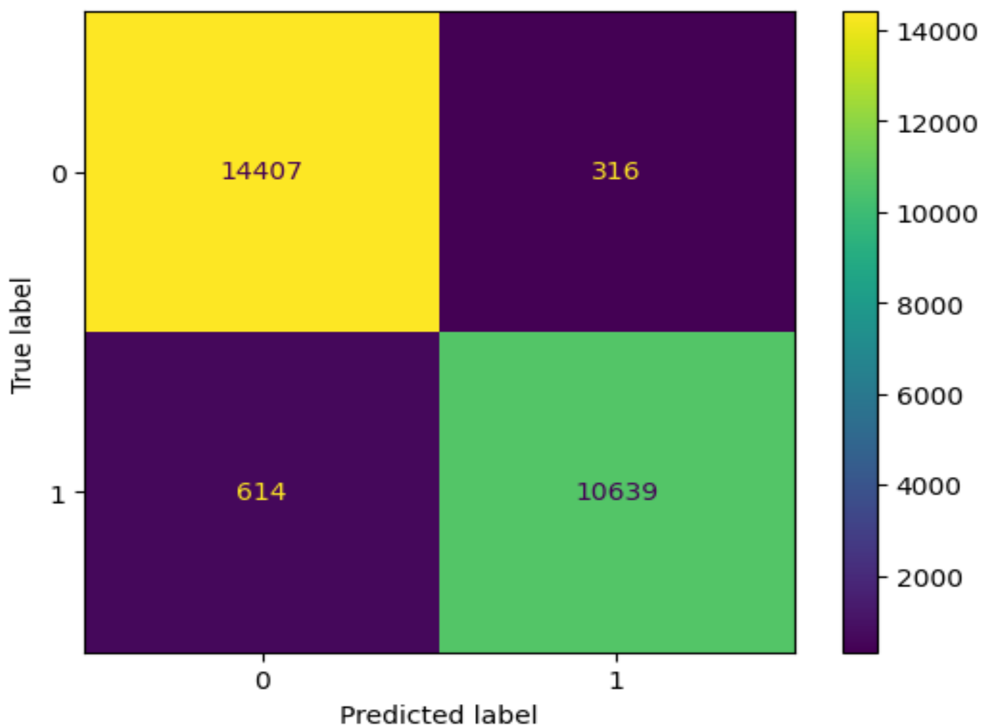


Figure 44: Confusion matrix of XGBoost

14,325 true negatives (correctly predicted class 0) and 10,596 true positives (correctly predicted class 1). However, there are areas for improvement, specifically in reducing false negatives (657 instances incorrectly labeled as class 0) and false positives (398 instances incorrectly labeled as class 1).

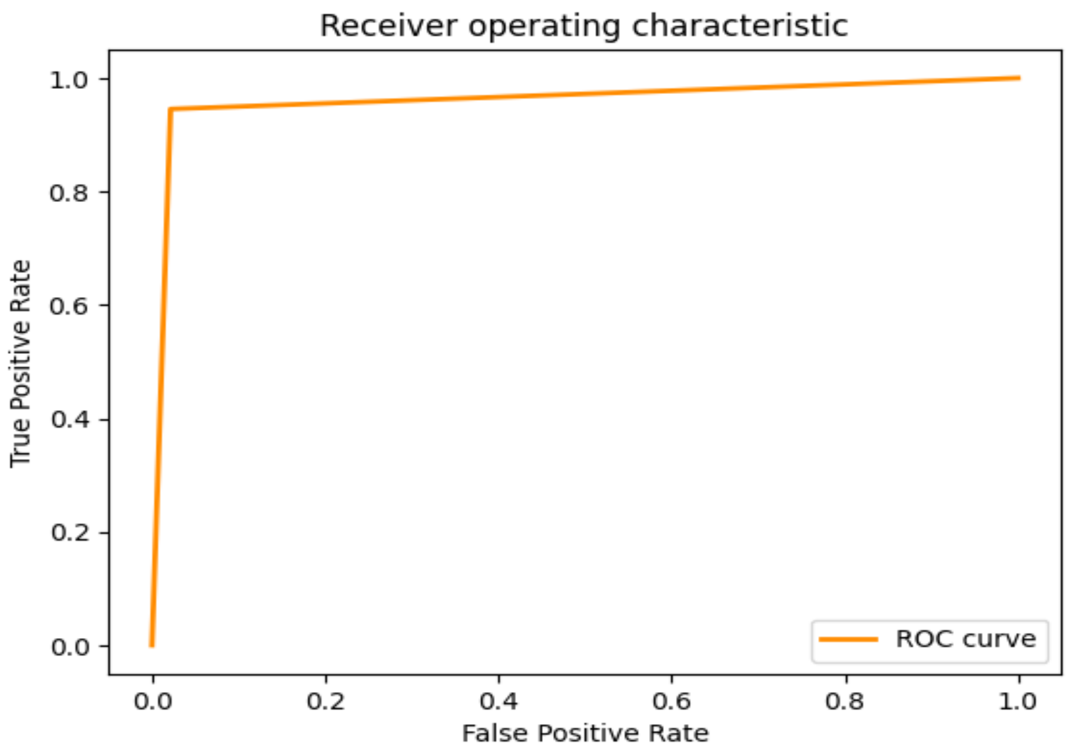


Figure 45: ROC curve of Gradient Boosting

The curve rises sharply towards the upper left corner, indicating a very high True Positive Rate (TPR) close to 1.0, while maintaining a low False Positive Rate (FPR), which approaches zero. This shape of the ROC curve is characteristic of models with high discriminative power, capable of effectively separating the positive class from the negative with minimal error.

9. SHAP Analysis

SHAP is a popular AI package using which we can infer which features contributing our model decisions and how they contributing. It also shows local and general prediction path.

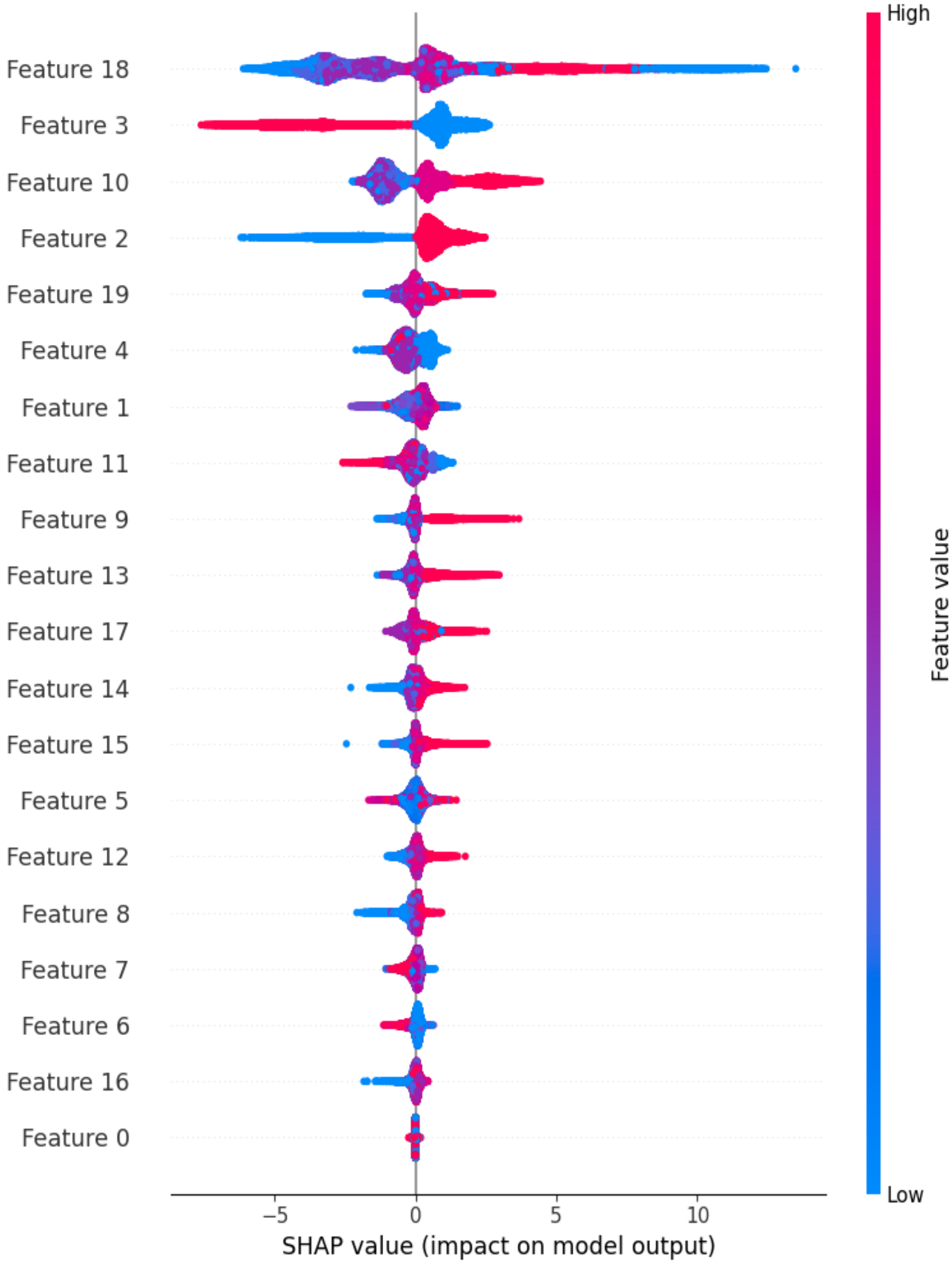


Figure 46: ROC curve of Gradient Boosting

Feature 18 (In-flight Wifi Service), Feature 3 (Type of Travel), and Feature 10 (Online Boarding) appear to be highly impactful, showing a large dispersion of SHAP values and differing by high and low feature values. Features like Feature 0 (Gender) and Feature 16 (Food and Drink) are lower on the plot, indicating they are less impactful on the model's prediction.

10. Comparing models

Model	Accuracy (%)	Execution Time (seconds)
Logistic Regression	87	5.130
Decision Tree	95	0.331
Naive Bayes	87	29.000
KNN	77	48.000
SVC	87	240.000
MLP	95	192.000
Random Forest	96	8.270
Bagging	96	2.030
Boosting	88	480.000
Voting	89	44.000
Gradient Boosting	96	58.000
XGBoost	96	0.273

Table 3: Accuracy and Execution Time

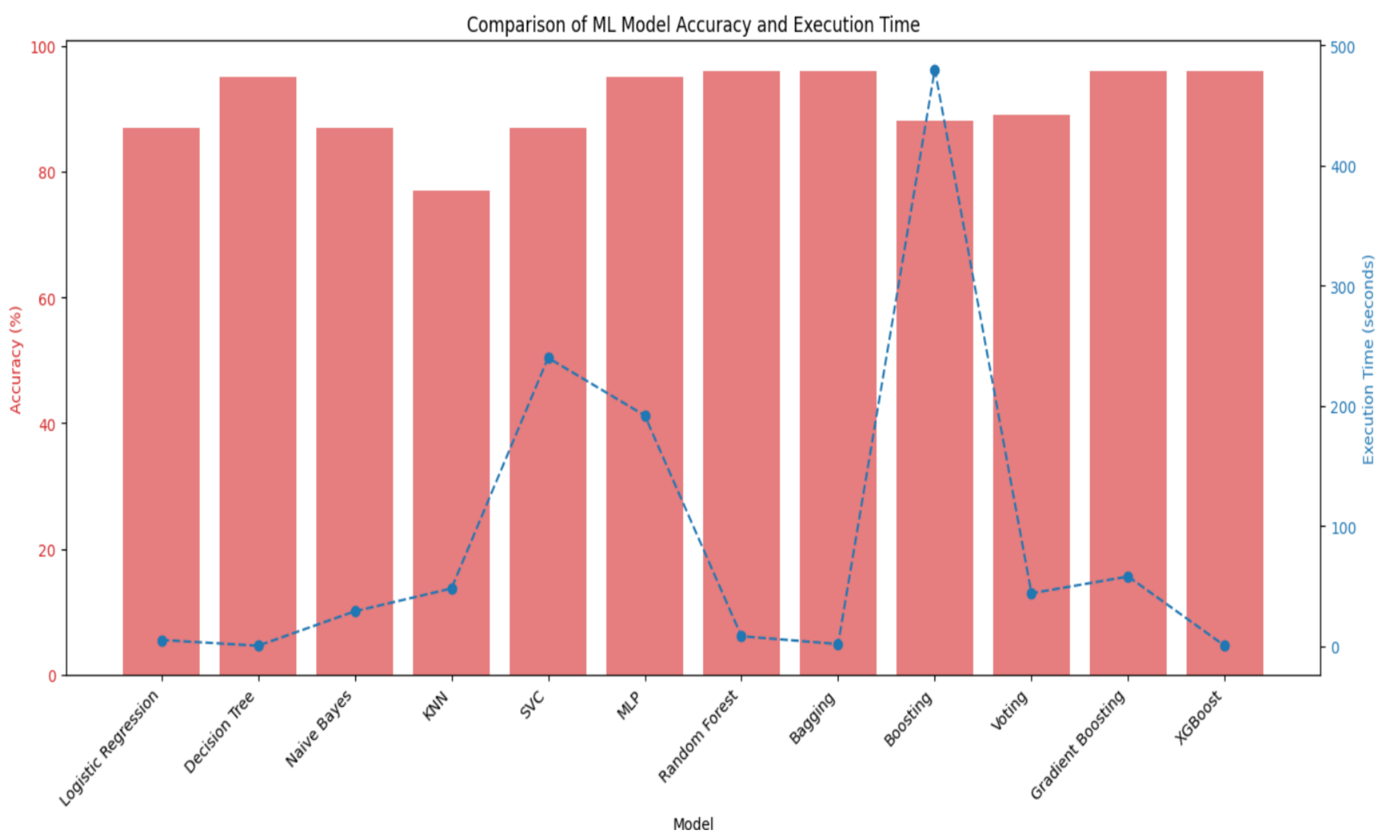


Figure 47: Comparison of ML model accuracy and execution time

This dual-axis chart illustrates a comprehensive comparison of model accuracy and execution time where the bar graph represents model accuracy (%) and the line graph represents execution time (seconds). Boosting and Gradient Boosting are both advanced ensemble techniques that focus on converting weak learners into strong ones, both achieving an accuracy of 96%. However, Boosting is particularly slow at 8 minutes, while Gradient Boosting takes about 58 seconds. XGBoost is incredibly fast, executing in just 0.273 milliseconds, and it matches the high accuracy of 96%. It's known for its performance and speed.

11. Conclusion and Discussion

The analysis reveals that ensemble methods outperform traditional classification models in accuracy, with decision trees and MLP also showing robust results. Enhancements in performance were achieved by removing features that are either highly correlated with others or have low correlation with the target variable. For the airline industry, the data suggests discontinuing gender-based promotions, as gender does not significantly influence customer satisfaction. Instead, attention should be given to age-related preferences, especially since younger customers tend to express more dissatisfaction. For first-time users, who often face challenges with system complexities, simplifying the booking and check-in processes or providing clearer guidance could enhance their experience. Furthermore, factors such as baggage handling, seat comfort, legroom, cleanliness, and food quality play pivotal roles in retaining customers and should be prioritized to improve satisfaction and loyalty. These recommendations are aimed at refining customer service strategies to boost overall satisfaction in the airline industry.

12. Bibliography & References

ACC	Accuracy
AdaBoost	Adaptive Boosting
ANN	Artificial neural network
AUC	Area under curve
CART	Classification and regression tree
DT	Decision tree
EDA	Exploratory data analysis
FNR	False negative rate
GD	Gradient descent
GB	Gradient Boosting
KNN	k-Nearest Neighbor
MLP	Multilayer perceptron
NB	Naïve bayes
NN	Neural network
PCA	Principal component analysis
RELU	Rectified linear unit
RF	Random Forest
ROC	Received Operating Characteristic
SVC	Support vector classifier
VIF	Variance inflation factor
XGB	eXtreme Gradient Boosting

- Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido
- https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models
- https://machinelearningmastery.com/xgboost-python-mini-course/
- https://machinelearningmastery.com/super-learner-ensemble-in-python/
- https://medium.com/@nhan.tran/the-chi-square-statistic-p-1-37a8eb2f27bb
- Shamsudin, M. F., Johari, N., Nayan, S., Esa, S. A., & Kadir, B. (2020). The influence of service quality, brand image, trust, and price on customer satisfaction: Case of airline services.