AIRLINE PASSENGER SATISFACTION

DATA MINING - TERM PROJECT

Işıl Deniz Öztürk

Hilal Ayışığı

Figure List

1.Introduction

The airline industry assembles an important part of the wider travel industry, by providing customers with the ability to purchase seats on flights and travel to different parts of the world. Industry has been growing considerably, since additional airports have opened, the number of flights has increased, tickets are more affordable, and continues to grow. There are a lot of airline companies that customers can choose from and because of that, the airline companies face challenges as it's hard to differentiate. The airline industry is growing significantly and there are now more than 5,000 airlines operating worldwide. As the growth continues, as in every sector, competition has occurred between rival airline companies and it is increasing as a major element in this competitive battle with each passing time. Given that the global airline industry is considered one of the most competitive markets, numerous attempts have been made to explore strategies for success in this industry. With customers being the main source of income and they bring lots of revenue, this makes them the most important factor for success. In order to be successful in the industry, companies need to understand customers' exceptions to deliver unique experiences and should consider high-grade satisfaction as a key factor so that they can retain that customer.

Satisfaction can change from person to person. But generally, if the product has at least met the needs of the consumer then it is said to be customer satisfaction. Therefore, companies need to understand customers' expectations, and need to deliver unique experiences in order to retain that customer. As the time passes, people are getting more used to flying and with that they are becoming more advanced about flying therefore have higher expectations from airline companies. This forces customer service quality to emerge as a fundamental factor in the design of a competitive strategy. Thus, the formation of a strategy, is based on the opinions of customers for this reason industry started collecting customers' opinions, by asking what they receive from the service which is often measured by their satisfaction ratings.

We attempt to understand the reasons for customer experience being satisfied or not by using a dataset from kaggle. As part of the analysis, we will be able to understand several factors which improve customer satisfaction level by using the obtained Airplane Passenger Satisfaction dataset. Dataset that we used is focused on customer ratings that they gave on various attributes such as cleanliness and they also made determination on whether or not they are satisfied overall which gives us the satisfaction of the customer.

2.Problem

All the airline companies would want to identify a customer satisfaction level because delivering the high service quality, meeting the needs and expectations of the customer, is essential for airlines’ survival and competitiveness. Based on understanding the reasons for customer experience being satisfied or not, improvements will be made to provide better service by the airline company which will cause the company to be successful.

High-grade customer satisfaction is the most important asset for air businesses. If the customer is not satisfied with the quality of service, they will probably don't want to use that company for further flights and will probably switch to another one which is a result that wouldn't be wanted by any company.

Nowadays, people can easily access all kinds of information they are looking for over the internet and easily share their experiences, especially through social media. Therefore, any miserable experience of a customer, which is shared, can become viral on the internet and reach many people, which may cause companies to lose many customers, seriously affecting the brand or goodwill of the company. So, the most important factor in the success of companies is customer satisfaction. The companies in the industry are identifying a customer’s satisfaction through a rating card.

The research is about classifying if the customer is satisfied or not by finding out the factors related to high satisfaction of customers with airline services and developing a better understanding of the main quality factors that affect customer satisfaction, through examination of customers’ feedback and ratings.

3.Dataset

We obtained our dataset which is Airline Passenger Satisfaction from Kaggle. By using this dataset we will try to understand the factors to satisfy the customer. The dataset contains a total of 129.880 observations and 25 attributes.

Dataset Source: https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction.

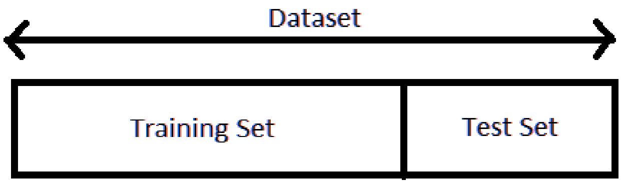


Figure 1:Dataset splitting into train-test set

The dataset we used in this research consists of an 80% training set and a 20% test set. In the training dataset, there are 103.904 rows and 25 columns and in the test dataset, there are 25.976 rows and 25 columns.

3.1 Content

Tablo 1: Content/attributes of the dataset.

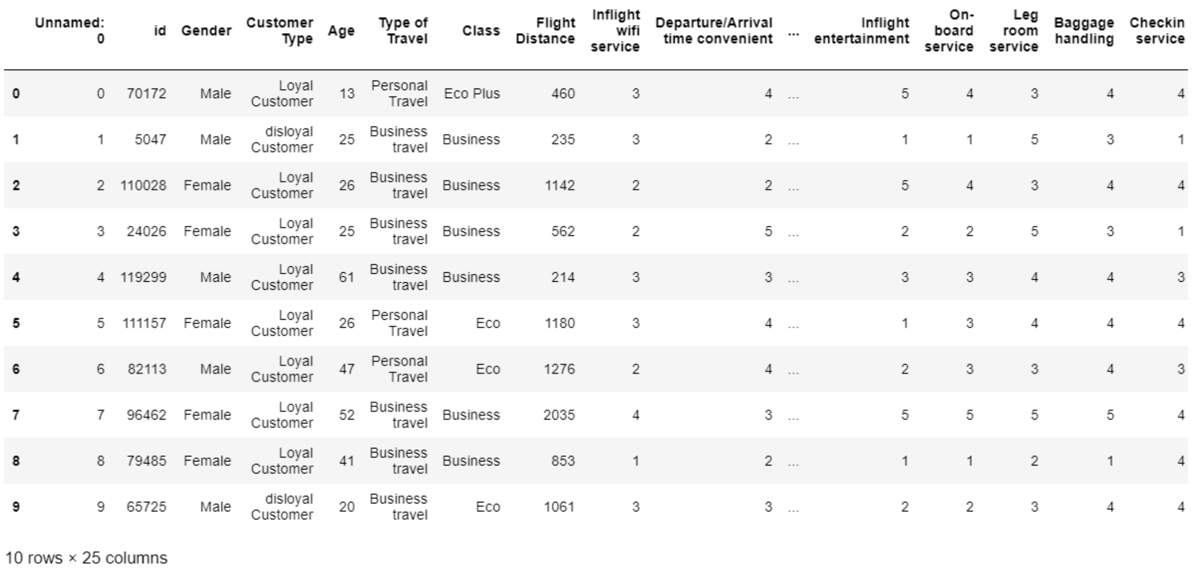
Variable	Variable Description	Variable Value Level	Variable	Variable Description	Variable Value Level
Gender	Gender of the passengers	Female, Male	**Satisfaction**	Airline satisfaction level	Satisfaction, neutral, or dissatisfaction
Age	The actual age of the passengers	–	**Type of Travel**	Purpose of the flight of the passengers	Personal Travel, Business Travel
Customer Type	The customer type	Loyal customer, disloyal customer	**Class**	Travel class in the plane of the passengers	Business, Eco, Eco Plus

Flight distance	The flight distance of this journey	–	**Inflight wifi service**	Satisfaction level of the inflight wifi service	Rating: 0 (least) - 5 (highest)
Cleanliness	Satisfaction level of Cleanliness	Rating: 0 (least) - 5 (highest)	**Ease of Online booking**	Satisfaction level of online booking	Rating: 0 (least) - 5 (highest)
Gate location	Satisfaction level of Gate location	Rating: 0 (least) - 5 (highest)	**Food and drink**	Satisfaction level of Food and drink	Rating: 0 (least) - 5 (highest)
Online boarding	Satisfaction level of online boarding	Rating: 0 (least) - 5 (highest)	**Seat comfort**	Satisfaction level of Seat comfort	Rating: 0 (least) - 5 (highest)
Inflight entertainment	Satisfaction level of inflight entertainment	Rating: 0 (least) - 5 (highest)	**On-board service**	Satisfaction level of On-board service	Rating: 0 (least) - 5 (highest)
Legroom service	Satisfaction level of Leg room service	Rating: 0 (least) - 5 (highest)	**Baggage handling**	Satisfaction level of baggage handling	Rating: 0 (least) - 5 (highest)
Check-in service	Satisfaction level of Check-in service	Rating: 0 (least) - 5 (highest)	**Inflight service**	Satisfaction level of inflight service	Rating: 0 (least) - 5 (highest)
Departure/ Arrival time convenient	Satisfaction level of Departure/ Arrival time convenient	Rating: 0 (least) - 5 (highest)	**Departure Delay in Minutes**	Minutes delayed when departure	——

Arrival Delay in Minutes	Minutes delayed when Arrival	—			

This table shows our dataset attributes and description about them with their value level. Out of all columns, 14 are survey entries where passengers rate the flight experience on a scale of 1 to 5.

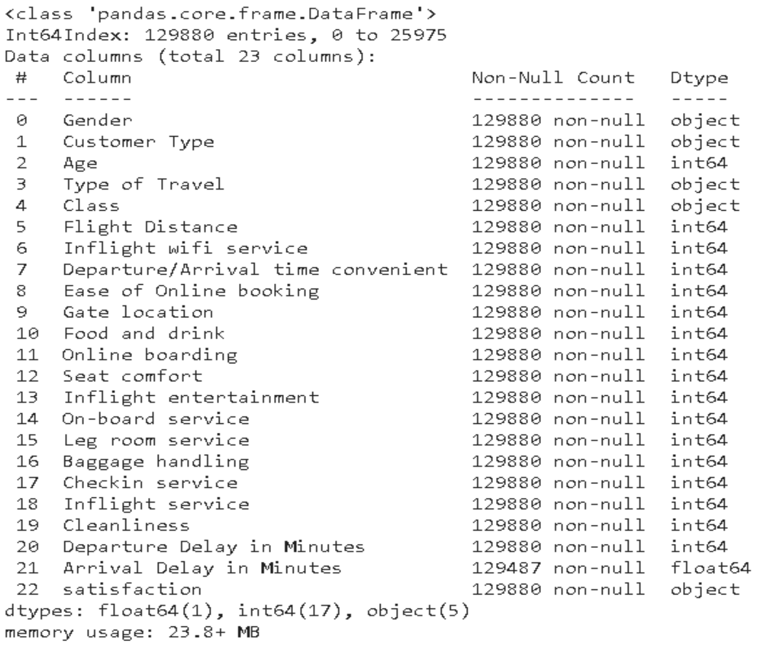
3.2 Data Preparation



	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	4
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	1
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4	3	4	4
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2	5	3	1
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...	3	3	4	4	3
5	5	111157	Female	Loyal Customer	26	Personal Travel	Eco	1180	3	4	...	1	3	4	4	4
6	6	82113	Male	Loyal Customer	47	Personal Travel	Eco	1276	2	4	...	2	3	3	4	3
7	7	96462	Female	Loyal Customer	52	Business travel	Business	2035	4	3	...	5	5	5	5	4
8	8	79485	Female	Loyal Customer	41	Business travel	Business	853	1	2	...	1	1	2	1	4
9	9	65725	Male	disloyal Customer	20	Business travel	Eco	1061	3	3	...	2	2	3	4	4

10 rows × 25 columns

Figure 2:The first 10 data from the training set

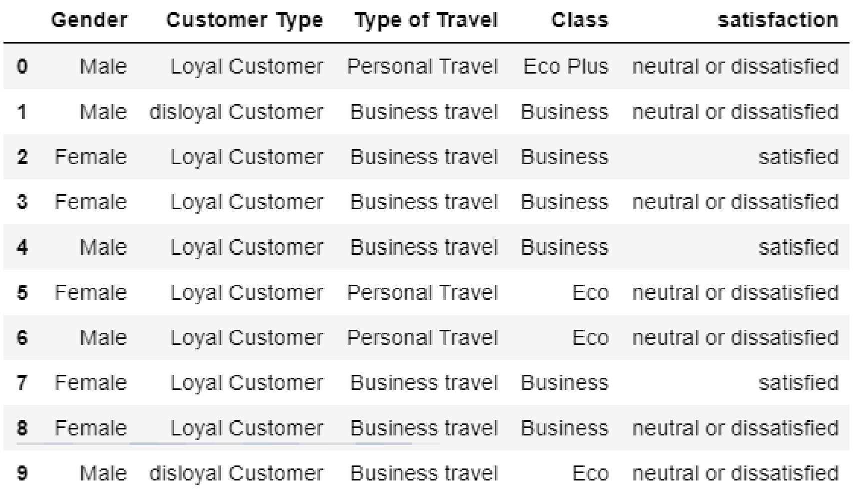


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129880 entries, 0 to 25975
Data columns (total 23 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Gender                             129880 non-null  object
 1   Customer Type                      129880 non-null  object
 2   Age                                129880 non-null  int64
 3   Type of Travel                     129880 non-null  object
 4   Class                              129880 non-null  object
 5   Flight Distance                    129880 non-null  int64
 6   Inflight wifi service              129880 non-null  int64
 7   Departure/Arrival time convenient  129880 non-null  int64
 8   Ease of Online booking             129880 non-null  int64
 9   Gate location                      129880 non-null  int64
 10  Food and drink                     129880 non-null  int64
 11  Online boarding                    129880 non-null  int64
 12  Seat comfort                       129880 non-null  int64
 13  Inflight entertainment             129880 non-null  int64
 14  On-board service                   129880 non-null  int64
 15  Leg room service                   129880 non-null  int64
 16  Baggage handling                   129880 non-null  int64
 17  Checkin service                    129880 non-null  int64
 18  Inflight service                   129880 non-null  int64
 19  Cleanliness                        129880 non-null  int64
 20  Departure Delay in Minutes         129880 non-null  int64
 21  Arrival Delay in Minutes           129487 non-null  float64
 22  satisfaction                       129880 non-null  object
dtypes: float64(1), int64(17), object(5)
memory usage: 23.8+ MB
```

We import the dataset into Python and observe the structure of the data. We observe the dataset having 129880 observations and 25 attributes. The Satisfaction level, which is our dependent variable, is represented as a factor ("Satisfied" and "Neutral or Dissatisfied"). This dataset contains an airline passenger satisfaction survey.

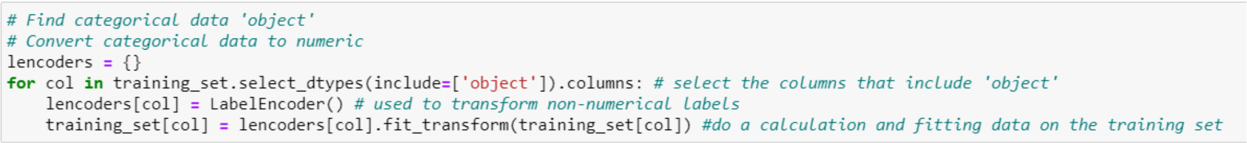
For preparing the data, first, we drop the unnecessary columns which we do not need in our analysis and model. These columns are ‘id’ and ‘Unnamed: 0’. After getting rid of these columns, we have 23 columns left for us to use in our study.

Some data mining algorithms cannot handle categorical variables, they require only ratio values which are in the category of numerical value so we checked the dataset for ‘Categorical values’ so that we can convert them into 'Numerical values'.



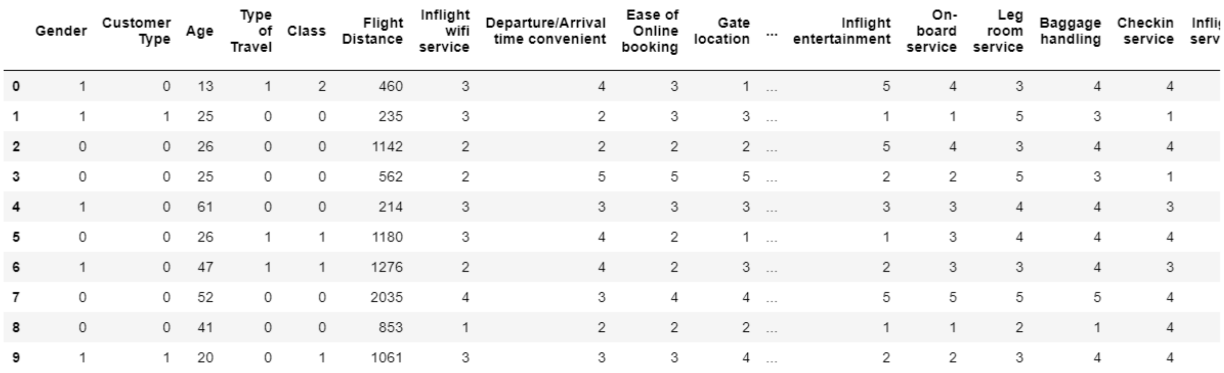
	Gender	Customer Type	Type of Travel	Class	satisfaction
0	Male	Loyal Customer	Personal Travel	Eco Plus	neutral or dissatisfied
1	Male	disloyal Customer	Business travel	Business	neutral or dissatisfied
2	Female	Loyal Customer	Business travel	Business	satisfied
3	Female	Loyal Customer	Business travel	Business	neutral or dissatisfied
4	Male	Loyal Customer	Business travel	Business	satisfied
5	Female	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied
6	Male	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied
7	Female	Loyal Customer	Business travel	Business	satisfied
8	Female	Loyal Customer	Business travel	Business	neutral or dissatisfied
9	Male	disloyal Customer	Business travel	Eco	neutral or dissatisfied

After finding categorical values, we convert them into numerical values with the help of LabelEncoder(). Which encodes target labels with values between 0 and n_classes-1.



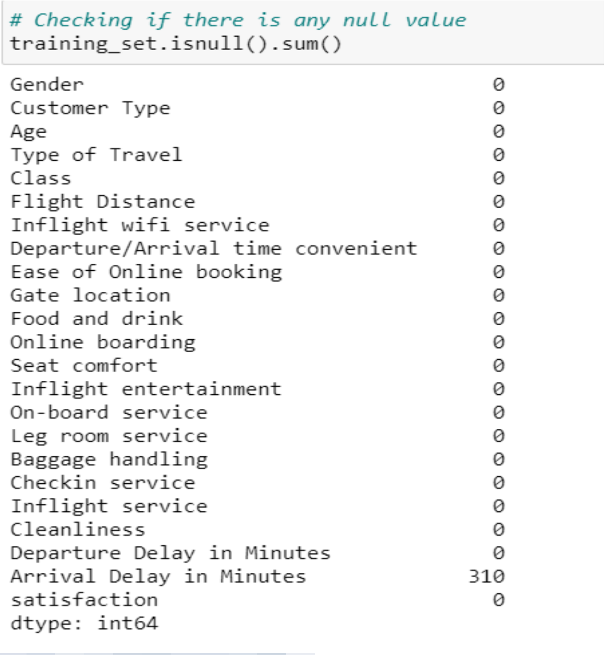
```
# Find categorical data 'object'
# Convert categorical data to numeric
lencoders = {}
for col in training_set.select_dtypes(include=['object']).columns: # select the columns that include 'object'
    lencoders[col] = LabelEncoder() # used to transform non-numerical labels
    training_set[col] = lencoders[col].fit_transform(training_set[col]) #do a calculation and fitting data on the training set
```

Figure 5: Code of LabelEncoder().



	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Infli serv
0	1	0	13	1	2	460	3	4	3	1	...	5	4	3	4	4	
1	1	1	25	0	0	235	3	2	3	3	...	1	1	5	3	1	
2	0	0	26	0	0	1142	2	2	2	2	...	5	4	3	4	4	
3	0	0	25	0	0	562	2	5	5	5	...	2	2	5	3	1	
4	1	0	61	0	0	214	3	3	3	3	...	3	3	4	4	3	
5	0	0	26	1	1	1180	3	4	2	1	...	1	3	4	4	4	
6	1	0	47	1	1	1276	2	4	2	3	...	2	3	3	4	3	
7	0	0	52	0	0	2035	4	3	4	4	...	5	5	5	5	4	
8	0	0	41	0	0	853	1	2	2	2	...	1	1	2	1	4	
9	1	1	20	0	1	1061	3	3	3	4	...	2	2	3	4	4	

Figure 6: Categorical values that convert to numerical values



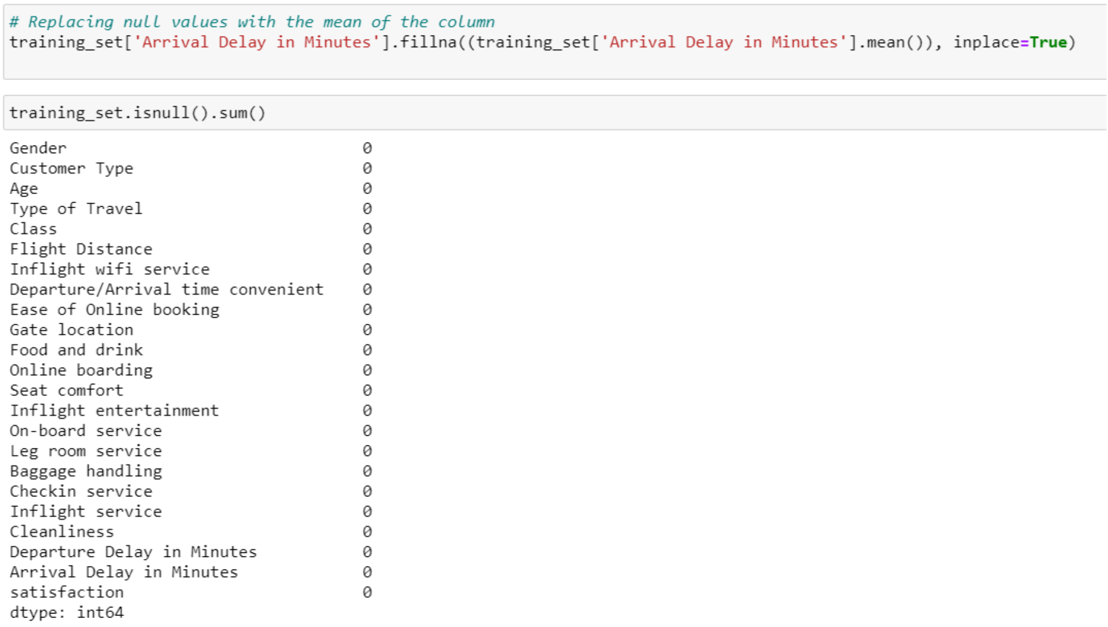
```
# Checking if there is any null value
training_set.isnull().sum()
```

```
Gender                                 0
Customer Type                          0
Age                                    0
Type of Travel                         0
Class                                  0
Flight Distance                        0
Inflight wifi service                  0
Departure/Arrival time convenient      0
Ease of Online booking                 0
Gate location                          0
Food and drink                         0
Online boarding                        0
Seat comfort                           0
Inflight entertainment                 0
On-board service                       0
Leg room service                       0
Baggage handling                       0
Checkin service                        0
Inflight service                       0
Cleanliness                            0
Departure Delay in Minutes             0
Arrival Delay in Minutes             310
satisfaction                           0
dtype: int64
```

The missing values can be a big problem because algorithms need to process the values. Therefore, we check our dataset for any possible NA values, which must be dealt with before we proceed with building the model.

With the IsNull() function we checked if there is any missing value. We observed a total of 310 NA values only in the attribute “Arrival Delay in Minutes”.

We decided to handle the null values by estimating the values and replaced these with the mean of that column.



```
# Replacing null values with the mean of the column
training_set['Arrival Delay in Minutes'].fillna((training_set['Arrival Delay in Minutes'].mean()), inplace=True)
```

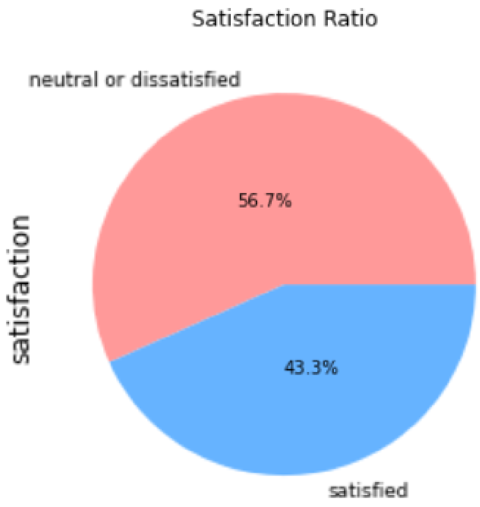
```
training_set.isnull().sum()
```

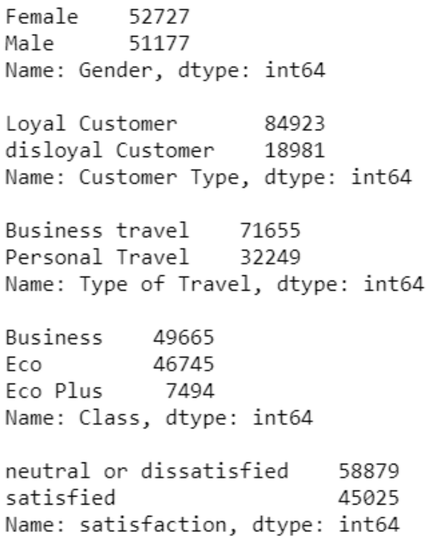
```
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Class                                0
Flight Distance                      0
Inflight wifi service                0
Departure/Arrival time convenient    0
Ease of Online booking               0
Gate location                        0
Food and drink                       0
Online boarding                      0
Seat comfort                         0
Inflight entertainment               0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Inflight service                     0
Cleanliness                          0
Departure Delay in Minutes           0
Arrival Delay in Minutes             0
satisfaction                         0
dtype: int64
```

Figure 8: Filling these NA values with the mean value

3.3 Data Exploratory Analysis

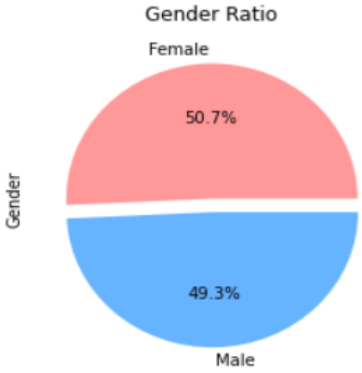
We did data analysis by looking at the distribution of different independent variables. Our key variables are age, gender, customer type, type of travel, class, satisfaction, etc and other various attributes that customers have given their ratings on such as seat comfort, inflight entertainment, etc. which they will also decide on whether or not customers are satisfied overall and at the end it gives us, the dependent variable called satisfaction. To verify and visualize the relationship between the satisfaction variable and other key independent variables, we made statistical research and a series of plots.



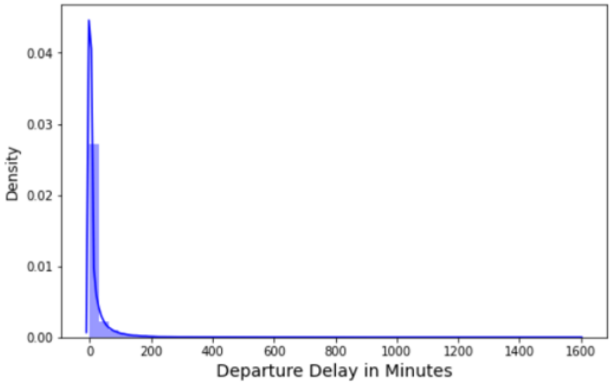


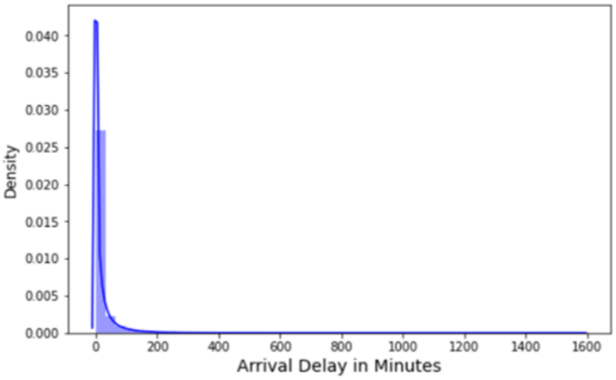
```
Female    52727
Male      51177
Name: Gender, dtype: int64

Loyal Customer       84923
disloyal Customer    18981
Name: Customer Type, dtype: int64

Business travel    71655
Personal Travel    32249
Name: Type of Travel, dtype: int64

Business    49665
Eco         46745
Eco Plus     7494
Name: Class, dtype: int64

neutral or dissatisfied    58879
satisfied                  45025
Name: satisfaction, dtype: int64
```

We can observe that more than half of people are neutral or dissatisfied. From an inferential perspective, these variables are considered the most significant in terms of airline customer satisfaction which is more likely the case as we observe the following visualizations.

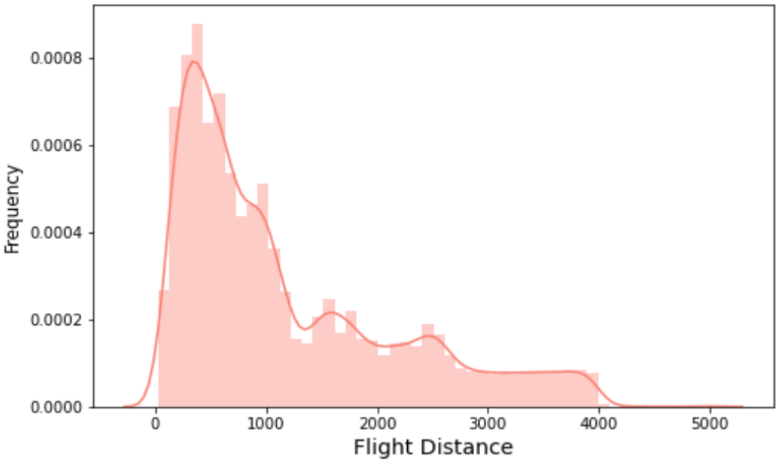


As you can see, the ratio of the total number of female customers is a little bit higher than male customers in this dataset.

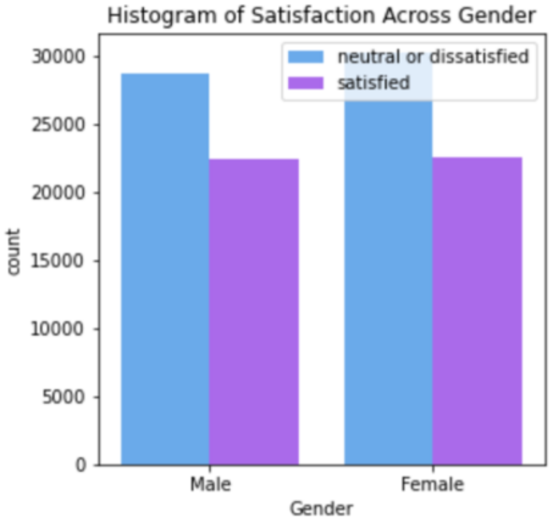


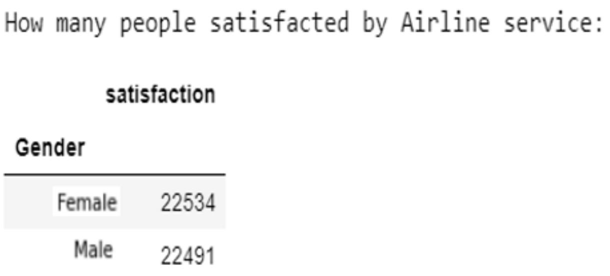


As it can be seen, there weren't extreme delays in both departure and arrival minutes. Mostly both on departure and arrival, delay minutes did not pass 200 minutes.



From this flight distance histogram, people mostly use airlines for short-distance flying.

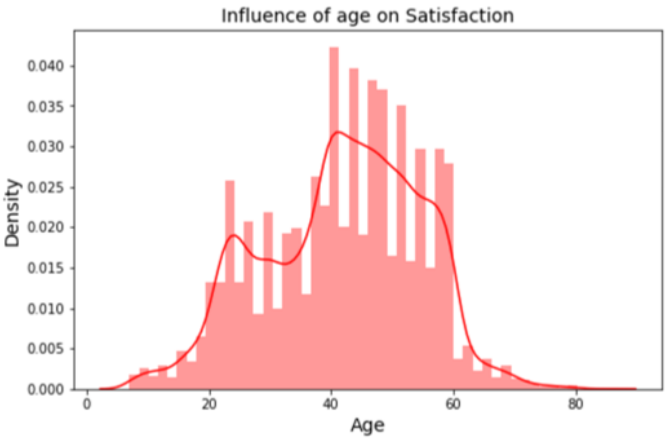


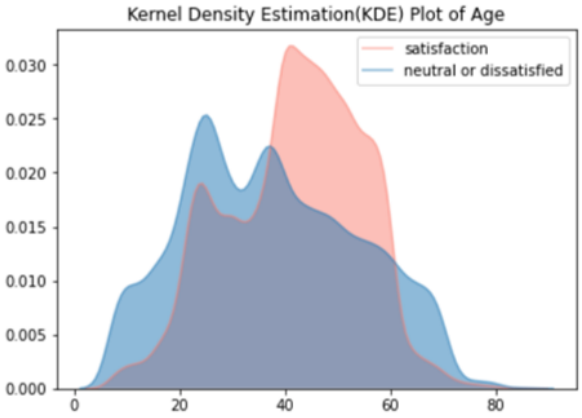

How many people satisfacted by Airline service:

	satisfaction
Gender	
Female	22534
Male	22491

Figure 14: Histogram of Satisfaction Across Gender and Number of Satisfaction Customers

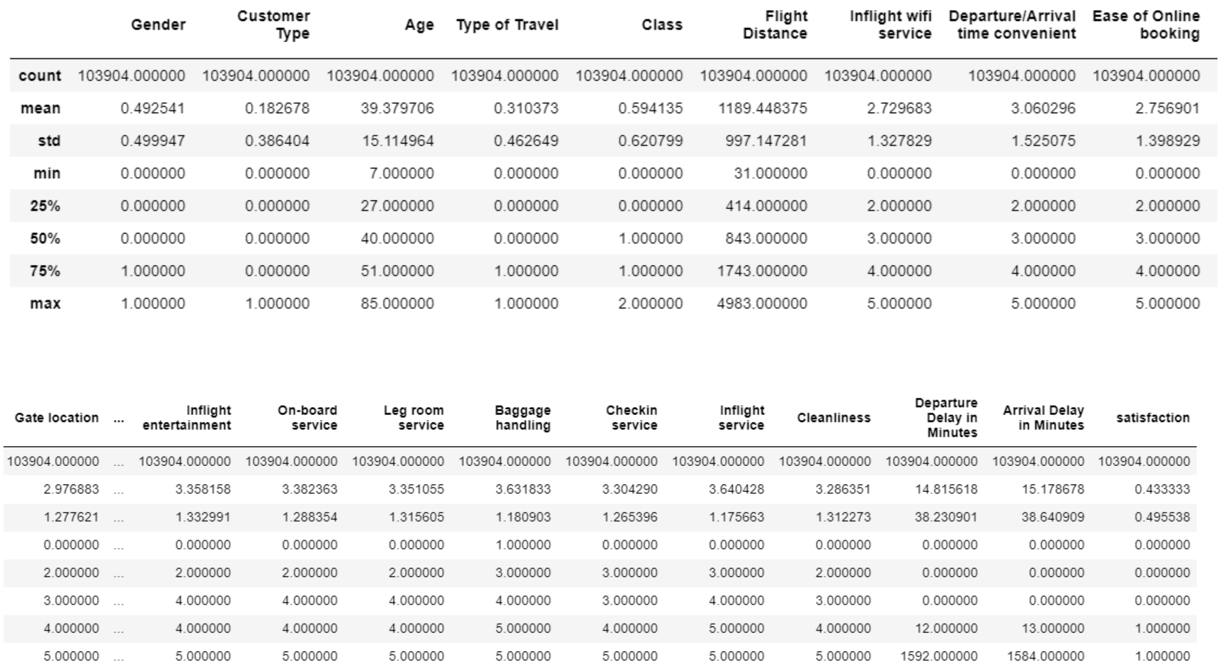
As it might be seen clearly that both female and male customers are mostly unsatisfied than satisfied but comparatively female customers are more unsatisfied than Male customers as in the histogram.





As can be seen, there is a nonlinear relationship between satisfaction and age. Middle-aged customers are more satisfied than young and old customers as in this histogram and KDE.

With the describe() function we calculated some statistical data like percentile, mean, min, max and std of the numerical values of DataFrame.



	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking
count	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000
mean	0.492541	0.182678	39.379706	0.310373	0.594135	1189.448375	2.729683	3.060296	2.756901
std	0.499947	0.386404	15.114964	0.462649	0.620799	997.147281	1.327829	1.525075	1.398929
min	0.000000	0.000000	7.000000	0.000000	0.000000	31.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	27.000000	0.000000	0.000000	414.000000	2.000000	2.000000	2.000000
50%	0.000000	0.000000	40.000000	0.000000	1.000000	843.000000	3.000000	3.000000	3.000000
75%	1.000000	0.000000	51.000000	1.000000	1.000000	1743.000000	4.000000	4.000000	4.000000
max	1.000000	1.000000	85.000000	1.000000	2.000000	4983.000000	5.000000	5.000000	5.000000

Gate location	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes	Arrival Delay in Minutes	satisfaction
103904.000000	...	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000
2.976883	...	3.358158	3.382363	3.351055	3.631833	3.304290	3.640428	3.286351	14.815618	15.178678	0.433333
1.277621	...	1.332991	1.288354	1.315605	1.180903	1.265396	1.175663	1.312273	38.230901	38.640909	0.495538
0.000000	...	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2.000000	...	2.000000	2.000000	2.000000	3.000000	3.000000	3.000000	2.000000	0.000000	0.000000	0.000000
3.000000	...	4.000000	4.000000	4.000000	4.000000	3.000000	4.000000	3.000000	0.000000	0.000000	0.000000
4.000000	...	4.000000	4.000000	4.000000	5.000000	4.000000	5.000000	4.000000	12.000000	13.000000	1.000000
5.000000	...	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	1592.000000	1584.000000	1.000000

Figure 16: Outputs of the describe() function

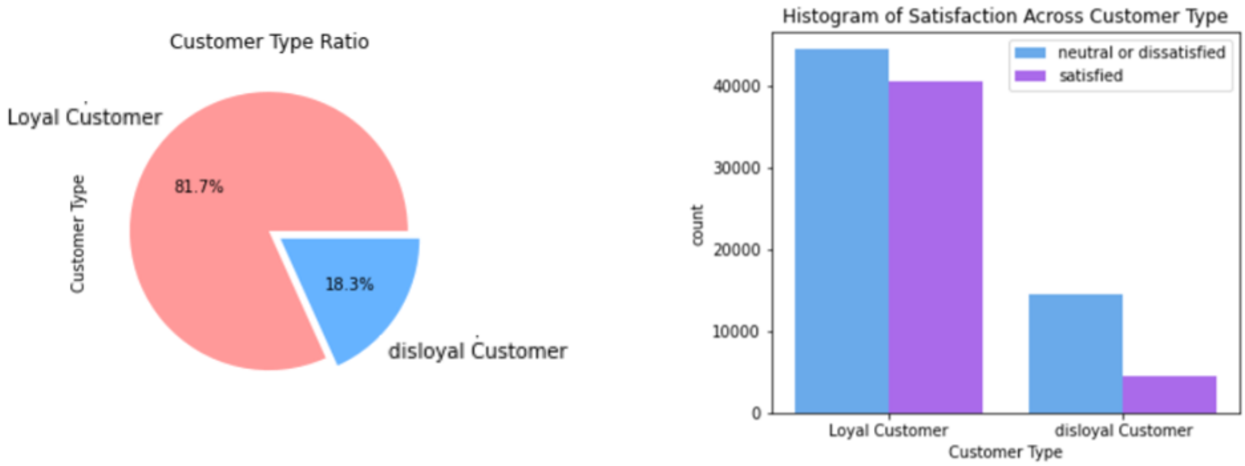


Figure 17: Customer Type Ratio and Histogram of Satisfaction Across Customer Type

We observed loyal customers are more satisfied than disloyal customers. But that does not mean most of the loyal customers are satisfied. Loyal customers or disloyal customers both are mostly unsatisfied.

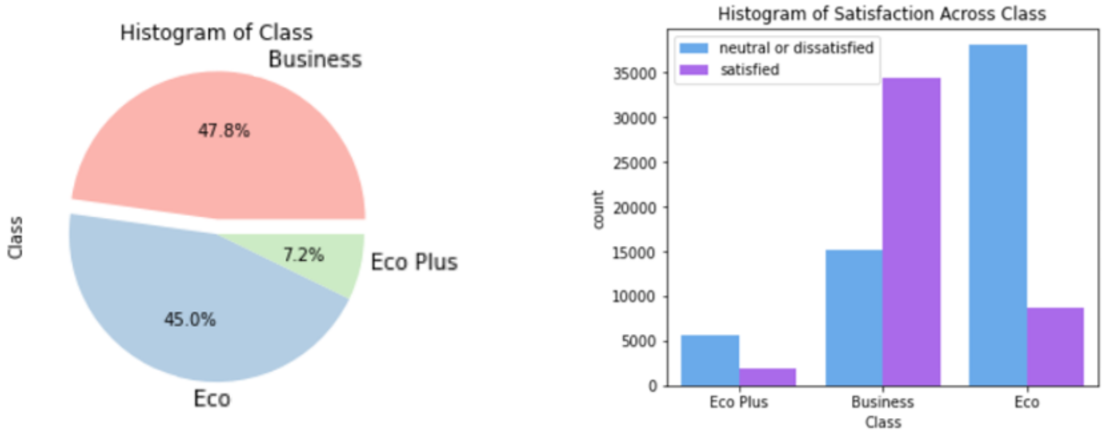


Figure 18: Class Ratio and Histogram of Satisfaction Across Class

Among the 3 types of cabin class types, the most satisfied customers are those who flew in the business class, and the most unsatisfied customers are those who flew in the eco class. People travel more via Eco rather than Eco Plus but still unsatisfied customers are higher than the business class.

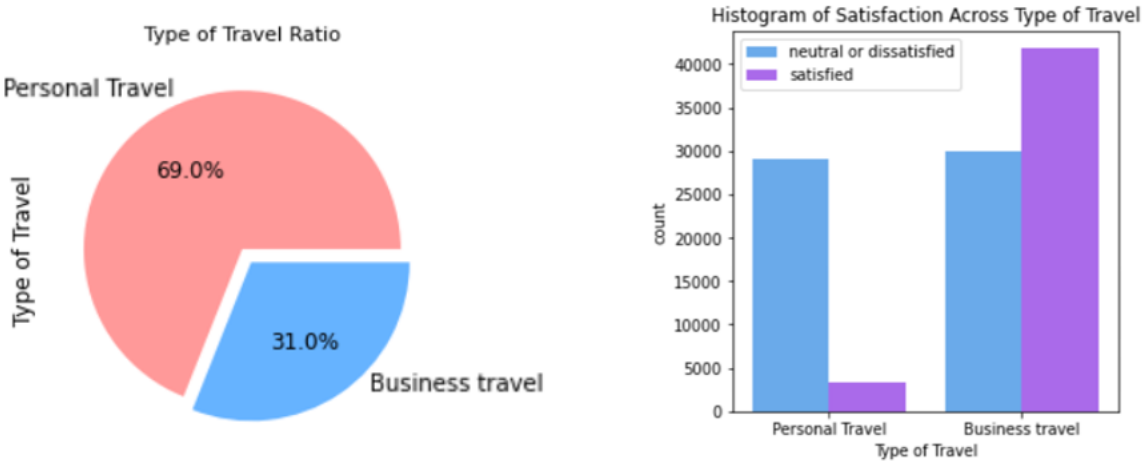
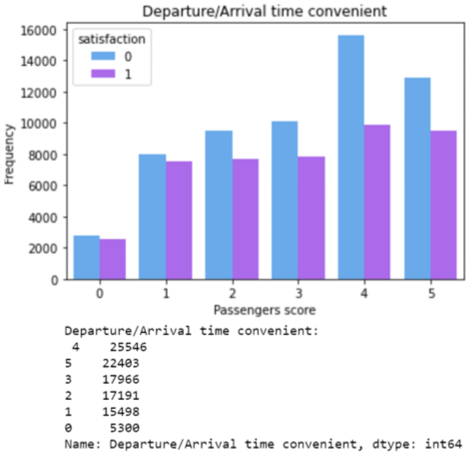
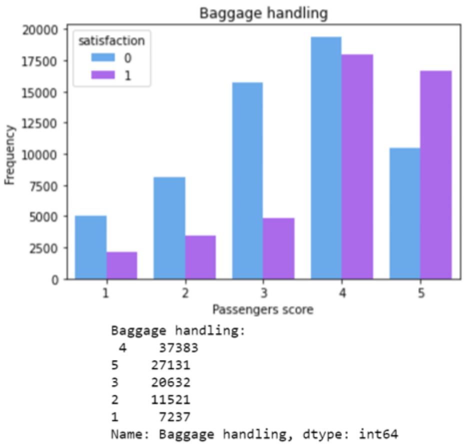


Figure 19: Type of Travel Ratio and Histogram of Satisfaction Across Type of Travel

The customers who flew with this airline for personal reasons, which is called personal travel, are the most unsatisfied rather than the customers who flew for business reasons which are called business travel. The reason for this dissatisfaction could be baggage handling because those who flew for personal traveling probably have more baggage than those who flew for business reasons and maybe they had a problem with the baggage, with just one point they can be unsatisfied for the whole journey.

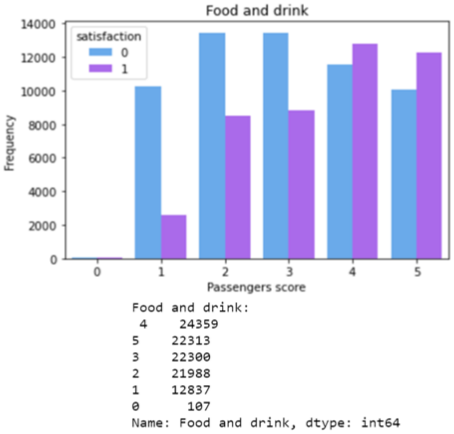


```
Departure/Arrival time convenient:
 4    25546
5    22403
3    17966
2    17191
1    15498
0     5300
Name: Departure/Arrival time convenient, dtype: int64
```

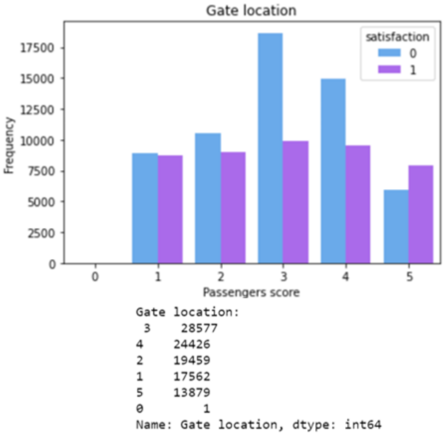


```
Baggage handling:
 4    37383
5    27131
3    20632
2    11521
1     7237
Name: Baggage handling, dtype: int64
```

Even though most people rate scores as 3, 4, or 5 for both of these variables, lots of the customers are overall unsatisfied with the flight. We observe that with departure/arrival time convenient, customers are overall much more likely to be unsatisfied. The Baggage Handling is having a significant effect on the customer satisfaction level, customers who rate a score of 5 for baggage handling are also overall satisfied with the flight.

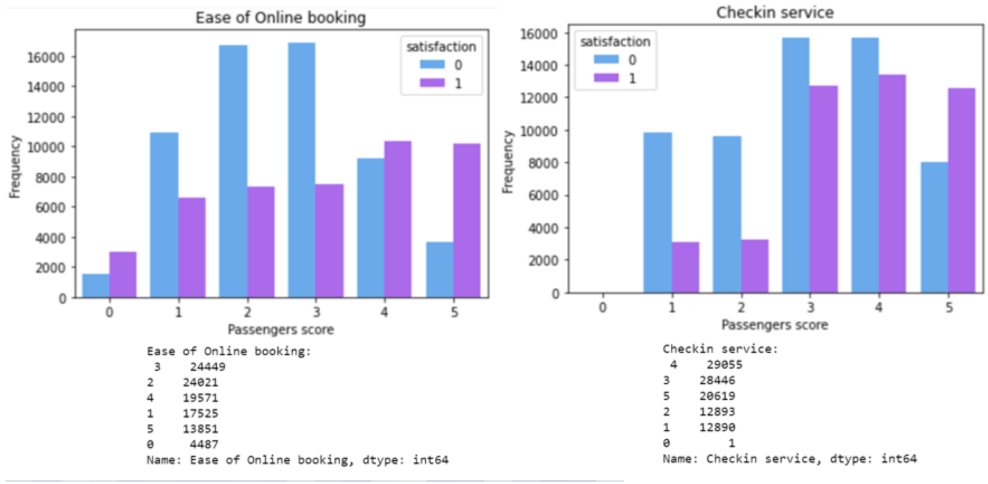


```
Food and drink:
 4    24359
5    22313
3    22300
2    21988
1    12837
0      107
Name: Food and drink, dtype: int64
```



```
Gate location:
 3    28577
4    24426
2    19459
1    17562
5    13879
0        1
Name: Gate location, dtype: int64
```

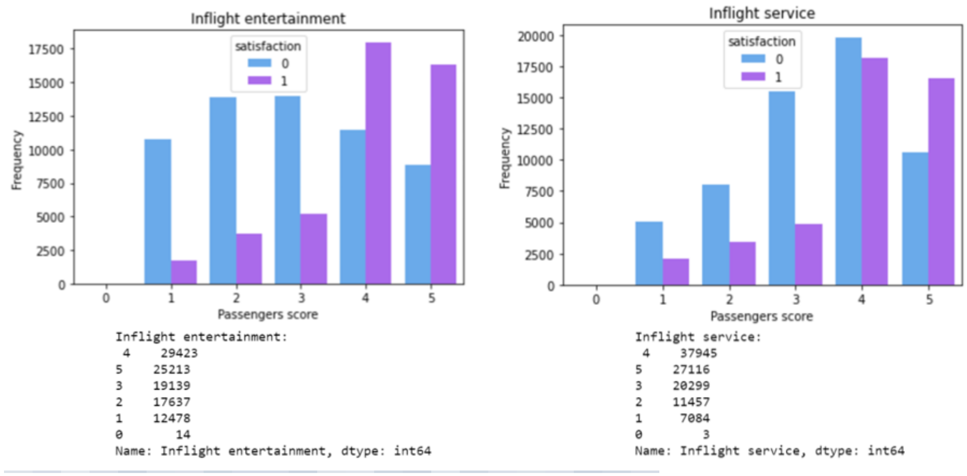
As shown, most of the customers like the food and drink. Although the food is great, many are not satisfied with the flight. Rather than those who rate a score of 4 or 5, customers are overall unsatisfied. As can be seen, most people did not like the Gate Location but it did not have enough effect on overall satisfaction. It seems that most people do not have an issue with the gate location but being satisfied with this is not enough to be satisfied with the flight in general.



```
Ease of Online booking:
 3    24449
2    24021
4    19571
1    17525
5    13851
0     4487
Name: Ease of Online booking, dtype: int64
```

```
Checkin service:
 4    29055
3    28446
5    20619
2    12893
1    12890
0        1
Name: Checkin service, dtype: int64
```

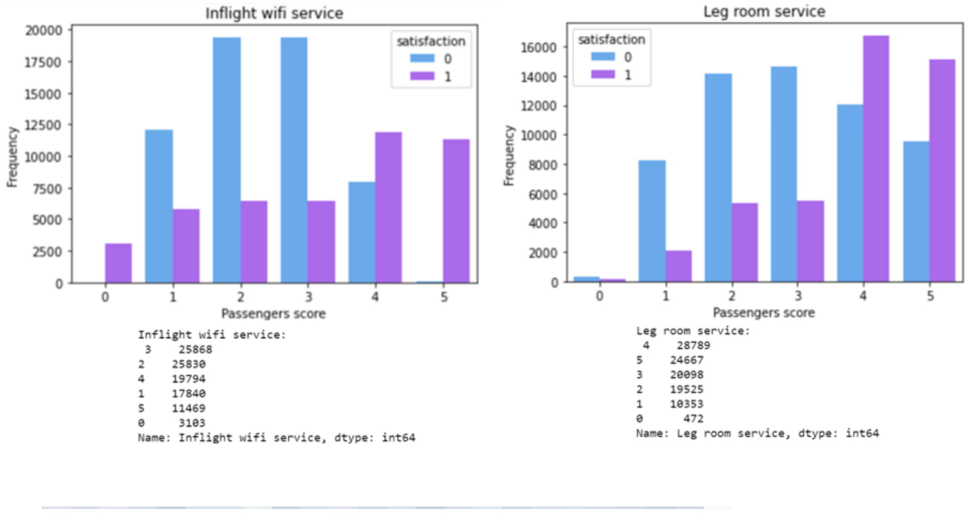
Although most of the customers are middle-aged, it seems lots of the customers have a problem with online booking. Perhaps those who gave a 4-5 are flying for business reasons and since they fly a lot, they are used to using online booking rather than the normal customers. We can see that customers rating rather than 4 or 5 on ease of online booking are overall mostly unsatisfied. As we can see, the check-in service worked better than the other parts, most people rate a score of 3,4, or 5 even though customers mostly were unsatisfied overall but mostly liked the check-in service.



```
Inflight entertainment:
 4    29423
5    25213
3    19139
2    17637
1    12478
0       14
Name: Inflight entertainment, dtype: int64
```

```
Inflight service:
 4    37945
5    27116
3    20299
2    11457
1     7084
0        3
Name: Inflight service, dtype: int64
```

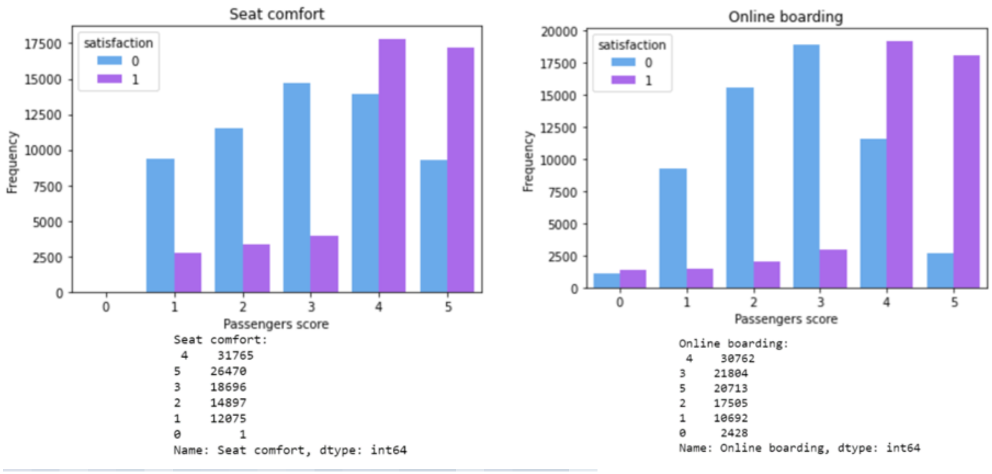
As shown by most customers like inflight entertainment and those who rate a score of 4 or 5 are overall much more likely to be satisfied than unsatisfied, which shows that it is one of the key flight attributes. Also with the inflight service, we observe that even though they mostly like it by rating it mostly 3,4 and 5, overall, they are mostly unsatisfied.



```
Inflight wifi service:
 3    25868
2    25830
4    19794
1    17840
5    11469
0     3103
Name: Inflight wifi service, dtype: int64
```

```
Leg room service:
 4    28789
5    24667
3    20098
2    19525
1    10353
0      472
Name: Leg room service, dtype: int64
```

As it might be seen, some customers did not like inflight Wi-Fi service that much and those who rate a score of 4 or 5 are overall much more likely to be satisfied. However, most of the customers like the legroom service, those who like it the most by the rate it as 4 or 5, overall satisfied with the flight, rather than the others.

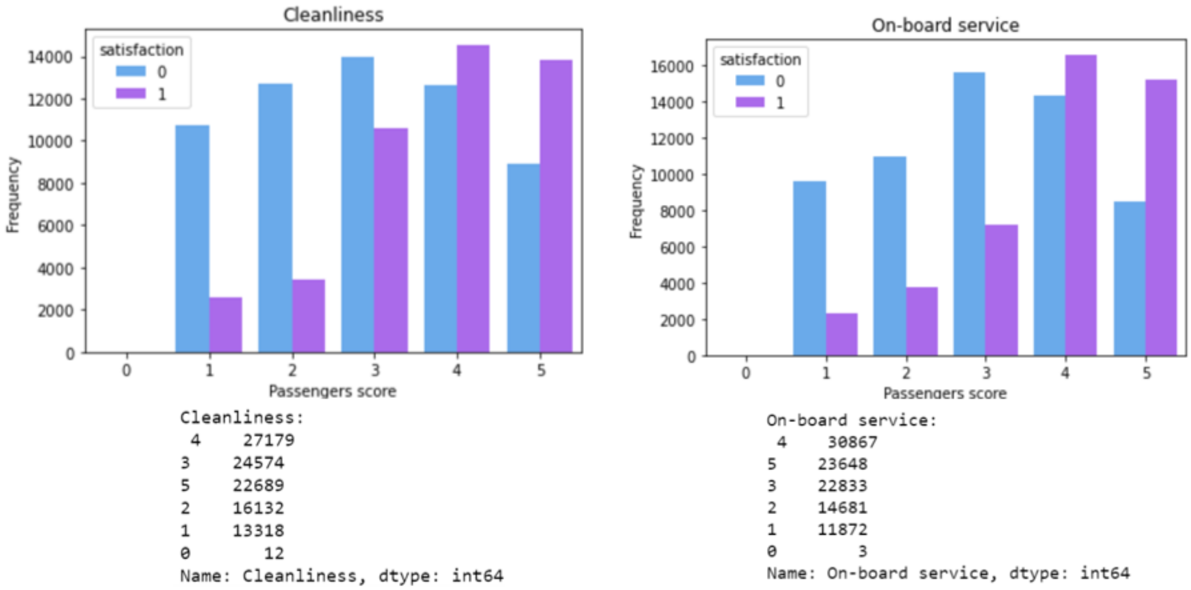


```
Seat comfort:
 4    31765
5    26470
3    18696
2    14897
1    12075
0        1
Name: Seat comfort, dtype: int64
```

```
Online boarding:
 4    30762
3    21804
5    20713
2    17505
1    10692
0     2428
Name: Online boarding, dtype: int64
```

Seat Comfort also has a big effect on satisfaction. In the histogram, lots of customers who are unsatisfied with the seat comfort also have unsatisfied flights too. On the other hand, those who gave 4-5 might fly in business class. Because when you fly business class you will have lots of opportunities like you can choose your seat from anywhere without paying any price so maybe that is why some of the people rate it with high scores.

Online also has a big effect on satisfaction. Almost all customers who were not satisfied with the online boarding system also did not like the flight overall. Maybe those who gave low scores were not good with technology or had a problem with the internet.



```
Cleanliness:
 4    27179
3    24574
5    22689
2    16132
1    13318
0       12
Name: Cleanliness, dtype: int64
```

```
On-board service:
 4    30867
5    23648
3    22833
2    14681
1    11872
0        3
Name: On-board service, dtype: int64
```

Figure 26: Cleanliness and On-board Service Histograms

We can see that most people like the on-board service and cleanliness. Those who gave scores of 4 or 5 are mostly also satisfied with the flight overall but rather than those, customers who gave low scores were mostly unsatisfied overall.

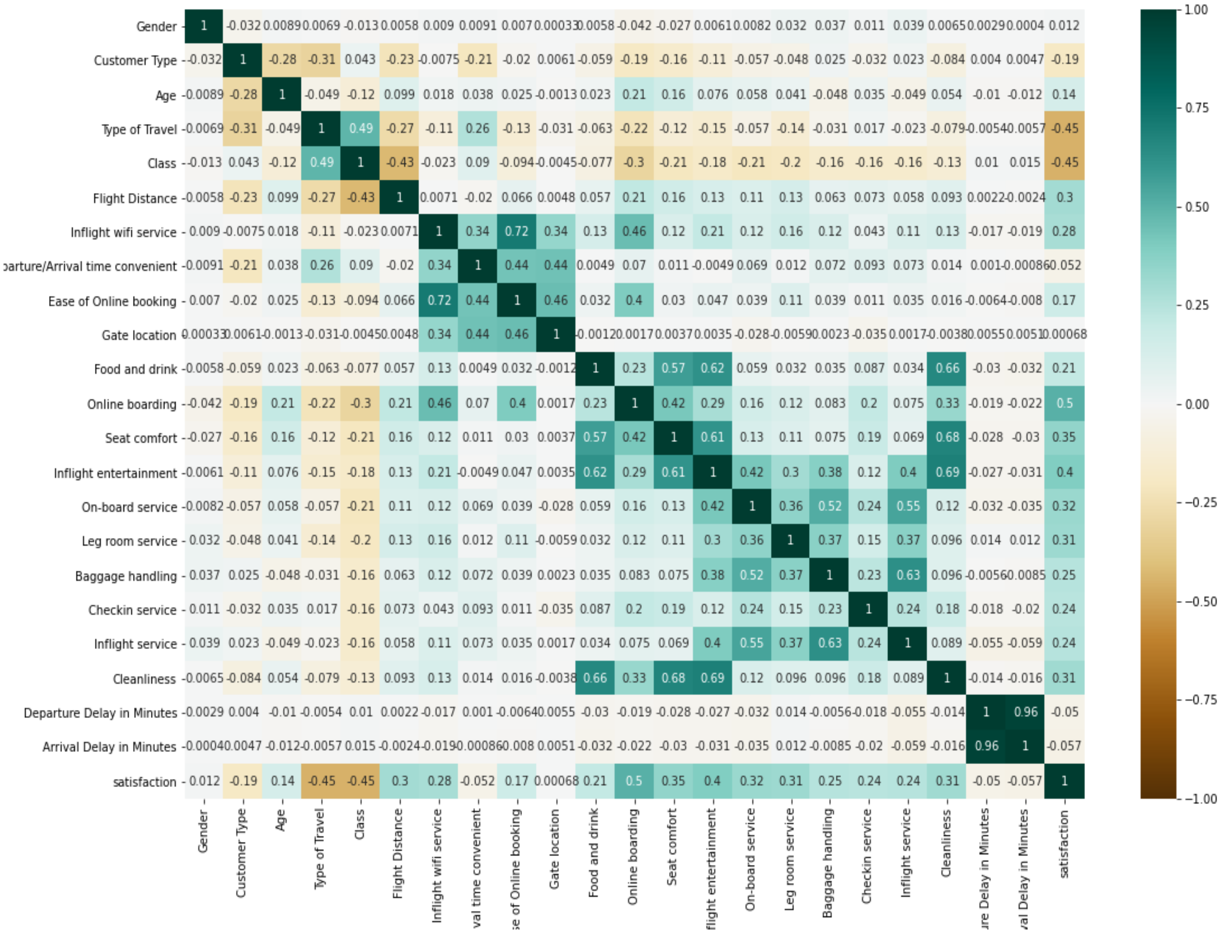


Figure 27: Correlation matrix of Dataset

Correlation matrix, summarize the data by showing the correlation coefficient between all the variables. Each cell shows the correlation between two variables, with that we can understand which pair of variables are related. We found that some of the more important variables for predicting satisfaction with the order are online boarding experience, inflight entertainment, seat comfort, onboarding service, leg room service, and cleanliness.

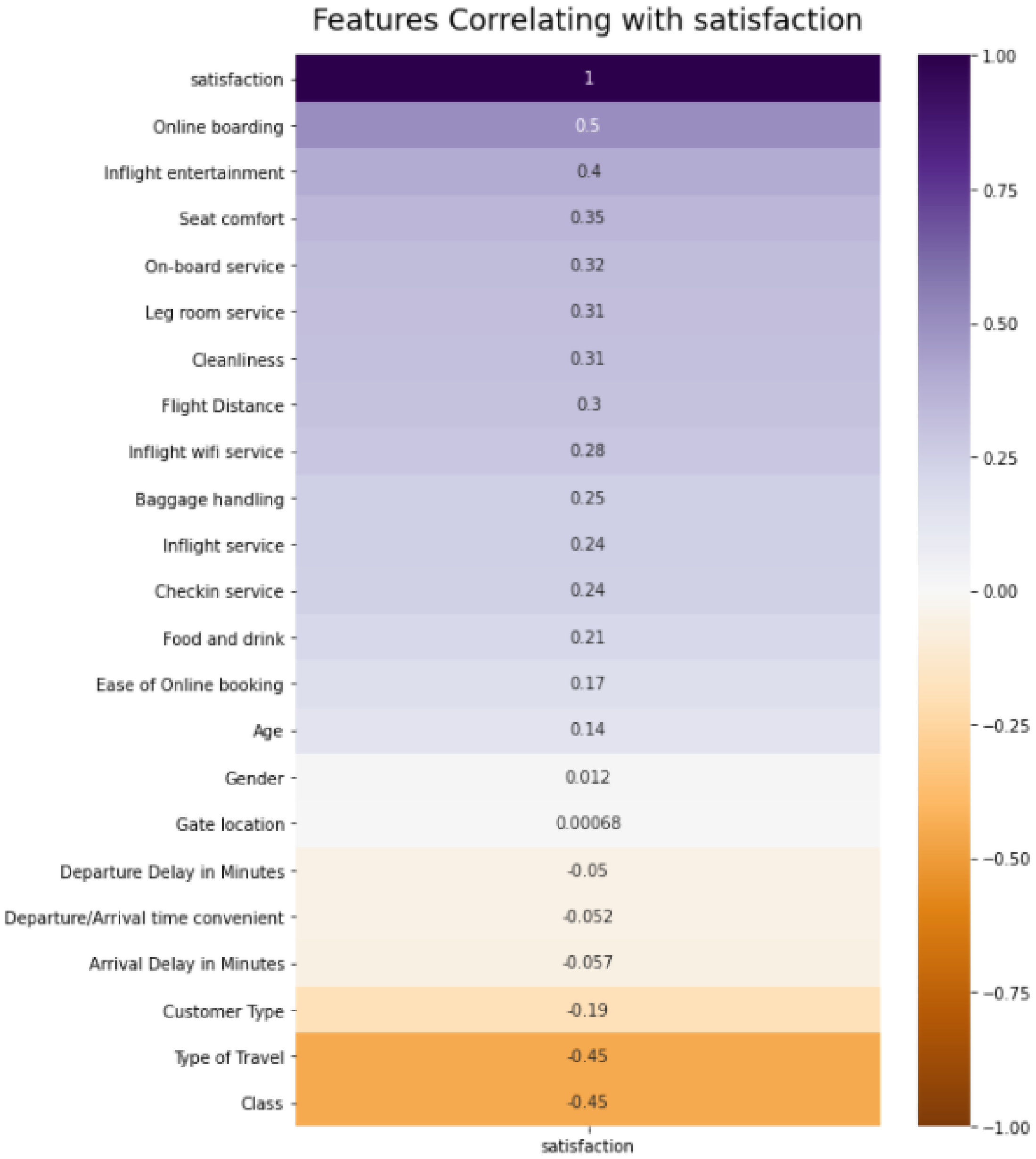
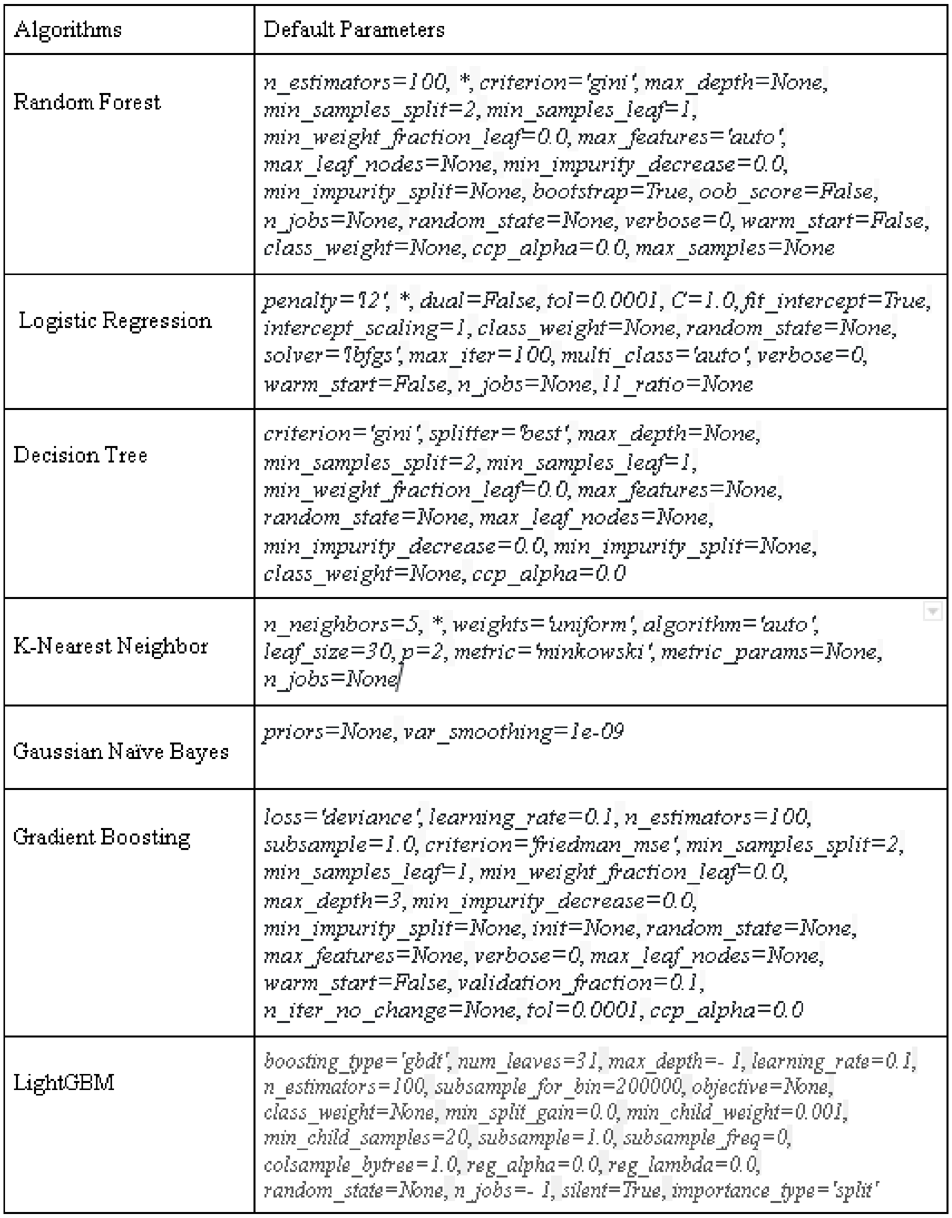


Figure 28: Features correlating with satisfaction

This correlation chart shows the important variables for satisfaction more clearly, we can see that it goes from what is most effective on satisfaction to one that has no effect which is the correlation between satisfaction and all of the key independent variables.

4. Data Mining Algorithms Used

In this study, we trained 7 different classification models even though there are a lot of classification algorithms. List of the classification algorithms we used are; Random Forest (RF), Logistic Regression (LR), Decision Tree (CART), K-Nearest Neighbor (KNN), Gaussian Naïve Bayes (GNB), Gradient Boosting (GBM) and LightGBM (LGBM). In addition, we ran all of the 7 classifiers with their default parameters. We compared these algorithms to find the best one to predict the satisfaction of the passengers.



Algorithms	Default Parameters
Random Forest	n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None
Logistic Regression	penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None
Decision Tree	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0
K-Nearest Neighbor	n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None
Gaussian Naïve Bayes	priors=None, var_smoothing=1e-09
Gradient Boosting	loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, min_impurity_split=None, init=None, random_state=None, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0
LightGBM	boosting_type='gbdt', num_leaves=31, max_depth=- 1, learning_rate=0.1, n_estimators=100, subsample_for_bin=200000, objective=None, class_weight=None, min_split_gain=0.0, min_child_weight=0.001, min_child_samples=20, subsample=1.0, subsample_freq=0, colsample_bytree=1.0, reg_alpha=0.0, reg_lambda=0.0, random_state=None, n_jobs=- 1, silent=True, importance_type='split'

1- Random Forest: It lies at the base of the Boruta algorithm, which selects important features in a dataset.

2- Logistic Regression: It is based on a set of independent variables, predicts the probability of action and obtains a value between 0 and 1.

3- Decision Tree: It splits into two or more homogeneous sets based on the most significant attributes making the groups as distinct as possible.

4- K-Nearest Neighbor: It is based on a feature similarity approach. It stores all existing cases to classify new cases by the majority vote of their neighbors. The status assigned to the class is the most common among K closest neighbors (Euclid, Manhattan, Minkowski, and Hamming), measured by a distance function.

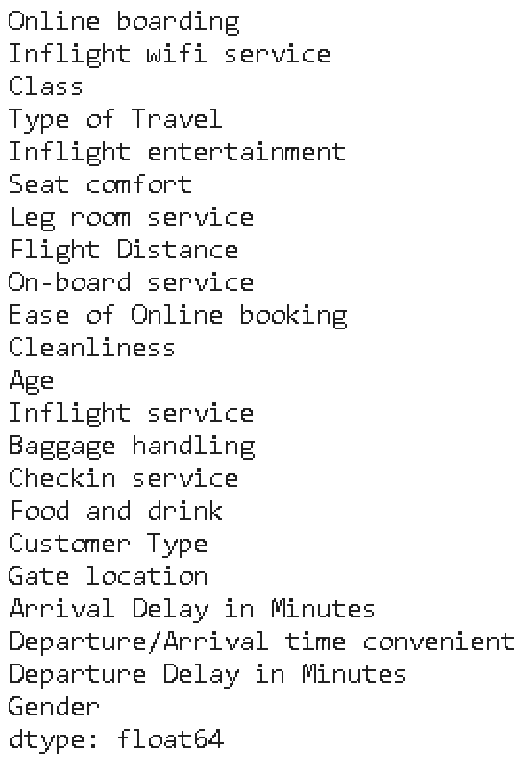
5- Gaussian Naïve Bayes : It assumes that the presence of a particular feature in a class has nothing to do with the existence of any other feature.

6- Gradient Boosting: It combines the predictions from multiple decision trees to generate the final predictions.

7- LightGBM: Light GBM is a gradient boosting framework that uses a tree based learning algorithm. It grows tree leaf-wise while other algorithms grow level-wise.

5. Experiments and Results

We created five different experiment datasets so that we can test classification models that we choose on these datasets to get a better understanding of satisfaction. It helped us to see better the attributes that have the most or no effect on passenger satisfaction. In order to select the attributes for experiment datasets, we used the values on the table below which is for partitioning the independent and dependent data.



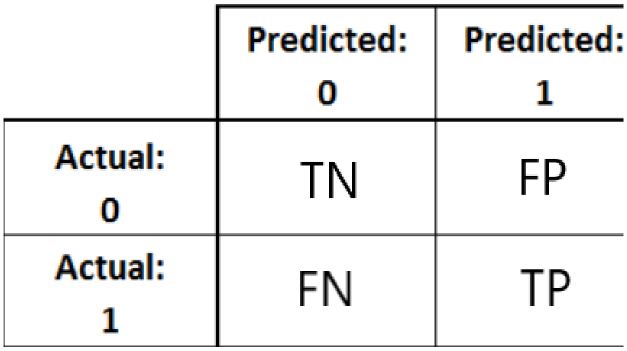
```
Online boarding                      0.216345
Inflight wifi service                0.164249
Class                                0.136638
Type of Travel                       0.113537
Inflight entertainment               0.093529
Seat comfort                         0.084464
Leg room service                     0.065138
Flight Distance                      0.060955
On-board service                     0.058090
Ease of Online booking               0.053799
Cleanliness                          0.051977
Age                                  0.046006
Inflight service                     0.045263
Baggage handling                     0.045051
Checkin service                      0.034748
Food and drink                       0.028352
Customer Type                        0.018536
Gate location                        0.015259
Arrival Delay in Minutes             0.006277
Departure/Arrival time convenient    0.005848
Departure Delay in Minutes           0.003282
Gender                               0.002484
dtype: float64
```

With using this table above, we choose gender, online boarding and inflight wifi service for using in the datasets.

In the table below, the experiment datasets and the details of the attributes that they contain or not are given.

	All Attributes	Gender	Online Boarding	Inflight wifi service	Online Boarding and Inflight wifi service
Dataset 1	+				
Dataset 2	+	x			
Dataset 3	+		x		
Dataset 4	+			x	
Dataset 5	+				x

We used Precision, Recall, F-Score, ROC Area and Accuracy metrics in order to measure algorithms' performance and compare them to each other for finding the best algorithm. If these metric values are higher on one of the algorithms we selected, then we can say that it has the higher performance and it is better for our study which is prediction of passenger satisfaction. Also for these metrics we used some basic terminologies which are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).



	Predicted: 0	Predicted: 1
Actual: 0	TN	FP
Actual: 1	FN	TP

T**rue Positive** : model correctly predicts that passengers are satisfied *overall*.

T**rue Negative:** model correctly predicts that passengers are neutral or dissatisfied.

F**alse Positive:** model incorrectly predicts that passengers are satisfied.(a "Type I error")

F**alse Negative**: model incorrectly predicts that passengers are neutral or dissatisfied. (a "Type I error".

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

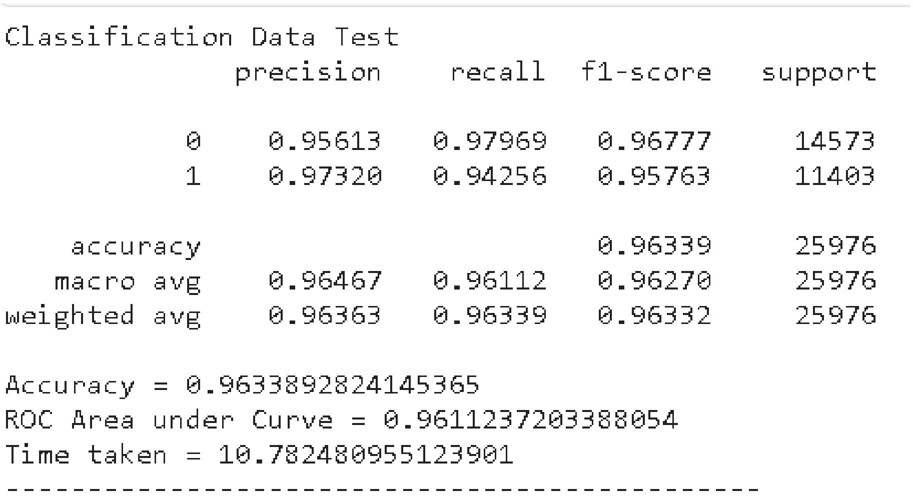
$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$

$$\text{F}_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

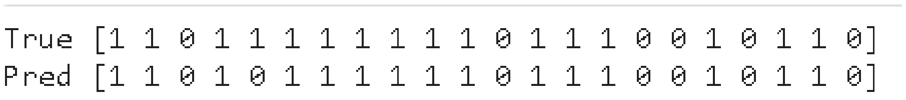
$$ROC\ Area = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

5.1 Random Forest



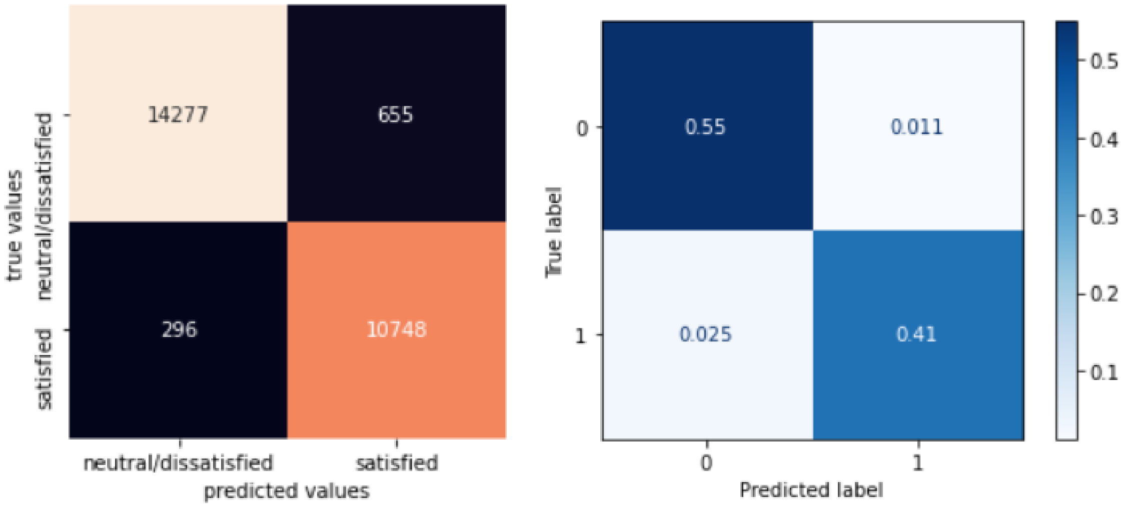
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.95613   0.97969   0.96777     14573
           1    0.97320   0.94256   0.95763     11403

    accuracy                        0.96339     25976
   macro avg    0.96467   0.96112   0.96270     25976
weighted avg    0.96363   0.96339   0.96332     25976

Accuracy = 0.9633892824145365
ROC Area under Curve = 0.9611237203388054
Time taken = 10.782480955123901
----------------------------------------------
```

We can see from the report that the RF model has high scores on all of the metrics.



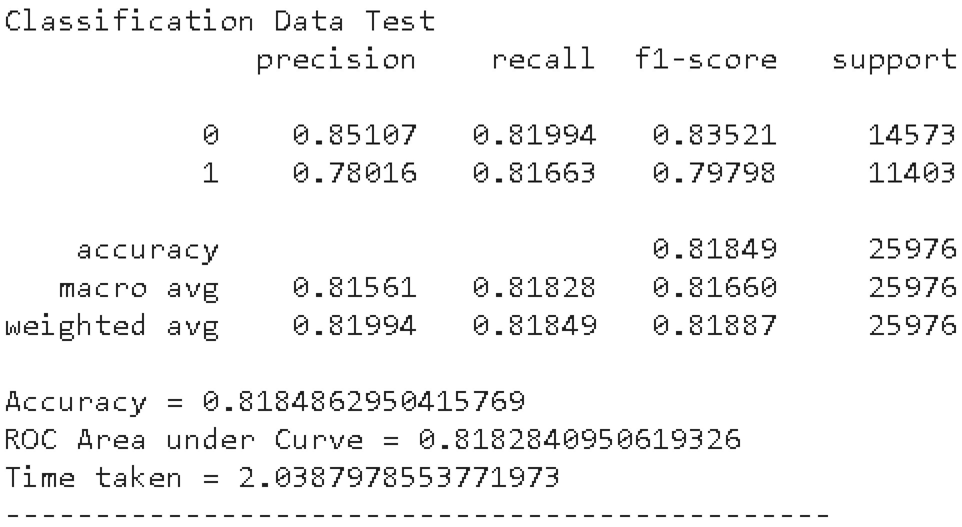
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
```

Here, we can clearly observe and compare the true values and predictions of the RF model.



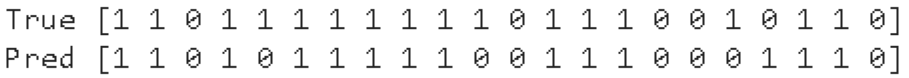
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts. First argument is true values[14277   296], the second argument is predicted values[655   10748].

5.2 Logistic Regression



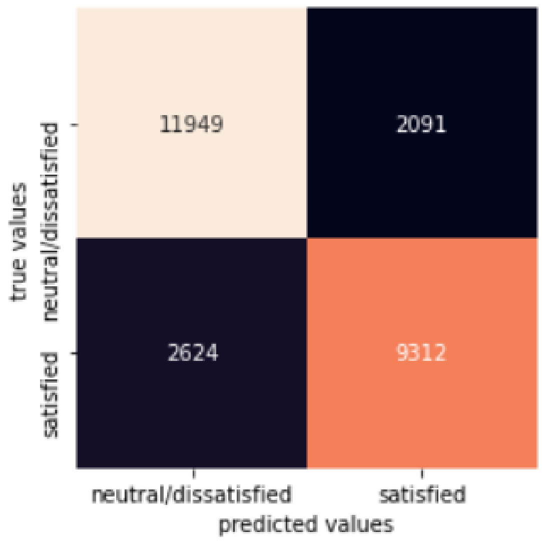
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.85107   0.81994   0.83521     14573
           1    0.78016   0.81663   0.79798     11403

    accuracy                        0.81849     25976
   macro avg    0.81561   0.81828   0.81660     25976
weighted avg    0.81994   0.81849   0.81887     25976

Accuracy = 0.8184862950415769
ROC Area under Curve = 0.8182840950619326
Time taken = 2.0387978553771973
----------------------------------------------
```

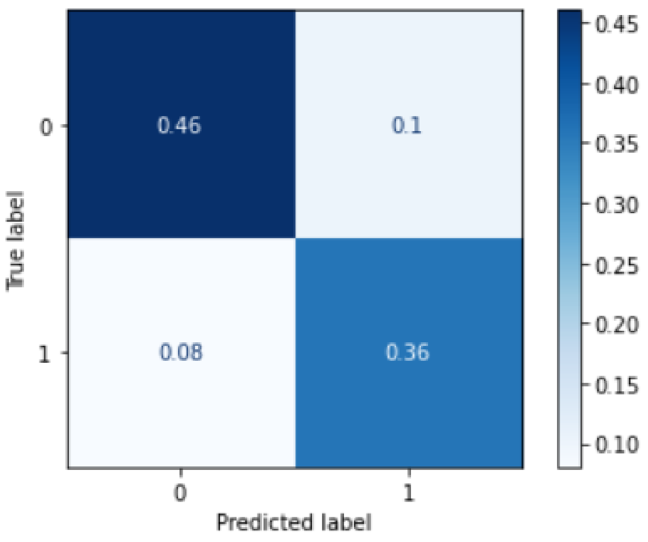
We can see from the report that the LR model has good scores on almost all of the metrics.



```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 0]
```

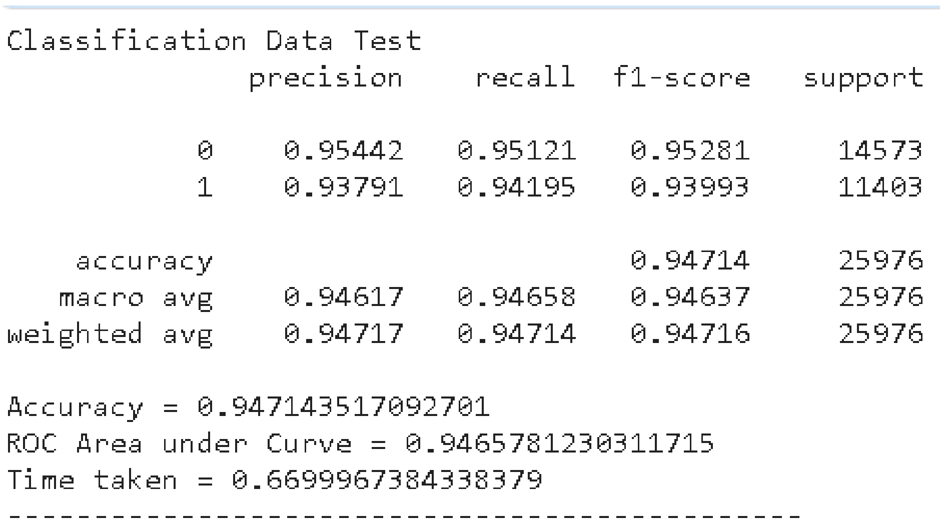
Here, we can clearly observe and compare the true values and predictions of the LR model.





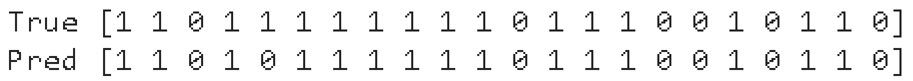
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.3 Decision Tree



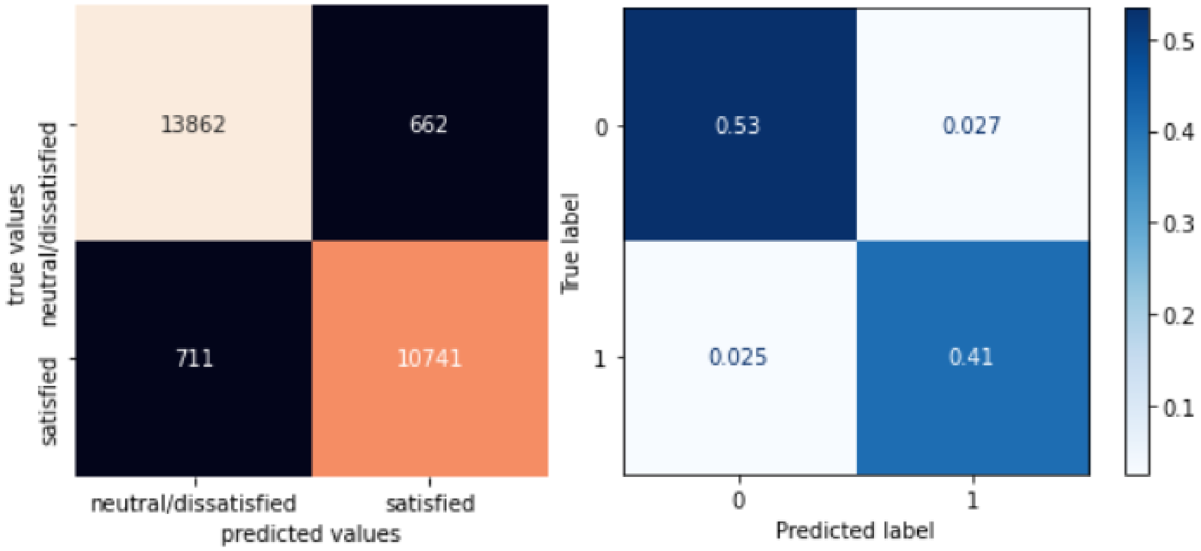
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.95442   0.95121   0.95281     14573
           1    0.93791   0.94195   0.93993     11403

    accuracy                        0.94714     25976
   macro avg    0.94617   0.94658   0.94637     25976
weighted avg    0.94717   0.94714   0.94716     25976

Accuracy = 0.947143517092701
ROC Area under Curve = 0.9465781230311715
Time taken = 0.6699967384338379
----------------------------------------------
```

We can see from the report that the CART model has high scores on all of the metrics.



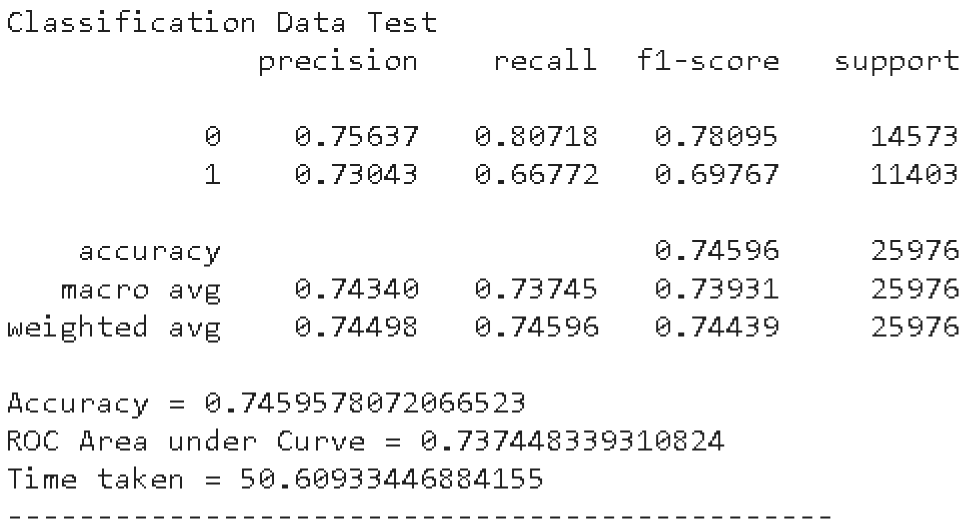
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
```

Here, we can clearly observe and compare the true values and predictions of the CART model.



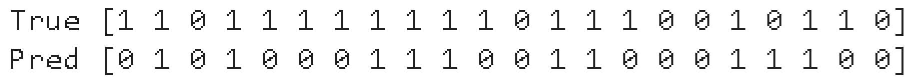
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.4 K-Nearest Neighbor



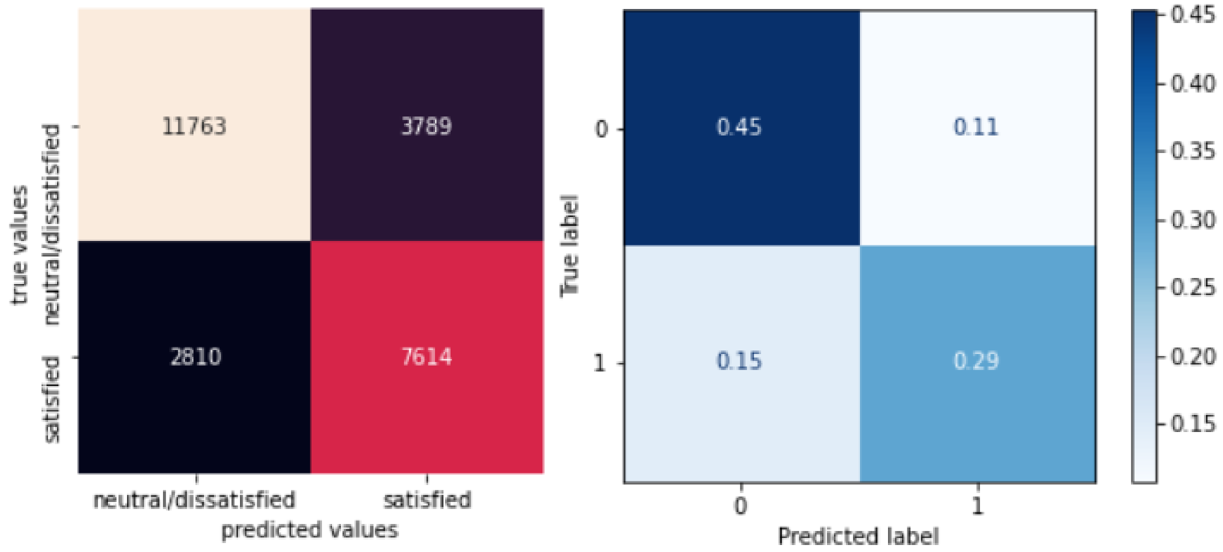
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.75637   0.80718   0.78095     14573
           1    0.73043   0.66772   0.69767     11403

    accuracy                        0.74596     25976
   macro avg    0.74340   0.73745   0.73931     25976
weighted avg    0.74498   0.74596   0.74439     25976

Accuracy = 0.7459578072066523
ROC Area under Curve = 0.737448339310824
Time taken = 50.60933446884155
----------------------------------------------
```

We can see from the report that the K-NN model has very low scores on most of the metrics.



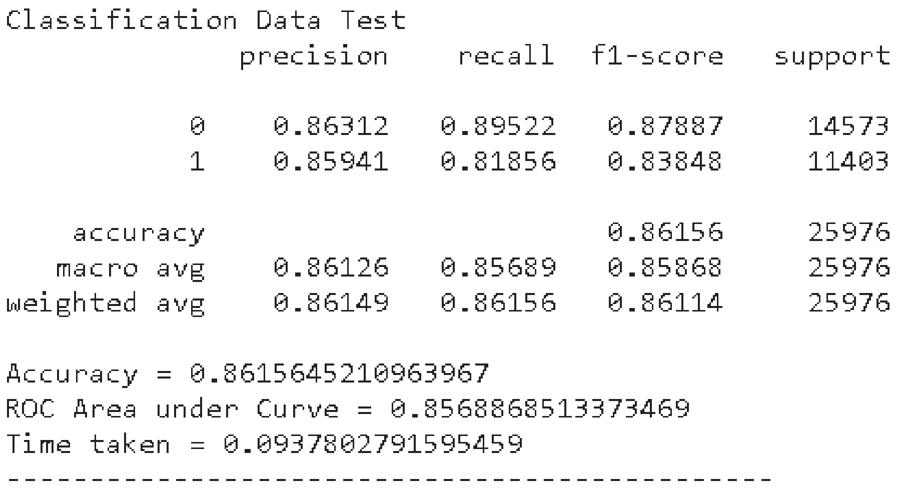
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [0 1 0 1 0 0 0 1 1 1 0 0 1 1 0 0 0 1 1 1 0 0]
```

Here, we can clearly observe and compare the true values and predictions of the K-NN model.



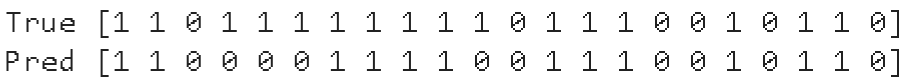
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.5 Gaussian Naïve Bayes



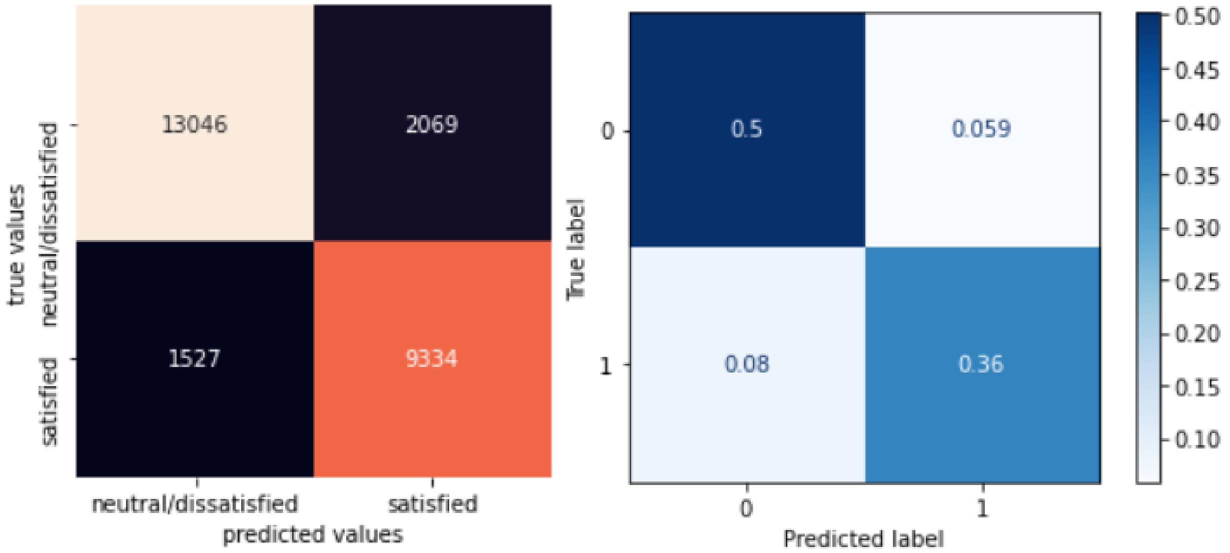
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.86312   0.89522   0.87887     14573
           1    0.85941   0.81856   0.83848     11403

    accuracy                        0.86156     25976
   macro avg    0.86126   0.85689   0.85868     25976
weighted avg    0.86149   0.86156   0.86114     25976

Accuracy = 0.8615645210963967
ROC Area under Curve = 0.8568868513373469
Time taken = 0.0937802791595459
----------------------------------------------
```

We can see from the report that the GNB model has good scores on most of the metrics.



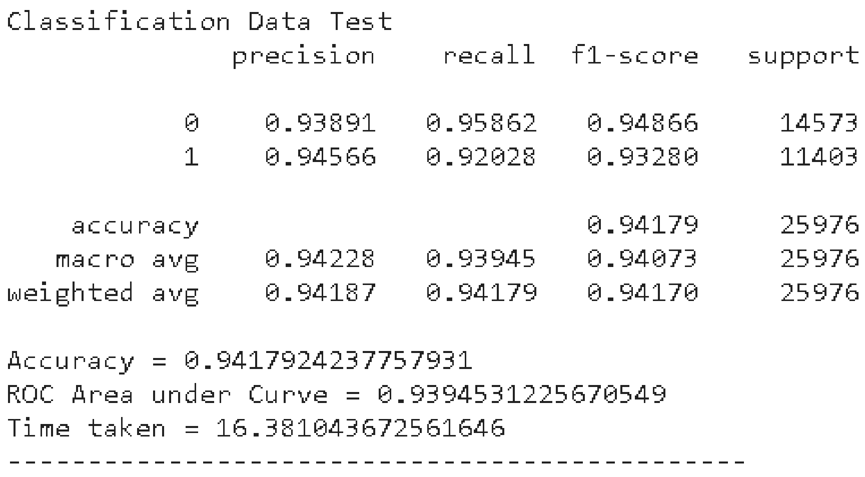
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0]
```

Here, we can clearly observe and compare the true values and predictions of the GNB model.



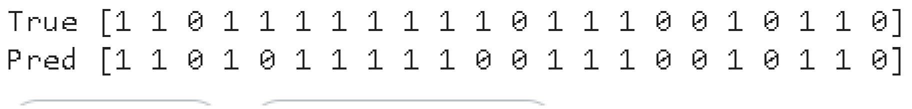
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.6 Gradient Boosting



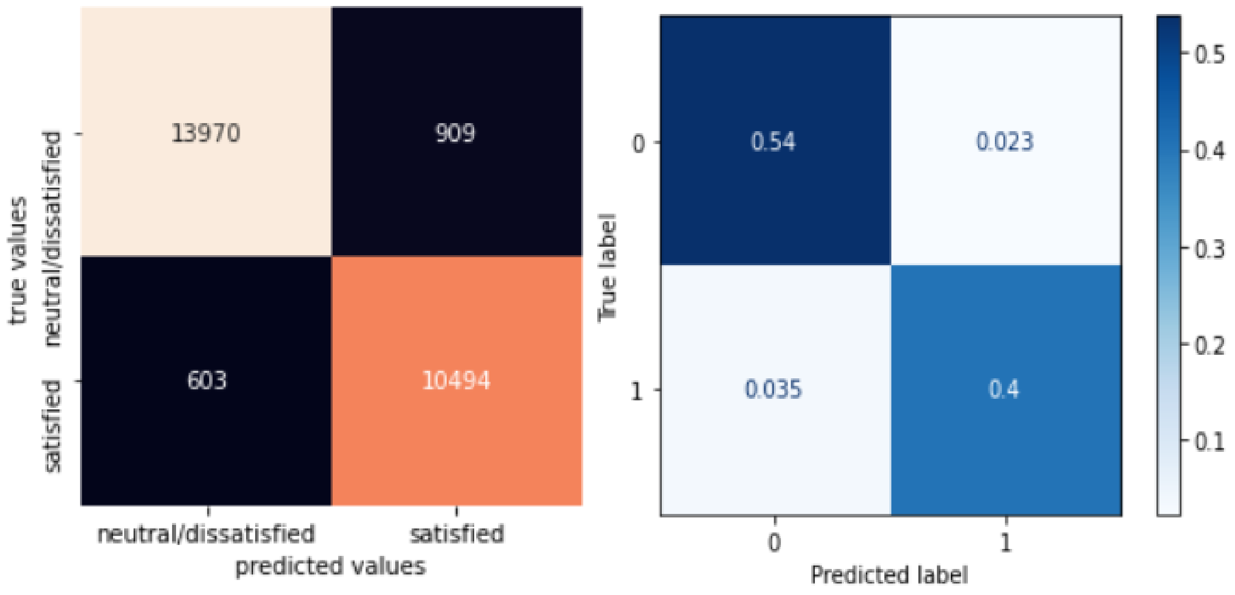
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.93891   0.95862   0.94866     14573
           1    0.94566   0.92028   0.93280     11403

    accuracy                        0.94179     25976
   macro avg    0.94228   0.93945   0.94073     25976
weighted avg    0.94187   0.94179   0.94170     25976

Accuracy = 0.9417924237757931
ROC Area under Curve = 0.9394531225670549
Time taken = 16.381043672561646
----------------------------------------------
```

We can see from the report that the GBM model has high scores on all of the metrics.



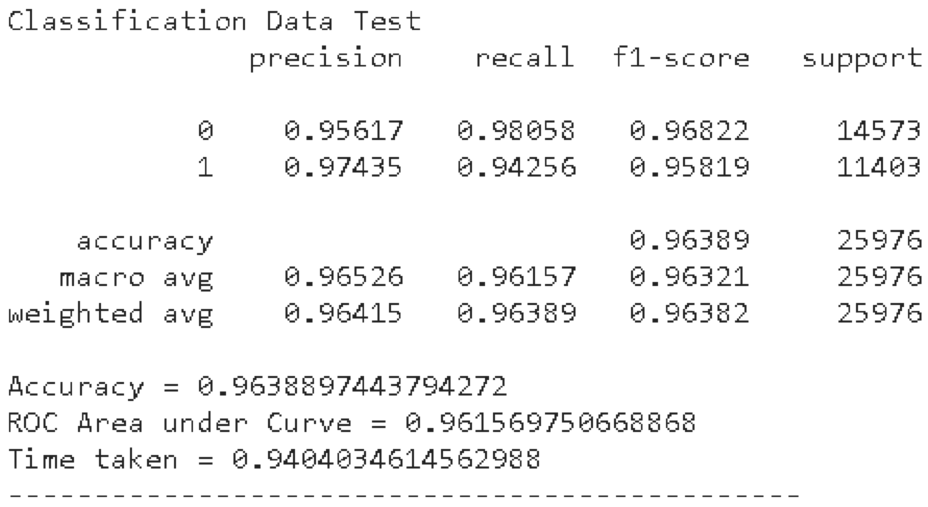
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0]
```

Here, we can clearly observe and compare the true values and predictions of the GBM model.



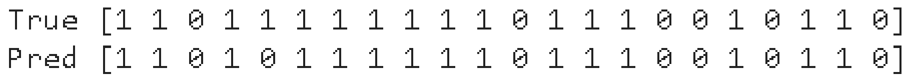
Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.7 LightGBM



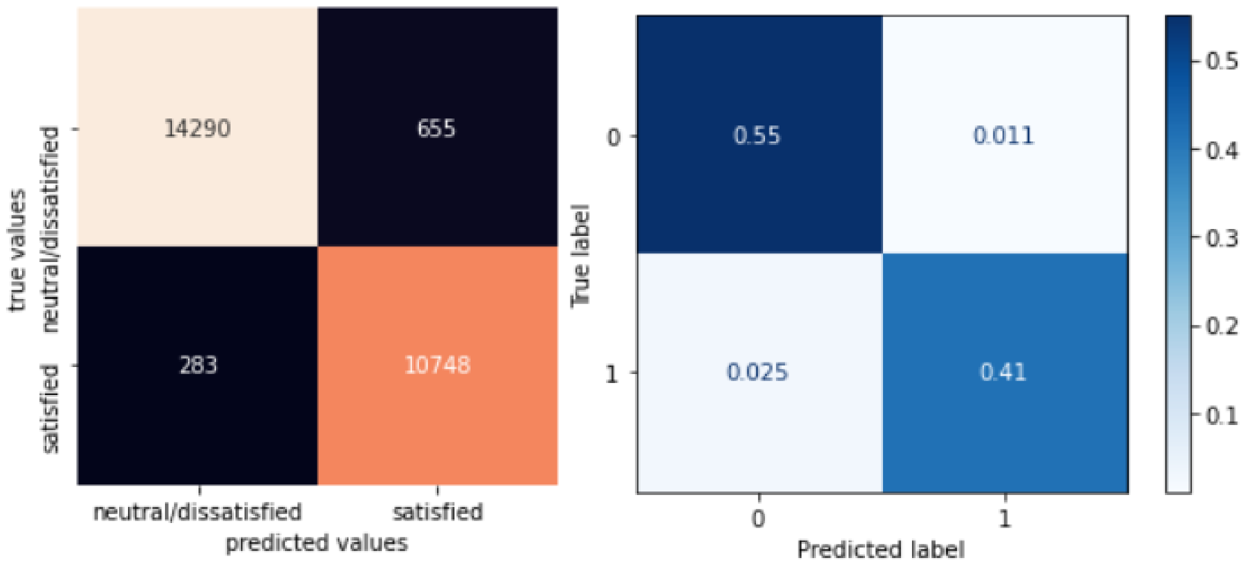
```
Classification Data Test
              precision    recall  f1-score   support

           0    0.95617   0.98058   0.96822     14573
           1    0.97435   0.94256   0.95819     11403

    accuracy                        0.96389     25976
   macro avg    0.96526   0.96157   0.96321     25976
weighted avg    0.96415   0.96389   0.96382     25976

Accuracy = 0.9638897443794272
ROC Area under Curve = 0.961569750668868
Time taken = 0.9404034614562988
---------------------------------------------
```

We can see from the report that the LGBM model has highest scores on all of the metrics.



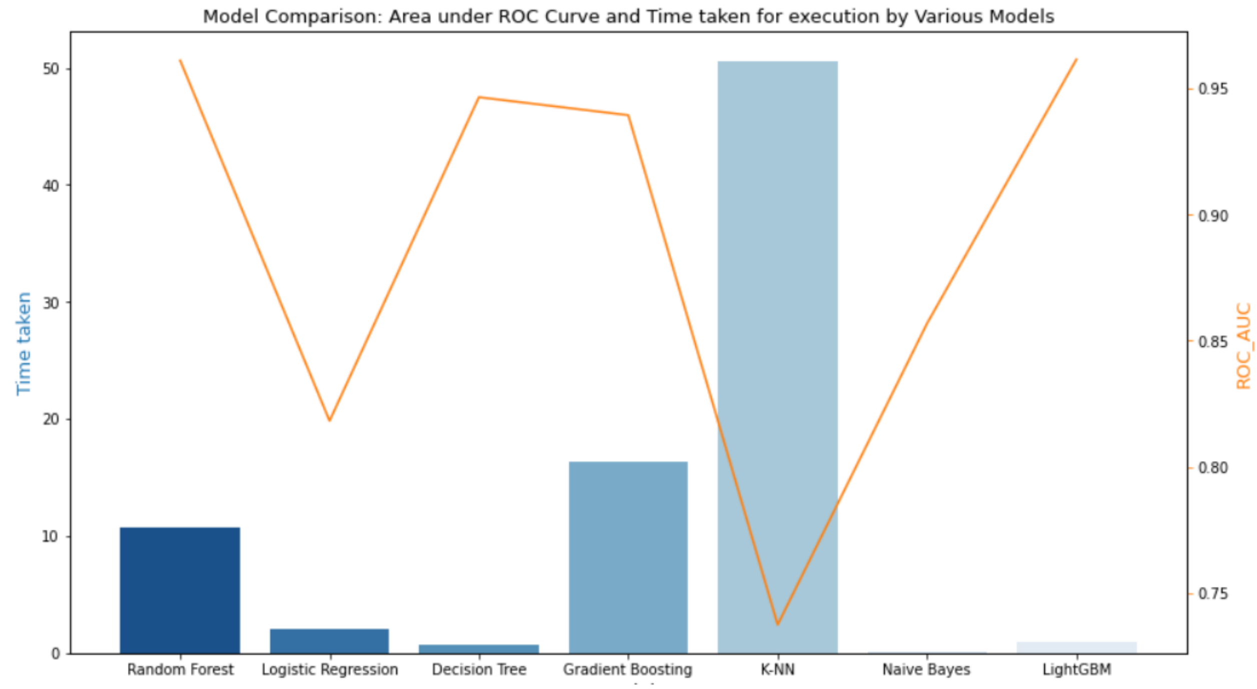
```
True [1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
Pred [1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 0]
```

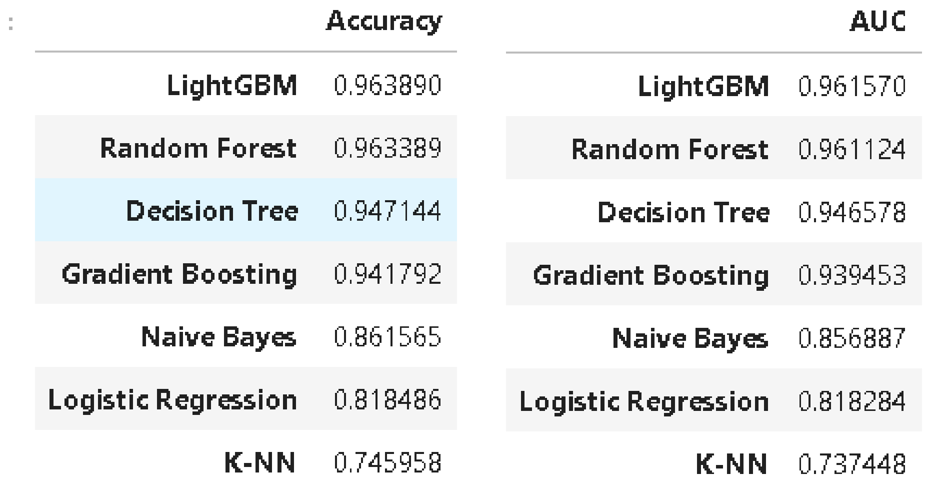
Here, we can clearly observe and compare the true values and predictions of the LGBM model.



Confusion matrix shows the satisfaction prediction results. The number of correct and incorrect results shown with counts.

5.8 Interpretation of the Results



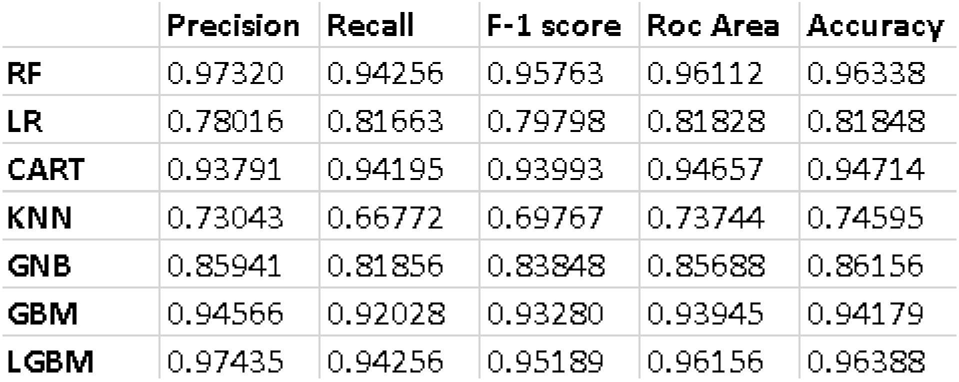


	Accuracy
LightGBM	0.963890
Random Forest	0.963389
Decision Tree	0.947144
Gradient Boosting	0.941792
Naive Bayes	0.861565
Logistic Regression	0.818486
K-NN	0.745958

	AUC
LightGBM	0.961570
Random Forest	0.961124
Decision Tree	0.946578
Gradient Boosting	0.939453
Naive Bayes	0.856887
Logistic Regression	0.818284
K-NN	0.737448

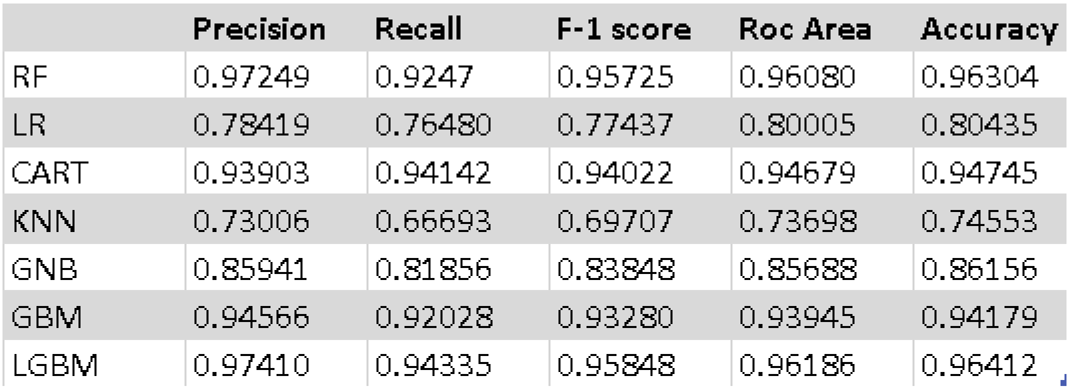
We examine the scores and we see that RF and LGBM, according to the high scores of accuracy and ROC Area had great performance, on the other hand with the lowest score K-NN has the worst performance. Other algorithms also had a good performance. We can see it clearly both from the graph and tables.

Dataset 1 - All attributes



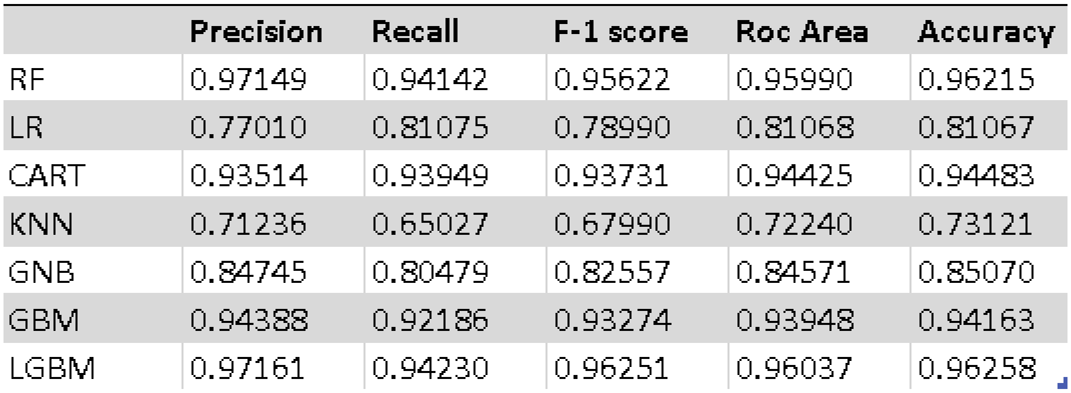
	Precision	Recall	F-1 score	Roc Area	Accuracy
RF	0.97320	0.94256	0.95763	0.96112	0.96338
LR	0.78016	0.81663	0.79798	0.81828	0.81848
CART	0.93791	0.94195	0.93993	0.94657	0.94714
KNN	0.73043	0.66772	0.69767	0.73744	0.74595
GNB	0.85941	0.81856	0.83848	0.85688	0.86156
GBM	0.94566	0.92028	0.93280	0.93945	0.94179
LGBM	0.97435	0.94256	0.95189	0.96156	0.96388

Dataset 2 - All features without ‘Gender’ attribute



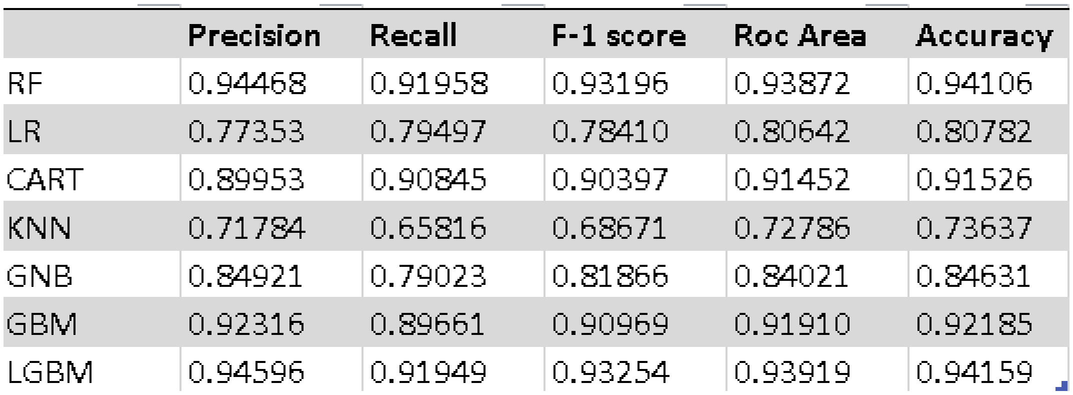
	Precision	Recall	F-1 score	Roc Area	Accuracy
RF	0.97249	0.9247	0.95725	0.96080	0.96304
LR	0.78419	0.76480	0.77437	0.80005	0.80435
CART	0.93903	0.94142	0.94022	0.94679	0.94745
KNN	0.73006	0.66693	0.69707	0.73698	0.74553
GNB	0.85941	0.81856	0.83848	0.85688	0.86156
GBM	0.94566	0.92028	0.93280	0.93945	0.94179
LGBM	0.97410	0.94335	0.95848	0.96186	0.96412

Dataset 3 - All features without ‘Online boarding’ attribute



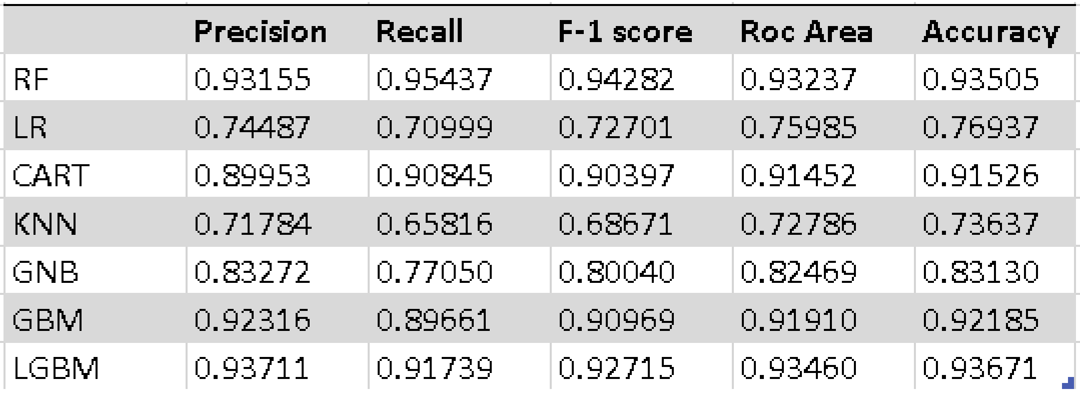
	Precision	Recall	F-1 score	Roc Area	Accuracy
RF	0.97149	0.94142	0.95622	0.95990	0.96215
LR	0.77010	0.81075	0.78990	0.81068	0.81067
CART	0.93514	0.93949	0.93731	0.94425	0.94483
KNN	0.71236	0.65027	0.67990	0.72240	0.73121
GNB	0.84745	0.80479	0.82557	0.84571	0.85070
GBM	0.94388	0.92186	0.93274	0.93948	0.94163
LGBM	0.97161	0.94230	0.96251	0.96037	0.96258

Dataset 4 - All features without ‘Inflight wifi service’ attribute.



	Precision	Recall	F-1 score	Roc Area	Accuracy
RF	0.94468	0.91958	0.93196	0.93872	0.94106
LR	0.77353	0.79497	0.78410	0.80642	0.80782
CART	0.89953	0.90845	0.90397	0.91452	0.91526
KNN	0.71784	0.65816	0.68671	0.72786	0.73637
GNB	0.84921	0.79023	0.81866	0.84021	0.84631
GBM	0.92316	0.89661	0.90969	0.91910	0.92185
LGBM	0.94596	0.91949	0.93254	0.93919	0.94159

Dataset 5 - All features without both ‘Online boarding’ and ‘Inflight wifi service’ attribute.



	Precision	Recall	F-1 score	Roc Area	Accuracy
RF	0.93155	0.95437	0.94282	0.93237	0.93505
LR	0.74487	0.70999	0.72701	0.75985	0.76937
CART	0.89953	0.90845	0.90397	0.91452	0.91526
KNN	0.71784	0.65816	0.68671	0.72786	0.73637
GNB	0.83272	0.77050	0.80040	0.82469	0.83130
GBM	0.92316	0.89661	0.90969	0.91910	0.92185
LGBM	0.93711	0.91739	0.92715	0.93460	0.93671

We can see that Dataset 2 is the one with all attributes without Gender attribute and these datasets’ metric scores almost as the as Dataset 1 which contain all attributes. With this information, we can say that passengers’ gender does not have any effect on the prediction of the algorithms.So, actually we do not need to know passengers gender for prediction and gender attribute can be excluded.

However, we observed that almost all metric scores in Dataset 3 and 4 are decreased compared to scores in Dataset 1 and 2, which shows that online boarding and inflight wifi service has a big effect on passenger satisfaction. Dataset 5 which contains the absence of both online boarding and inflight wifi service attributes also has low metric scores.We need both of these attributes to estimate passenger satisfaction and they will be needed while creating the real-world model.

We observed that RF and LGBM performed very well according to all of the high scores on the datasets. On the other hand with the lowest score K-NN had the lowest scores on all of the datasets which failed in passenger satisfaction prediction. LGBM stands out as the best algorithm for our study as it has slightly higher scores than RF in most of the datasets and scores.

6. Conclusion

The airline industry assembles an important part of the wider travel industry and with customers being the main source of income, this makes them the most important factor for success. In order to be successful they need to make passengers satisfied from their experience. Therefore, they need to understand customers' expectations to deliver unique experiences and should consider high-grade satisfaction as a key factor to retain that passenger. Based on understanding the reasons for passengers being satisfied or not, improvements will be made to provide better service by the airline company which will cause the company to be successful. So it is important to predict the satisfaction, in order to make improvements.

In this study we used Airline Passenger Satisfaction dataset and classification models to help to get better satisfaction from passengers, and evaluate a solution for their dissatisfaction. After cleaning the dataset and doing some preprocessing steps like dropping unnecessary columns, we make it ready for classification algorithms use. We select 7 models for predicting satisfaction and choose the best one according to some metric scores. Also we compared them with Some of the models who had good scores and performed very well and one of them failed badly compared to the other 6. However, RF and LGBM performed the highest scores on all of the cases, datasets and of course one of them has to be the best one which is the LGBM, it had a scores slightly higher than RF in some datasets, and happened to be the best model for airline passenger satisfaction prediction.

Appendix

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)


import seaborn as sns # Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics

import matplotlib.pyplot as plt # collection of functions that make matplotlib work like MATLAB


%matplotlib inline

from sklearn.preprocessing import LabelEncoder # Encode target labels with values between 0 and n_classes-1.

import warnings

warnings.filterwarnings('ignore') #to ignore deprecation warnings


import os

for dirname, _, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))


# Importing data into variables

training_set = pd.read_csv('../input/airline-passenger-satisfaction/train.csv')

test_set = pd.read_csv('../input/airline-passenger-satisfaction/test.csv')


# Row and column count of training and test sets
```

```
print("Shape of train dataset:  ",training_set.shape )

print("Shape of test dataset:  ",test_set.shape )

 # Getting the first 10 data from training set

training_set.head(10)


# Data Preparation

# Getting rid of the unnecessary columns in both training and test sets

training_set.drop(labels = ['Unnamed: 0', 'id'], axis = 1, inplace= True)

test_set.drop(labels = ['Unnamed: 0', 'id'], axis = 1, inplace= True)


dataset = training_set.append(test_set)


# Getting more details regarding data

# Print a concise summary of a DataFrame

# To get a quick overview of the dataset

dataset.info()


#find categorical data

categorical_data = training_set.select_dtypes(exclude= np.number)

categorical_col = categorical_data.columns

categorical_data.head(10)


# Find categorical data 'object'

# Convert categorical data to numeric

lencoders = {}
```

```
for col in training_set.select_dtypes(include=['object']).columns: # select the columns that include 'object'
    lencoders[col] = LabelEncoder() # used to transform non-numerical labels
     training_set[col] = lencoders[col].fit_transform(training_set[col]) #do a calculation and fitting data on the training set


lencoders = {}
for col in test_set.select_dtypes(include=['object']).columns: # select the columns that include 'object'
    lencoders[col] = LabelEncoder() # used to transform non-numerical labels
     test_set[col] = lencoders[col].fit_transform(test_set[col]) #do a calculation and fitting data on the training set


training_set.head(10)


# Checking if there is any null value
training_set.isnull().sum()


# Replacing null values with the mean of the column
training_set['Arrival Delay in Minutes'].fillna((training_set['Arrival Delay in Minutes'].mean()), inplace=True)


test_set['Arrival Delay in Minutes'].fillna((test_set['Arrival Delay in Minutes'].mean()), inplace=True)


features = ['Gender','Customer Type', 'Age', 'Type of Travel', 'Class',
       'Flight Distance', 'Inflight wifi service',
    'Departure/Arrival time convenient', 'Ease of Online booking',
    'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
    'Inflight entertainment', 'On-board service', 'Leg room service',
    'Baggage handling', 'Checkin service', 'Inflight service',
    'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes']
target = ['satisfaction']


# Split into test and train
X_train = training_set[features]
y_train = training_set[target].to_numpy()
X_test = test_set[features]
y_test = test_set[target].to_numpy()


X_test.head()


dataset = training_set.append(test_set)
training_set.isnull().sum()


training_set.describe() # calculating some statistical data like percentice, mean and std of the numerical values of DataFrame


numeric =['Gender',    'Customer Type',        'Type of Travel',         'Class', 'satisfaction']


for count in numeric:
  print("{} \n".format(training_set[count].value_counts())) # formats the specified value(s) and insert them inside the string's placeholder. The placeholder is defined using curly brackets: {}.
```

```
# 0- Female, 1- Male

colors = ['#ff9999','#66b3ff']

ax_worktype2=training_set['Gender'].value_counts().plot(kind='pie', rot=0, title='Gender Ratio', colors = colors,autopct='%1.1f%%')


# 0- Loyal Customer, 1- Disloyal Customer

training_set['Customer Type'].value_counts().plot(kind='bar', rot=0, title='Histogram of Customer Type')


plt.figure(figsize = (8, 5))

age_hist = sns.distplot(training_set['Age'], color = 'blue')

plt.ylabel('Density', size = 12)

plt.xlabel('Age', size = 14)


# 0- Business Travel, 1- Personal Travel

training_set['Type of Travel'].value_counts().plot(kind='bar', rot=0, title='Histogram of Type of Travel')


# 0- Business, 1- Eco, 2- Eco Plus

# People travel more via Eco rather than Eco Plus

training_set['Class'].value_counts().plot(kind='bar', rot=0, title='Histogram of Class')


# 0- neutral/dissatisfied, 1- satisfied

print('People are more neutral/dissatisfied than satisfied with the airline')
```

```
training_set['satisfaction'].value_counts().plot(kind='bar', rot=0, title='Histogram of Satisfaction', color = ['salmon','#66b3ff'])
print('How many people satisfacted by Airline service: ')
training_set.groupby('Gender')[['satisfaction']].sum()

print('Percentage of satisfacted people : ')
training_set.groupby('Gender')[['satisfaction']].sum()/ training_set.groupby('Gender')[['satisfaction']].count()

# 0- Female, 1 - Male,  0- Dissatisfaction, 1-satisfaction
plt.figure(figsize = (10,10))
plt.subplot(2,2,1)
sns.countplot(data = training_set, x="Gender" , hue="satisfaction", palette="cool")
plt.title("Histogram of Satisfaction Across Gender")
plt.legend()

# 0- Business, 1- Eco, 2- Eco Plus
print('Satisfaction of people depending on the class: ')
training_set.groupby('Class')[['satisfaction']].sum()

# 0- Business, 1- Eco, 2- Eco Plus
print('Number of people for each class:')
training_set.groupby('Class')[['satisfaction']].count()

# 0- Business, 1- Eco, 2- Eco Plus
print('Percentage of satisfacted people depending from the class:')
```

```
training_set.groupby('Class')[['satisfaction']].sum()/
training_set.groupby('Class')[['satisfaction']].count()


# 0- Business, 1 - Eco, 2- Eco Plus

plt.figure(figsize = (10,10))

plt.subplot(2,2,1) #  which creates a single subplot within a grid. As you can see, this command takes three integer arguments—the number of rows, the number of columns, and the index of the plot to be created in this scheme

sns.countplot(data = training_set, x="Class" , hue="satisfaction", palette="cool") # countplot method is used to Show the counts of observations in each categorical bin using bars.

plt.title("Histogram of Satisfaction Across Class")

plt.legend() # legend is an area describing the elements of the graph


# 0-Business Travel, 1- Personal Travel

print('Percentage of satisfacted people depending from the type of travel:')

training_set.groupby('Type of Travel')[['satisfaction']].sum()/ training_set.groupby('Type of Travel')[['satisfaction']].count()


# 0-Business Travel, 1- Personal Travel

plt.figure(figsize = (10,10))

plt.subplot(2,2,1)

sns.countplot(data = training_set, x="Type of Travel" , hue="satisfaction", palette="cool")

plt.title("Histogram of Satisfaction Across Type of Travel")

plt.legend()


print('Percentage of satisfacted people depending from the customer type:')
```

```
training_set.groupby('Customer Type')[['satisfaction']].sum()/ training_set.groupby('Type of Travel')[['satisfaction']].count()


# 0- Loyal Customer, 1- Disloyal customer

plt.figure(figsize = (10,10))

plt.subplot(2,2,1)

sns.countplot(data = training_set, x="Customer Type" , hue="satisfaction", palette="cool")

plt.title("Histogram of Satisfaction Across Customer Type")

plt.legend()


plt.figure(figsize = (8, 5))

fligth_dist_hist = sns.distplot(training_set['Flight Distance'], color = 'blue')

plt.ylabel('Frequency', size = 8)

plt.xlabel('Flight Distance', size = 14)


plt.figure(figsize = (8, 5))

fligth_dist_hist = sns.distplot(training_set['Departure Delay in Minutes'], color = 'blue')

plt.ylabel('Density', size = 12)

plt.xlabel('Departure Delay in Minutes', size = 14)


plt.figure(figsize = (8, 5))

fligth_dist_hist = sns.distplot(training_set['Arrival Delay in Minutes'], color = 'blue')

plt.ylabel('Density', size = 12)

plt.xlabel('Arrival Delay in Minutes', size = 14)
```

```
# kdeplot method for visualizing the distribution of observations in a dataset,
plt.figure(figsize=(10,10))
plt.subplot(2,2,1)

sns.kdeplot(training_set.loc[training_set["satisfaction"]==1]["Age"],alpha=0.5,label="satisfaction")
sns.kdeplot(training_set.loc[training_set["satisfaction"]==0]["Age"],alpha=0.5,label="neutral or dissatisfied")
plt.title("Satisfaction vs Age")
plt.legend()

plt.figure(figsize=(8, 5))
age_satisfaction = sns.distplot(training_set[training_set['satisfaction']==1]['Age'],color='red')
plt.ylabel('Density',size=14)
plt.xlabel('Age',size=14)

def bar_plot(variable):
  var = training_set[variable]
  var_value= var.value_counts()

  plt.figure(figsize= (9,3))
  plt.bar(var_value.index, var_value.values)

  plt.xlabel("Passengers score")
  plt.ylabel("Frequency")
  plt.title(variable)
  plt.show()


  print("{}: \n {}".format(variable,var_value))



category1 = [ "Inflight wifi service", "Departure/Arrival time convenient", "Ease of Online booking", "Gate location", "Food and drink", "Online boarding", "Seat comfort", "Inflight entertainment", "On-board service", "Leg room service", "Baggage handling", "Checkin service", "Inflight service", "Cleanliness",]


for name in category1:
  bar_plot(name)


def bar_plot(variable):
  var = training_set[variable]
  var_value= var.value_counts()
  sns.countplot(data=training_set, x= training_set[variable] , hue="satisfaction", palette="cool")


  plt.xlabel("Passengers score")
  plt.ylabel("Frequency")
  plt.title(variable)
  plt.show()


  print("{}: \n {}".format(variable,var_value))
```

```
category1 = [ "Inflight wifi service", "Departure/Arrival time convenient", "Ease of Online
booking", "Gate location", "Food and drink", "Online boarding", "Seat comfort", "Inflight
entertainment", "On-board service", "Leg room service", "Baggage handling", "Checkin service",
"Inflight service", "Cleanliness", "Arrival Delay in Minutes", "Departure Delay in Minutes",
"Flight Distance"]


for name in category1:
  bar_plot(name)


#Plotting the correlation coefficients of all the Features
#training_set.corr() - Pearson Correlation Coefficients
plt.figure(figsize=(18,12))
sns.heatmap(training_set.corr(),vmin=-1, vmax=1, annot=True,cmap='PuOr')


plt.figure(figsize=(16,10))
# define the mask to set the values in the upper triangle to True
mask = np.triu(np.ones_like(dataset.corr(), dtype=np.bool))
heatmap = sns.heatmap(dataset.corr(), mask=mask, vmin=-1, vmax=1, annot=True,
cmap='PuOr')
heatmap.set_title('Triangle Correlation Heatmap', fontdict={'fontsize':18}, pad=16);
plt.figure(figsize=(8, 12))
heatmap = sns.heatmap(training_set.corr()[['satisfaction']].sort_values(by='satisfaction',
ascending=False), vmin=-1, vmax=1, annot=True, cmap='PuOr')
heatmap.set_title('Features Correlating with satisfaction', fontdict={'fontsize':18}, pad=16);


X_train.shape, y_train.shape
```

```
X_test.shape, y_test.shape


# **Build the Classification Algorithms**


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, plot_confusion_matrix
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
import sklearn.metrics as metrics
import time


def classAlg(model):
    t0=time.time()


    model.fit(X_train,y_train)
    prediction_test = model.predict(X_test)


    print('Classification Data Test')
    print(classification_report(y_test,prediction_test, digits=5))
    accuracy = accuracy_score(y_test, prediction_test)
    roc_auc = roc_auc_score(y_test, prediction_test)
    time_taken = time.time()-t0
    print("Accuracy = {}".format(accuracy))
    print("ROC Area under Curve = {}".format(roc_auc))
    print("Time taken = {}".format(time_taken))
    print("---------------------------------------------")
    print('True', y_test[0:21])
    print('Pred', prediction_test[0:21])
    print("---------------------------------------------")
    print("First argument is true values, second argument is predicted values")
    print(metrics.confusion_matrix(y_test, prediction_test))

    return model, accuracy, roc_auc, time_taken

def models_result(model, X_test, y_test):
    labels = model.predict(X_test)
    matrix = confusion_matrix(y_test, labels)
    ax = sns.heatmap(matrix.T, square=True, annot=True, fmt='d', cbar=False)
    ax.set(xlabel='predicted values',ylabel='true values')
    ax.set_xticklabels(['neutral/dissatisfied', 'satisfied'])
    ax.set_yticklabels(['neutral/dissatisfied', 'satisfied', ])

model_RF = RandomForestClassifier()
```

```
model_rf, accuracy_rf, roc_auc_rf, tt_rf= classAlg(model_RF)
models_result(model_RF, X_test, y_test)

plot_confusion_matrix(model_RF, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

model_LR = LogisticRegression()
model_lr, accuracy_lr, roc_auc_lr, tt_lr= classAlg(model_LR)
models_result(model_LR, X_test, y_test)
plot_confusion_matrix(model_LR, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

model_TREE = DecisionTreeClassifier()
model_dt, accuracy_dt, roc_auc_dt, tt_dt= classAlg(model_TREE)
models_result(model_TREE, X_test, y_test)

plot_confusion_matrix(model_TREE, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

model_GBM = GradientBoostingClassifier()
model_gb, accuracy_gb, roc_auc_gb, tt_gb= classAlg(model_GBM)
models_result(model_GBM, X_test, y_test)

plot_confusion_matrix(model_GBM, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

model_KNN = KNeighborsClassifier()
model_kn, accuracy_kn, roc_auc_kn, tt_kn= classAlg(model_KNN)
models_result(model_KNN, X_test, y_test)
```

```
plot_confusion_matrix(model_KNN, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

model_GNB = GaussianNB()
model_nb, accuracy_nb, roc_auc_nb, tt_nb= classAlg(model_GNB)
models_result(model_GNB, X_test, y_test)

plot_confusion_matrix(model_GNB, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

import lightgbm as lgb
model_LGB = lgb.LGBMClassifier()
model_lb, accuracy_lb, roc_auc_lb, tt_lb= classAlg(model_LGB)
models_result(model_LGB, X_test, y_test)

plot_confusion_matrix(model_LGB, X_test, y_test,cmap=plt.cm.Blues, normalize = 'all')

#partioning the independent and dependent data
X=training_set.drop('satisfaction',axis=1)
y=training_set['satisfaction']

#Univariate Feature Selection
from sklearn.feature_selection import mutual_info_classif
mic=mutual_info_classif(X,y)

mic=pd.Series(mic)
```

```
mic.index=X.columns

mic.sort_values(ascending=False)


roc_auc_scores = [roc_auc_rf, roc_auc_lr, roc_auc_dt, roc_auc_gb,  roc_auc_kn, roc_auc_nb, roc_auc_lb]

accuracy_score = [accuracy_rf, accuracy_lr, accuracy_dt, accuracy_gb,  accuracy_kn, accuracy_nb, accuracy_lb]

classifiers=['Random Forest','Logistic Regression','Decision Tree','Gradient Boosting','K-NN','Naive Bayes','LightGBM']

model_scores_auc = pd.DataFrame(roc_auc_scores, index=classifiers, columns=['AUC'])

model_scores_auc.sort_values(by='AUC',ascending=False).head(7)


model_scores_acc = pd.DataFrame(accuracy_score, index=classifiers, columns=['Accuracy'])

model_scores_acc.sort_values(by='Accuracy',ascending=False).head(7)

roc_auc_scores = [roc_auc_rf, roc_auc_lr, roc_auc_dt, roc_auc_gb,  roc_auc_kn, roc_auc_nb, roc_auc_lb]

tt = [tt_rf, tt_lr, tt_dt, tt_gb, tt_kn, tt_nb, tt_lb]

model_data = {'Model': ['Random Forest','Logistic Regression','Decision Tree','Gradient Boosting','K-NN','Naive Bayes','LightGBM'],

          'ROC_AUC': roc_auc_scores,

          'Time taken': tt}

data = pd.DataFrame(model_data)


fig, ax1 = plt.subplots(figsize=(14,8))

ax1.set_title('Model Comparison: Area under ROC Curve and Time taken for execution by Various Models', fontsize=13)

color = 'tab:blue'
```

```
ax1.set_xlabel('Model', fontsize=13)

ax1.set_ylabel('Time taken', fontsize=13, color=color)

ax2 = sns.barplot(x='Model', y='Time taken', data = data, palette='Blues_r')

ax1.tick_params(axis='y')

ax2 = ax1.twinx()

color = 'tab:orange'

ax2.set_ylabel('ROC_AUC', fontsize=13, color=color)

ax2 = sns.lineplot(x='Model', y='ROC_AUC', data = data, sort=False, color=color)

ax2.tick_params(axis='y', color=color)
```