Customer Satisfaction

Ahmed Ibrahim

Data definition:

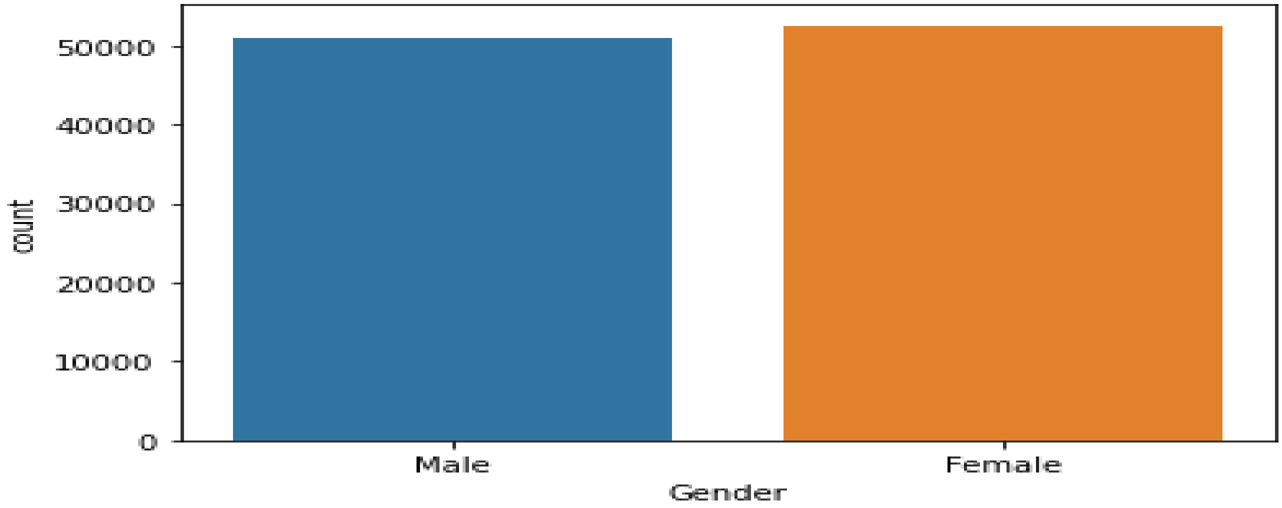
Airline customers satisfaction dataset consists of 1 csv file. This data contains the records of the flight information with 25 features for each flight and 103904 rows. So, it is needed first to list the features (Columns) to explore the data and for better understanding. The features can be split into two parts, Numerical features, and categorical features. Numerical features are the features that contain numbers as data. Categorical features that contain data divided into groups. The first two features are the ID and the index so it will be ignored. “Gender” whether male or female, “Customer type” whether loyal or disloyal, “type of travel” whether personal or business, “Class” whether eco plus, business or eco and the last categorical features is the “satisfaction” the most important because it is the target class for analysis and classification model. The numerical features are 'Age', 'Flight Distance', 'Inflight Wi-Fi 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', 'Check-in service', 'Inflight service', 'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes'.

STATISTICAL ANALYSIS:

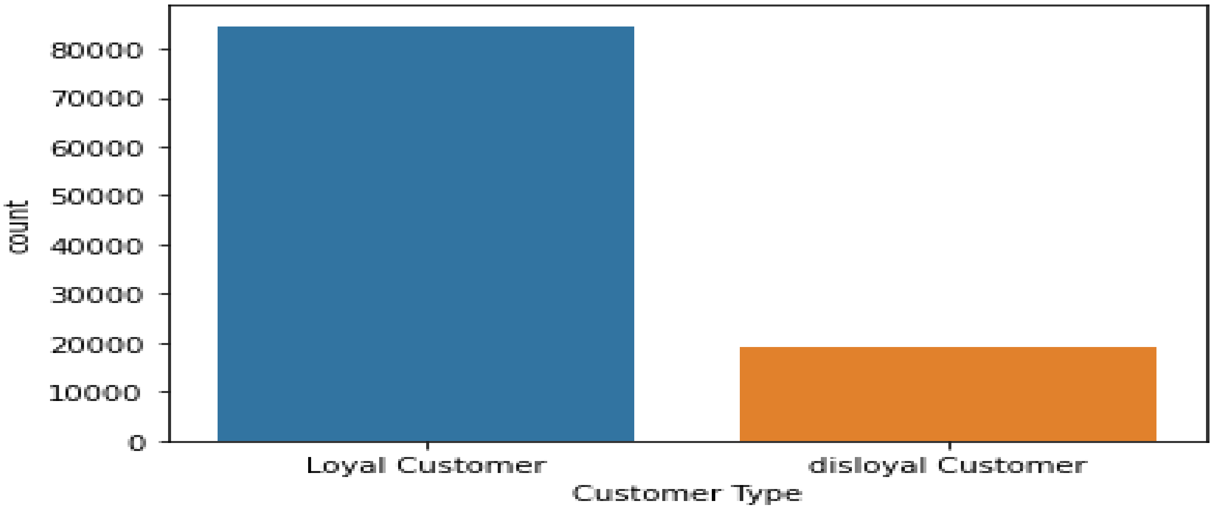
Female travelers outnumber male passengers by 52727. 50% of the flights in this data were not delayed because the median number of delays is 0. Flight delays average 15 minutes, with a standard variation of 38 minutes.

General Visualization:

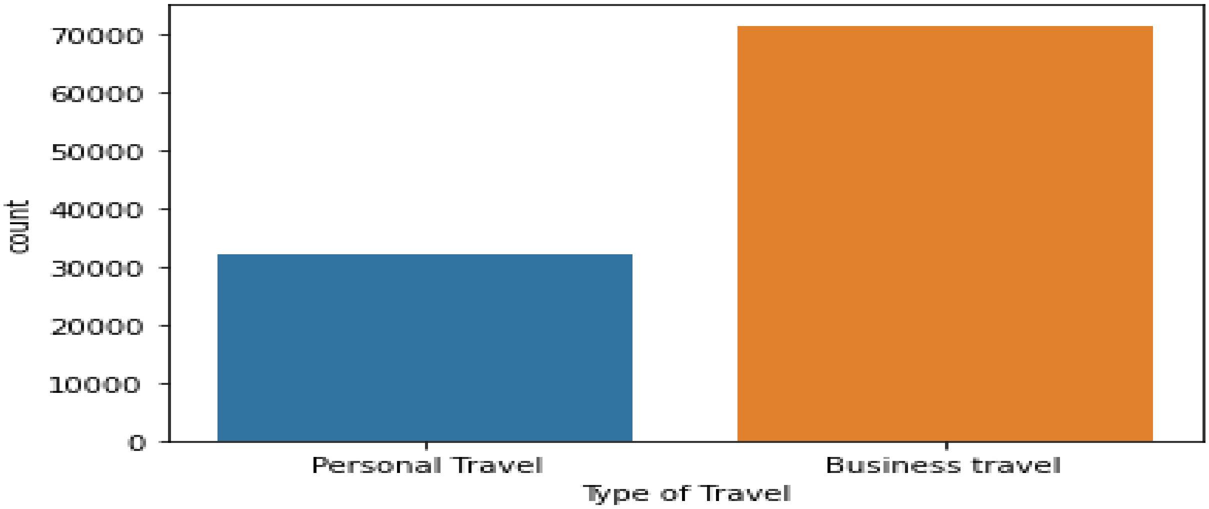
As shown in the plot below the number of travelers of female and male is almost the same.



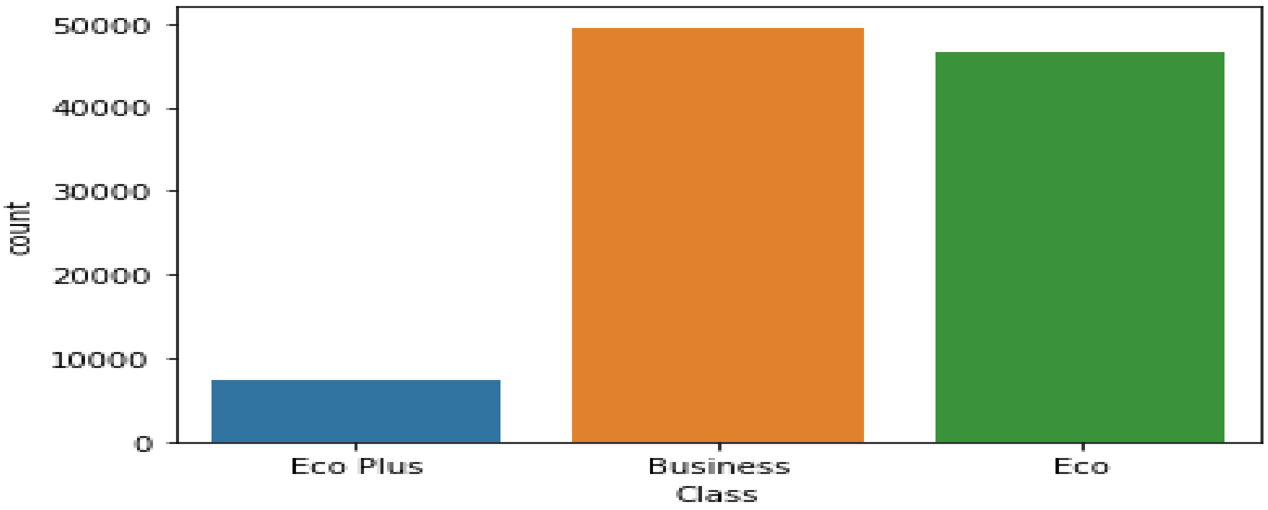
The number of loyal customers is clearly higher than disloyal customers.



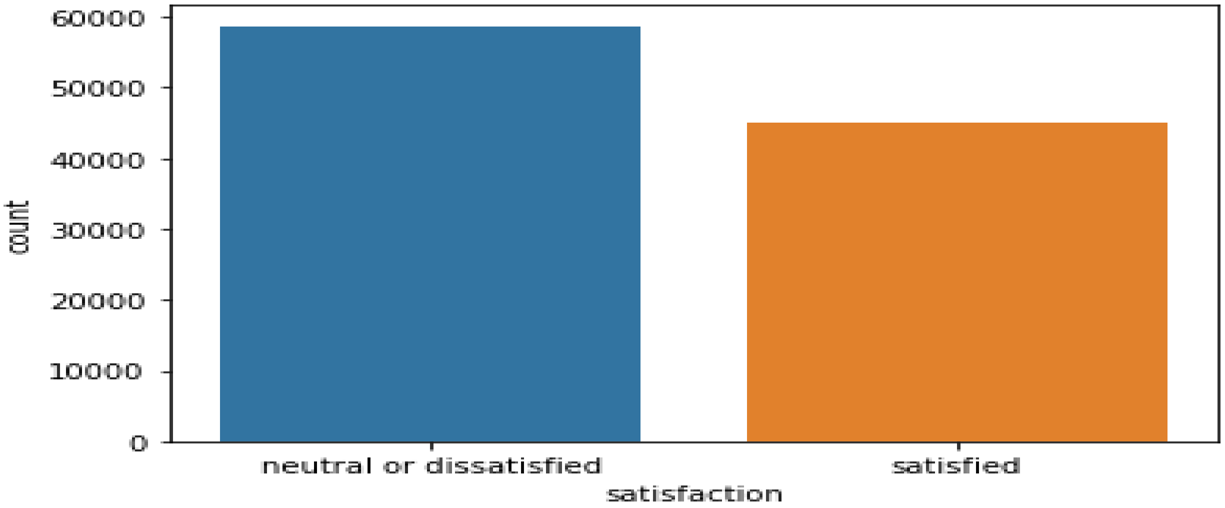
The number of business travels is much higher than personal travel.



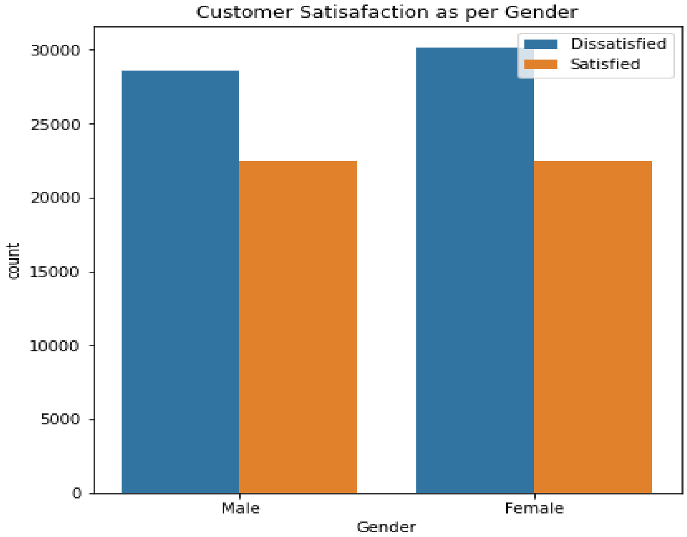
Business class and eco class is almost the same but the eco plus class is clearly lower than the other two classes.

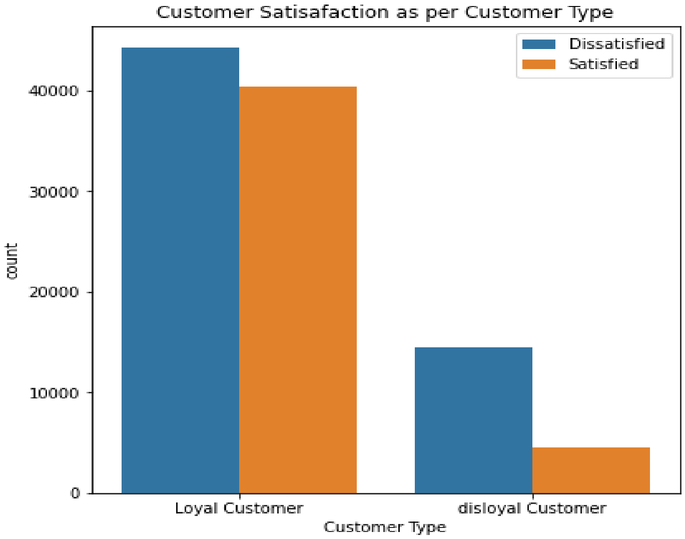


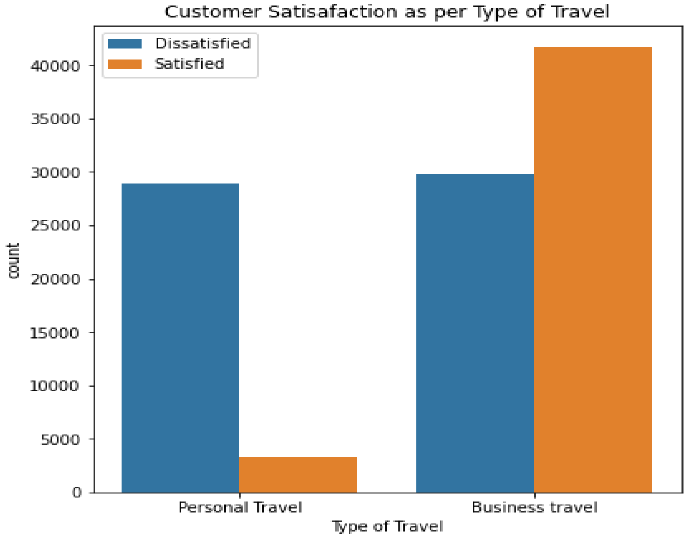
It is shown the neutral or dissatisfied class is little more than satisfied but **does not** consider as unbalanced data.

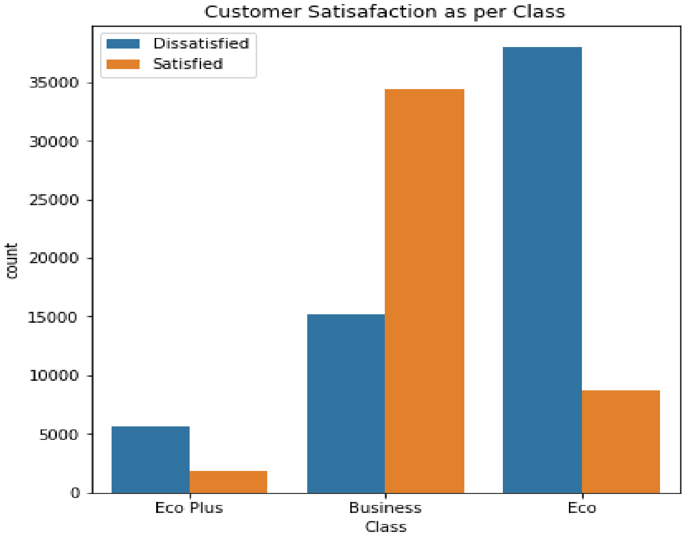


Extracting useful insights:





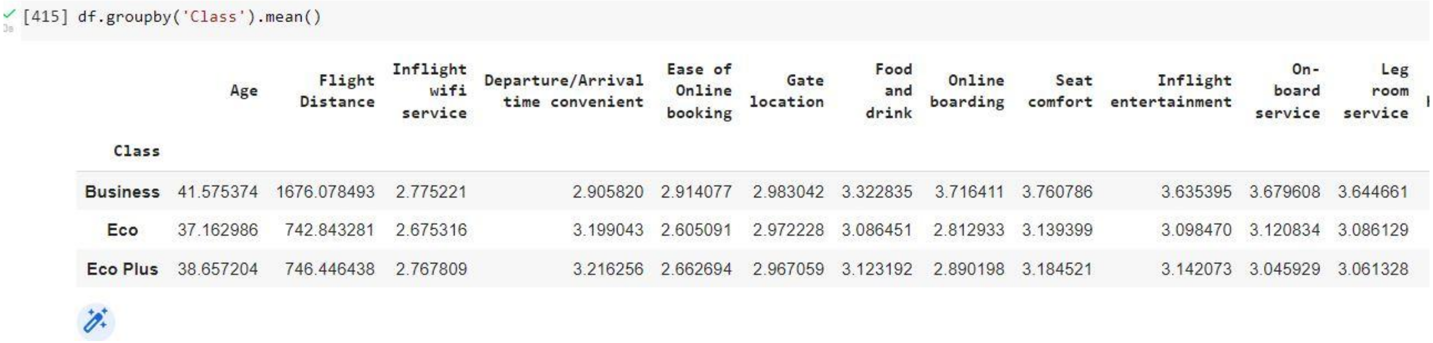




As shown in above plots, males and females are the same number of dissatisfied. Disloyal customers are more dissatisfied the loyal customers. Personel Travel customers are hugely dissatisfied and business travelers are very satisfied. The eco class customers are insanely dissatisfied, the business class customers are highly satisfied and a high number of customers in the eco plus class are dissatisfied.

Conclusion of insights:

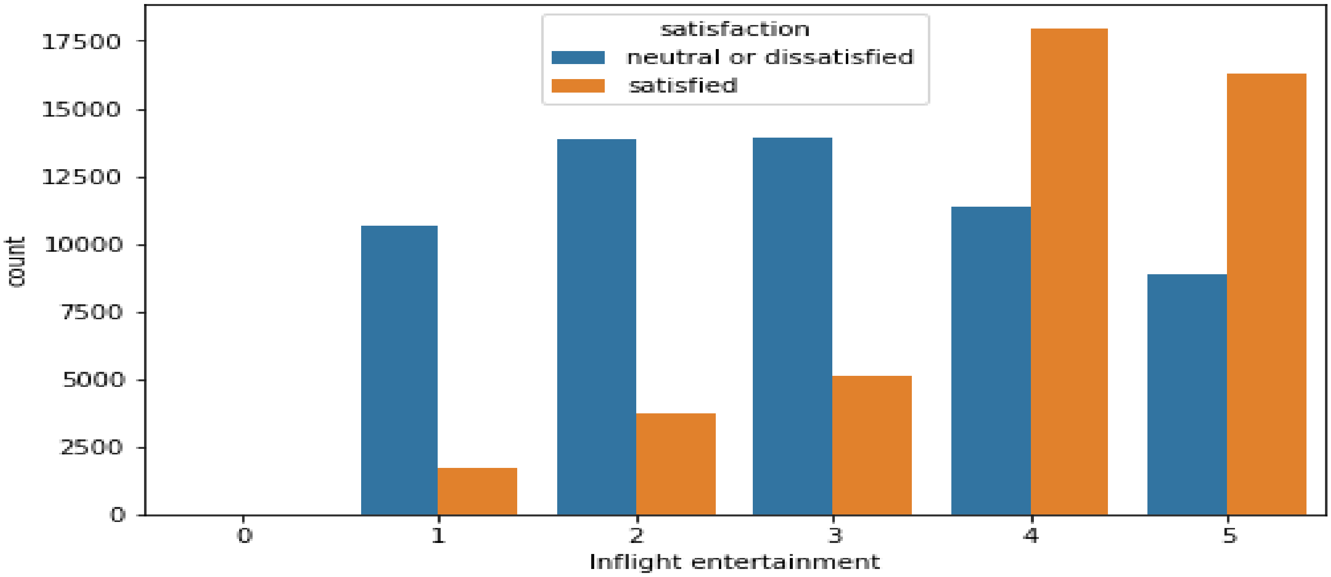
- Men and women appear to be equally worried about the same factors, hence gender does not appear to have a significant impact in satisfaction.
- Although this airline has a large number of loyal customers, dissatisfaction is prevalent regardless of loyalty. The airline will have to work hard to keep its loyal customers.
- Business travelers appear to be happier with their flights than personal Travellers.
- People in business class appear to be the most satisfied, while those in economy class appear to be the least satisfied.



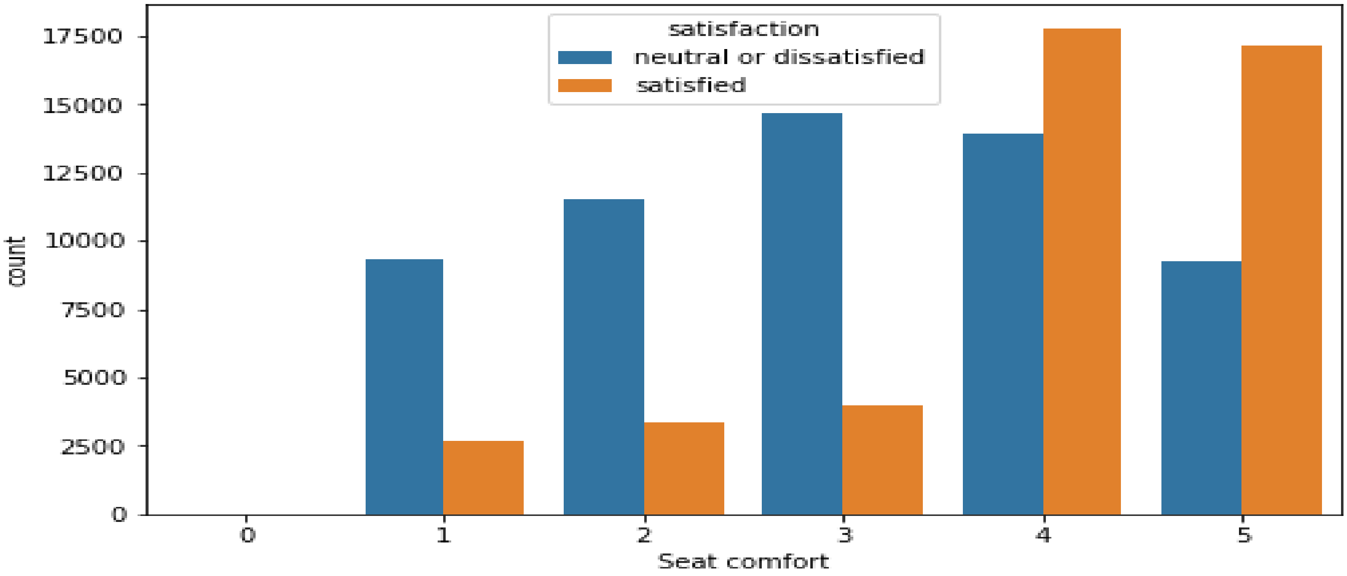
```
[415] df.groupby('Class').mean()
```

	Age	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
Class												
Business	41.575374	1676.078493	2.775221	2.905820	2.914077	2.983042	3.322835	3.716411	3.760786	3.635395	3.679608	3.644661
Eco	37.162986	742.843281	2.675316	3.199043	2.605091	2.972228	3.086451	2.812933	3.139399	3.098470	3.120834	3.086129
Eco Plus	38.657204	746.446438	2.767809	3.216256	2.662694	2.967059	3.123192	2.890198	3.184521	3.142073	3.045929	3.061328

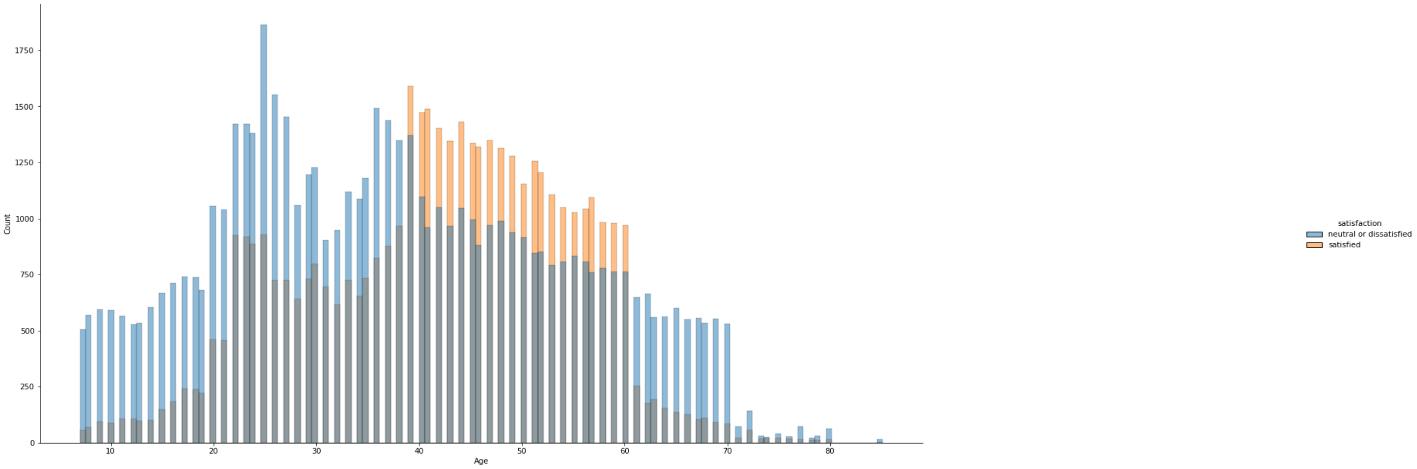
In comparison to Eco and Eco plus, business class customers have been given superior evaluations for all services supplied. As a result, the class of travel should play a significant role in overall satisfaction.



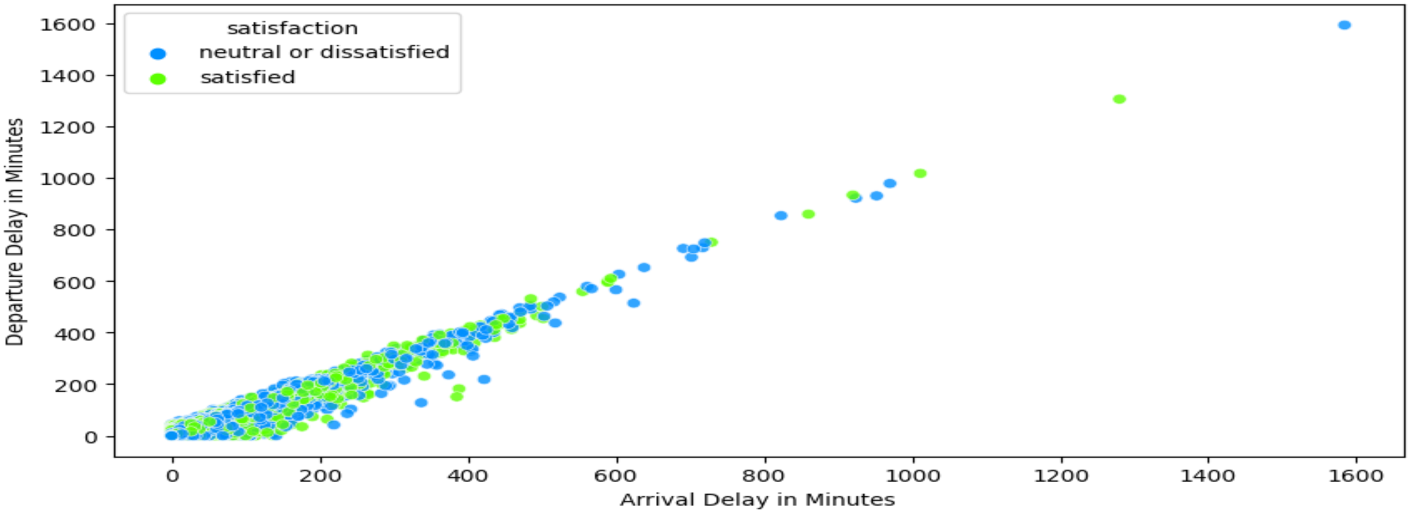
As shown, there is a direct correlation between in-flight entertainment and satisfaction.



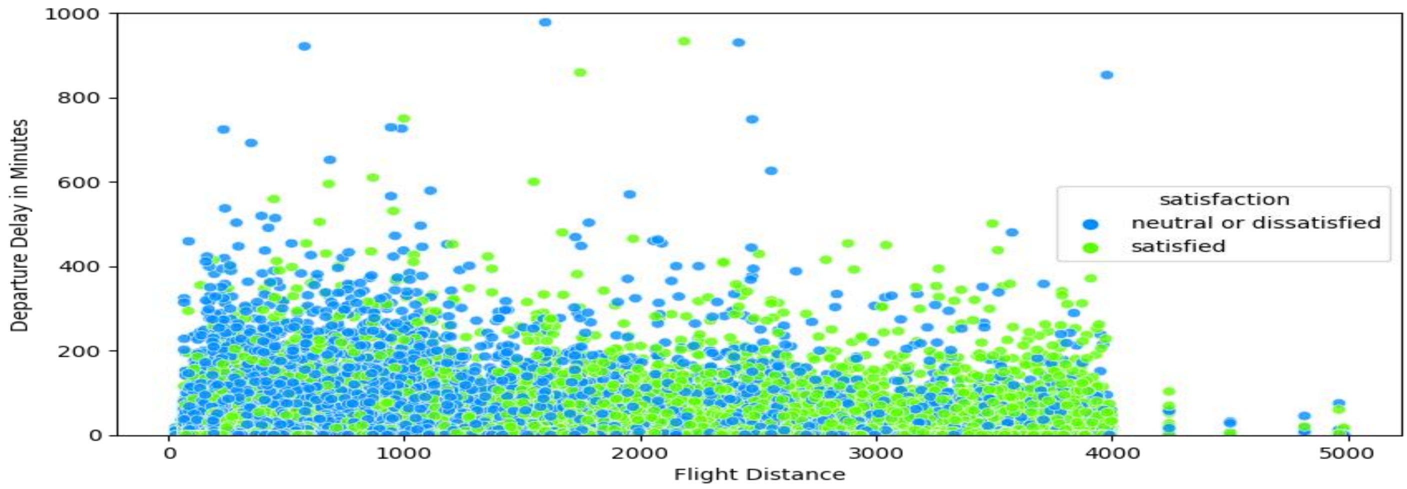
As shown, there is a direct correlation between seat comfort and satisfaction.



Travelers between the ages of 38 and 60 are more satisfied than customers of other ages.

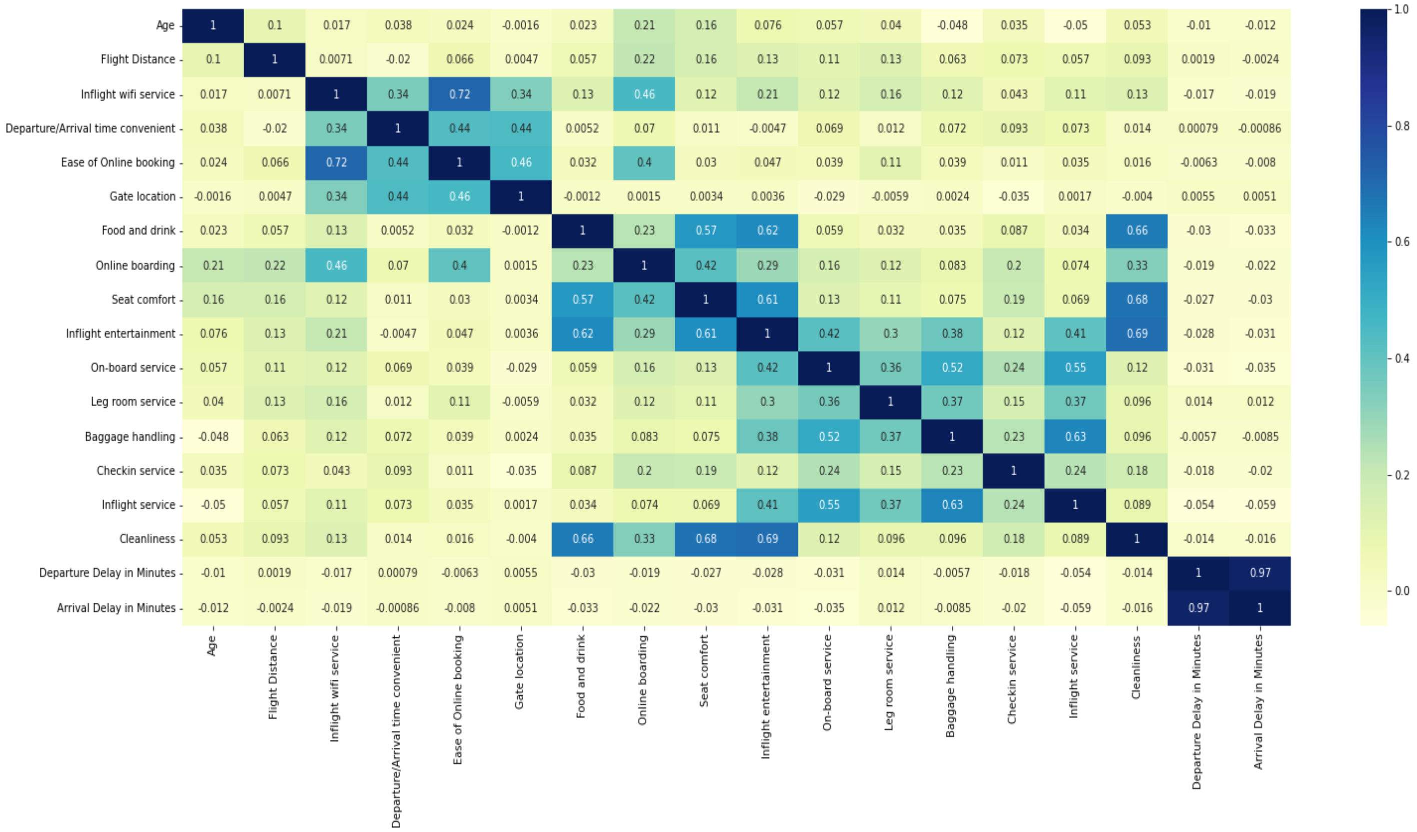


The arrival and departure delays follow a linear relationship.



The most essential point from this plot is that the greater the trip distance, the more passengers are willing to accept a short delay in departure. As a result, departure delays are less of an issue for long-distance flights; however, short-distance travelers do not appear to be concerned about departure delays.

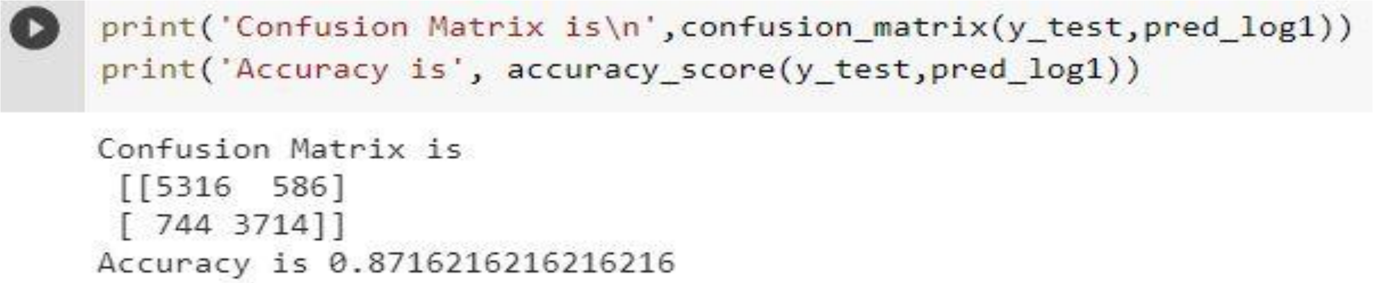
Feature selection:



Features like Flight Distance, Departure/Arrival time convenient, Gate location, Departure Delay in Minutes and Arrival Delay in Minutes have very low impact on customer satisfaction and the correlation is weak. So, it’s going to drop those features to decrease model complexity.

Model training:

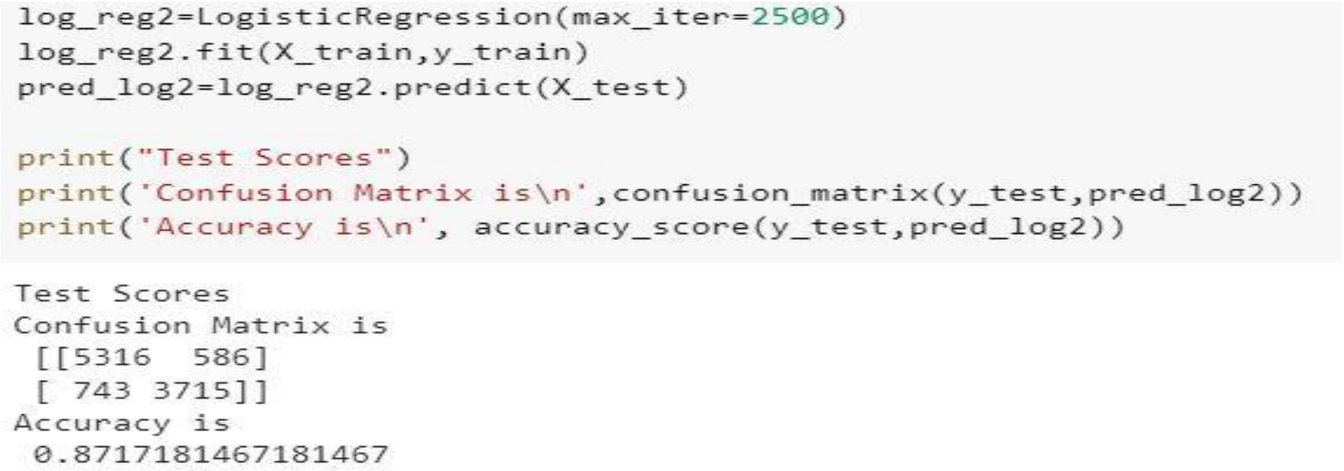
The first Model is Logistic regression and the model trained twice. The first train was using scaling method and the second train without any scaling method. The difference between them was very small and unnoticeable. But in confusion matrix the scaled data has slightly overfit, so it’s used the unscaled data.



```
print('Confusion Matrix is\n',confusion_matrix(y_test,pred_log1))
print('Accuracy is', accuracy_score(y_test,pred_log1))
```

```
Confusion Matrix is
 [[5316  586]
 [ 744 3714]]
Accuracy is 0.8716216216216216
```

Scaled



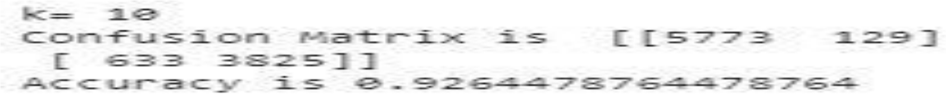
```
log_reg2=LogisticRegression(max_iter=2500)
log_reg2.fit(X_train,y_train)
pred_log2=log_reg2.predict(X_test)

print("Test Scores")
print('Confusion Matrix is\n',confusion_matrix(y_test,pred_log2))
print('Accuracy is\n', accuracy_score(y_test,pred_log2))
```

```
Test Scores
Confusion Matrix is
 [[5316  586]
 [ 743 3715]]
Accuracy is
 0.8717181467181467
```

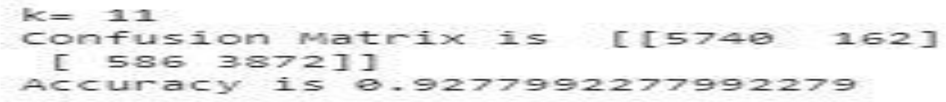
Unscaled

The second model was (KNN) K nearest neighbor. To identify the suitable number of K, there’s a loop iterate from 10 to 18 to choose the best value of K.

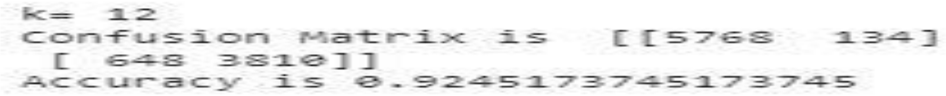
Logistic Regression



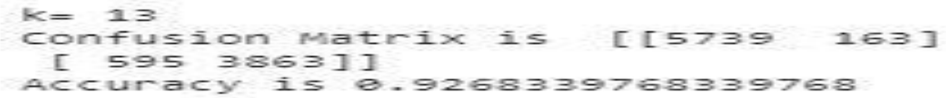
```
k= 10
Confusion Matrix is  [[5773  129]
 [ 633 3825]]
Accuracy is 0.9264478764478764
```



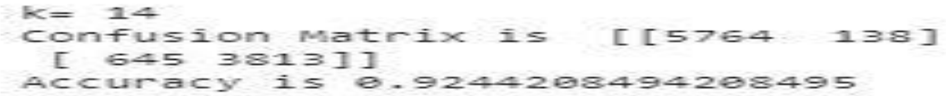
```
k= 11
Confusion Matrix is  [[5740  162]
 [ 586 3872]]
Accuracy is 0.9277992277992279
```



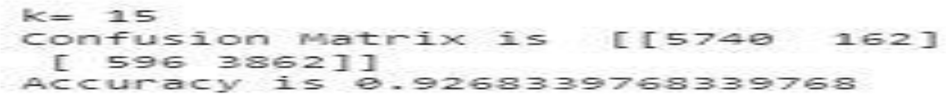
```
k= 12
Confusion Matrix is  [[5768  134]
 [ 648 3810]]
Accuracy is 0.9245173745173745
```



```
k= 13
Confusion Matrix is  [[5739  163]
 [ 595 3863]]
Accuracy is 0.9268339768339768
```



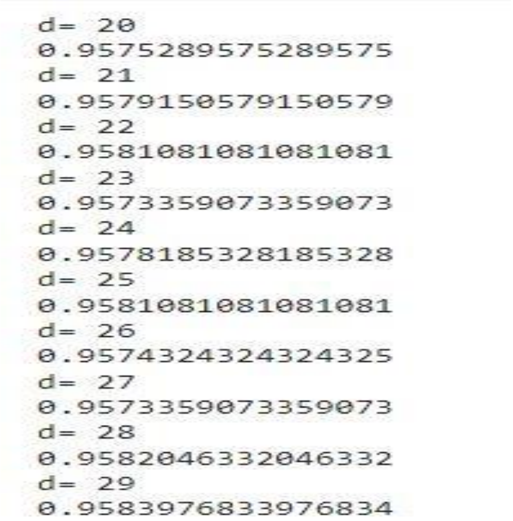
```
k= 14
Confusion Matrix is  [[5764  138]
 [ 645 3813]]
Accuracy is 0.9244208494208495
```



```
k= 15
Confusion Matrix is  [[5740  162]
 [ 596 3862]]
Accuracy is 0.9268339768339768
```

As shown below the best K value is 11.

The third model is the Decision tree classifier. It uses a loop to set the best depth for the tree from to 20 to 30.

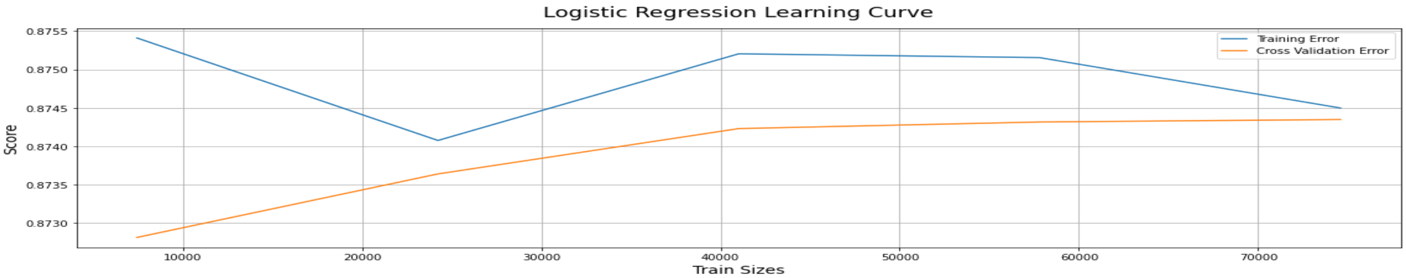


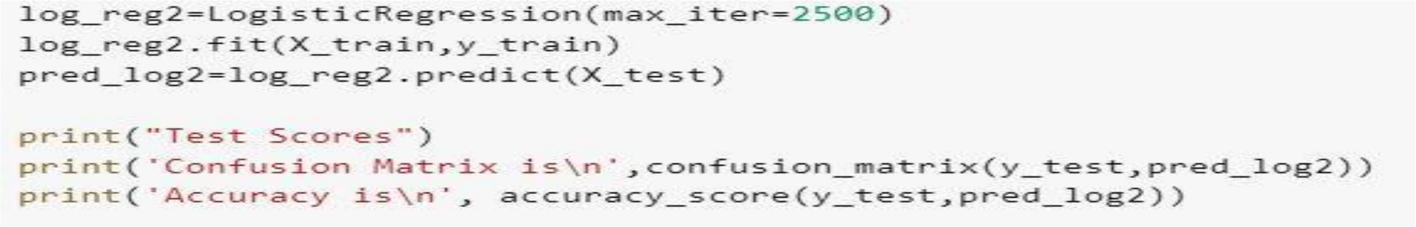
```
d= 20
0.9575289575289575
d= 21
0.9579150579150579
d= 22
0.9581081081081081
d= 23
0.9573359073359073
d= 24
0.9578185328185328
d= 25
0.9581081081081081
d= 26
0.9574324324324325
d= 27
0.9573359073359073
d= 28
0.9582046332046332
d= 29
0.9583976833976834
```

As shown the best depth is 25.

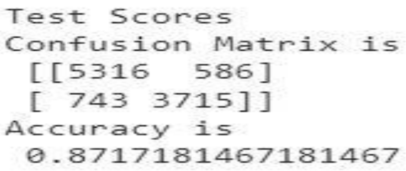
Model Evaluation:

Logistic Regression



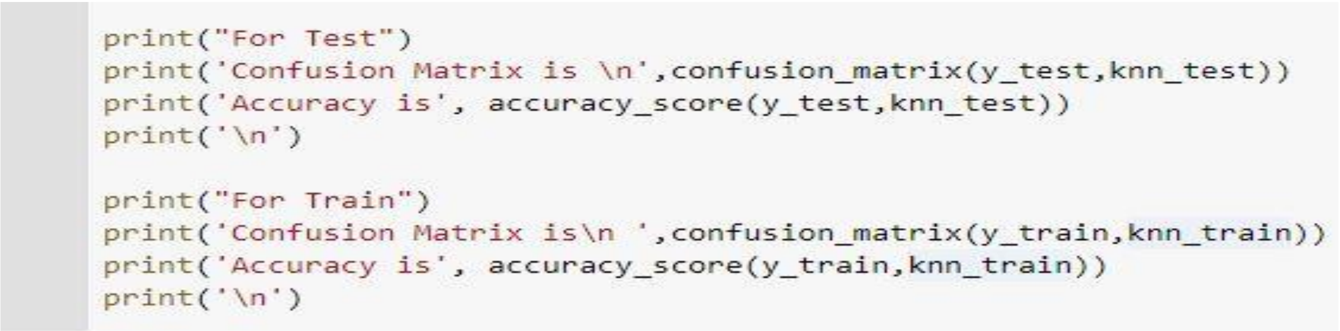


```
log_reg2=LogisticRegression(max_iter=2500)
log_reg2.fit(X_train,y_train)
pred_log2=log_reg2.predict(X_test)

print("Test Scores")
print('Confusion Matrix is\n',confusion_matrix(y_test,pred_log2))
print('Accuracy is\n', accuracy_score(y_test,pred_log2))
```

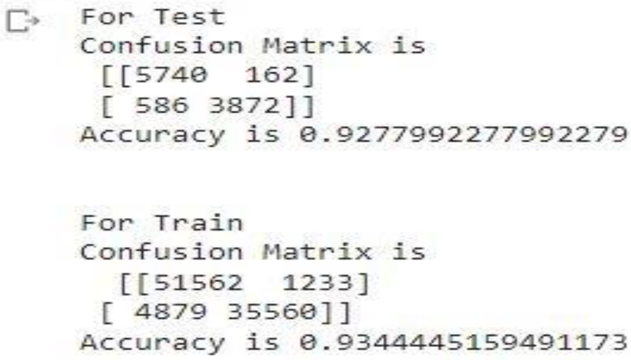


```
Test Scores
Confusion Matrix is
 [[5316  586]
 [ 743 3715]]
Accuracy is
 0.8717181467181467
```

KNN

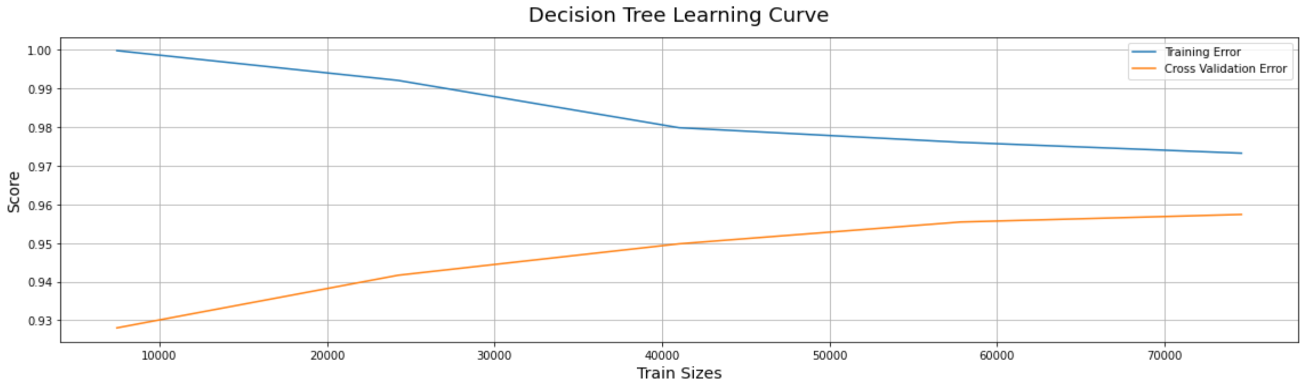


```
print("For Test")
print('Confusion Matrix is \n',confusion_matrix(y_test,knn_test))
print('Accuracy is', accuracy_score(y_test,knn_test))
print('\n')

print("For Train")
print('Confusion Matrix is\n ',confusion_matrix(y_train,knn_train))
print('Accuracy is', accuracy_score(y_train,knn_train))
print('\n')
```

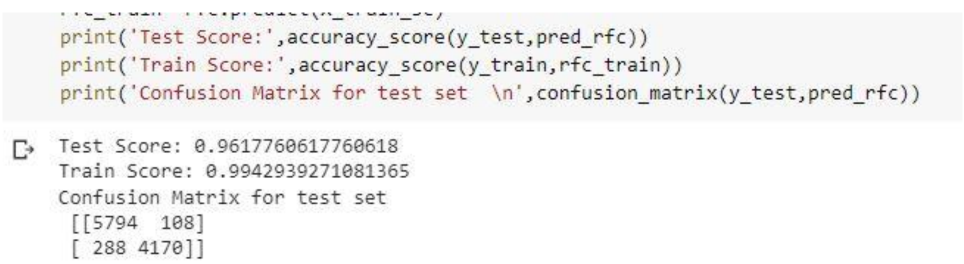


```
For Test
Confusion Matrix is
 [[5740  162]
 [ 586 3872]]
Accuracy is 0.9277992277992279


For Train
Confusion Matrix is
  [[51562  1233]
 [ 4879 35560]]
Accuracy is 0.9344445159491173
```

Decision Tree



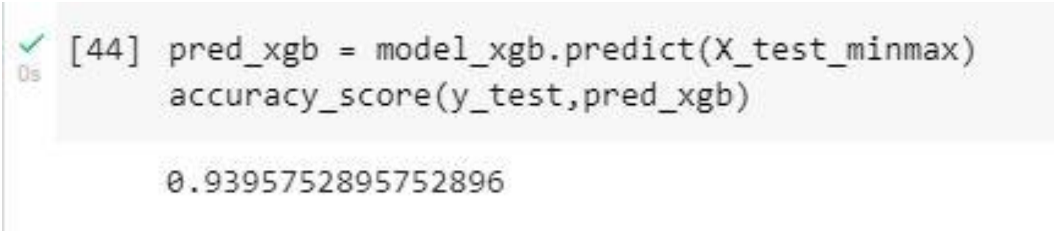
Random Forest



```
rfc_train = rfc.predict(X_train_sc)
print('Test Score:',accuracy_score(y_test,pred_rfc))
print('Train Score:',accuracy_score(y_train,rfc_train))
print('Confusion Matrix for test set  \n',confusion_matrix(y_test,pred_rfc))
```

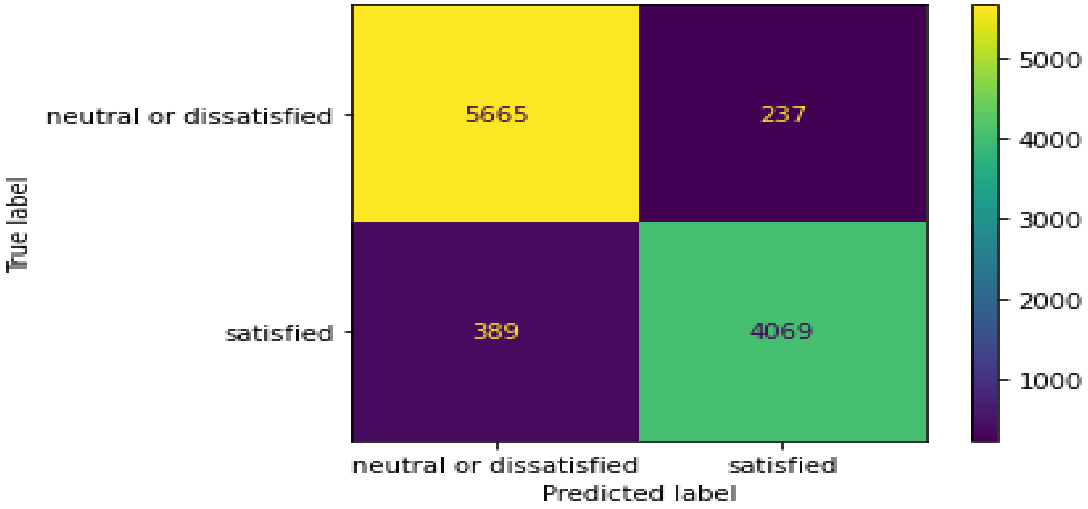
```
Test Score: 0.9617760617760618
Train Score: 0.9942939271081365
Confusion Matrix for test set
 [[5794  108]
 [ 288 4170]]
```

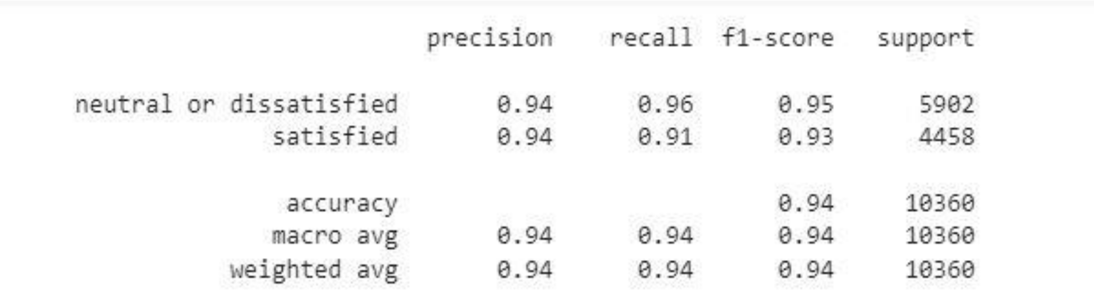
XGBoost



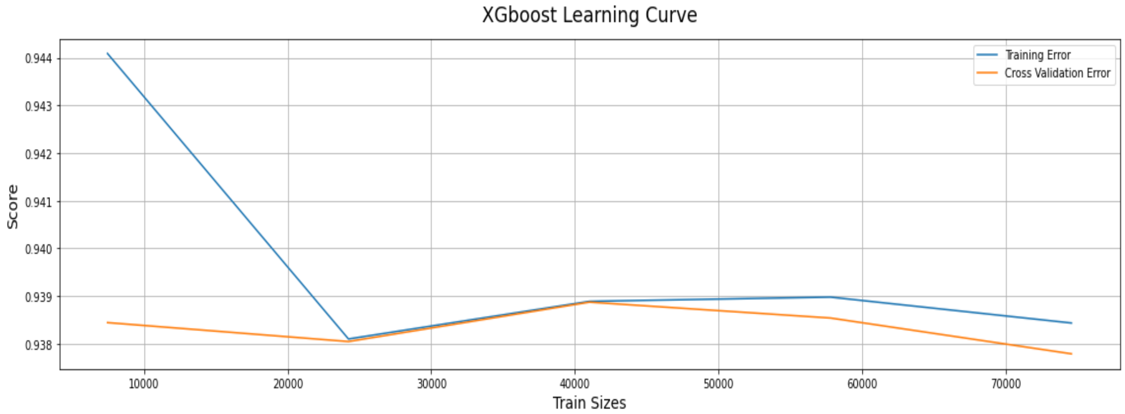
```
[44] pred_xgb = model_xgb.predict(X_test_minmax)
     accuracy_score(y_test,pred_xgb)
```

```
0.9395752895752896
```





```
                         precision    recall  f1-score   support

neutral or dissatisfied       0.94      0.96      0.95      5902
              satisfied       0.94      0.91      0.93      4458

               accuracy                           0.94     10360
              macro avg       0.94      0.94      0.94     10360
           weighted avg       0.94      0.94      0.94     10360
```



The best model is the XGBoost