Analyzing the Satisfaction of American Airline Customers

Dina Haji Zeynaly Biooky

University of Waterloo, Canada

dhajizey@uwaterloo.ca

	Content	
Abstract		**1**
Introduction		1
Related work		2
Data		3
	Class Variable	3
	Fact Features	3
	Respond Features	4
	Flight Features	5
Results		6
	Preprocessing	6
	Classification	7
	Clustering	9
	Association	9
Conclusion		10
Reference		11
Appendix		12

Abstract

Customer satisfaction is a critical issue for all kinds of businesses. Similarly, unsatisfied or neural passengers would cause fewer flights and less revenue for airlines. What improves customer satisfaction is an excellent experience at every travel. Cleanliness, on board services, on-time flights, in-flight entertainment, more (and better) snacks, and more legroom have always been obvious contributors to a good experience and loyalty. Besides all these factors, some other contributors such as Inflight WIFI, ease of online booking, online support have played a role in passenger satisfaction over past years.

Using a real dataset, we demonstrate some factors have a major effect on customer satisfaction. We investigate the relations of these factors together and with satisfaction. Some of these factors are not a strong source of satisfaction as they might seem to be at first look. We found out Decision Tree is the best model of classification of our dataset. We also found that some of our features are associated, and some others can be clustered in meaningful groups.

Introduction

In this article we will study most of the important factors in customer satisfaction and some important information about each of those customers from an American Airline. This dataset includes 129880 records from different passengers and 24 different columns. We reduced and grouped these 24 columns to 21 independent features and 1 class variable. We do not eliminate any of the records and the reasons of this will be discussed later.

In order to mention and check out our variables easily, we consider them as 4 different groups based on their similarity in types and roles in our study.

Class Variable	• Satisfaction	Categorical	Two levels: Satisfied - Neural or dissatisfied
Facts Features	• Gender	Categorical	Two levels: Female – Male
	• Customer Type		Two levels: Royal – Disloyal
	• Travel Type		Two levels: Business – Personal
	• Class Type		Two levels: Business - Eco or Eco Plus
	• Age	Numerical	Ranges from 7 – 85 Years
Flight Features	• Flight Distance	Numerical	Ranges from 50 – 6951 Miles
	• Real Delay		Ranges from 0 – 180 Minutes
Respond Features	• Seat comfort • Departure/Arrival time convenient • Food and drink • Gate location • Inflight WIFI service • Inflight entertainment • Online support • Ease of Online booking • On-board service • Leg room service • Baggage handling • Check in service • Cleanliness • Online boarding	Ordinal	Six Levels: Ranges from 0-5

In this dataset we will first investigate the distribution of all features and their relations with our class variable, then we will use classification models to predict our class variables based on features. Later we will do these classifications more specifically on some different subsets. We also plan to do clustering on Flight Features. We will finally try to find whether any association rule exists among Respond Features or not.

Related work

Hansemark and Albinson (2004) defined Customer Satisfaction as an overall customer attitude towards a service provider, or an emotional reaction to the difference between what customers anticipate and what they receive regarding the fulfillment of some needs, goals or desire.

There is no universal list of features to determine customer satisfaction. Customer satisfaction may be measured in different industries differently. Even in same industry every time different factors might be considered.

J.D. Power 2019 North America Airline Satisfaction Study found out the reservation and check-in experiences are the most satisfying portions of the airline experience, while In-flight services, such as seatback entertainment, food service and Wi-Fi continue to be the lowest-ranked part of the air traveler experience. This conclusion was based on responses of 5,966 passengers who flew on a major North American airline between March 2018 and March 2019.

Another literature review performed by Hongwei and Yahua (2016) done on four major China's airline market discusses the impacts of the service quality on customer satisfaction. This article also tries to find the linkage of customer satisfaction with brand loyalty, which is out of scope of our investigations. This article uses In-flight entertainment, In-flight food and drinks, Comfort of aircraft seat, and many other features thar are used in this project as factors of satisfaction. Findings of this study show that most of service quality features, except, Price Ticket and Gender, have an effect on customer satisfaction but not customer loyalty.

(ORSEA) INTERNATIONAL CONFERENCE (2012) indicated that pre-flight, in-flight and post-flight services had a significant effect on passenger satisfaction. In addition to that, passenger satisfaction as a mediating variable also had a significant effect on passenger loyalty. The study results imply that airline marketers should develop various strategies to improve service quality for example meeting passengers' desired service levels, improving the quality of in-flight meals, solving service problems effectively, developing convenient reservation and ticketing systems, making convenient schedules for passengers and reducing the effect of service failures as these directly affect passenger satisfaction and loyalty.

Percentage of on-time arrival, passengers denied boarding, mishandled baggage and customer complaints are examples of customer satisfaction factors in another study, Service Quality and Customer Satisfaction in the Airline Industry (2013). This study measures these factors at two levels Expected service and Perceived service, and compares customer satisfaction of different airlines: Hawaiian, Alaska, US, Delta, etc. The finding shows Industry performance for all four measurements was better in 2011 compared to its four previous years.

Norazah MohdSuki (2014) in his study aimed to examine the effects of aspects of airline service quality, such as airline tangibles, terminal tangibles, and empathy on levels of customer satisfaction. This airline survey was conducted among the population of the Federal Territory of Labuan, Malaysia. This study finds out that customer satisfaction is widely influenced by empathy, which is why flight punctuality and good

transportation links between city venues and airports are prioritized by providers. This finding is in contrast with our findings that Arrival/Departure Delay does not effect satisfaction significantly.

There are many other studies that work on customer satisfaction. Some of them are specifically on airline industry. We did not find any article that uses this dataset especially.

Data

This is an online dataset and is collected through surveys among passengers of American Airline in 2015. American Airline is founded on April 15, 1926; 94 years ago (earliest predecessor airline as American Airways, Inc.). It is the world's largest airline when measured by fleet size, scheduled passengers carried, and revenue passenger mile. American, together with its regional partners, operates an extensive international and domestic network with almost 6,800 flights per day to nearly 350 destinations in more than 50 countries.

As mentioned earlier, this study contains 129880 records of different passengers and 21 finalized independent features and 1 class variable. For the sake of simplicity, we grouped our variables in 4 different categories.

- **Class Variable:**

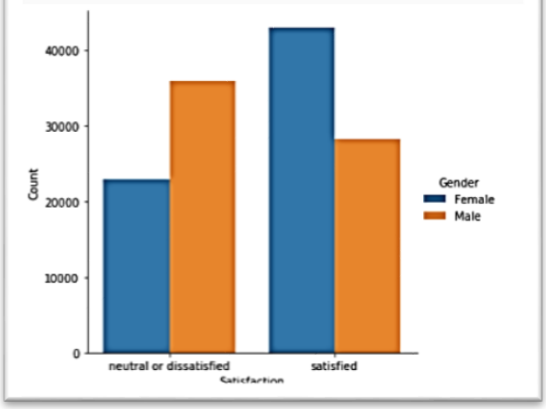
Satisfaction is our class variable. It has two categories 'Satisfied' with 71087 instances (about %55) and 'Neutral or dissatisfied' with 58793 instances (about %45). As we can see our class variable is equally distributed. This can augment our results since we have enough information from each category.

- **Facts Features:**

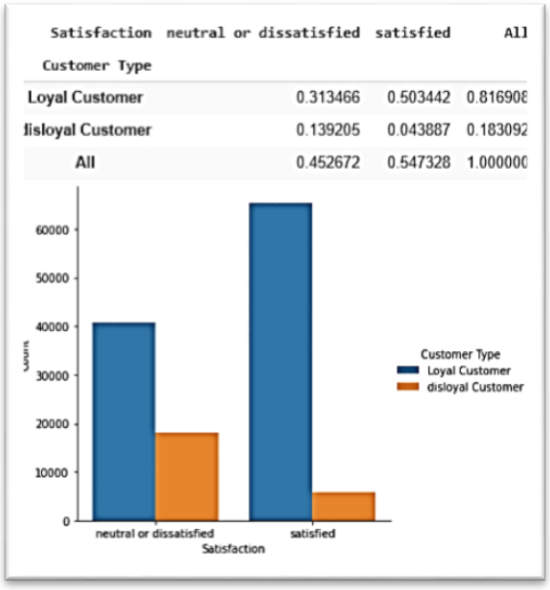
Fact features are some basic information about each passenger. Like Satisfaction, both **Gender** and **Class Type** are equally distributed; however, Class Type originally had three categories (Eco/ Eco-Plus/Business), and we merged eco and eco-plus. These two classes have fairly similar passenger types and locations in an airplane. Furthermore, Eco-plus passengers contain only %7 of data and most of superiority reasons of this class is already reflected in questions such as Seat comfort, Leg room service, etc. Therefore, if the comfort of this class has an effect on satisfaction, this effect is already captured by other parts of survey.

We have more satisfied female (%33 out of %51) and more not satisfied male (%27 out of %49). This, besides the fact that generality is equally distributed, shows that gender might have an effect on customer satisfaction and female might get satisfied more easily in similar condition. We also have more satisfied passenger in business Class (%34 out of %48) and more not satisfied passenger in eco or eco-plus class (%31 out of %52). Passenger seat classes might also have an effect on satisfaction, too. This completely makes sense since more services are provided in business part.

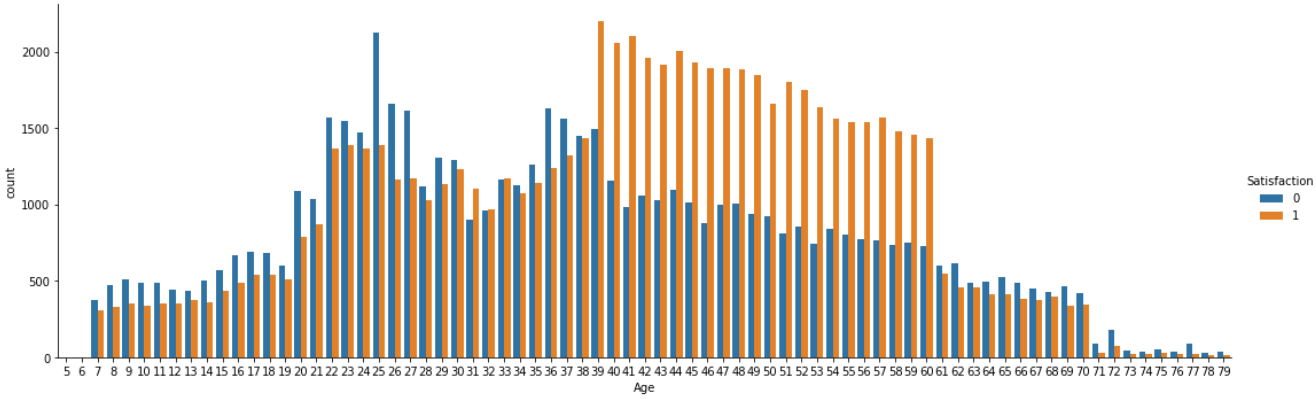
Satisfaction	neutral or dissatisfied	satisfied	All
Gender			
Female	0.176863	0.330520	0.507384
Male	0.275808	0.216808	0.492616
All	0.452672	0.547328	1.000000



Unlike other categorical variables, **Customer Type** and **Travel Type** are not equally distributed. We have %82 Loyal customers out of them %50 is satisfied. %69 of customers has business purposes for traveling out of them %40 is satisfied. These figures (%50 and %40) can not cause us to jump to the conclusion that Customer Type and Travel Type might have an effect on satisfaction since as we mentioned these variables are not equally distributed, the number of disloyal customers and personal travelers are so low, %18 and %31 of whole records respectively.

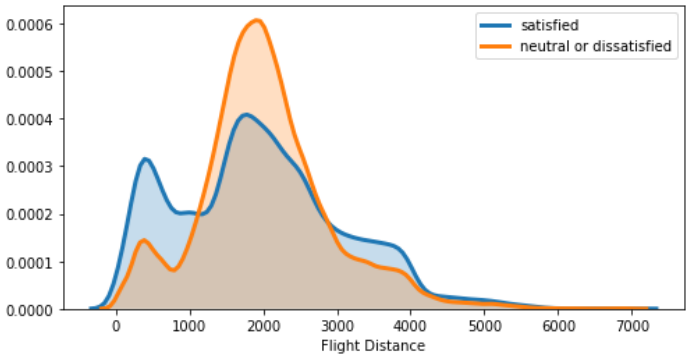


Age is the last fact feature and ranges from 7 to 85 years old. This variable is normally distributed with 39.5 years old mean and 15 years old standard deviation. As we can see in the following graph the number of satisfied customers is twice of that of not satisfied customers between those who aged 40 to 60 years old. Other than this period, satisfaction is distributed fairly equally among customers. The effect of age on satisfaction needs more investigation and graphing is not enough.



- **Flight Features:**

Flight Distance is fairly skewed to the right, ranging from 50 to 6951 miles with 1981 and 1027 mean and standard deviation respectively. Although this variable is not normally distributed, its median and mean are fairly equal (median=1925) and just about %0.28 of its observations are above of 3 standard deviation. We did not consider them as outliers since this airline has flights to 50 countries all around word and reaching these distances are not out of mind. As we can see in the following graph, most of not



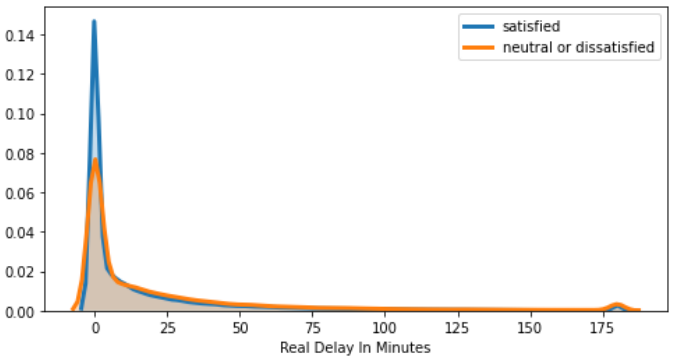
satisfied passengers experienced around 2000 miles travel (fairly equal to mean). The more orange graph skewed to left, the deeper effect distances will have on dissatisfactions. In our experiment this skewedness is not so much drastic. It also seems that Flight Distance is built from 2 fairly normal distributions. The graph its is not normal, though.

In our original datasets, we had 2 highly correlated variables (with 0.97 correlation); Departure Delay in Minutes and Arrival Delay in Minutes. This high level of correlation will cause multicollinearity. We believed that delay is reported on either one of the columns or both columns. Therefore, we merge the value of these two variables in a single variable, Real Delay, by following formula:

Real Delay = Arrival Delay in Minutes + | Arrival Delay in Minutes - Departure Delay in Minutes|

Before this merger, we also replace 393 missed values of Arrival Delay in Minutes (%0.3 of all records) with zero. We believed that because of high correlation of Departure Delay in Minutes with Arrival Delay in Minutes, and Real Delay's formula, the effect of missing values is acquirable. This feature is the only feature with missing values in our dataset.

Real Delay is completely skewed to the right. This variable ranged originally from 0 to 1592 minutes with %75 of records less than 22 minutes. The diagram and description table showed that it might contain lots of outliers, so we checked more specifically and found that about %2.12 of observations are just above +3 standard deviation. This amount is 42 times greater than the standard amount %0.05. We did not have enough reasons to eliminate them (it is logical and common that a flight has even 26.5 hours delay), so we just replaced them with 180 minutes as threshold. Finally, as we can see in following graph, most of satisfied passengers experience no delay in their flight; however, we still have many not satisfied passengers with zero delay.



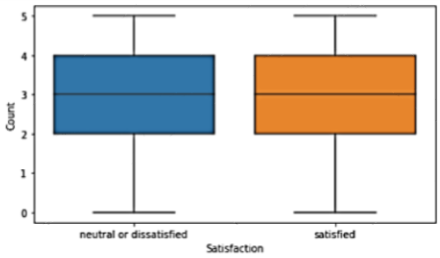
- **Respond Features:**

Seat comfort, Departure/Arrival time convenient, Food and drink, Gate location, Inflight WIFI service, Inflight entertainment, Online support, Ease of Online booking, On-board service, Leg room service, Baggage handling, Check in service, Cleanliness, and **Online boarding** reflect the opinion of each passenger about different properties of U.S Airline. These variables are ordinal and range from 0 (worst service) to 5 (best service). Some of these variables are correlated to each other, but this correlation is not that high to cause problems (from -0.0006 to 0.72). Furthermore, these relations are logical. For instance, an airline that pay a lot of attention to Inflight WIFI service will definitely consider Ease of Online booking as an important factor, so passengers would rank both of them either high or low. On the other hand, a geek passenger might rate Inflight WIFI service and Ease of Online booking similarly low, while a normal passenger's perception from the same services might be high.

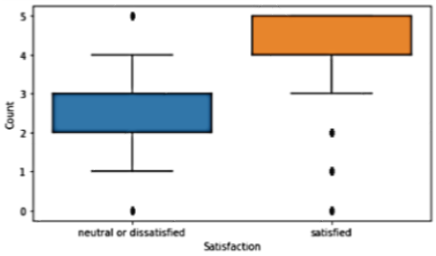
The means of Respond Features are between 2.8 and 3.7 and their medians are 3 or 4. This shows that most of options are chosen by customers, and our results are trustable. 4 and 0 are the most and the least checked options respectively.

Based on relational graphs and crosstab tables we found out that Departure/Arrival time convenient, Food and drink, and Gate location's values have fairly equal behavior for satisfied and not satisfied customers at different levels. For instance, %11.5, %12, %10, %9, %9, and %5 of satisfied customers chose 5 to 0 for Departure/Arrival time convenient respectively. These figures for not satisfied customers are only 1-2 percent lower at %9, %11, %8, %8, %7, %7, and %3 respectively. This shows that these three variables might not have a huge effect on our class variable (Satisfaction).

Satisfaction	neutral or dissatisfied	satisfied	All
Departure/Arrival time convenient			
0	0.023506	0.027803	0.051309
1	0.066346	0.094018	0.160363
2	0.080705	0.094795	0.175500
3	0.082214	0.096289	0.178503
4	0.108331	0.119518	0.227849
5	0.091569	0.114906	0.206475
All	0.452672	0.547328	1.000000



Satisfaction	neutral or dissatisfied	satisfied	All
Inflight entertainment			
0	0.007792	0.015137	0.022929
1	0.071797	0.019125	0.090922
2	0.122513	0.025185	0.147698
3	0.149253	0.037073	0.186326
4	0.090306	0.232137	0.322444
5	0.011010	0.218671	0.229681
All	0.452672	0.547328	1.000000



On the other hand, the number of passengers that ranked the remaing respond features at 4 or 5 is much higher in satisfied customers than in not satisfied customers. For instance, %22 of satisfied customers ranked Inflight Entertainment at 5, while only %1 of not satisfied customers chose 5 for this feature. This, besides the fact that satisfaction is equally distributed, suggests that these remaining Respond Features might have a huge effect on satisfaction.

Results

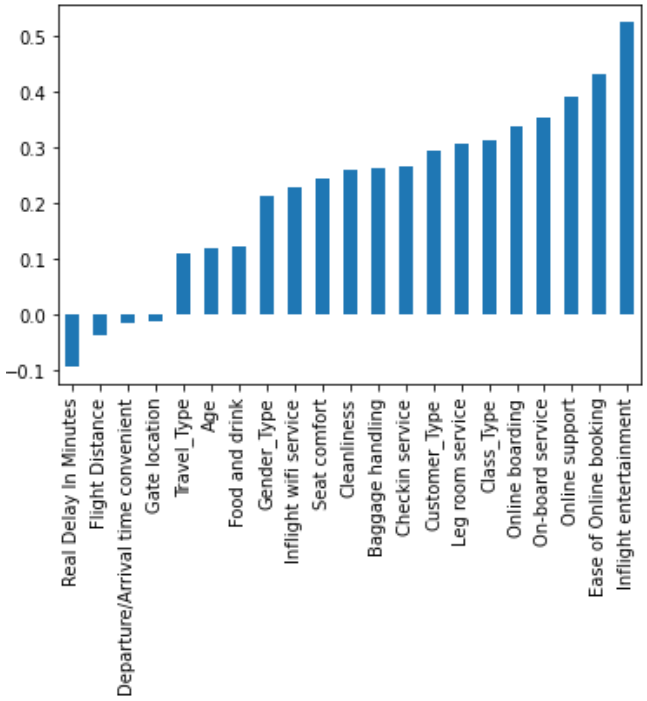
In this study we will try to classify class variable, Satisfaction, based on several algorithms and feature sets. We want to compare different classification models based on metric results and chosen subsets. Then, we will do clustering on Flight features. Finally, we will do association rule mining on Respond features to find out whether there is a relation among high ranked features or not.

- **Preprocessing:**

Before doing any supervised or unsupervised learning tasks, we transformed all categorical variables to numeric ones. 1 is defined as satisfied and 0 is defined as neutral or not satisfied for Satisfaction. Likewise,

1-loyal and 0-disloyal for Customer Type, 1-female and 0-male for Gender, 1-business travel and 0-personal travel for Travel Type, and 1-business class and 0-eco or eco plus class for Class Type.

After transformation, we took a look at relation of all explanatory variables with Satisfaction based on their correlations. As we can see in the following graph, Inflight Entertainment and Ease of Online booking have the highest correlations with satisfaction at 0.52 and 0.43 respectively, while Gate location and Departure/Arrival time convenient have the lowest at -0.012 and -0.016 respectively. Correlation results are approximately at same direction with our distributional graph conceptions. For instance, as we guessed earlier, Class Type has a higher effect on Satisfaction than Travel Type, this proved with 0.31 and 0.11 correlations. Furthermore, most of Respond Features are good predictors of Satisfaction except Gate location, Departure/Arrival time convenient, and Food and drink.



- **Classification:**

Based on correlation results and feature natures, we did classifications on three levels:

- **First level: All variables.** We did classification for all variables. We did not eliminate any columns at this level and wanted to see the effect of all features in our prediction accuracy regardless of their types and individual relations with class variable.
- **Second level: Respond Features.** We did classifications only for Respond variables. We believed these are the variables that airlines can improve the most. Fact features are usually related to passengers themselves. Airlines can not change someone's gender or even their travel purpose. Besides, lots of factors have an effect on Flight features. For instance, it is not logical to recommend an airline to decrease its flight distance to get a higher customer satisfaction.
- **Third level: High Correlated Features.** We did classifications only for those features that the absolute value of their correlations with satisfaction is more than 0.2. We wanted to find out how much our models will be improved if just critical explanatory variables remained. We omitted six variables at this level: Travel Type, Age, Flight Distance, Real Delay, Gate location, Departure/Arrival time convenient, and Food and drink.

We then split datasets to train and test for each level. On each train part, we did 4 different model of classifications: **Logistic Regression, Naïve Bayes, K-Nearest Neighbors,** and **Decision Tree**. Each level has same train and test, so comparison is possible between different classification methods. We evaluated models based on **Classification Scores** and **Roc Curve Graph.**

✓ **Classification Score Table**

Following table is the score results on test part. Although accuracy is a good score for comparison in this dataset, data is fairly balanced, we tried to consider other scores, too.

Level	Classification	Accuracy	F1 Score	Precision	Recall
All Variables	Logistic Regression	0.78	0.77	0.78	0.78
	Naïve Bayes	0.82	0.82	0.82	0.82
	K-Nearest Neighbors	0.70	0.70	0.70	0.70
	Decision Tree	0.94	0.94	0.94	0.94
Respond Variables	Logistic Regression	0.81	0.80	0.80	0.80
	Naïve Bayes	0.79	0.78	0.78	0.79
	K-Nearest Neighbors	0.91	0.91	0.92	0.91
	Decision Tree	0.93	0.93	0.93	0.93
High Correlated Variables	Logistic Regression	0.83	0.83	0.83	0.83
	Naïve Bayes	0.82	0.82	0.82	0.82
	K-Nearest Neighbors	0.92	0.92	0.92	0.92
	Decision Tree	0.93	0.93	0.93	0.93

As we can see, Decision Tree is the best model among all classifiers at all levels. After that K-Nearest Neighbors model has the best prediction for Respond and High Correlated levels. The second-best predictor for All variables is Naïve Bayes, though.

We can also conclude that eliminating features have not a huge impact on improvement of model prediction. Metric scores of Decision Tree model is fairly equal at different levels. By changing from All variables to Respond variables, Accuracy score of Logistic Regression increased from 0.78 to 0.81; however, this score decreased for Naïve Bayes from 0.82 to 0.79. Metric scores of High Correlated variables are fairly higher than those of other levels for each classification, but the differences are not significant, on average 2-3 percent.

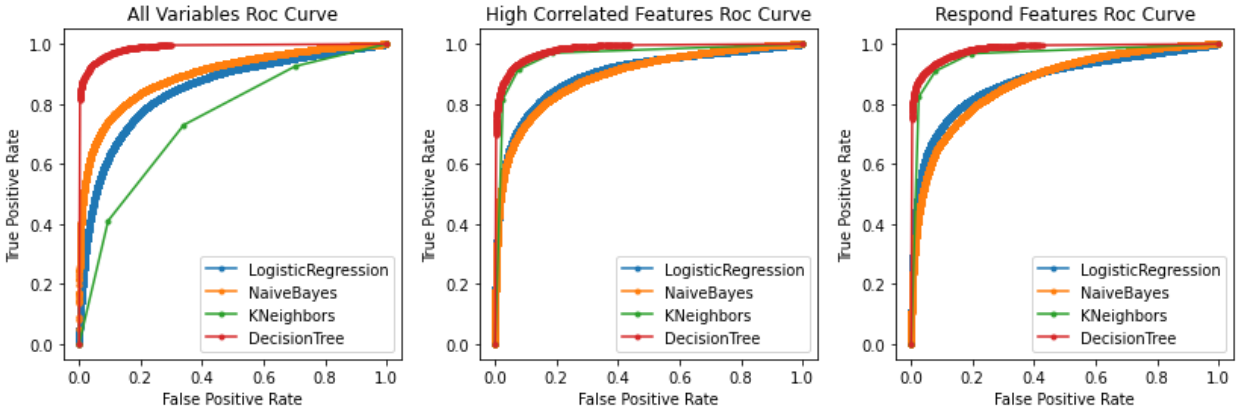
We can conclude that we could improve our predictions by changing our classification model, but not by eliminating variables. It seems even those variables with little impact on Satisfaction can help us for better prediction.

✓ **Roc Curve Graph**

As we can see in the following graphs, Decision Tree is the best model at all levels, and there is not much differences in using either Logistic Regression or Naïve Bayes at all levels. Finally, K-Nearest Neighbors model is the second-best model at Respond and High Correlated levels, while this model is the poorest model among four at All Variables level.

We can also see that changing from one model to another model results in more improvement in All Variables level than other levels. This changing brings not much improvement for Respond Features and

High Correlated Features especially when we use Decision Tree instead of K-Nearest Neighbors or Logistic Regression instead of Naïve Bayes.

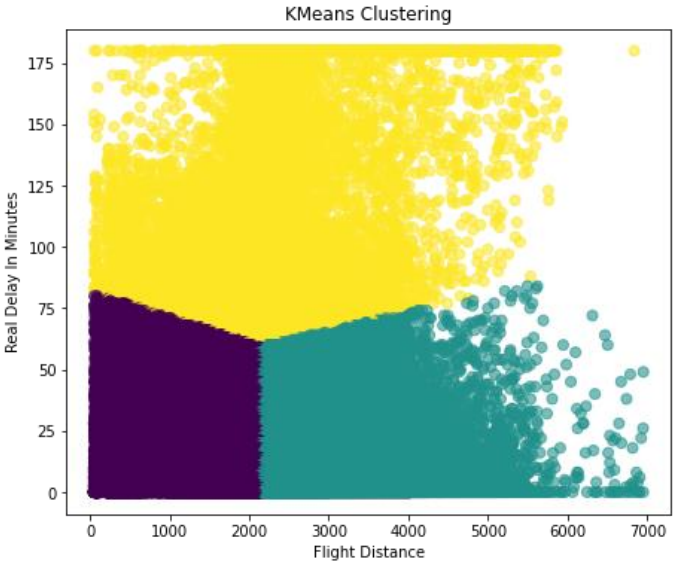


- **Clustering:**

We clustered on Flight Features, Flight Distance and Real Delay. Number of clusters is chosen 3 based on elbow method. Before the experiment, we expected deviation in these factors be a strong source of dissatisfaction; however, as we found out in preprocessing these two variables have fairly no correlation, about -0.015, with satisfaction. Therefore, we decided to check their similarity by assigning them into clusters. We expected that passengers who experienced long distances and long waiting time be grouped in same cluster and those who had short delay and short journey be grouped together.

As we can see, Flight Distance can differentiate among passengers who experienced delay lower than 75 minutes. 2100 miles is approximately this differentiation threshold. %92 of passengers are in these two clusters, Green and Purple parts. Therefore, we can say that our expectation is met halfway. The waiting time causes no differentiation for majority of passengers.

On contrary, the Flight Distance causes no differentiation for minority of passengers, %7. There is no vertical threshold in yellow part.



- **Association:**

At our final analysis part, we decided to find out whether there is a similarity among Respond Features or not. We saw in data part that some of Respond Features have similar graphs at different levels of satisfaction. Therefore, there might be an association among choosing them. We first define each of these features as important, features that are ranked 4 or 5, and not important, features that are ranked less than 4. Then we tried different support and confidence thresholds to find out whether customers pay

equal attention at ranking these features or not. For instance, a passenger who is happy with Inflight WIFI is more likely to be happy with Ease of Online booking.

As we can see On-board service, Cleanliness, and Baggage handling are completely associated at 0.75 confidence and 0.45 support levels. Passengers who ranked any of them high usually ranked the other one high, too. Leg room service can approximately be in this subset. The only rule that this threshold can not cover is Leg room service as antecedent and On-board service as consequent. The connection among importance of these four features makes logical sense. People who are strike toward the quality of Leg room service are more likely to have same attitude toward Cleanliness. Knowing these association rules help airlines to not focus only on one part. They have to improve different related parts together to get customer satisfaction.

	antecedents	consequents	support	confidence
8	(On-board service)	(Cleanliness)	0.485148	0.870330
14	(Baggage handling)	(Cleanliness)	0.559686	0.865505
15	(Cleanliness)	(Baggage handling)	0.559686	0.858118
6	(On-board service)	(Baggage handling)	0.476917	0.855564
1	(Ease of Online booking)	(Online support)	0.487096	0.854261
4	(Ease of Online booking)	(Cleanliness)	0.473121	0.829753
0	(Online support)	(Ease of Online booking)	0.487096	0.820832
3	(Ease of Online booking)	(Baggage handling)	0.466246	0.817694
12	(Leg room service)	(Cleanliness)	0.459478	0.805542
10	(Leg room service)	(Baggage handling)	0.457545	0.802154
9	(Cleanliness)	(On-board service)	0.485148	0.743835
7	(Baggage handling)	(On-board service)	0.476917	0.737510
5	(Cleanliness)	(Ease of Online booking)	0.473121	0.725396
2	(Baggage handling)	(Ease of Online booking)	0.466246	0.721008
11	(Baggage handling)	(Leg room service)	0.457545	0.707553
13	(Cleanliness)	(Leg room service)	0.459478	0.704478

Ease of Online booking and Online support are associated, too. This is logical since these two questions are asking about same fact.

Conclusions

In this project we investigate different effective factors on passenger satisfaction in American Airline. We found out some factors such as Delay, Flight Distance, Food and drink, etc. that might seem critical at first look, are not much determinative. On the other hand, passengers pay more attention to factors such as Inflight Entertainment, Ease of Online booking, Online support, etc.

We also found that eliminating features does not guarantee a better result in our prediction of satisfaction but changing classification does. Decision Tree, with an accuracy around 0.94, is the best model for predicting whether a passenger is satisfied with airline or not.

We also saw that although we can group majority of customers in two clusters: lower than 75 minutes Delay and lower than 2100 miles Flight, or lower than 75 minutes Delay and more than 2100 miles Flight, these two clusters do not have high Inter-cluster distance; their thresholds were stick to each other.

Finally, there is association rules among those features that passengers ranked high, but these rules are restricted to some 4 or 5 features out of 14. Therefore, they cannot bring significant conclusions.

Airline companies can use these findings to improve their services. They can predict which infrastructure might be more beneficial for them to invest.

References

1. Hongwei Jiang a, and Yahua Zhang, “An investigation of service quality, customer satisfaction and loyalty in China's airline market”, Journal of Air Transport Management, 2016 Elsevier Ltd.
2. Orsea-Tanzania, Chapter in Collaboration with The University of Dare S Salaam Business School (UDBS), International Conference.
3. Hansemark and Albinson, “Customer satisfaction and retention”, 2004 Journal of Service Theory and Practice.
4. David Mc. A Baker, “Service Quality and Customer Satisfaction in the Airline Industry: A Comparison between Legacy Airlines and Low-Cost Airlines”, Harmon College of Business and Professional Studies, University of Central Missouri.
5. Norazah MohdSuki, “Passenger satisfaction with airline service quality in Malaysia: A structural equation modeling approach”,2014 Labuan School of International Business & Finance, University Malaysia Sabah, Labuan, Malaysia.
6. Ove C. Hansemark , Marie Albinsson “Customer satisfaction and retention: The experiences of individual employees”,2004,Journal of Service Theory and Practice
7. https://www.wikipedia.org/
8. https://www.elsevier.ca/
9. https://www.kaggle.com/
10. https://www.udemy.com/
11. https://www.jdpower.com/business/
12. https://scikit-learn.org/stable/
13. https://www.tutorialspoint.com/python/
14. https://stackoverflow.com/

Appendix

```
# -*- coding: utf-8 -*-
"""623 project.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1VPkf1rVQNVdsJLUS3V1n7F0rzIXG0ubL
"""

# Commented out IPython magic to ensure Python compatibility.
#Importing Required Libraries And Dataset
import pandas as pd
df0=pd.read_csv('satisfaction.csv')
import seaborn as sns
import matplotlib.pyplot as plt
# %matplotlib inline
import numpy as np

#Looking At Dataset
df0.head()

#Dropping Unsueful Column
df0=df0.drop('id',1)
df0.head()

df0.shape

#Take a look at Datasets Measures of Central Tendency
df0.describe()

#Finding about Missing Data
np.sum(df0.isnull())

#Calculating the percentage of missing data (%0.3 of arrival delay is missed)
df0.isnull().sum()/len(df0)

#Finding whether features are independent or not
plt.figure(figsize=(20,10))
corrMatrix = df0.corr()
sns.heatmap(corrMatrix,annot=True,cmap='coolwarm')
plt.show()

## Take a look at Departure/Arrival variables distribution
plt.hist(df0['Departure Delay in Minutes'])
```

```
plt.xlabel('Departure Delay in Minutes')
plt.show()

plt.hist(df0['Arrival Delay in Minutes'])
plt.xlabel('Arrival Delay in Minutes')
plt.show()

## Take a  better look at Departure/Arrival variables distribution
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(y='Departure Delay in Minutes',data=df0)
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
sns.boxplot(y='Arrival Delay in Minutes',data=df0)

#Checking the relation of Arrival and Depart Delay
sns.lmplot(x='Departure Delay in Minutes',y='Arrival Delay in Minutes',data=df0)

#Replacing missing value of Arrival Delay with zero since we believe this amount is reported in Departe Delay
df0['Arrival Delay in Minutes'] = df0['Arrival Delay in Minutes'].replace(np.nan, 0)

#Joing the information of Arrival & Departure Delay in a single column to prevent dependency of featurears
df0['Real Delay In Minutes']=df0['Arrival Delay in Minutes']+abs(df0['Arrival Delay in Minutes']-df0['Departure Delay in Minutes'])
df0=df0.drop(['Arrival Delay in Minutes','Departure Delay in Minutes'],axis=1)

#Finding about maximum Real delay variables
df0.sort_values(by=['Real Delay In Minutes'], inplace=True, ascending=False)
df0.head(10)

df0.describe()

#Checking percent of outliers
print(np.sum(df0['Flight Distance']>5359))
print(np.sum(df0['Flight Distance']>5359)/len(df0))

print(np.sum(df0['Real Delay In Minutes']>88))
print(np.sum(df0['Real Delay In Minutes']>88)/len(df0))
print(np.sum(df0['Real Delay In Minutes']>149))
print(np.sum(df0['Real Delay In Minutes']>149)/len(df0))

#Converting all categorical features  to  numeric variables
def Delay_Check(x):
  if x >= 180:
    return 180
  else:
```

```
    return x
df0['Real Delay In Minutes']=df0['Real Delay In Minutes'].apply(Delay_Check)

print(np.sum(df0['Real Delay In Minutes']>75))
print(np.sum(df0['Real Delay In Minutes']>75)/len(df0))

df0.sort_values(by=['Real Delay In Minutes'], inplace=True, ascending=False)
df0.head()

df0.describe()

#Finding the frequency and percentage of categorical variables
print(df0['satisfaction_v2'].value_counts())
print(df0['satisfaction_v2'].value_counts()/len(df0))

print(df0['Gender'].value_counts())
print(df0['Gender'].value_counts()/len(df0))

print(df0['Customer Type'].value_counts())
print(df0['Customer Type'].value_counts()/len(df0))

print(df0['Type of Travel'].value_counts())
print(df0['Type of Travel'].value_counts()/len(df0))

print(df0['Class'].value_counts())
print(df0['Class'].value_counts()/len(df0))

## Take a look at Class Variable
plt.hist(df0['satisfaction_v2'])
plt.xlabel('satisfaction_v2')
plt.show()

## Take a look at Fact Features distribution
plt.hist(df0['Gender'])
plt.xlabel('Gender')
plt.show()

plt.hist(df0['Customer Type'])
plt.xlabel('Customer Type')
plt.show()

plt.hist(df0['Type of Travel'])
plt.xlabel('Type of Travel')
plt.show()

plt.hist(df0['Class'])
plt.xlabel('Class')
plt.show()
```

```
plt.hist(df0['Age'])
plt.xlabel('Age')
plt.show()

## Take a look at Flight Features distributions
plt.hist(df0['Flight Distance'])
plt.xlabel('Flight Distance')
plt.show()

plt.hist(df0['Real Delay In Minutes'])
plt.xlabel('Real Delay In Minutes')
plt.show()

## Take a look at Respond Features distributions
plt.hist(df0['Seat comfort'])
plt.xlabel('Seat comfort')
plt.show()

plt.hist(df0['Departure/Arrival time convenient'])
plt.xlabel('Departure/Arrival time convenient')
plt.show()

plt.hist(df0['Food and drink'])
plt.xlabel('Food and drink')
plt.show()

plt.hist(df0['Gate location'])
plt.xlabel('Gate location')
plt.show()

plt.hist(df0['Inflight wifi service'])
plt.xlabel('Inflight wifi service')
plt.show()

plt.hist(df0['Inflight entertainment'])
plt.xlabel('Inflight entertainment')
plt.show()

plt.hist(df0['Online support'])
plt.xlabel('Online support')
plt.show()

plt.hist(df0['Ease of Online booking'])
plt.xlabel('Ease of Online booking')
plt.show()

plt.hist(df0['On-board service'])
```

```
plt.xlabel('On-board service')
plt.show()

plt.hist(df0['Leg room service'])
plt.xlabel('Leg room service')
plt.show()

plt.hist(df0['Baggage handling'])
plt.xlabel('Baggage handling')
plt.show()

plt.hist(df0['Checkin service'])
plt.xlabel('Checkin service')
plt.show()

plt.hist(df0['Cleanliness'])
plt.xlabel('Cleanliness')
plt.show()

plt.hist(df0['Online boarding'])
plt.xlabel('Online boarding')
plt.show()

#Take a look at  Flight Features with  Satisfaction

#Real Delay vs Satisfaction
#Flight Distance vs Satisfaction

plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Real Delay In Minutes',data=df0)

plt.subplot(1,2,2)
sns.boxplot(x='satisfaction_v2', y='Flight Distance',data=df0)

s=['satisfied','neutral or dissatisfied']
plt.figure(figsize=(8,4))
for i in s:
  m=df0[df0['satisfaction_v2']==i]
  sns.distplot(m['Real Delay In Minutes'],
kde=True,hist=False,kde_kws={'shade':True,'linewidth':3},label=i)

s=['satisfied','neutral or dissatisfied']
plt.figure(figsize=(8,4))
for i in s:
  N=df0[df0['satisfaction_v2']==i]
  sns.distplot(N['Flight Distance'], kde=True,hist=False,kde_kws={'shade':True,'linewidth':3},label=i)
```

```
#Take a look at Fact features with Satisfaction
# Gender vs Satisfaction
ax=sns.catplot(x='satisfaction_v2',kind='count',data=df0, hue='Gender')
ax.set(xlabel="Satisfaction", ylabel = "Count")
pd.crosstab(df0.Gender,
df0.satisfaction_v2,margins=True,rownames=['Gender'],colnames=['Satisfaction']).apply(lambda r:
r/len(df0))

# Customer Type vs Satisfaction
ax_2=sns.catplot(x='satisfaction_v2',kind='count',data=df0, hue='Customer Type')
ax_2.set(xlabel="Satisfaction", ylabel = "Count")
pd.crosstab(df0['Customer Type'], df0.satisfaction_v2,margins=True,rownames=['Customer
Type'],colnames=['Satisfaction']).apply(lambda r: r/len(df0))

# Travel Type vs Satisfaction
sns.catplot(x='satisfaction_v2',kind='count',data=df0, hue='Type of Travel')
pd.crosstab(df0['Type of Travel'] , df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

# Class vs Satisfaction
sns.catplot(x='satisfaction_v2',kind='count',data=df0, hue='Class')
pd.crosstab(df0.Class, df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Age vs Satisfaction
sns.boxplot(x='satisfaction_v2', y='Age',data=df0)

sns.catplot("Age", data=df0, aspect=3.0, kind='count', hue='satisfaction_v2', order=range(5, 80))

#Look at each Respond features with Satisfaction
#Seat Comfort vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Seat comfort',data=df0)
pd.crosstab(df0['Seat comfort'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Departure/Arrival time convenient vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
ax_5=sns.boxplot(x='satisfaction_v2', y='Departure/Arrival time convenient',data=df0)
ax_5.set(xlabel="Satisfaction", ylabel = "Count")
pd.crosstab(df0['Departure/Arrival time convenient'],
df0.satisfaction_v2,margins=True,colnames=['Satisfaction']).apply(lambda r: r/len(df0))

#Food and drink vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
ax_4=sns.boxplot(x='satisfaction_v2', y='Food and drink',data=df0)
ax_4.set(xlabel="Satisfaction", ylabel = "Count")
```

```
pd.crosstab(df0['Food and drink'],
df0.satisfaction_v2,margins=True,colnames=['Satisfaction']).apply(lambda r: r/len(df0))

#Gate location vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Gate location',data=df0)
pd.crosstab(df0['Gate location'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Inflight wifi service vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
sns.boxplot(x='satisfaction_v2', y='Inflight wifi service',data=df0)
pd.crosstab(df0['Inflight wifi service'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Inflight entertainment vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
ax_6=sns.boxplot(x='satisfaction_v2', y='Inflight entertainment',data=df0)
ax_6.set(xlabel="Satisfaction", ylabel = "Count")
pd.crosstab(df0['Inflight entertainment'],
df0.satisfaction_v2,margins=True,colnames=['Satisfaction']).apply(lambda r: r/len(df0))

#Online support vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
ax_3=sns.boxplot(x='satisfaction_v2', y='Online support',data=df0)
ax_3.set(xlabel="Satisfaction", ylabel = "Count")
pd.crosstab(df0['Online support'],
df0.satisfaction_v2,margins=True,colnames=['Satisfaction']).apply(lambda r: r/len(df0))

#Ease of Online booking vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Ease of Online booking',data=df0)
pd.crosstab(df0['Ease of Online booking'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#On-board service vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
sns.boxplot(x='satisfaction_v2', y='On-board service',data=df0)
pd.crosstab(df0['On-board service'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Leg room service vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Leg room service',data=df0)
pd.crosstab(df0['Leg room service'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))
```

```
#Baggage handling vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
sns.boxplot(x='satisfaction_v2', y='Baggage handling',data=df0)
pd.crosstab(df0['Baggage handling'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Checkin service vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Checkin service',data=df0)
pd.crosstab(df0['Checkin service'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Cleanliness vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,2)
sns.boxplot(x='satisfaction_v2', y='Cleanliness',data=df0)
pd.crosstab(df0['Cleanliness'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

#Online boarding vs Satisfaction
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.boxplot(x='satisfaction_v2', y='Online boarding',data=df0)
pd.crosstab(df0['Online boarding'], df0.satisfaction_v2,margins=True).apply(lambda r: r/len(df0))

df1=df0

#Converting class variable to a numeric variable
def Satisfaction_Check(x):
  if x =='satisfied':
    return 1
  else:
    return 0
df1['Satisfaction']=df1['satisfaction_v2'].apply(Satisfaction_Check)
df1=df1.drop(['satisfaction_v2',],axis=1)

#Converting all categorical features  to  numeric variables
def Customer_Check(x):
  if x =='Loyal Customer':
    return 1
  else:
    return 0
df1['Customer_Type']=df1['Customer Type'].apply(Customer_Check)
df1=df1.drop(['Customer Type',],axis=1)

def Gender_Check(x):
  if x =='Female':
    return 1
```

```
  else:
    return 0
df1['Gender_Type']=df1['Gender'].apply(Gender_Check)
df1=df1.drop(['Gender',],axis=1)

def Travel_Check(x):
  if x =='Business travel':
    return 1
  else:
    return 0
df1['Travel_Type']=df1['Type of Travel'].apply(Travel_Check)
df1=df1.drop(['Type of Travel',],axis=1)

def Class_Check(x):
  if x =='Business':
    return 1
  else:
    return 0
df1['Class_Type']=df1['Class'].apply(Class_Check)
df1=df1.drop(['Class',],axis=1)

#Checking 'class' relation with satisfaction after making eco and eco plus one grooup
sns.catplot(x='Satisfaction',kind='count',data=df1, hue='Class_Type')
pd.crosstab(df1.Class_Type, df1.Satisfaction,margins=True).apply(lambda r: r/len(df1))

#Rearranging the order of variables in table
df1=df1[['Satisfaction','Gender_Type','Customer_Type','Travel_Type','Class_Type','Age','Flight
Distance','Real Delay In Minutes','Seat comfort','Departure/Arrival time convenient','Food and
drink','Gate location','Inflight wifi service','Inflight entertainment','Online support','Ease of Online
booking','On-board service','Leg room service','Baggage handling','Checkin service','Cleanliness','Online
boarding']]

#Take a look at all variables to find whether all are numeric or not
df1.head()

df1.describe()

#Finding about centarl tendency of measurements for different values of class
df1.groupby('Satisfaction').mean()

#Finding about centarl tendency of measurements for different values of class
df1.groupby('Satisfaction').min()
df1.groupby('Satisfaction').max()
df1.groupby('Satisfaction').std()

#Checking the relations(correlation) of features with satisfaction
plt.figure(figsize=(20,15))
corrMatrix2 = df1.corr()
```

```
mask = np.triu(np.ones_like(corrMatrix2, dtype=np.bool))
sns.heatmap(corrMatrix2, annot=True,cmap='coolwarm', mask=mask, vmax=None,
center=0,square=True,linewidths=0.6, cbar_kws={"shrink": .9})
plt.show()

#Checking the relations of features with satisfaction
corr_dia=df1.corr()['Satisfaction'].sort_values().drop('Satisfaction').plot(kind='bar')
corr_dia

df1.shape

#Importing Libraries for train/test split
from sklearn.model_selection import train_test_split
# For evaluating our ML results
from sklearn import metrics

#Defining features(X) and class variable(Y) for All Variables
Y=df1['Satisfaction']
X=df1.drop(['Satisfaction'],axis=1)

X.head()

Y.head()

# Split  train/test data for All variables
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.30)

#Seprating Respond Features
df_Q=pd.DataFrame()

df_Q=df1[['Satisfaction','Seat comfort','Departure/Arrival time convenient','Food and drink','Gate
location','Inflight wifi service','Inflight entertainment','Online support','Ease of Online booking','On-
board service','Leg room service','Baggage handling','Checkin service','Cleanliness','Online boarding']]

df_Q.head()

#Defining features(X) and class variable(Y) for Respond features
Y_Q=df_Q['Satisfaction']
X_Q=df_Q.drop(['Satisfaction'],axis=1)

# Split train/test data for Respond features
X_Q_train, X_Q_test, Y_Q_train, Y_Q_test = train_test_split(X_Q,Y_Q,test_size=0.30)

#Seprating High correlated Features

df_C=pd.DataFrame()
```

```
df_C=df1[['Satisfaction','Gender_Type','Customer_Type','Class_Type','Seat comfort','Inflight wifi service','Inflight entertainment','Online support','Ease of Online booking','On-board service','Leg room service','Baggage handling','Checkin service','Cleanliness','Online boarding']]

df_C.head()

#Defining features(X) and class variable(Y) for high correlated features
Y_C=df_C['Satisfaction']
X_C=df_C.drop(['Satisfaction'],axis=1)

# Split train/test data for High correlated features
X_C_train, X_C_test, Y_C_train, Y_C_test = train_test_split(X_C,Y_C,test_size=0.30)

# Machine Learning Imports
from sklearn.linear_model import LogisticRegression

# Make a log_model For All Variables
log_model = LogisticRegression()

# Now fit the new model
log_model.fit(X_train,Y_train)

# Predict the classes of the testing data set
class_predict_log = log_model.predict(X_test)


# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_test,class_predict_log))
print('Report', metrics.classification_report(Y_test,class_predict_log))

# Use zip to bring the column names and the np.transpose function to bring together the coefficients from the model
coeff_df = pd.DataFrame(zip(X_train.columns,np.transpose(log_model.coef_)))
coeff_df.columns=['Features','Coefficient']
coeff_df.sort_values(by='Coefficient', ascending=False)

# Make a log_model for Respond features
log_model_Q= LogisticRegression()

# Now fit the new model for  Respond features
log_model_Q.fit(X_Q_train,Y_Q_train)

# Predict the classes of the testing data set
class_predict_log_Q= log_model_Q.predict(X_Q_test)


# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_Q_test,class_predict_log_Q))
```

```
print('Report', metrics.classification_report(Y_Q_test,class_predict_log_Q))

# Use zip to bring the column names and the np.transpose function to bring together the coefficients
from the model
coeff_df_Q = pd.DataFrame(zip(X_Q_train.columns,np.transpose(log_model_Q.coef_)))
coeff_df_Q.columns=['Features','Coefficient']
coeff_df_Q.sort_values(by='Coefficient', ascending=False)

# Make a log_model for High correlated  features
log_model_C= LogisticRegression()

# Now fit the new model for High correlated features
log_model_C.fit(X_C_train,Y_C_train)

# Predict the classes of the testing data set
class_predict_log_C= log_model_C.predict(X_C_test)


# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_C_test,class_predict_log_C))
print('Report', metrics.classification_report(Y_C_test,class_predict_log_C))

# Use zip to bring the column names and the np.transpose function to bring together the coefficients
from the model
coeff_df_C = pd.DataFrame(zip(X_C_train.columns,np.transpose(log_model_C.coef_)))
coeff_df_C.columns=['Features','Coefficient']
coeff_df_C.sort_values(by='Coefficient', ascending=False)

# Machine Learning Imports
from sklearn.naive_bayes import GaussianNB

# Make a NaiveBayes_model For All Variables
naive_bayes = GaussianNB()

# Now fit the new model for High correlated features
naive_bayes.fit(X_train,Y_train)

# Predict the classes of the testing data set
class_predict_naive = naive_bayes.predict(X_test)


# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_test,class_predict_naive))
print('Report', metrics.classification_report(Y_test,class_predict_naive))

# Make a NaiveBayes_model For Respond Features
naive_bayes_Q = GaussianNB()
```

```
# Now fit the new model for Respond Features
naive_bayes_Q.fit(X_Q_train,Y_Q_train)

# Predict the classes of the testing data set
class_predict_naive_Q = naive_bayes_Q.predict(X_Q_test)


# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_Q_test,class_predict_naive_Q))
print('Report', metrics.classification_report(Y_Q_test,class_predict_naive_Q))

# Make a NaiveBayes_model For High Correlated Features
naive_bayes_C= GaussianNB()

# Now fit the new model for High Correlated Features
naive_bayes_C.fit(X_C_train,Y_C_train)

# Predict the classes of the testing data set
class_predict_naive_C= naive_bayes_C.predict(X_C_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_C_test,class_predict_naive_C))
print('Report', metrics.classification_report(Y_C_test,class_predict_naive_C))

# Machine Learning Imports
from sklearn.neighbors import KNeighborsClassifier

# Make a KNeighbors_model For All Variables
KNeighbors_model=KNeighborsClassifier(n_neighbors=3)

# Now fit the new model for All Variables
KNeighbors_model.fit(X_train,Y_train)

# Predict the classes of the testing data set
class_predict_KNeighbors=KNeighbors_model.predict(X_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_test,class_predict_KNeighbors))
print('Report', metrics.classification_report(Y_test,class_predict_KNeighbors))

# Make a KNeighbors_model For Respond Features
KNeighbors_model_Q=KNeighborsClassifier(n_neighbors=3)

# Now fit the new model for Respond Features
KNeighbors_model_Q.fit(X_Q_train,Y_Q_train)

# Predict the classes of the testing data set
class_predict_KNeighbors_Q=KNeighbors_model_Q.predict(X_Q_test)
```

```
# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_Q_test,class_predict_KNeighbors_Q))
print('Report', metrics.classification_report(Y_Q_test,class_predict_KNeighbors_Q))

# Make a KNeighbors_model For High Correlated Features
KNeighbors_model_C=KNeighborsClassifier(n_neighbors=3)

# Now fit the new model for High Correlated Features
KNeighbors_model_C.fit(X_C_train,Y_C_train)

# Predict the classes of the testing data set
class_predict_KNeighbors_C=KNeighbors_model_C.predict(X_C_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_C_test,class_predict_KNeighbors_C))
print('Report', metrics.classification_report(Y_C_test,class_predict_KNeighbors_C))

# Machine Learning Imports
from sklearn.tree import DecisionTreeClassifier

# Make a DecisionTree_model For All Variables
DecisionTree_model=DecisionTreeClassifier(min_samples_split=100)

# Now fit the new model for For All Variables
DecisionTree_model.fit(X_train,Y_train)

# Predict the classes of the testing data set
class_predict_DecisionTree=DecisionTree_model.predict(X_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_test,class_predict_DecisionTree))
print('Report', metrics.classification_report(Y_test,class_predict_DecisionTree))

#Graphing Decision Tree
from sklearn import tree
import graphviz
from sklearn.tree import export_graphviz
tree.plot_tree(DecisionTree_model)
treedata=tree.export_graphviz(DecisionTree_model,out_file=None)
n=graphviz.Source(treedata)
n.render("Retail")

# Make a DecisionTree_model For Respond Features
DecisionTree_model_Q=DecisionTreeClassifier(min_samples_split=100)

# Now fit the new model for Respond Features
DecisionTree_model_Q.fit(X_Q_train,Y_Q_train)
```

```
# Predict the classes of the testing data set
class_predict_DecisionTree_Q=DecisionTree_model_Q.predict(X_Q_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_Q_test,class_predict_DecisionTree_Q))
print('Report', metrics.classification_report(Y_Q_test,class_predict_DecisionTree_Q))

# Make a DecisionTree_model For High Correlated Features
DecisionTree_model_C=DecisionTreeClassifier(min_samples_split=100)

# Now fit the new model For High Correlated Variables
DecisionTree_model_C.fit(X_C_train,Y_C_train)

# Predict the classes of the testing data set
class_predict_DecisionTree_C=DecisionTree_model_C.predict(X_C_test)

# Compare the predicted classes to the actual test classes
print('Accuracy', metrics.accuracy_score(Y_C_test,class_predict_DecisionTree_C))
print('Report', metrics.classification_report(Y_C_test,class_predict_DecisionTree_C))

#Importing Roc-Curve metric
#Defining function for graphing Roc-Curve
from sklearn.metrics import  roc_curve
def graph(Model, xtest, ytest ,Label):
  y_probable=Model.predict_proba(xtest)
  y_probable=y_probable[:,1]
  fpr, tpr, _ = roc_curve(ytest, y_probable)
  plt.plot(fpr, tpr, marker='.', label=Label)

#Graphing Roc-Curve For All Variables
plt.figure(figsize=(4,4))
graph(Model=log_model, xtest=X_test ,ytest=Y_test,Label='LogisticRegression' )
graph(Model=naive_bayes, xtest=X_test ,ytest=Y_test,Label='NaiveBayes' )
graph(Model=KNeighbors_model, xtest=X_test ,ytest=Y_test,Label='KNeighbors' )
graph(Model=DecisionTree_model, xtest=X_test ,ytest=Y_test,Label='DecisionTree' )
# axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('All Variables Roc Curve')
# show the legend
plt.legend()
# show the plot
plt.show()

#Graphing Roc-Curve For Respond Features
plt.figure(figsize=(4,4))
graph(Model=log_model_Q, xtest=X_Q_test ,ytest=Y_Q_test,Label='LogisticRegression' )
```

```
graph(Model=naive_bayes_Q, xtest=X_Q_test ,ytest=Y_Q_test,Label='NaiveBayes' )
graph(Model=KNeighbors_model_Q, xtest=X_Q_test ,ytest=Y_Q_test,Label='KNeighbors' )
graph(Model=DecisionTree_model_Q, xtest=X_Q_test ,ytest=Y_Q_test,Label='DecisionTree' )
# axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Respond Features Roc Curve')
# show the legend
plt.legend()
# show the plot
plt.show()

#Graphing Roc-Curve For High Correlated Features
plt.figure(figsize=(4,4))
graph(Model=log_model_C, xtest=X_C_test ,ytest=Y_C_test,Label='LogisticRegression' )
graph(Model=naive_bayes_C, xtest=X_C_test ,ytest=Y_C_test,Label='NaiveBayes' )
graph(Model=KNeighbors_model_C, xtest=X_C_test ,ytest=Y_C_test,Label='KNeighbors' )
graph(Model=DecisionTree_model_C, xtest=X_C_test ,ytest=Y_C_test,Label='DecisionTree' )
# axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('High Correlated Features Roc Curve')
# show the legend
plt.legend()
# show the plot
plt.show()

# Import Libraries for Clustering
import sklearn
from sklearn.cluster import KMeans
from sklearn.preprocessing import scale

#Clustering Numeric variables

#Flight Distance - Real Delay
Distance_Delay=pd.concat([df1['Flight Distance'],df1['Real Delay In Minutes']],axis=1)
Distance_Delay_Clusters= scale(Distance_Delay)

#Elbow method
distortations = {}
for k in range(1,10):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(Distance_Delay_Clusters)
  distortations[k] = kmeans.inertia_

plt.plot(list(distortations.keys()),list(distortations.values()))
plt.title('Elbow method on Airline dataset')
plt.xlabel('Number of clusters')
```

```
plt.ylabel('Within-cluster SSE')
plt.show()

clustering  = KMeans(n_clusters=3)
clustering.fit(Distance_Delay_Clusters)

#color_theme = np.array(['darkgray','lightsalmon','powderblue'])
plt.figure(figsize=(16,6))
#plt.subplot(1,2,1)
#plt.scatter(x=df1['Flight Distance'],y=df1['Real Delay In Minutes'],c=[df1['Satisfaction']],s=50)
#plt.xlabel('Flight Distance')
#plt.ylabel('Real Delay In Minutes')
#plt.title('Ground Truth Classification')

plt.subplot(1,2,2)
plt.scatter(x=df1['Flight Distance'],y=df1['Real Delay In Minutes'],c=[clustering.labels_],alpha=0.6,s=50)
plt.xlabel('Flight Distance')
plt.ylabel('Real Delay In Minutes')
plt.title('KMeans Clustering')

#Age - Real Delay

Age_Delay=pd.concat([df1['Age'],df1['Real Delay In Minutes']],axis=1)
Age_Delay_Clusters= scale(Age_Delay)

#Elbow method
distortations = {}
for k in range(1,10):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(Age_Delay_Clusters)
  distortations[k] = kmeans.inertia_

plt.plot(list(distortations.keys()),list(distortations.values()))
plt.title('Elbow method on Airline dataset')
plt.xlabel('Number of clusters')
plt.ylabel('Within-cluster SSE')
plt.show()

clustering  = KMeans(n_clusters=3)
clustering.fit(Age_Delay_Clusters)

#color_theme = np.array(['darkgray','lightsalmon','powderblue'])
plt.figure(figsize=(16,6))

plt.subplot(1,2,1)
plt.scatter(x=df1['Age'],y=df1['Real Delay In Minutes'],c=[df1['Satisfaction']],s=50)
plt.title('Ground Truth Classification')
```

```
plt.subplot(1,2,2)
plt.scatter(x=df1['Age'],y=df1['Real Delay In Minutes'],c=[clustering.labels_],s=50)
plt.title('K-Means Classification')

#Age- Flight Distance

Age_Distance=pd.concat([df1['Age'],df1['Flight Distance']],axis=1)
Age_Distance_Clusters= scale(Age_Distance)

#Elbow method
distortations = {}
for k in range(1,10):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(Age_Distance_Clusters)
  distortations[k] = kmeans.inertia_

plt.plot(list(distortations.keys()),list(distortations.values()))
plt.title('Elbow method on Airline dataset')
plt.xlabel('Number of clusters')
plt.ylabel('Within-cluster SSE')
plt.show()

clustering  = KMeans(n_clusters=3)
clustering.fit(Age_Distance_Clusters)

#color_theme = np.array(['darkgray','lightsalmon','powderblue'])
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.scatter(x=df1['Age'],y=df1['Flight Distance'],c=[df1['Satisfaction']],s=50)
plt.title('Ground Truth Classification')

plt.subplot(1,2,2)
plt.scatter(x=df1['Age'],y=df1['Flight Distance'],c=[clustering.labels_],s=50)
plt.title('K-Means Classification')

df2=pd.DataFrame()

#Finding Association Rules Among  Respond Features
def Transform(x):
  if x >=4:
    return True
  else:
    return False
df2['SeatComfort']=df1['Seat comfort'].apply(Transform)
df2['Departure/Arrival time convenient']=df1['Departure/Arrival time convenient'].apply(Transform)
df2['Food and drink']=df1['Food and drink'].apply(Transform)
df2['Gate location']=df1['Gate location'].apply(Transform)
df2['Inflight wifi service']=df1['Inflight wifi service'].apply(Transform)
```

```
df2['Inflight entertainment']=df1['Inflight entertainment'].apply(Transform)
df2['Online support']=df1['Online support'].apply(Transform)
df2['Ease of Online booking']=df1['Ease of Online booking'].apply(Transform)
df2['On-board service']=df1['On-board service'].apply(Transform)
df2['Leg room service']=df1['Leg room service'].apply(Transform)
df2['Baggage handling']=df1['Baggage handling'].apply(Transform)
df2['Checkin service']=df1['Checkin service'].apply(Transform)
df2['Cleanliness']=df1['Cleanliness'].apply(Transform)
df2['Online boarding']=df1['Online boarding'].apply(Transform)

df2.head()

#Import Libraries
from mlxtend.frequent_patterns import apriori, association_rules

#Defining Support level
frequent_itemsets = apriori(df2,min_support=0.45,use_colnames=True)
print(frequent_itemsets)

Rules=pd.DataFrame()

#Defining Confidence Level
Rules = association_rules(frequent_itemsets,metric='confidence',min_threshold=0.7)
Rules_Head=Rules[['antecedents','consequents','support','confidence']]

Rules_Head.sort_values(by=['confidence'], inplace=True, ascending=False)
Rules_Head
```