CSE 454 DATA MINIG

2021-2021 FALL SEMESTER

FINAL PROJECT REPORT

AIRLINE PASSENGER SATISFACTION WITH PREDICTIONAL CLASSIFICATION TECHNIQUES

1801042103

OZAN GECKİN

Project Definition:

In this project, the satisfaction status of the airline passengers is estimated with feature engineering, feature selection and classification techniques on the satisfaction surveys.

The aim of this project is to analyze and classify the available data in order to increase customer satisfaction of the airline company and to find out which classification method is suitable for estimating the satisfaction status with classification techniques.

In this Project, Random Forest, Logistic Regression, XGB, Gaussian Navie Bayes(I implemented) classification techniques were used and accuracy values were compared.

Dataset Used in the Project:

Airline Passenger Satisfaction dataset is used in my project. If you want to access the dataset, you can access it from the references section.

This dataset has 23 attributes.

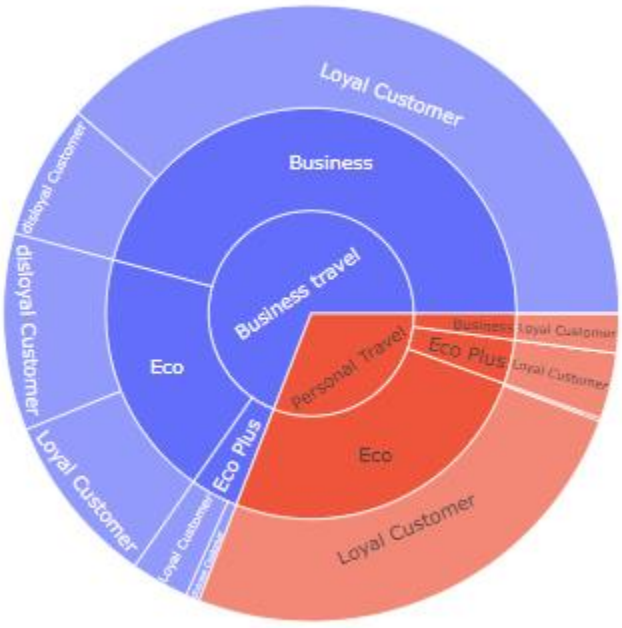
Gender: Gender of the passengers (Female, Male)
Customer Type: The customer type (Loyal customer, disloyal customer)
Age: The actual age of the passengers
Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)
Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)
Flight distance: The flight distance of this journey
Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)
Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient
Ease of Online booking: Satisfaction level of online booking
Gate location: Satisfaction level of Gate location
Food and drink: Satisfaction level of Food and drink
Online boarding: Satisfaction level of online boarding
Seat comfort: Satisfaction level of Seat comfort
Inflight entertainment: Satisfaction level of inflight entertainment
On-board service: Satisfaction level of On-board service
Leg room service: Satisfaction level of Leg room service
Baggage handling: Satisfaction level of baggage handling
Check-in service: Satisfaction level of Check-in service
Inflight service: Satisfaction level of inflight service
Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure
Arrival Delay in Minutes: Minutes delayed when Arrival
Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

Data table:

	Unnamed: 0	id	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-b ser
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	1	5	3	5	5	
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	3	1	3	1	1	
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	2	5	5	5	5	
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	5	2	2	2	2	
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	3	4	5	5	3	

Sunburst Graph (Type of Travel, Class, Customer Type):



Data Cleaning:

I'll start with removing the Unnamed and id column. Cause my guess is they don't matter.

```
[ ]  dftrain= dftrain.drop(["Unnamed: 0","id"],axis=1)
     dftest = dftest.drop(["Unnamed: 0","id"],axis=1)
```

I detected the places with total nulls in my dataset. And I filled them in with the mean value. Because I did not want to reduce the number of data I have.

```
print(dftrain.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             306
satisfaction                           0
dtype: int64
```

```
df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].median(), inplace = True)
return df
```

I converted the string data in my dataset into binary. A necessary process for my analysis.

```
def transform_gender(x):
    if x == 'Female':
        return 1
    elif x == 'Male':
        return 0
    else:
        return -1

def transform_customer_type(x):
    if x == 'Loyal Customer':
        return 1
    elif x == 'disloyal Customer':
        return 0
    else:
        return -1

def transform_travel_type(x):
    if x == 'Business travel':
        return 1
    elif x == 'Personal Travel':
        return 0
    else:
        return -1

def transform_class(x):
    if x == 'Business':
        return 2
    elif x == 'Eco Plus':
        return 1
    elif x == 'Eco':
        return 0
    else:
        return -1

def transform_satisfaction(x):
    if x == 'satisfied':
        return 1
    elif x == 'neutral or dissatisfied':
        return 0
    else:
        return -1

def process_data(df):
    df['Gender']=df['Gender'].apply(transform_gender)
    df['Customer Type'] = df['Customer Type'].apply(transform_customer_type)
    df['Type of Travel'] = df['Type of Travel'].apply(transform_travel_type)
    df['Class'] = df['Class'].apply(transform_class)
    df['satisfaction'] = df['satisfaction'].apply(transform_satisfaction)
```

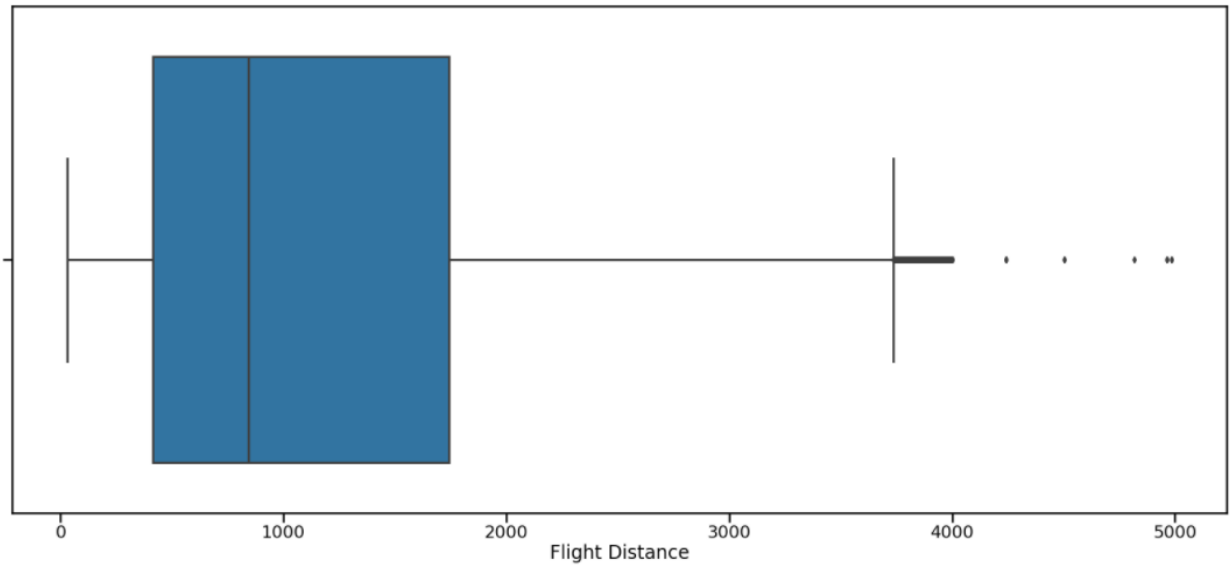
Outlier Detection:

In my project, I observed the general state of my datasets with the describe function.

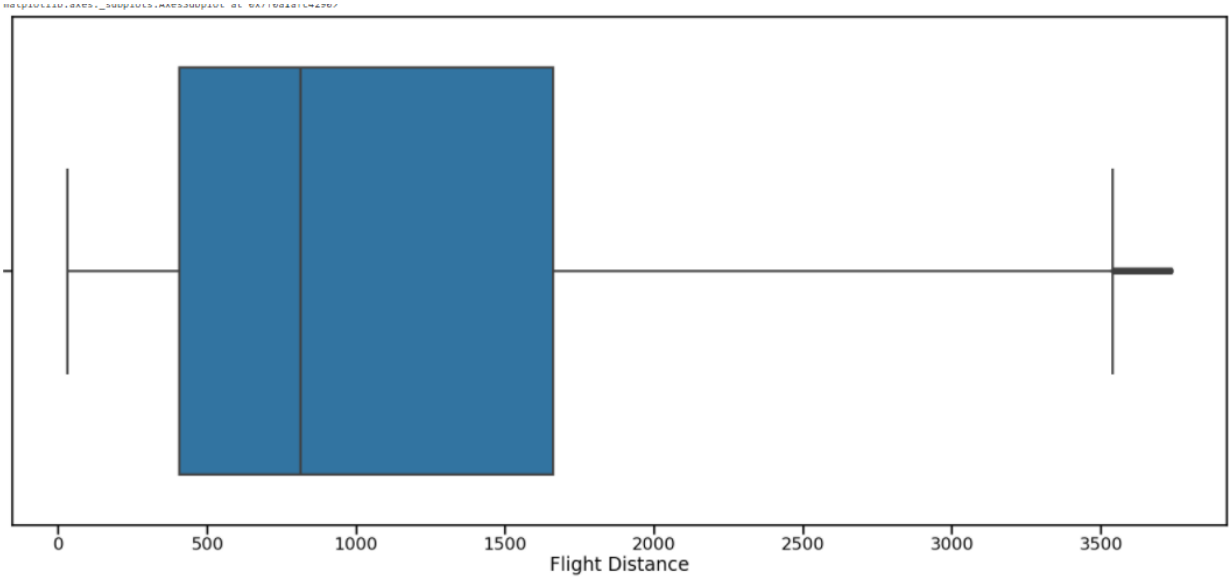
	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	ent
count	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103904.000000	103
mean	39.379706	1189.448375	2.729683	3.060296	2.756901	2.976883	3.202129	3.250375	3.439396	
std	15.114964	997.147281	1.327829	1.525075	1.398929	1.277621	1.329533	1.349509	1.319088	
min	7.000000	31.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	27.000000	414.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	
50%	40.000000	843.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	4.000000	
75%	51.000000	1743.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	5.000000	
max	85.000000	4983.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	

When we examine the table, we can see the maximum, minimum and average values. If there are maximum or minimum data too far from the mean values, we can say that they are outliers.

When we look at this table, there are too many deviations from the average value in the "Flight Distance" attribute.

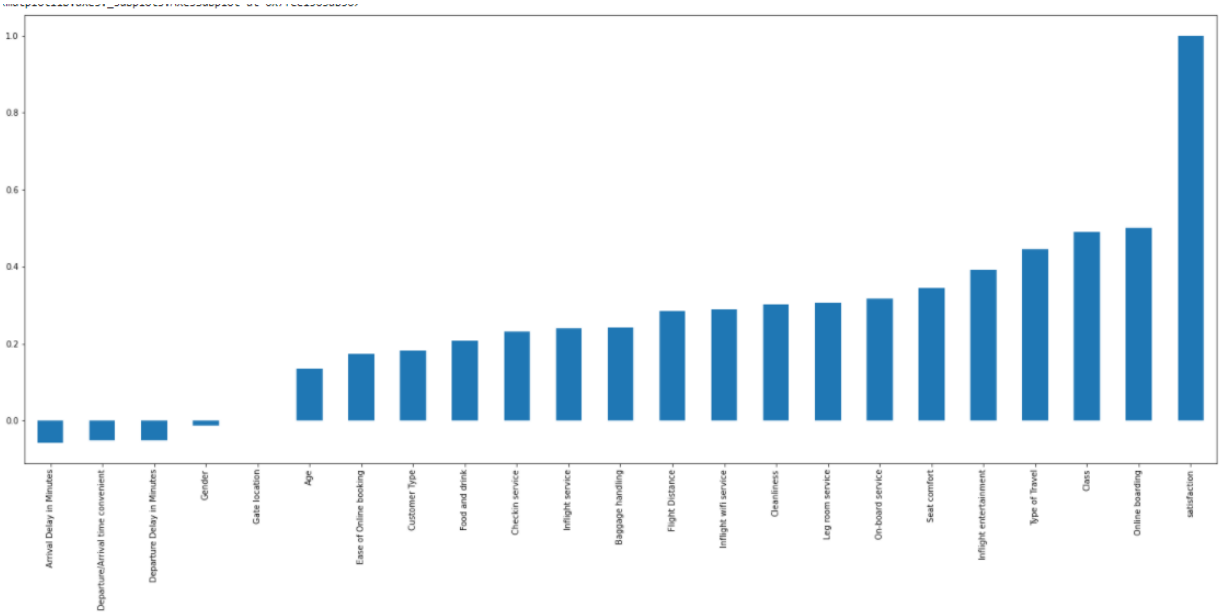


The graphic representation of the Outliers is like this. I'm getting rid of outlier values with a function I wrote myself. The logic of the function I wrote is to find a certain range and set it as an outlier.



Feature Selection:

Pearson correlation



I used the "Person correlation" method to observe the relationship of my target satisfaction attribute with other attributes.

Person Correlation analysis is a statistical method used to determine whether there is a linear relationship between two numerical data, and if so, what is the direction and size of this relationship. The negative places in this graph have an inverse relationship with the target satisfaction value of our project. If it is positive, there is a linear relationship.

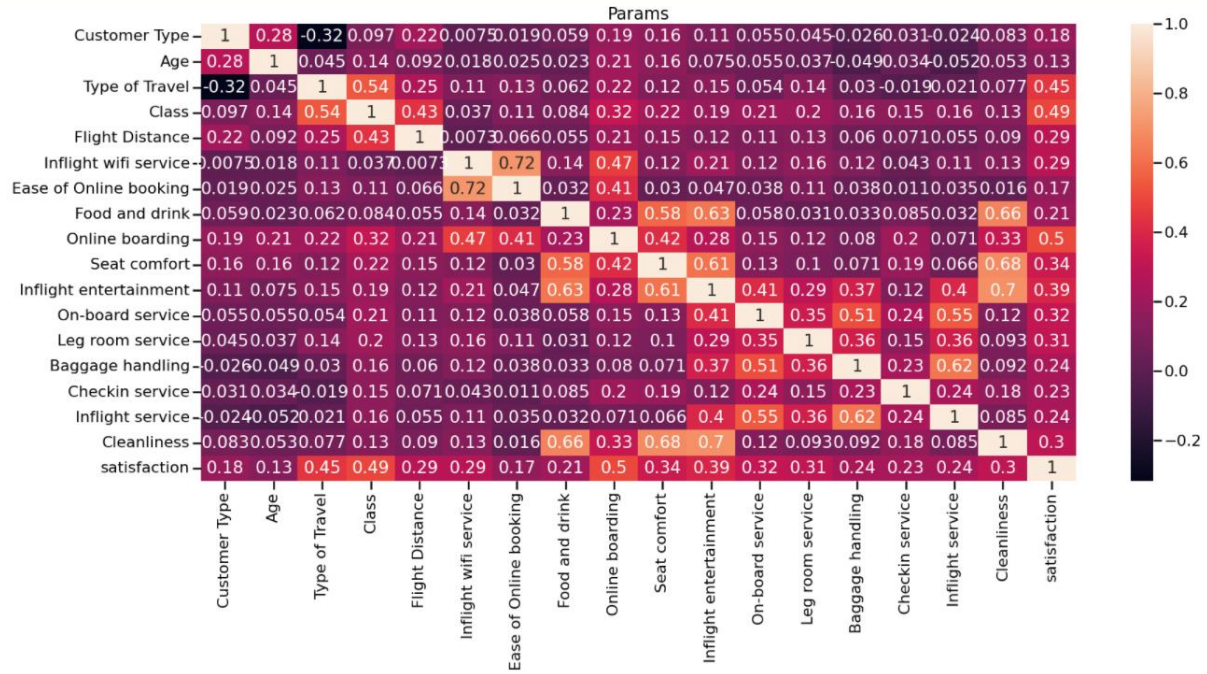
When I look at the chart, I see that the first 4 attributes are inversely related, and I remove them from my datasets.

The attributes I removed are "Arrival Delay in Minutes", "Departure/Arrival time convenient", "Departure Delay in Minutes", "Gender", "Gate location".

```
dftrain=dftrain.drop(columns=["Arrival Delay in Minutes","Departure/Arrival time convenient","Departure Delay in Minutes","Gender","Gate location"],axis=1)
dftest=dftest.drop(columns=["Arrival Delay in Minutes","Departure/Arrival time convenient","Departure Delay in Minutes","Gender","Gate location"],axis=1)
```

Correlation Matrix:

I am doing a correlation matrix, I learn more about data, I learn the relationship between param eters.



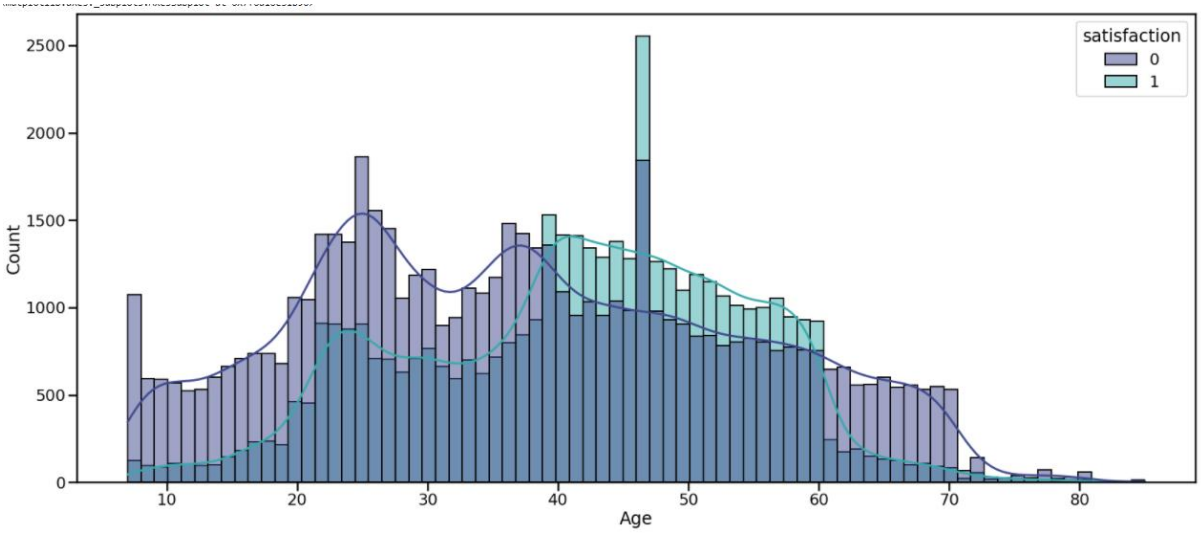
Let's take a look at a correlation heatmap to see which attributes correlate well with customer satisfaction.

Best features - Online Booking, Class, and Type of Travel

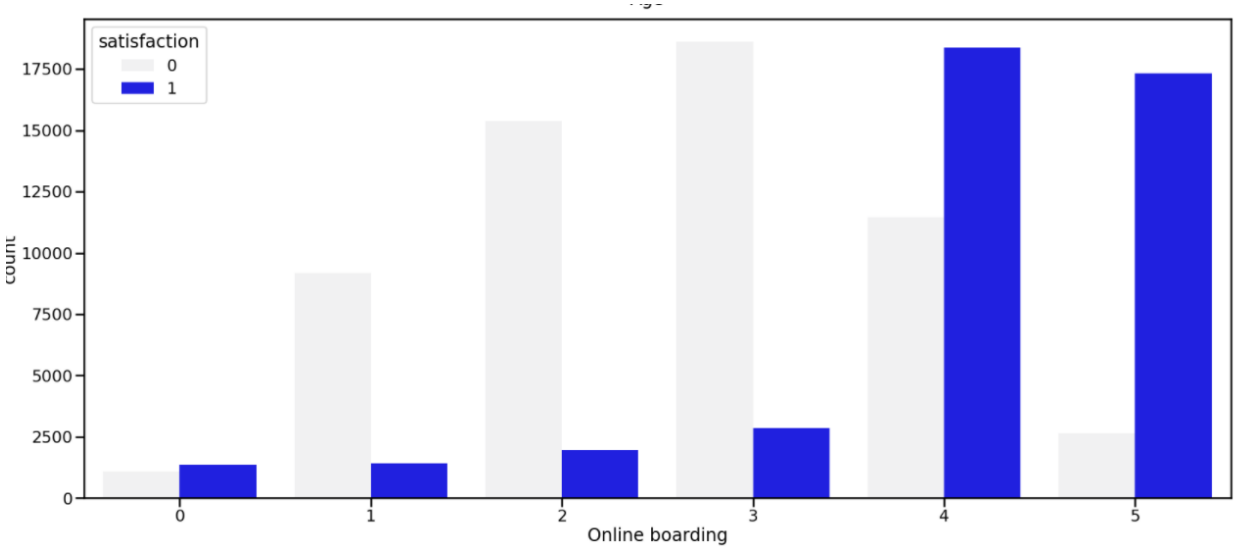
Worst features – Age, Ease of Online booking, Customer Type

I made graphs showing the relationships between the Target and some attributes to be predicted.

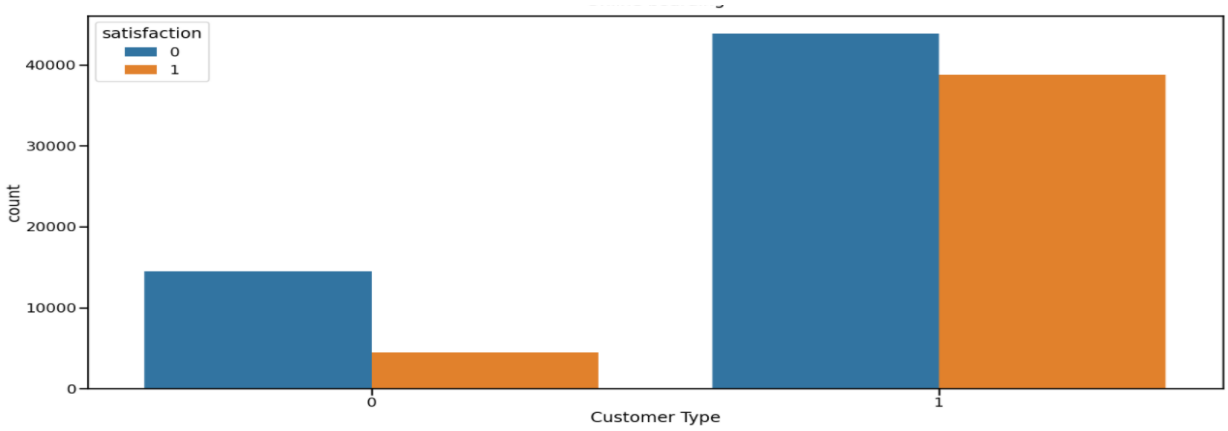
Age-Satisfaction:



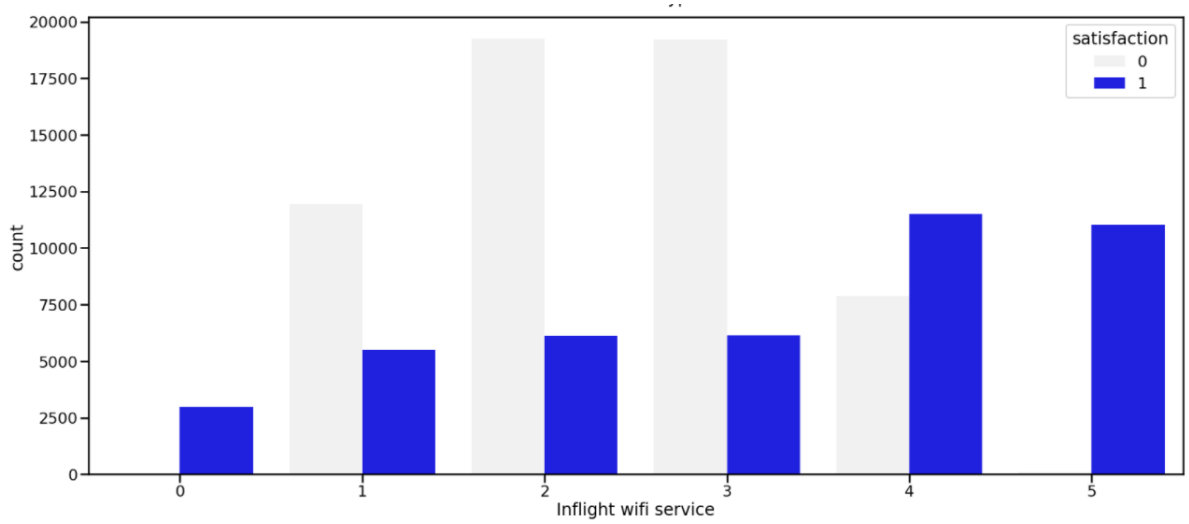
Online Boarding-Satisfaction:



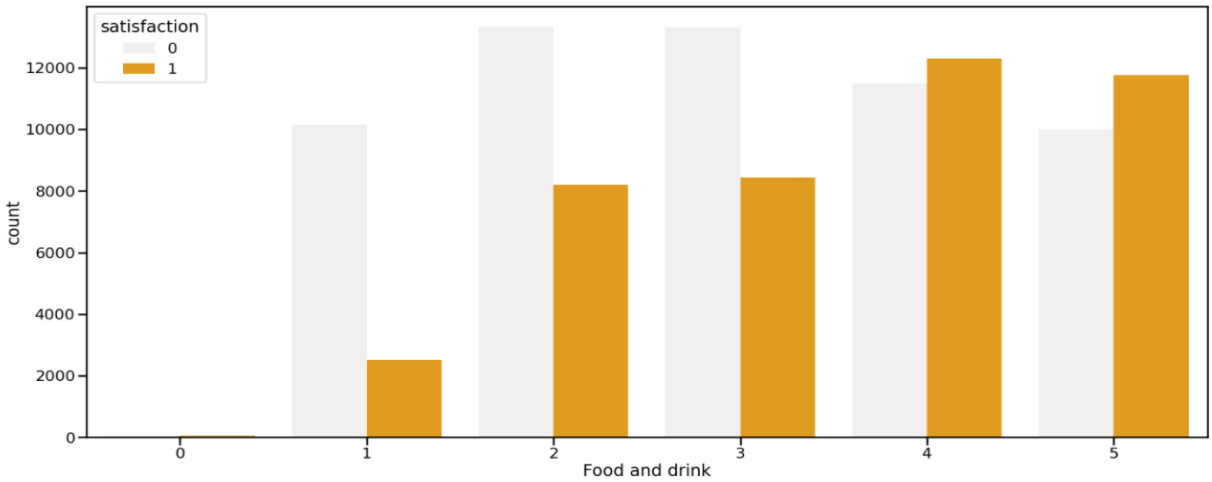
Customer Type-Satisfaction:



Inflight wifi service-Satisfaction:



Food and drink-Satisfaction:



The tables here can observe those who are satisfied or not satisfied with the attribute names under the table.

CLASSIFICATION:

In the project, I used Random Forest, Naive Bayes, Logistic Regression , XGBoost from classification techniques and extracted the results of these techniques over metrics. And I compared these results. Since our homework is on cluster algorithms, I chose my final project on classification.

Random Forest:

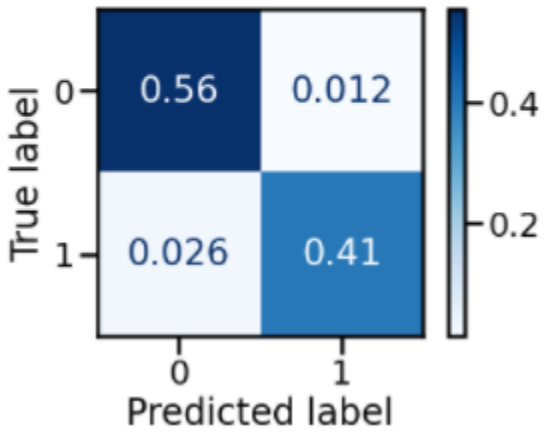
The Random Forest classification technique creates several decision trees and combines them to obtain a more accurate and stable prediction. This technique enables fast and automatic identification of relevant information from extremely large datasets. The greatest strength of the algorithm is that it collects(bagging) the results of many decision trees rather than relying on a single decision tree, and randomly selects (feature randomness) the attribute during its predictions.

I used it by implementing it from the Sklearn library.

Prediction Result:

ROC_AUC = 0.9593881257908613

	precision	recall	f1-score	support
0	0.95501	0.97956	0.96713	14432
1	0.97213	0.93922	0.95539	10957
accuracy			0.96215	25389
macro avg	0.96357	0.95939	0.96126	25389
weighted avg	0.96240	0.96215	0.96206	25389



Logistic Regression:

Logistic Regression is a regression method for classification. It is used to classify categorical or numerical data. It works if the dependent variable, namely the result, can only take 2 different values.

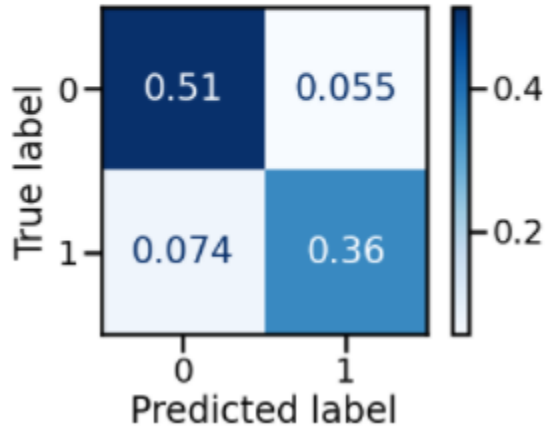
I used it by implementing it from the Sklearn library.

Prediction Result:

```
ROC_AUC = 0.8651701606127319
              precision    recall  f1-score   support

           0    0.87327   0.90292   0.88785     14432
           1    0.86615   0.82742   0.84634     10957

    accuracy                        0.87034     25389
   macro avg    0.86971   0.86517   0.86710     25389
weighted avg    0.87020   0.87034   0.86994     25389
```



XGBoost:

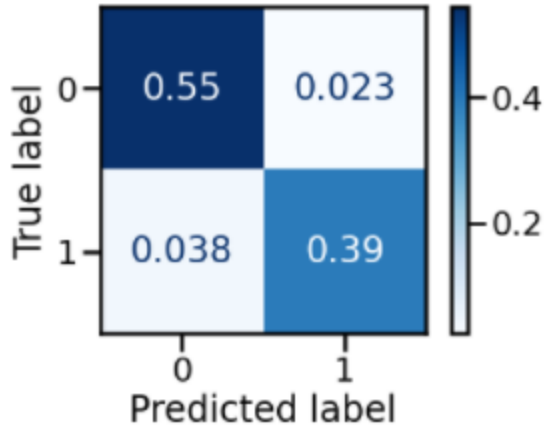
XGBoost (extreme Gradient Boosting) is a high-performance version of the Gradient Boosting algorithm optimized with various tweaks. XGBoost builds decision trees in all possible scenarios to maximize the earnings score for each variable. Such algorithms are called “Greedy Algorithm”. This process can take a very long time in large datasets. Instead of examining each value in the data, XGBoost divides the data into pieces (quantiles) and works according to these pieces. As the amount of parts is increased, the algorithm will look at smaller intervals and make better predictions. Of course, this will increase the learning time of the model.

I used it by implementing it from the Sklearn library.

```
ROC_AUC = 0.9351128400639711
              precision    recall  f1-score   support

           0    0.93426   0.95912   0.94653     14432
           1    0.94420   0.91111   0.92736     10957

    accuracy                        0.93840     25389
   macro avg    0.93923   0.93511   0.93694     25389
weighted avg    0.93855   0.93840   0.93825     25389
```



Gaussian Naive Bayes İmplementation:

Naïve Bayes classification aims to determine the class, or category, of the data presented to the system with a series of calculations defined according to probability principles. It is a statistical classification method. It is based on Bayes' theorem.

Bayes' theorem is an important subject studied in probability theory. This theorem shows the relationship between conditional probabilities and marginal probabilities within the probability distribution for a random variable.

Bayes Teoremi

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

P (A | B) = The probability of event A occurring when event B occurs

P (A) = The probability of event A occurring

P (B | A) = The probability of event B occurring when event A occurs

P (B) = The probability of event B occurring

Naive Bayes Classification Model:

A classification problem consists of many features and an outcome (target) variable. C represents the given target and F represents our properties. The naive Bayesian classifier is simply the product of all conditional probabilities.

$$p(C|F_1,...F_n) = \frac{P(C)p(F_1,...F_n|C)}{p(F_1,...F_n)}$$

Gaussian Naive Bayes: If our properties are continuous values, we assume that these values are sampled from a gaussian distribution or, in other words, from a normal distribution.

The way the algorithm works is it calculates the probability of each state for an element and classifies it according to the highest probability value.

I wrote my Naïve Bayes implementation in a class. In this class, I have methods named def Fit, def classify, def predict and def callikehood, def calPrior to help in calculations.

def fit:

```
def fit(self, X_train, y_train):
  self.X_train,self.y_train = X_train, y_train
  self.classes = np.unique(y_train)
  self.param=[]
  for i in range(2):
    self.param.append([])
    for row in X_train:
      param={"mean":row.mean(),"var":row.var()}
      self.param[i].append(param)
```

This function calculates the mean and variance of each feature for each class and add the mean and variance for each feature.

Def classify:

```
def classify(self,model):
    posts = []
    for i, c in enumerate(self.classes):
        post = self.calPrior(c)
        for attributeVal, params in zip(model, self.param[i]):
            possible = self.calpossibility(params["mean"], params["var"], attributeVal)
            post *= possible
        posts.append(post)
    return self.classes[np.argmax(posts)]
```

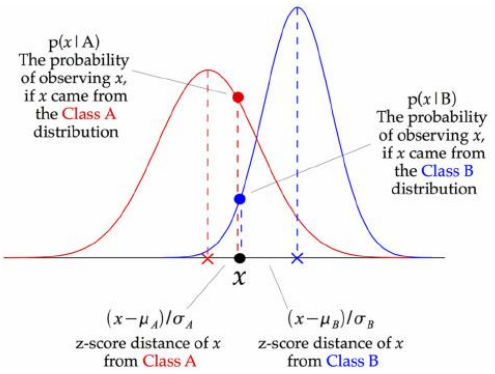
I explained above that Bayes rule is used in this classification technique. Classifies as the class that most likely results in this function. At first I calculate P(C). Then I calculate P(F) with all attributes. I keep these values in an array and then return the largest result.

Def predict:

```
def predict(self, X_test):
  y_pred = [self.classify(model) for model in X_test]
  return y_pred
```

This function takes the X_test data and sends it to the def classify function to perform the estimation.

Def callikehood :



$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

```
def callikehood(self, mean, var, x):
  eps = 1e-4
  coeff = 1.0 / math.sqrt(2.0 * math.pi * var + eps)
  exponent = math.exp(-(math.pow(x - mean, 2) / (2 * var + eps)))
  return coeff * exponent
```

Here is the function that performs the gaussian operation. Calculates the probability of a function occurring in a series using the mean and standard deviation of a series.

Project Demo Link:

https://www.youtube.com/watch?v=3iZ7FWd6YtQ

ARTICLE SUMMARY:

International Journal of Electrical and Computer Engineering (IJECE)
Vol.8, No.6, December 2018, pp. 5153~5161
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i6.pp5153-5161

Analysis of Mobile Service Providers Performance Using Naive Bayes Data Mining Technique

M. A. Burhanuddin1 , Ronizam Ismail2, Nurul Izzaimah3, Ali Abdul-Jabbar Mohammed4, Norzaimah Zainol5
Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia
Faculty of Science and Technology, Kolej Universiti Islam Melaka, Malaysia

In this research, the performance differences between mobile service providers in Malaysia were compared using twitter data. Naïve Bayes management, which is classification management, was used to compare these performance differences. Twitter data includes customer feedback to telecommunication service providers.

The method they use as data mining is CRISP-DM. They explained this method through diagrams. The reason they prefer this method is that this model is well known in data mining. The project plan they drew for themselves by adhering to this method is Data Collecting, Feature/ attribute Selection, Pre-processing, Training Data, Classification, Data Visualization. I can give the following information here, and I planned my operations in this way in my own project. They said this business plan is the best analysis method for telecommunication business operation. They wrote their programs in the R programming language. The data to be used during the test were obtained with the Twitter API. After analyzing the data they obtained, they used the Navie Bayes algorithm to reach the most accurate result. They used the Navie Bayes technique in 5 steps and first determined the test data. Since it was Twitter data, they analyzed it over words. They took out the Frequency table of the words. And they classified the sentences as positive and negative. As the third step, they calculated the priority values. As the fourth step, they calculated the conditional property of each feature. For example, they made calculations such as the probability of how many words are positive. They saved the Twitter data in csv format. They created separate datasets for each phone provider. They analyzed the files in their hands together with the model and observed the weights of the words as positive, negative and neutral. These results can be used among telecommunication companies to provide a better service to customers. Navie bayes was found to be successful as the classification method that works independently among the features. Visualizations were made with R libraries and the results were visualized.

As a result of this project, it has been shown how twitter data can be analyzed and which visuals can be used to become more understandable. At the same time, it has been shown what kind of competition can be between telecommunication companies in Malaysia and what are the customer returns.

REFERENCE

Dataset:

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

Classification Methods:

https://scikit-learn.org/

Gaussian Naive Bayes:

https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4

Article:

https://d1wqtxts1xzle7.cloudfront.net/64005710/40%204Jul%2011Mar%2012252-21997-1-ED-with-cover-page-v2.pdf?Expires=1642816722&Signature=Bh2-198miXkMz~QzXUg8QIp1kVNILp002hwYyHur1T58oMQCLZQXiSiwXhDGirvPcg9xusfSVWPeCR7pYfv~6eUbMp3WFSKQ3pzsVnEfkKdffUwhTNd38XBY7PImFLBPsNzM1alxRCR5vlWkDC4uG~AV1rtYZbW0vgj9DnDj-ol-ofMJ5zsrRW-~mV0~qcGaFKMAHeqUZMp281ZNLPf1TXkA7zvIr7S-53H2LFomcDghmHDsY8Zbt~aT64m7vuUDhoqtfoAvCvW-VUFJ1rPRh6UehviqfpKUKV2uE-RxcsP9aeN7vuiTWtOu7G2-aDg-K2dkOy0Ywv~nYSCVEGboew__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA