12-2022

Predicting & Optimizing Airlines Customer Satisfaction Using Classification

Mhd Ridwan AlHabbal
mga1863@g.rit.edu

Predicting & Optimizing Airlines Customer Satisfaction Using Classification

by

Mhd Ridwan AlHabbal

A Capstone Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Professional Studies:

Data Analytics

Department of Graduate Programs & Research

Rochester Institute of Technology
RIT Dubai
December 2022

RIT

Master of Science in Professional Studies:

Data Analytics

Graduate Capstone Approval

Student Name**: Mhd Ridwan AL-Habbal**

Graduate Capstone Title**:**

Predicting & Optimizing Airlines Customer Satisfaction Using Classification

Graduate Capstone Committee:

Name:	**Dr. Sanjay Modak** **Chair of committee**	**Date:**
Name:	**Dr. Ehsan Warriach** **Member of committee**	**Date:**

Acknowledgments

I would like to embrace this opportunity to express my sincere thanks and gratitude to my mentor professor Ehssan Warriach for his endless support and guidance in this project and his vital points and valid remarks that made this project successful. Furthermore, I would like to express my deepest appreciation to chair of committee Dr. Sanjay Modak for his continuous support and guidance towards guiding the whole department and helping students achieving their goals and have solid education background. I also would like to thank all professors in Dubai and New York campuses who taught me throughout this journey. I really did enjoy my journey at RIT Dubai, and they really made it a success story. Finally, I would like to thank my colleagues and family for their contribution and support during this master's degree journey.

Abstract

This research is going to be a machine learning project that aims to study the various factors that may play a role in forming customer satisfaction response and tries to figure out which attributes or combination of them are the driver of positive customer satisfaction. The research is going to use initially some dataset from Kaggle (explained in the section of data source) in order to run machine learning algorithms and creating a predictor that would help airlines in predicting which customers are satisfied and trying to have a proactive reaction in case of negative feedback, so we can make it up to the annoyed customer and get him satisfied. The research is going to examine several classification algorithms and tries to tune them in order to get the best result. Then will do experiments on resulting models and tries to find the optimal one among the others.

Keywords: Machine Learning, Classification, CRISP-DM, K-Nearest Neighbor, Support Vector Machine, Random Forest, Logistic Regression, Confusion Matrix, Artificial Neural Network.

Table of Contents

Table of Figures

Table of Tables

List of Acronyms

Acronym	Description
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
AUC	Area Under Curve
ROC	Receiver Operating Characteristic curve
RF	Random Forest
LR	Logistic Regression
CM	Confusion Matrix
CSV	comma-separated values
ME	Mean Error
MAE	Mean Absolute Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
CRISP-DM	Cross Industry Standard Process for Data Mining
ANN	Artificial Neural Network
DNN	Deep Learning Neural Network
CNN	Convolutional Neural Network
CNN-LSTM	CNN-Long Short-Term Memory Network
ML	Machine Learning
MLP	Multilayer Perceptron
NLP	Natural Language Processing
SMO	Sequential Minimal Optimization
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
EDA	Exploratory Data Analysis
RFE	Recursive Feature Elimination

Chapter 1

1.1 Introduction

Customer satisfaction is an essential target for any organization especially for commercial ones. It is a metric for both business owners and business marketers to check if their services are performing well from customers perspective. It could also let you know if the customers are willing to come back repeatedly. In addition, it could give warning signs that the customers are unsatisfied and potentially at risk of quitting consuming your services. As a result, customer satisfaction metric provides extremely vital information to realize if they are moving in the right direction or not. Airlines industry is not any exception of this theory.

1.1.1 Pandemic Challenge

Airlines are on the cusp of facing their third year dealing with the Coronavirus pandemic. The restrictions and challenges are still exist, and government regulations and restrictions are still effective. Although we can observe remarkable ease of restrictions compared to 2020. However, the challenges have continued relentlessly, ranging from new variants of the virus, to shifting government policies on travel restrictions and testing. The impact on customer confidence and on airlines' ability to plan and operate predictable schedules has hit revenues and finances severely. Such hectic conditions reduce the number of passengers and raise the level of competition between various airlines around the globe. Hence, customer satisfaction is highly important especially in such period of time, and there is a real need to forecast what will satisfy the customers, and how to get them back over and over again.

1.1.2 Customer Satisfaction factors

Customer satisfaction in airlines industry is not an easy thing to have. Compared to other industries, we can easily find that with airlines it would require a lot of hard work to achieve customer satisfaction goal. The service is starting from the departure airport and facilities it has, the scheduled timing of the trip, and whether it is convenient or not. online/mobile boarding is important, baggage handling, check-in, inflight services, and delay in either departure or arriving. The case is highly critical, and it really needs creative solutions to manage using newest technologies.

1.2 Project goals

This project goals to help airlines to achieve the desired customer satisfaction by allowing them to focus on most important things that might be the most highly desirable thing by the customer. Its main goal to answer two questions:

1. Predicting whether a customer is satisfied with the overall service provided by the airlines.
2. Focusing on the most important factors that leads to customer satisfaction, and as a result more customers to come back on the same airlines.

As a result, it tries to answer those questions.

1.3 Aims and Objectives

For long decades, airlines were competing together in order to get more customers to achieve more revenues. To do so, airlines were doing their best to offer the best services at best price to win the competition. However, due to variety of people and having different aims of their travel and interacting a lot of factors in such process such as airports, customs, immigration… etc. it is not always straightforward to achieve such goals. This project aims to utilize machine learning techniques to help companies focusing on main factors that help to gain customer satisfaction and to be proactive and predict if a customer is satisfied or not, so they can give extra benefits or offers in order to make it up to him or her.

1.4 Research Methodology

The Methodology would depend on Machine Learning techniques using R Programming language and Tubule for visualization

I will follow CRISP-DM methodology with the following phases:

1. Business understanding – Deep understanding of the busing people expectations?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What are the modeling techniques that we should apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – The last phase to put the solution in production for stakeholders?

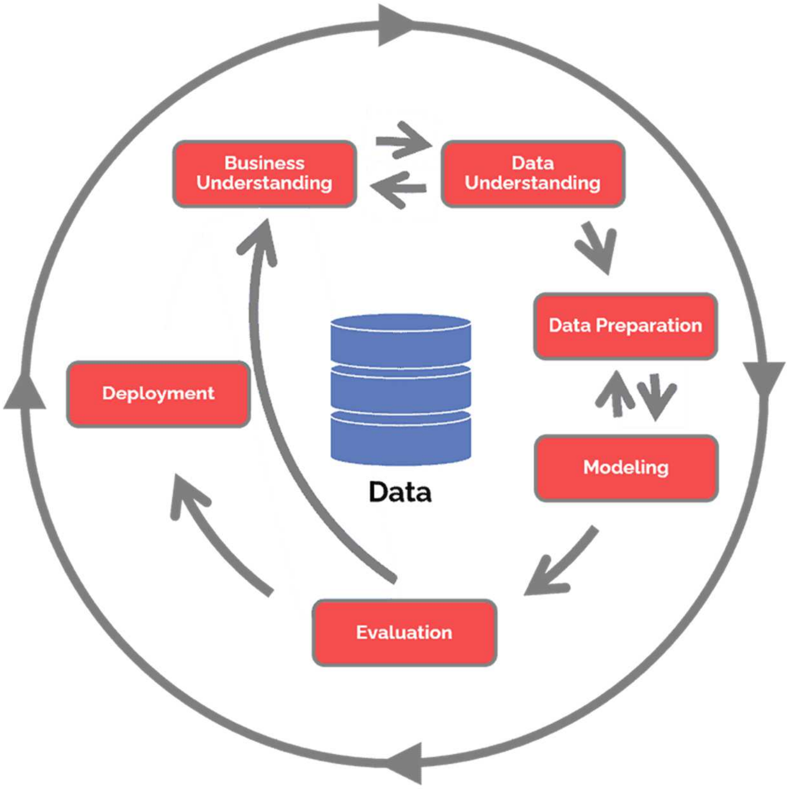


Figure 1 CRISP DM Methodology

1. **Business Understanding:** During this phase, we will try to understand the whole conditions that might participate in affecting the customer satisfaction and might result with negative impression and dissatisfaction or might encourage the customer to come back over and over again. Having such business understanding of all possible parameters that may intervene in shaping the customer opinion and how he might make a decision to book repeatedly would lead to have realistic results. The resulting model might have great results; however, it could not be valid because the actual variables are not there. So, it is vital to focus on all potential circumstances that might face the passenger during his trip.
2. **Data Understanding:** In this stage we will examine our dataset and try to check if it covers the whole set of figures that might affect customer satisfaction. The dataset is informing when the customer is satisfied along with all other variables such as in-flight services, airport facilities, and whether the timing was convenient or not. We need to distinguish the categorical variables from numerical variables.
3. **Data Preparation:** This is the step when we start working on our dataset. We need to keep in mind, that data preparation is essential in producing and good model. Having a lot of NAs and inaccurate data would lead to abnormal result. We will perform:

   Data Cleaning: we will detect NAs and trying to reduce them by predicting possible results using different ways, in addition, we might handle outliers to have good model.

   Data Transformation: We will be doing normalization of data and scaling the attributes wherever it is necessary to do so.

   Data Integration: The data is going to be supported and integrated by several data whenever it is needed to do so.
4. **Modeling:** once data is prepared, so we can start building the prediction model. We are going to build different kind of models and we will focus on classification algorithms in order to predict the customer satisfaction. We are using different models to evaluate later and check which one is giving the most convenient and accurate results.

   In this step we will also build the online analysis and batch analysis to allow different kind of analysis to our model
5. **Evaluation:** Once we have finished building our models, it is time to evaluate them and try to optimize them. Various methodologies would be used to evaluate the models that would help us to make sur that the model is the best and is tuned to be performing very well.
6. **Deployment:** The last step is deployment. Once we finish all activities and make sure the model is optimized and has good results. Moreover, the results are fine with the business needs, then it is time to push it to production. The ability to perform online analysis and batch analysis is essential in the deployment process.

1.5 Limitations of the Study

The study tends to be neutral and give guidance to the airlines about the major factors that really leads to customer satisfaction. As a result, they would be able to focus more on such factors. However, there are other dimensions that could be considered. Such factors could be the region of the world, purpose of travel, and other factors that may play role in customer satisfaction. These other factors have to be taken into account. However, due to limitation of the dataset and time, we will limit our study to the boundaries listed above.

Chapter 2 – Literature Review

2.1 Introduction

In this section, we give outlines about the most remarkable previous studies and research. There are many several approaches that target to study customer satisfaction and particularly for airlines customer satisfaction. Some of researchers tend to measure customer satisfaction based on twitter sentiment analysis. Some others went deeper to check how much the reviews are beneficial in terms coming back customers.

2.2 Literature Review

Hwanga, Kima, Parka & Kwonb (2020) tried to answer the following question: "Can users' affective expressions on airline services be useful in estimating their return visits to the services?" It is not mandatory at all that if we found positive feedback about an airline that the customer is going to come back. They conducted a survey to check feedback from people who booked their flights online, they collected 309,331 customer's feedback. They validated these feedbacks and dropped invalid ones. In addition, dropped the responses of delayed or cancelled flights. Based on these procedures, 178,951 responses (57.9%) were validated. One year later, in November 2018, the customers with validated responses were asked if they used airline services for the past year. "The customers who responded positively (133,872 responses, 43.3%) were subsequently asked if they used the same airline service as stated in the initial survey. Then they performed several machine learning classifiers, The seven classifiers, *Decision Tree*, *Gaussian Naive Bayes*, *Logistic Regression*, *Random Forest*, *KNeighbors*, *Support Vector Machine*, and *XGBoost.*

"The study found out that both emotional and linguistic aspects of users' comments on a particular service are valuable when predicting and determining their future behavior related to the service". "In addition, the results of predicting return visits showed relatively high accuracy regardless of the length of reviews".

Ulkhaq, Adyatama, Fidiyanti, Rozaq, and Raharjo in (February 2020) conducted a study to predict customer loyalty to online travel agency (OTA) using the artificial neural network (ANN) approach. People habits has changed these days. No body nowadays are attending to a shop to buy travel services. People these days are using online websites to buy travel services in particular. So, it is vital that online travel agencies are performing well in order to gain customer satisfaction. The study main goal is to predict customer loyalty and satisfaction and possibility of revisiting once again or giving referral based on six dimensions. Each dimension consists of several attributes. The dimensions are: (1) ease of use (2) Security/privacy (3) information/content (4) Responsiveness of the website (5) visual appeal. (6) "fulfillment. It refers to the delivery of products and/or services within a service level promised".

Artificial Neural Network ANN was the approach used to make the prediction. "A quantitative-based survey was conducted to accomplish the objective of the study". The survey has three main parts: first is collect data about demographic data about the respondents, the second part asks for 31 attributes about the website quality under the six dimensions mentioned above. The customer may give value from 1-5 where 5 is the best evaluation and 1 is the worst. The third last part asks if the customer would revisit/recommend/ or provide positive referral to others. The results were compared to logistic regression model. As a result, the results of ANN, according to the study, was much better than the results of logistic regression in every dimension, since the results have higher accuracy and lower RMSE values. We have to keep in mind that all of the six dimensions mentioned above are effective in the study. The order of top three dimensions below is in descending order stating from the strongest dimension.

1. Information/content
2. Security/privacy
3. Visual appeal

Overtveld & Balsingh (2019) conducted research using DNN to **predict errors in delay estimations** of airlines trips in AMSTERDAM AIRPORT SCHIPHOL. Airports usually announce in advance that some aircrafts are delayed, but many times that announcement is not accurate. Predicting the error in delay estimation aims to reduce the actual delay and set it to minimum. As a result, reduce the cost of delay and increase the customer satisfaction which is the subject matter of this proposal. There are several classifications of the delay. Flight phases must also be defined to determine the type of delay. To predict flight delays, studies were conducted on the distributions (Normal & Poisson distribution) of the different flight phases to predict flight delays. As a result, it is essential to predict the error in initial delay estimates to reduce flight delays. To do so, they used Deep Learning neural network (DNN). The dataset contains the predicted/actual time of arrival & departure times. In addition, there is new message of delay prediction message is sent, and the new prediction of departure /arrival time. They used feed-forward neural network because the given flight data is not as complex as speech. They built two different networks, one for departure and one for arrival. However, both of them have the same structure. "The input layer takes the standardized feature columns from the processed data frame, and the output layer gives a predicted error on the expected delay". The study concluded that the estimation of error in departure is much lower than the error estimation in arrival. This is because of the following three justifications:

1. Time for departure is much shorter than arrival time. As a result, it would be much easier to predict, with less error, the departure time than to predict arrival time.
2. In departing phase, the aircraft is still on the ground and is not as prone to weather changes or other, random, or environmental causes of delay.
3. Since operations on ground are automated, so these supports having low variations in the data due to the consistency of machines. As a result, it is likely to have different in accuracy in prediction of times of arriving and departing.

Kaur (2021) has issued a paper discussing sentiment analysis derived from tweets. The target of the research is to check the sentiment analysis of the airline customers and check if they have negative or positive feedback. He had a dataset used in this paper is taken from Twitter. He got 11533 tweets are there in this dataset about Six Airlines It has 2361 number of Positive views and 9172 number of negative views. After removing null tweets and removing stop words as well as preprocessing the tweets he ended by clean set of tweets and then applied Support Vector Machines SVM, and Random Forest. Then validated and verified by the Naïve Bayes algorithm. As for results, he depended on the factors such as: Precision, Recall and F1- score of various classifiers. He found that combined algorithms resulted the best performance over all other classifiers. In details: he found SVM giving the best results for binary classification and the accuracy was over 90%, as well as Random Forest. However, when he used the combined approach, he got a great accuracy that was 98%.

Kumar & Zymbler (2019) conducted research to check sentiment analysis of tweets published by passengers after the trip. The study aims to check “if there is a continuous trend of negative tweets for an airline, then it may put a negative impact to the economic growth of the airline company”. “it is important to understand the issues that give rise to negative tweets so that the respective airline company can take appropriate action on time”. As the number of tweets is very big, so it is important to use big data technologies as well as machine learning techniques. Support Vector Machine (SVM) and Artificial Neural Networks were trained on the reprocessed tweets. In action, they used convolutional neural network (CNN) and compared its results to best model among SVM and ANN models. “In addition, association rule mining is used to map the relationship between several issues related to passenger’s comfort during flight with the nature of emotions (positive or negative)”. The total number of tweets for different airlines () was 146,731. AMA (American Airlines) has the highest number of tweets that consists of 44.13% of all tweets downloaded. After pre-processing, and deleting retweeted ones, they got 120,766 tweets. They tested ANN with different configurations and the final result is that they got better performance for both SVM (76.5%) and ANN4 (79.4%) model. 4 in ANN4 is the number of configurations used for ANN. CNN performed well for both training and test dataset. It achieved an accuracy of 92.3%. For context of the tweets and the association rules in tweets, they considered 7 dimensions (6-word categories and 1 additional sentiment category: 1-positive, 0-negative). The results shows that the Cabin Crew Behavior (CCB) then Food Quality (FQL) have the major impact on customer’s feedback. The lowest effect is for dimension Loss of Baggage (LOB).

Park, Kim, Kim & Park (2022) identified that most of the studies that dealt with airline services capes impact on airline customer propensities so far have limited the human service to only the cabin crew. Therefore, there is a limitation in that the derived results are limited to the linear relationship. “The ultimate purpose of this study is to connect airline customer propensities with brand loyalty by extending the human service to the viewpoint of passengers”. They conducted a survey of the effect of airline services cape on customer churn risk and satisfaction. “A total of 340 Korean adults, who have used airplanes at least one time within the last five years, those persons responded to the 50 questions related to the physical and social environment of the airlines, brand experience, brand loyalty, and customer satisfaction.” The algorithms used are KNN and decision tree, ensemble learning models, such as RF and XGBoost, and deep learning models, such as CNN and CNN-LSTM. As for results, “**The first goal is to determine the most accurate machine learning and deep learning models for the prediction of airline customer propensities”**. For machine learning models, the RF model achieves the highest accuracy of 84% and 86% in predicting customer churn risk and customer satisfaction, respectively. For deep learning models, the CNN-LSTM model achieves the highest accuracy of 94% and 90% in predicting customer churn risk and customer satisfaction, respectively.

"The second goal is to investigate the influence of different airline servicescapes on the accuracy of machine learning and deep learning models". For machine learning models, the prediction accuracy of the KNN model jumped significantly from 74% to 84% in predicting customer churn risk and from 76% to 84% in predicting customer satisfaction when considering both physical and social servicescapes. For deep learning models, the CNN-LSTM model achieved the most significant improvement of prediction accuracy (i.e., from 87% to 94% in predicting customer churn risk and from 81% to 90% in predicting customer satisfaction) when considering both physical and social servicescapes.

García, Florencia-Juárez, Sánchez-Solís, Rivera-Zarate & Contreras-Masse (2019). They demonstrated ensembles of regression techniques. The ensembles models have emerged because there is not an overall best algorithm for dealing with a problem. As a result, it has been demonstrated that ensemble models perform better than single prediction methods in classification/regression problems. The problem they are trying to resolve is that customer satisfaction by exploring the use of ensembles of regression models. They analyze the performance of the k-nearest neighbor (k-NN) model for regression as base classifier in the BAGGING (Bootstrap AGGregatING) ensemble model. You could check https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ for more outlines about the ensemble learning algorithms. They conducted the research over real-database constructed from 129,889 surveys supplied by several airline companies. "By combining individual regression models, the ensemble approaches aiming to minimize the error on problems where the output variable is continuous." The results show that the individual regression model achieves the highest error values ($RMSE$ ≈ 0.7985, MAE ≈ 0.6365). Whereas, for ensemble regression models the error values ($RMSE$ ≈ 0.7650 and MAE ≈ 0.5664 when number or regression models is set to 30). As a result, "customer satisfaction prediction problem can be handled better using ensembles approaches than single models. Like- wise, when the number of base models is increased, the error decrease".

Lucini, Tonetto, Fogliatto & Anzanello (2019) submitted a paper that uses text mining techniques to explore **dimensions** of airline customer satisfaction from the analysis of Online Customer Reviews explore Online Customers Reviews (OCRs) to help airlines to increase competitiveness. The process of this paper has five main points "(i) identify and extract dimensions of customer satisfaction expressed in OCRs; (ii) verify the distribution and importance of those dimensions in OCRs from different groups of airline customers; (iii) identify and extract adjectives used to describe perceptions in those dimensions and calculate the adjectives' sentiment scores; and (iv) test and validate the dimensions and adjectives in (iii) through regression analysis"

Eritier, Bocamazo, Delahaye & Acuna-Agost (2019) have introduce the Multinomial Logit (MNL) models that were used to help airlines and travel agencies to adapt offers to market conditions and customer needs. MNL model has good simplicity and readability, However, their disadvantage is the lack of flexibility to handle collinear attributes and correlations between alternative offers. To resolve such issue, they introduced machine learning model that is based on non-parametric model to segment customers automatically by taking in consideration the non-linear relationship between attributers of alternatives and characteristics of the decision maker. They used Random Forest for the Machine Learning approach. The did two experiments: one by using alternative features only, and another by including attributes to model individual heterogeneity. The results of the machine learning model exceeded the results of MNL models in terms of prediction accuracy and computation time.

Ustebay, Yelmen & Zontul (2019) performed a study to group airlines customers into specific sections, so an airline can manage responses based on customer segmentations. It also aims to guid companies to behave and react appropriately to customer and not waist money and effort The study used data from an airline firm between 2017 and 2018. They tried to find ML algorithm to group customers into clusters based on similar sales tendencies. They have used K-Means++ algorithms. For newly customers (unknown) have used K-Nearest Neighbor and Random Forest classification algorithms to classify them and to assign them the right cluster. They used attributes such behavior, Socio-Demographic, Geographical, Lifestyle, Needs-Based, etc. in addition to that, they used: number of bought flights by a customer annually, semi-annual, or quarterly periods, the amount of tickets purchased and the total mileage they fly etc. The original dataset contains 3,232,527 observations. As 65% of records contain missing values or unaccepted values such negative values, so the total records remained are 1,099,934. The results showed that **3** clusters are the best number of clusters based on the Elbow method, also they used Random Forest and KNN classification algorithms. The percentage of 70%/30% for training/testing was used, and the accuracy was 95%.

W Baswardono, D Kurniadi, A Mulyani and D M Arifin (2019) tried to make Comparative analysis of decision tree algorithms. This analysis is to compare results from two different machine learning algorithms (Random Forest and C4.5) for airlines customer satisfaction classification. In order to do so, they selected the same dataset for customer evaluation of the airline's services. After that, they created different models using machine learning algorithms and compared the results. The comparative analysis was made using three splits (training | testing) of data. The first is 70|30, then the second split 80:20, and the last is 90|10. For both algorithms the last split had the best accuracy, but practically all are the same because the difference was so small. Accuracy for Random Forest was (93.30, 93.31, 93.32) for the three splits, and for C4.5 the accuracies were (92.21, 92.22, 92.55). As a result, we can conclude that the accuracy of Random Forest is a bit higher than C4.5.

Ann-Nee Wong, Booma Poolan Marikannan (2020) tried to study the key factors that lead to customer satisfaction. Although this paper is not directly related to airlines. However, its core study is to know which factors do result into customer satisfaction. In other words, they are trying to answer the question: what are the key drivers which influence the customers and predict the likelihood of satisfaction. In order to create a machine learning model, they used a dataset from multiple sources to apply Decision Tree, Random Forest, Support Vector Machine and Artificial Neural Network algorithms. The dataset contains attributes of the order identification followed by the fields of customer satisfaction. The ranges of accuracy of the four algorithms ranges from 87.0% to 87.5% even with various data pre-processing methods and feature engineering. Random forest was the best in terms of accuracy. Delivery performance was marked as number one important factor in customer satisfaction. Where purchase_delivery_days came as the second factor.

Moulay smail Bouzakraoui, Abdelalim Sadiq, Abdessamad Youssfi Alaoui (2020) tackled customer satisfaction using an extremely advance method. Most of the other researchers did that using regular sentiment analysis over tweets or reviews. However, who said that all people leave comments, and how could we really know that the reviews are not hiding something. They tried to check satisfaction based on Facial Expressions. They depended on extracting geometric features of the customer's faces. They calculated the distances between landmark points and used distances between the neutral side and the negative or positive feedback. Once they finished collecting the data, they applied several classifiers, namely Support Vector Machine (SVM), KNN, Random Forest, Adaboost, and Decision Tree. They verified the algorithms against JAFFE dataset. it turned out that the best performance was using SVM with an accuracy over than 98%.

HARI MOHAN PANDEY, Prayag Tiwari, Aditya Khamparia, Sachin Kumar (2019) used tweeter as a source of tweets for identifying the feelings and emotions of customers about the airplane services using machine learning models. The target of their study is to have sentiment analysis about the tweets and classify them as positive, negative, or neutral. They used only two classification algorithms. Namely: Random Forest (RF) and Logistic Regression (LR). They first prepossessed the tweets by removing unnecessary stuff such hashtags, punctuation, and stop words. Then they stemmed the words and lower them to lower cases. The number of tweets were 14500 ones of passengers of US airlines. They applied the two algorithms then and performed K-fold validation due to the limited dataset. The results of the classifiers were promising they got precision or 80% and 82% for both LR and RF respectively.

So-Hyun Park, Mi-Yeon Kim, Yeon-Ji Kim, and Young-Ho Park (2022) tried to answer two questions. First question: what is the relationship between various factors that may influence airlines passengers. Second Questions: what is the influence of social servicescape on the propensity of the airline's customers. They applied deep learning techniques such as KNN and decision tree, Extreme Gradient Boosting (XGBoost), ensemble learning models, such as Random Forest (RF). In addition, they used, deep learning models, such as CNN Long Short-Term Memory Networks (CNN-LSTM) and Convolutional Neural Networks (CNN). As a result of evaluation, the result shows more accuracy in using deep learning models compared to machine learning models. The accuracy increased by 9% to 10% in the prediction of customer satisfaction and churn risk respectively.

P. Rethina Sabapathi; K.P. Kaliyamurthie (2022) conducted research to find the feedback on certain products on Amazon such as Amazon kindle app, fire TV stick and more. The analysis is based on text mining and sentiment analysis. They corrected the reviews, pre-processed them by removing the HTMLs and hashtags …etc. then did sentimental analysis extraction by specifying the tags for the words. After that they did sentimental scoring. This is done by giving 1 to the most positive review, and -1 for the most negative review and 0 for the neutral review. If a review contains mixed lines, then there is a formula to calculate the sentiment score and then the ratio of it. Once they finished all these steps, they did

Sentiment polarity categorization. The results of the research said that 45% of the reviews are positive, whereas 25% are negative. On the other hand, 30% of the reviews are neutral.

Yung-Chun Chang, Chih-Hao Ku, Duy-Duc Le Nguyen (2022) conducted a study that aims to study the lack of consumer confidence in airlines industry, even when flights started to resume after COVID-19 pandemic. The lack of confidence is because the restrictions of COVID-19 incurred a lot of flights delays and cancelation, and these two issues are the core of airline industry failure. The methodology is to collect a considerable amount of reviews (191,123 reviews) on Tripadvisor.com of the top 10 ranking airlines in 2020. They used text mining and sentiment analysis. They found that the comments and reviews were great till February 2020. However, after the pandemic announcement by WHO. The is a remarkable decline in customer confidence. The results of the study of the reviews shows that the negative reviews raised for all aspects in the recent years, and especially for year 2020.

Eyden Samunderu, Michael Farrugia (2022) conducted a study to help airlines predicting the purpose of travel. Understanding the purpose of travel (business or leisure) is essential for customer of airlines because it has impact on the elasticity of the passenger regarding the price of the trip. To perform the task of understanding the purpose of travel, the study had to create a predictor to guess what the purpose of travel is. The second thing to do is to perform segmentation of travelers to understand each group properties. The algorithms used for the predictor were Logistic Regression, Decision Trees, SVM, as well as Random Forest. The last one won the performance over all other classifiers with 100 trees. The Gini index was used for splitting. For clustering, they used k-means.

Kun Gao, Ying Yang, Xiaobo Qu (2021) submitted a research based on quantitative data to find out which factors of airlines services are most important to the majority of passengers. In addition, they wanted to predict the satisfaction of the passenger. They grouped the factors in the following groups: pre-flight, during flight, post-flight, and other attributes. To resolve such problem, they applied several machine learning algorithms such as: Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Multi-layer Perceptron (MLP), and Random Forest (RF). The most important feature was Type of travel. Then inflight Wi-Fi service. The third was customer type.

Wajdi Aljedaani, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, Imran Ashraf (2022) had a deep study to reveal the sentiment feedback about services of six American airlines companies using tweets. Although that approach was used before. however, it lacks the accuracy, and it tries to perform this analysis using a hybrid sentiments analysis approach. In other words, they are using lexicon-based methods along with deep learning models in order to improve accuracy. They introduced TextBlob for annotation. TextBlob is lexicon-based library model. It is used for sentiment analysis. It could be used in Python to provide simplified text processing. TextBlob is used to avoid any subjective manual annotations provided by the experts. The algorithms used are deep learning models. Specifically, they are GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and CNN-LSTM augmented with lexicon-based technique TextBlob. This is in order to check and investigate how could TextBlob method affect the accuracy of the classification models. The models results says that the model could perform in a better way when these models are trained using the TextBlob assigned sentiments. This could be noticeable when compared to the original sentiment dataset. The best algorithm was LSTMGRU in its performance. It scored highest 0.97 accuracy and 0.96 F1 scores against other studies. They concluded that TextBlob annotation can not replace humans. However, TextBlob-annotated labels can assist human annotators to avoid bias, subjectivity, and error-proneness generated by purely expert human annotators.

Balasubramanian Thiagarajan, Lakshminarasimhan Srinivasan, Aditya Vikram Sharma, Dinesh Sreekanthan, Vineeth Vijayaraghavan (2017) had a study to predict delay in flights in order to enhance airlines customer satisfaction. They realized that such delay would be inconvenient and would be a main cause of customer unsatisfaction. The problem they were trying to resolve had two parts. First part would a delay occur? so this was a binary classification problem. The later part of the problem was about predicting continuous value using regression. In other words, they wanted not just to predict if there would be a delay, but to predict how long the delay would be. In terms of data, they used historical data for 5 years based on flight schedule and weather data.

For classification they used the following classification algorithms:

A. Gradient Boosting Classifier
B. Random Forest Classifier
C. Extra-Trees Classifier
D. AdaBoost Classifier

In addition, for regression they used:

A. Extra-Trees Regressor
B. Random Forest Regressor
C. Gradient Boosting Regressor
D. Multilayer Perceptron (MLP)

In the classification stage, Gradient Boosting Classifier was the best classifier (accuracy 86.48%) and in the regression stage, Extra-Trees Regressor performed the best (accuracy 93.73%).

Siavash Farzadnia, Iman Raeesi Vanani (2022) tried to study customer satisfaction in Middle East airlines (Emirates, Flydubai, Etihad, Oman, Saudi, Gulf Air, Kuwait, Qatar, Royal Jordanian, Turkish airlines) using sentiment analysis. The problem they try to resolve is to identify the trends of opinion of customers of those 10 airlines. The methodology used is to cluster the main topics at the beginning of the study. Then, we identify the customer feedback against each of the identified topic for each airline. The sentiment analysis score would be reported for each service for each carrier. For topic clustering (unsupervised technique) they used algorithms: Nonnegative matrix factorization (NMF), Latent Dirichlet allocation (LDA), and latent semantic indexing (LSI). LSI has been revealed to be better than LDA. The score against each topic for each carrier is done through polarity analysis for each airline and topic. The Level of passenger satisfaction for each topic was identified.

Cem Baydoğan & Bilal Alatas (2019) conducted research to automatically determine the customers feedback from online comments. This seems a traditional text mining and classification study. However, the most remarkable point in this paper is that it focuses on unbalanced dataset. Generally speaking, classification problems are likely to deal with such data in problems like: Disease diagnosis, Customer churn prediction, Fraud detection, and Natural disaster. When we calculate accuracy of models of such problems in regular way, the percentage might be extremely high, however the actual detection and prediction of negative cases such has disaster or fraud is pretty low. So, they did their best to take this point into account while applying classification models. After applying text mining techniques, the problem because a classification problem of three classes (positive, negative, neutral), however, with remarkable unbalanced data sets. They applied algorithms Sequential Minimal Optimization (SMO) classifier, then KNN with k = (3, 5), Decision Table algorithm, Multi Class Classification Algorithm, which is used especially in unbalanced and multiclass data, and finally the used J48. The algorithms were executed against datasets of 6 selected airlines separately. The top performance was obtained from SMO and Multi Class Classification Algorithm.

Taufiqul Haque Khan Tusar, Touhidul Islam (2021) performed research on understanding the public opinion and trends from tweets using Natural Language Processing (NLP) and Machine Learning (ML). The research was to get the sentiment analysis of the US airlines and understand the opinion of customers about topics, services, and products. In order to use NLP, they used two techniques: Bag-of-Words and TF-IDF. And for classification part the used variety of machine learning models such as Support Vector Machine, Multinomial Naive Bayes, Random Forest, Logistic Regression. They divided the tweets into Positive, Negative, and Neutral. So, as it sounds, it was not a binary classification. The best algorithms were Support Vector Machine and Logistic which performed better than others with accuracy 77%. This was along with the Bag-of-Words technique for NLP.

Murat Demircana, Adem Seller, Fatih Abut, Mehmet Fatih Akay (2021) did research to examine sentiment analysis of customers for an e-commerce. Although this is not about airlines customers. However, this project paradigm is pretty similar. They are examining if a customer has positive, negative, or neutral feedback on a specific product. The research extracted reviews from a Turkish e-commerce website along with reviews scores. Instead of using lexicon-based sentiment analysis approach, they used supervised learning methods. Supervised learning methods are more efficient in determine the document's semantic orientation as they are not using the polarities of the words or phrases to determine the polarity of the text. They used five machine learning models (SVM, RF, DT, LR, and KNN). It turned out that SVM and RF have outperformed other three modes used, specifically DT, LR, and KNN.

2.3 Conclusion

Most of previous literature reviews are focusing on text mining techniques to predict and evaluate the level of services. These methodologies are giving overview about the services and tell the airlines if they are performing well in the past or not. However, they are not responding to the customers individually. A customer might have negative experience, and it really worth to try to make it up to him and predict whether he is satisfied or not based on his evaluation of the services provided. As a result, this project comes in between. It helps airlines to predict the customer satisfaction and will also help to focus on the real factors that mostly the customers are looking for.

As a result, we will check which dimension, or the service does influence the customers. Below are the outlines of the studies, and why this capstone is selected.

- Most of the previous studies concentrate on text mining of the customer reviews and sentiment analysis of the online reviews of the customers. This indicates the general feedback of people who commented and doesn't give a detailed analysis of how to improve particular services which does make the difference and would raise the feedback much higher.
- There is no direct evaluation for individual passenger experience. Predicting if a customer would be unsatisfied, especially for loyal customers, would be vital for any airlines to respond directly to such customer and make it up to them.
- Text mining and sentiment analysis through reviews give overview feedback of the service. However, this feedback analysis but it is not proactive. It merely indicates how people feel regarding the service. But it doesn't not help to guide how to provide better and effective service.

Chapter 3 - Project Description

3.1 Introduction

The project main target is to help airlines companies utilize machine learning techniques to achieve their customer satisfaction and be proactive in this regard. Airlines tend to ask customers to give feedbacks on their websites and do sentiment analysis or text mining toward analyzing their feedback and act appropriately. The approach we had in our project is to obtain data of customer evaluation of each factor of the trip such as food & drink, gate location and so on. In addition to that, there is so personal data such as gender, and purpose of travel, and some details about the trip itself. Based on these all attributes the, there is a binary classification of the customer satisfaction. We are going to use this data to generate several models to help airlines determine in advance the satisfaction of all new customers and act proactively before getting unsatisfied customers. So, we will collect the data, then we are going to explore the data and preprocess data wherever it is necessary. The data would be splitted into two categories: training category and testing category. The training category would be used for creating several machine learning models. These models would be classifiers that result a binary value. For each model, we will examine using the test data and provide comparison of the generated models. The study would also examine the important of each predictor and conclude which predictor is more important than the others so the airlines would be able to concentrate on the most important factors that play the most important role in customer satisfaction.

3.2 Data Source

The data source used in this project is provided from https://www.kaggle.com/najibmh/us-airline-passenger-satisfaction-survey. The dataset has about 130 thousand entries. Each entry represent feedback from a customer and his evaluation of all attributes of the flight and some information about the passenger as well as the flight.

3.3 Data Collection

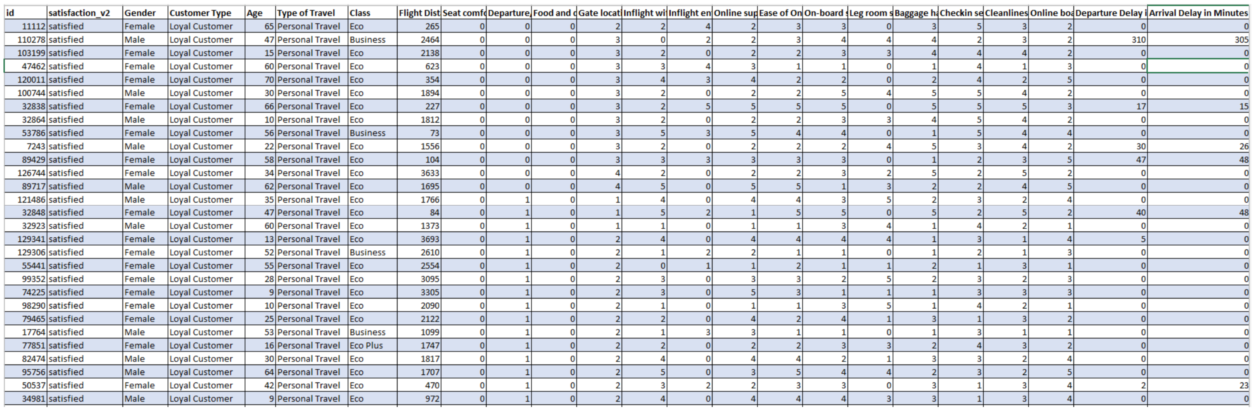
Below is a table that summarize the description of the attributes of dataset and the datatype of each as well a brief description of each column.

#	Name	Type	Desc
1	Id	Integer	Customer Id
2	Satisfaction	String	Class column
3	Gender	String	Male/Female
4	Customer Type	String	Loyal or not
5	Age	Integer	Age in years
6	Type of Travel	String	Personal/Business
7	Class	String	Eco/Eco Plus/Business
8	Flight Distance	Integer	Distance in KM
9	Seat comfort	Integer	Level from 0 to 5
10	Departure/Arrival time convenient	Integer	Level from 0 to 5
11	Food and Drink	Integer	Level from 0 to 5

12	Gate Location	Integer	Level from 0 to 5
13	Inflight Wi-Fi service	Integer	Level from 0 to 5
14	Inflight entertainment	Integer	Level from 0 to 5
15	Online support	Integer	Level from 0 to 5
16	Ease of online booking	Integer	Level from 0 to 5
17	Onboard service	Integer	Level from 0 to 5
18	Leg room service	Integer	Level from 0 to 5
19	Baggage handling	Integer	Level from 0 to 5
20	Check in service	Integer	Level from 0 to 5
21	Cleanness	Integer	Level from 0 to 5
22	Online boarding	Integer	Level from 0 to 5
23	Departure Delay in minutes	Integer	Departure Delay in minutes
24	Arrival Delay in minutes	Integer	Arrival Delay in minutes

Table 1 Dataset Structure

In addition, here I present a sample of the data, with all columns shown below:



id	satisfaction_v2	Gender	Customer Type	Age	Type of Travel	Class	Flight Dist	Seat comf	Departure/	Food and d	Gate locati	Inflight wi	Inflight en	Online sup	Ease of On	On-board s	Leg room s	Baggage ha	Checkin se	Cleanlines	Online boa	Departure Delay i	Arrival Delay in Minutes
11112	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	2	2	4	2	3	3	0	3	5	3	2	0	0
110278	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	3	0	2	2	3	4	4	4	2	3	2	310	305
103199	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	3	2	0	2	2	3	3	4	4	4	2	0	0
47462	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	3	3	4	3	1	1	0	1	4	1	3	0	0
120011	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	3	4	3	4	2	2	0	2	4	2	5	0	0
100744	satisfied	Male	Loyal Customer	30	Personal Travel	Eco	1894	0	0	0	3	2	0	2	2	5	4	5	5	4	2	0	0
32838	satisfied	Female	Loyal Customer	66	Personal Travel	Eco	227	0	0	0	3	2	5	5	5	5	0	5	5	5	3	17	15
32864	satisfied	Male	Loyal Customer	10	Personal Travel	Eco	1812	0	0	0	3	2	0	2	2	3	3	4	5	4	2	0	0
53786	satisfied	Female	Loyal Customer	56	Personal Travel	Business	73	0	0	0	3	5	3	5	4	4	0	1	5	4	4	0	0
7243	satisfied	Male	Loyal Customer	22	Personal Travel	Eco	1556	0	0	0	3	2	0	2	2	2	4	5	3	4	2	30	26
89429	satisfied	Female	Loyal Customer	58	Personal Travel	Eco	104	0	0	0	3	3	3	3	3	3	0	1	2	3	5	47	48
126744	satisfied	Female	Loyal Customer	34	Personal Travel	Eco	3633	0	0	0	4	2	0	2	2	3	2	5	2	5	2	0	0
89717	satisfied	Male	Loyal Customer	62	Personal Travel	Eco	1695	0	0	0	4	5	0	5	5	1	3	2	2	4	5	0	0
121486	satisfied	Male	Loyal Customer	35	Personal Travel	Eco	1766	0	1	0	1	4	0	4	4	3	5	2	3	2	4	0	0
32848	satisfied	Female	Loyal Customer	47	Personal Travel	Eco	84	0	1	0	1	5	2	1	5	5	0	5	2	5	2	40	48
32923	satisfied	Male	Loyal Customer	60	Personal Travel	Eco	1373	0	1	0	1	1	0	1	1	3	4	1	4	2	1	0	0
129341	satisfied	Female	Loyal Customer	13	Personal Travel	Eco	3693	0	1	0	2	4	0	4	4	4	4	1	3	1	4	5	0
129306	satisfied	Female	Loyal Customer	52	Personal Travel	Business	2610	0	1	0	2	1	2	2	1	1	0	1	2	1	3	0	0
55441	satisfied	Female	Loyal Customer	55	Personal Travel	Eco	2554	0	1	0	2	0	1	1	2	1	1	2	1	3	1	0	0
99352	satisfied	Female	Loyal Customer	28	Personal Travel	Eco	3095	0	1	0	2	3	0	3	3	2	5	2	3	2	3	0	0
74225	satisfied	Female	Loyal Customer	9	Personal Travel	Eco	3305	0	1	0	2	3	0	5	3	1	1	1	3	3	3	0	0
98290	satisfied	Female	Loyal Customer	10	Personal Travel	Eco	2090	0	1	0	2	1	0	1	1	3	5	1	4	2	1	0	0
79465	satisfied	Female	Loyal Customer	25	Personal Travel	Eco	2122	0	1	0	2	2	0	4	2	4	1	3	1	3	2	0	0
17764	satisfied	Male	Loyal Customer	53	Personal Travel	Business	1099	0	1	0	2	1	3	3	1	1	0	1	3	1	1	0	0
77851	satisfied	Female	Loyal Customer	16	Personal Travel	Eco Plus	1747	0	1	0	2	2	0	2	2	3	3	2	4	3	2	0	0
82474	satisfied	Male	Loyal Customer	30	Personal Travel	Eco	1817	0	1	0	2	4	0	4	4	2	1	3	3	2	4	0	0
95756	satisfied	Male	Loyal Customer	64	Personal Travel	Eco	1707	0	1	0	2	5	0	3	5	4	4	2	3	2	5	0	0
50537	satisfied	Female	Loyal Customer	42	Personal Travel	Eco	470	0	1	0	2	3	2	2	3	3	0	3	1	3	4	2	23
34981	satisfied	Male	Loyal Customer	9	Personal Travel	Eco	972	0	1	0	2	4	0	4	4	4	3	3	1	3	4	0	0

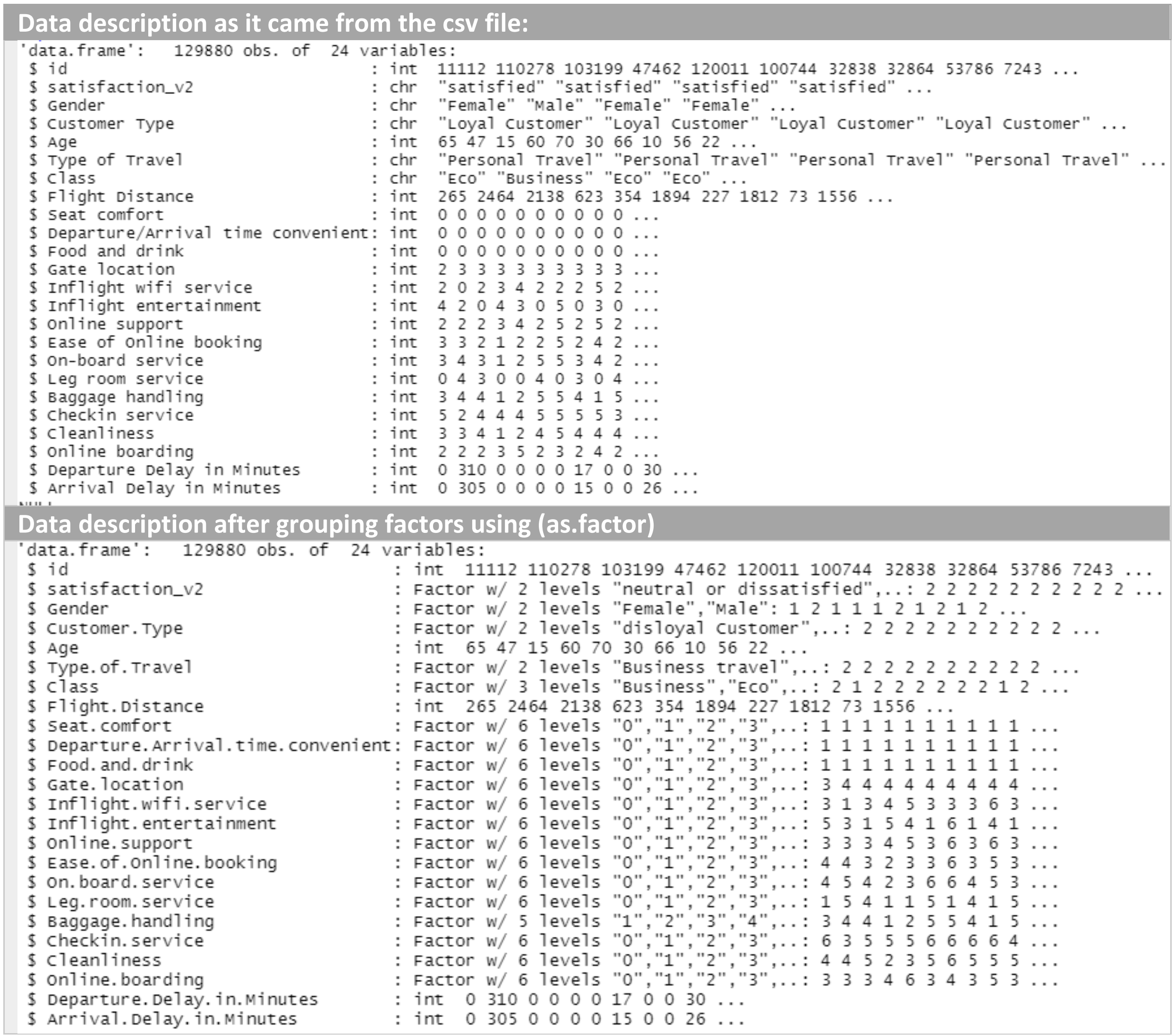
Figure 2 Dataset Sample

Chapter 4 - Data Analysis

4.1Data Preparation

The data used for this project has only one dataset. I found it sufficient for the scope of this project to depend on this dataset only, so there would be no binding process to produce the final dataset. In order to start our study, we need to load the data into our RStudio. Once we load data from CSV file, we notice that names of the columns have spaces, which may create problems in coding. So, we use the function (make.names) in R to replace the spaces with periods (.). In addition, when we examine the loaded data, we find data not grouped as categorical. So, we use the function (as.factor) to fix that.

The result could be compared as below:



Data description as it came from the csv file:

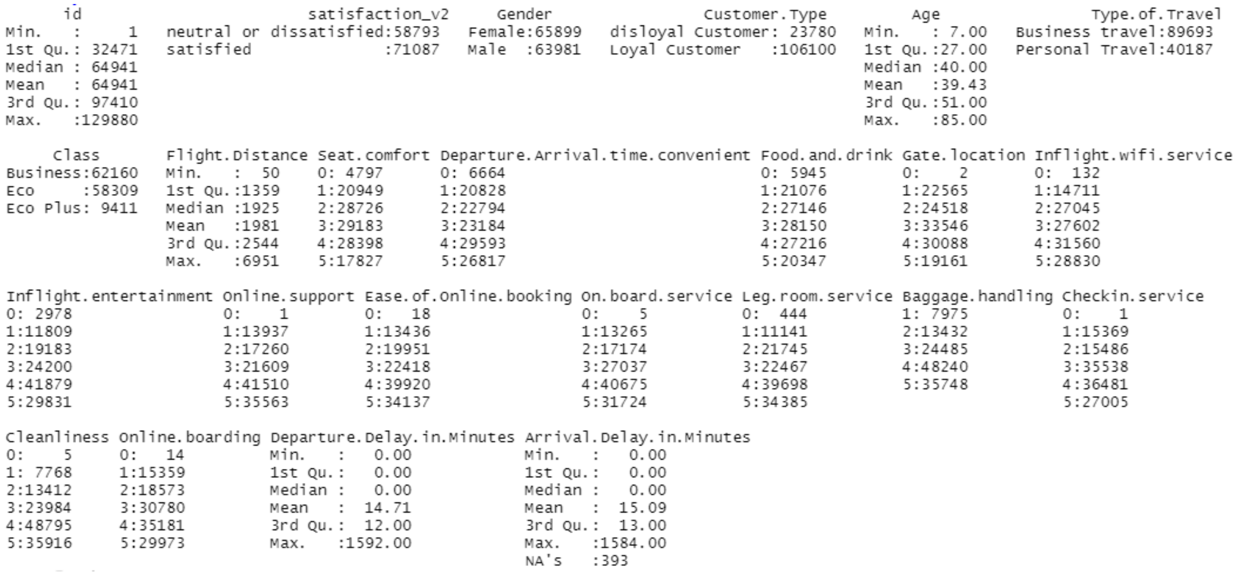
```
'data.frame':	129880 obs. of  24 variables:
 $ id                               : int  11112 110278 103199 47462 120011 100744 32838 32864 53786 7243 ...
 $ satisfaction_v2                  : chr  "satisfied" "satisfied" "satisfied" "satisfied" ...
 $ Gender                           : chr  "Female" "Male" "Female" "Female" ...
 $ Customer Type                    : chr  "Loyal Customer" "Loyal Customer" "Loyal Customer" "Loyal Customer" ...
 $ Age                              : int  65 47 15 60 70 30 66 10 56 22 ...
 $ Type of Travel                   : chr  "Personal Travel" "Personal Travel" "Personal Travel" "Personal Travel" ...
 $ Class                            : chr  "Eco" "Business" "Eco" "Eco" ...
 $ Flight Distance                  : int  265 2464 2138 623 354 1894 227 1812 73 1556 ...
 $ Seat comfort                     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Departure/Arrival time convenient: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Food and drink                   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Gate location                    : int  2 3 3 3 3 3 3 3 3 3 ...
 $ Inflight wifi service            : int  2 0 2 3 4 2 2 2 5 2 ...
 $ Inflight entertainment           : int  4 2 0 4 3 0 5 0 3 0 ...
 $ Online support                   : int  2 2 2 3 4 2 5 2 5 2 ...
 $ Ease of Online booking           : int  3 3 2 1 2 2 5 2 4 2 ...
 $ On-board service                 : int  3 4 3 1 2 5 5 3 4 2 ...
 $ Leg room service                 : int  0 4 3 0 0 4 0 3 0 4 ...
 $ Baggage handling                 : int  3 4 4 1 2 5 5 4 1 5 ...
 $ Checkin service                  : int  5 2 4 4 4 5 5 5 5 3 ...
 $ Cleanliness                      : int  3 3 4 1 2 4 5 4 4 4 ...
 $ Online boarding                  : int  2 2 2 3 5 2 3 2 4 2 ...
 $ Departure Delay in Minutes       : int  0 310 0 0 0 0 17 0 0 30 ...
 $ Arrival Delay in Minutes         : int  0 305 0 0 0 0 15 0 0 26 ...
```

Data description after grouping factors using (as.factor)

```
'data.frame':	129880 obs. of  24 variables:
 $ id                               : int  11112 110278 103199 47462 120011 100744 32838 32864 53786 7243 ...
 $ satisfaction_v2                  : Factor w/ 2 levels "neutral or dissatisfied",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Gender                           : Factor w/ 2 levels "Female","Male": 1 2 1 1 1 2 1 2 1 2 ...
 $ Customer.Type                    : Factor w/ 2 levels "disloyal Customer",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Age                              : int  65 47 15 60 70 30 66 10 56 22 ...
 $ Type.of.Travel                   : Factor w/ 2 levels "Business travel",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Class                            : Factor w/ 3 levels "Business","Eco",..: 2 1 2 2 2 2 2 2 1 2 ...
 $ Flight.Distance                  : int  265 2464 2138 623 354 1894 227 1812 73 1556 ...
 $ Seat.comfort                     : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Departure.Arrival.time.convenient: Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Food.and.drink                   : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Gate.location                    : Factor w/ 6 levels "0","1","2","3",..: 3 4 4 4 4 4 4 4 4 4 ...
 $ Inflight.wifi.service            : Factor w/ 6 levels "0","1","2","3",..: 3 1 3 4 5 3 3 3 6 3 ...
 $ Inflight.entertainment           : Factor w/ 6 levels "0","1","2","3",..: 5 3 1 5 4 1 6 1 4 1 ...
 $ Online.support                   : Factor w/ 6 levels "0","1","2","3",..: 3 3 3 4 5 3 6 3 6 3 ...
 $ Ease.of.Online.booking           : Factor w/ 6 levels "0","1","2","3",..: 4 4 3 2 3 3 6 3 5 3 ...
 $ On.board.service                 : Factor w/ 6 levels "0","1","2","3",..: 4 5 4 2 3 6 6 4 5 3 ...
 $ Leg.room.service                 : Factor w/ 6 levels "0","1","2","3",..: 1 5 4 1 1 5 1 4 1 5 ...
 $ Baggage.handling                 : Factor w/ 5 levels "1","2","3","4",..: 3 4 4 1 2 5 5 4 1 5 ...
 $ Checkin.service                  : Factor w/ 6 levels "0","1","2","3",..: 6 3 5 5 5 6 6 6 6 4 ...
 $ Cleanliness                      : Factor w/ 6 levels "0","1","2","3",..: 4 4 5 2 3 5 6 5 5 5 ...
 $ Online.boarding                  : Factor w/ 6 levels "0","1","2","3",..: 3 3 3 4 6 3 4 3 5 3 ...
 $ Departure.Delay.in.Minutes       : int  0 310 0 0 0 0 17 0 0 30 ...
 $ Arrival.Delay.in.Minutes         : int  0 305 0 0 0 0 15 0 0 26 ...
```

Figure 3 Dataset Structure

The summary of the dataset could be displayed using the summary function, and below the result is:



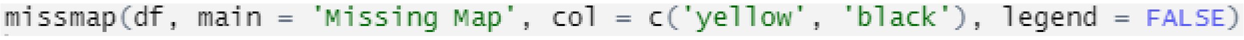
```
      id                        satisfaction_v2     Gender                Customer.Type         Age                Type.of.Travel
Min.   :     1   neutral or dissatisfied:58793   Female:65899   disloyal Customer: 23780   Min.   : 7.00   Business travel:89693
1st Qu.: 32471   satisfied              :71087   Male  :63981   Loyal Customer   :106100   1st Qu.:27.00   Personal Travel:40187
Median : 64941                                                                             Median :40.00
Mean   : 64941                                                                             Mean   :39.43
3rd Qu.: 97410                                                                             3rd Qu.:51.00
Max.   :129880                                                                             Max.   :85.00

     Class       Flight.Distance Seat.comfort Departure.Arrival.time.convenient Food.and.drink Gate.location Inflight.wifi.service
Business:62160   Min.   :  50    0: 4797      0: 6664                           0: 5945        0:    2       0:  132
Eco     :58309   1st Qu.:1359    1:20949      1:20828                           1:21076        1:22565       1:14711
Eco Plus: 9411   Median :1925    2:28726      2:22794                           2:27146        2:24518       2:27045
                 Mean   :1981    3:29183      3:23184                           3:28150        3:33546       3:27602
                 3rd Qu.:2544    4:28398      4:29593                           4:27216        4:30088       4:31560
                 Max.   :6951    5:17827      5:26817                           5:20347        5:19161       5:28830

Inflight.entertainment Online.support Ease.of.Online.booking On.board.service Leg.room.service Baggage.handling Checkin.service
0: 2978                0:    1        0:   18                0:    5          0:  444          1: 7975          0:    1
1:11809                1:13937        1:13436                1:13265          1:11141          2:13432          1:15369
2:19183                2:17260        2:19951                2:17174          2:21745          3:24485          2:15486
3:24200                3:21609        3:22418                3:27037          3:22467          4:48240          3:35538
4:41879                4:41510        4:39920                4:40675          4:39698          5:35748          4:36481
5:29831                5:35563        5:34137                5:31724          5:34385                           5:27005

Cleanliness Online.boarding Departure.Delay.in.Minutes Arrival.Delay.in.Minutes
0:    5     0:   14         Min.   :   0.00            Min.   :   0.00
1: 7768     1:15359         1st Qu.:   0.00            1st Qu.:   0.00
2:13412     2:18573         Median :   0.00            Median :   0.00
3:23984     3:30780         Mean   :  14.71            Mean   :  15.09
4:48795     4:35181         3rd Qu.:  12.00            3rd Qu.:  13.00
5:35916     5:29973         Max.   :1592.00            Max.   :1584.00
                                                       NA's   :393
```

Figure 4 Dataset Summary

4.2 Data Preprocessing

4.2.1 Discovering NAs:

NAs could be found by looking at the summary of the dataset. However, plotting NAs would be move convenient way to find them out. We could do so by using the command below:



```
missmap(df, main = 'Missing Map', col = c('yellow', 'black'), legend = FALSE)
```

After executing it, we will get the below plot that clearly indicate the attributes that has NAs.

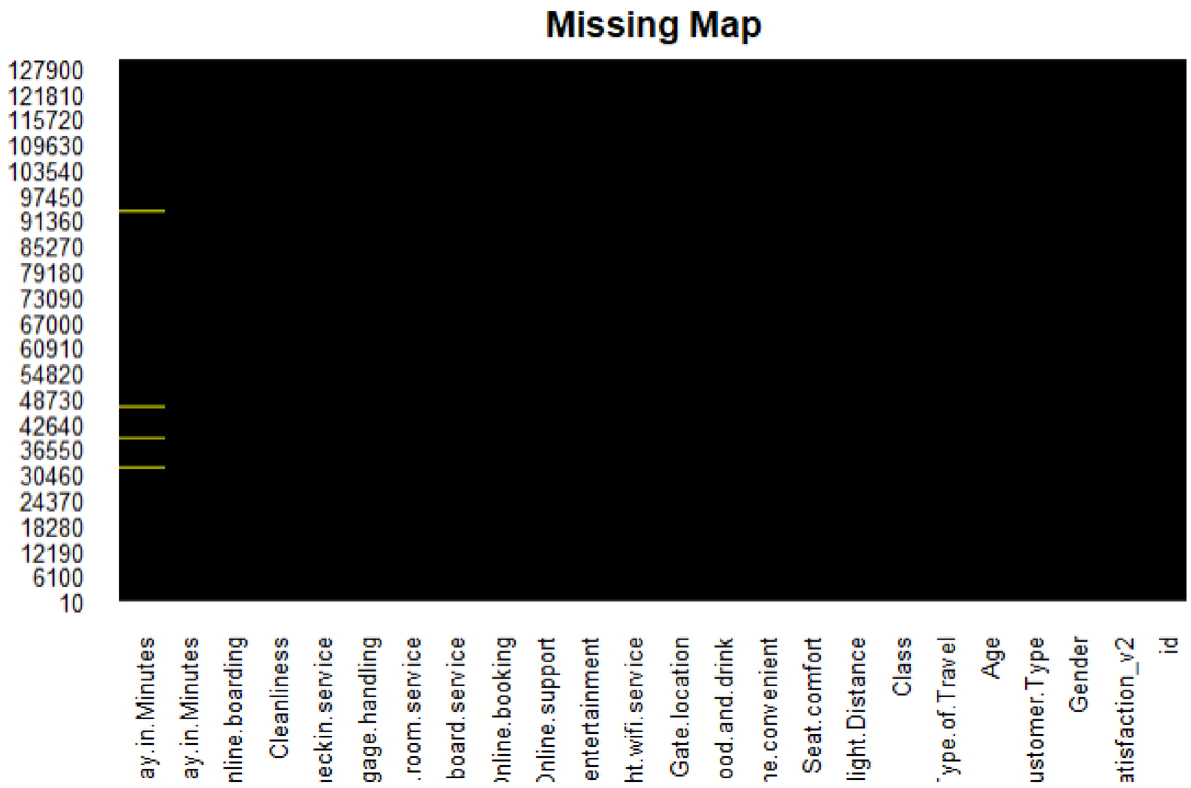
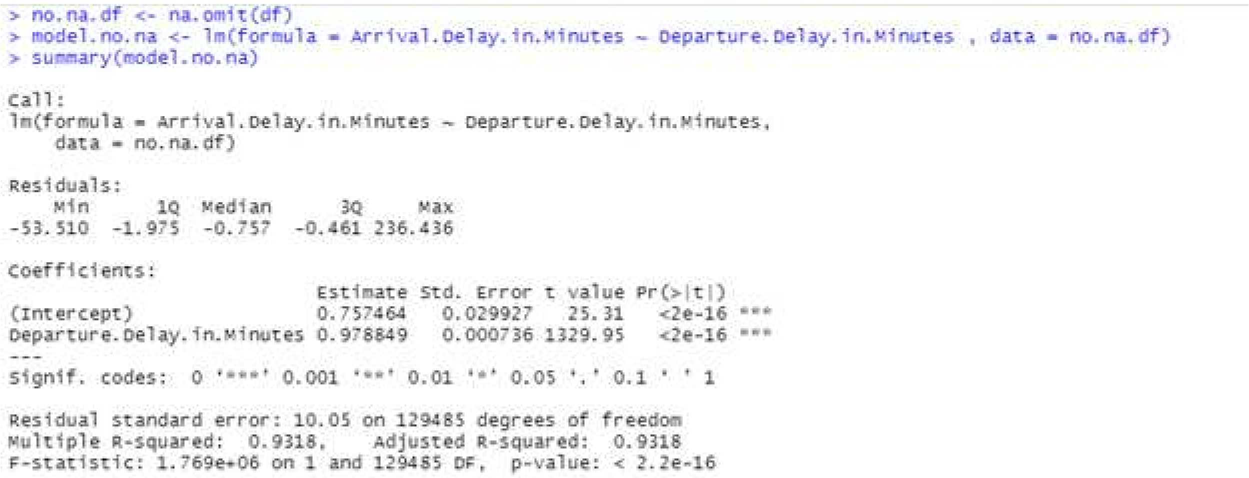


Figure 5 Missing Map

4.2.2 Removing NAs:

The first step towards data preprocessing is to remove NAs. When we examine the data, we find out that data is almost clean and the only column that contains NAs is (Arrival.Delays.in.Minutes). The count of NAs is merely 393 observations out of 129880 one which is too much small subset. Someone may suggest to either delete these rows or simply set the NAs to zero (0). But if we looked at the values, we find that almost in every delay in departure, there is a delay in arrival. This is a fact, and it is logic as well. As a result, we will study the relationship between delay in departure and delay in arrival. Let's build a linear regression model to predict the values of the NAs using the sub-dataset without NAs. We apply a small machine learning model to predict the values of Arrival.Delay.in.Minutes using a Linear Regression Model. We consider the dependent variable as (Arrival.Delay.in.Minutes) and the independent variable as (Departure.Delay.in.Minutes). The (df) below, represent the data frame from the original CSV. Then we deleted all rows containing any NAs, and in our case, it is only the 393 rows in Arrival.Delay.in.Minutes. so, total we have 129487 clean observations, and these are in data frame called: no.na.df.



```
> no.na.df <- na.omit(df)
> model.no.na <- lm(formula = Arrival.Delay.in.Minutes ~ Departure.Delay.in.Minutes , data = no.na.df)
> summary(model.no.na)

Call:
lm(formula = Arrival.Delay.in.Minutes ~ Departure.Delay.in.Minutes,
    data = no.na.df)

Residuals:
    Min      1Q  Median      3Q     Max
-53.510  -1.975  -0.757  -0.461 236.436

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.757464   0.029927   25.31   <2e-16 ***
Departure.Delay.in.Minutes 0.978849   0.000736 1329.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.05 on 129485 degrees of freedom
Multiple R-squared:  0.9318,    Adjusted R-squared:  0.9318
F-statistic: 1.769e+06 on 1 and 129485 DF,  p-value: < 2.2e-16
```

Figure 6 Arrival Delay Prediction Model

The result above shows a very strong dependency on Departure Delay. The p-value is quite small, and we can see the 3-starts indication about the strong correlation. In addition, this is logical because if there is delay in departure, then, certainly there would be delay in arrival. In addition, this relationship is much stronger than any other potential relationship with any other attribute in the dataset. Let's list the values of errors here and later we will explain each of them:

Error Name	Formula	Error Value
ME	$\frac{1}{N}\sum_{i=1}^{N} Y_i - \hat{Y_\iota}$	-7.45259261220834e-14
MAE	$\frac{1}{N}\sum_{i=1}^{N} \lvert Y_i - \hat{Y_\iota}\rvert$	5.28889248527955
MSE	$\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y_\iota})^2$	100.927499434029
RMSE	$\sqrt[2]{\frac{1}{n}\sum_{i=1}^{n} (\hat{y_\iota} - y_i)^2}$	10.0462679356082
R^2	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_\iota})^2}{\sum_{i=1}^{n}(y_i - \bar{y_\iota})^2}$	0.9318
R$_{\text{adjusted}}$	$1 - [\frac{(1 - R^2)\,(n-1)}{n-k-1}]$	0.9318

Table 2 Regression Model Residuals Errors

Let’s plot the regression line and check the observations in the below plot. We see that what we have discussed is a matter of fact.

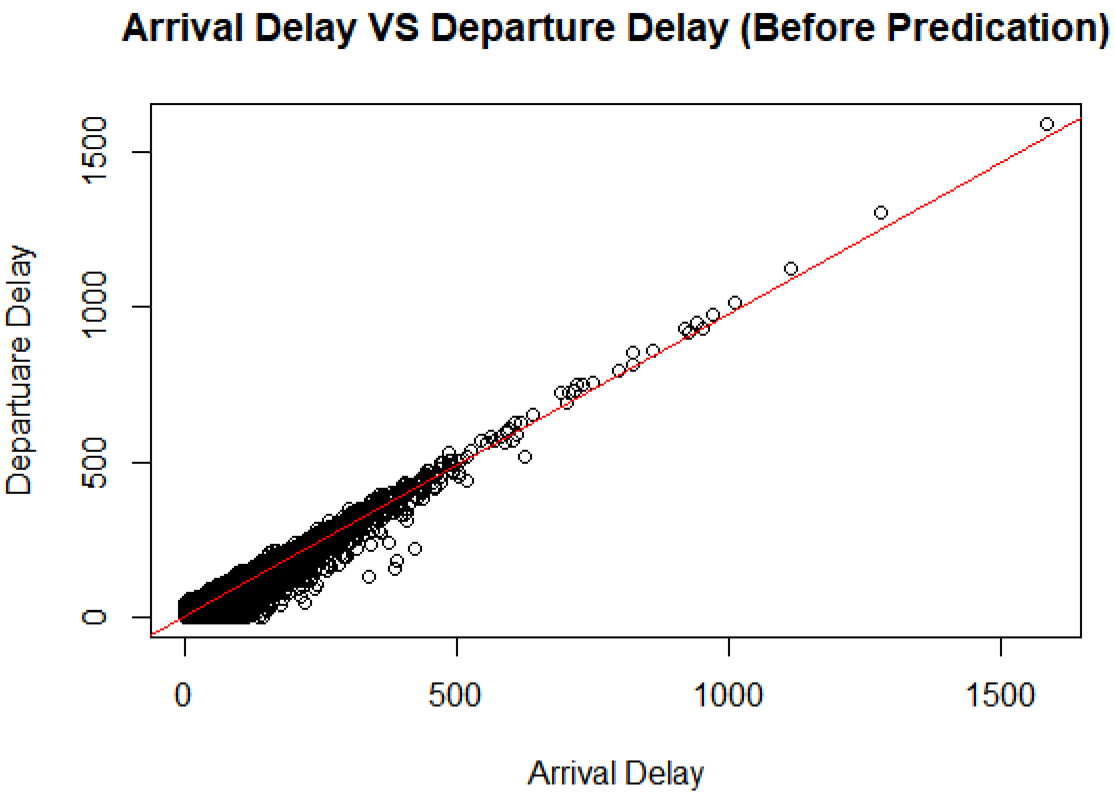
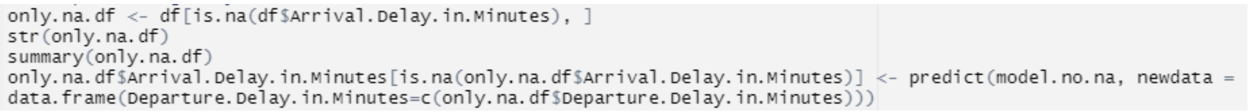


Figure 7 Arrival Delay vs Departure Delay (Before Prediction)

So, in order to clean the NAs in (Arrival.Delay.in.Minutes) we simply set the null values with the predicted values from the linear regression model using the value of (Departure.Delay.in.Minutes).

In the below code, we fill the NAs with the predicated values from the regression model we just created.



```
only.na.df <- df[is.na(df$Arrival.Delay.in.Minutes), ]
str(only.na.df)
summary(only.na.df)
only.na.df$Arrival.Delay.in.Minutes[is.na(only.na.df$Arrival.Delay.in.Minutes)] <- predict(model.no.na, newdata =
data.frame(Departure.Delay.in.Minutes=c(only.na.df$Departure.Delay.in.Minutes)))
```

Then we plot the changed values and as we could see in the next plot that the predicted values (Arrival Delay) are directly proportional original values (Departure Delay)

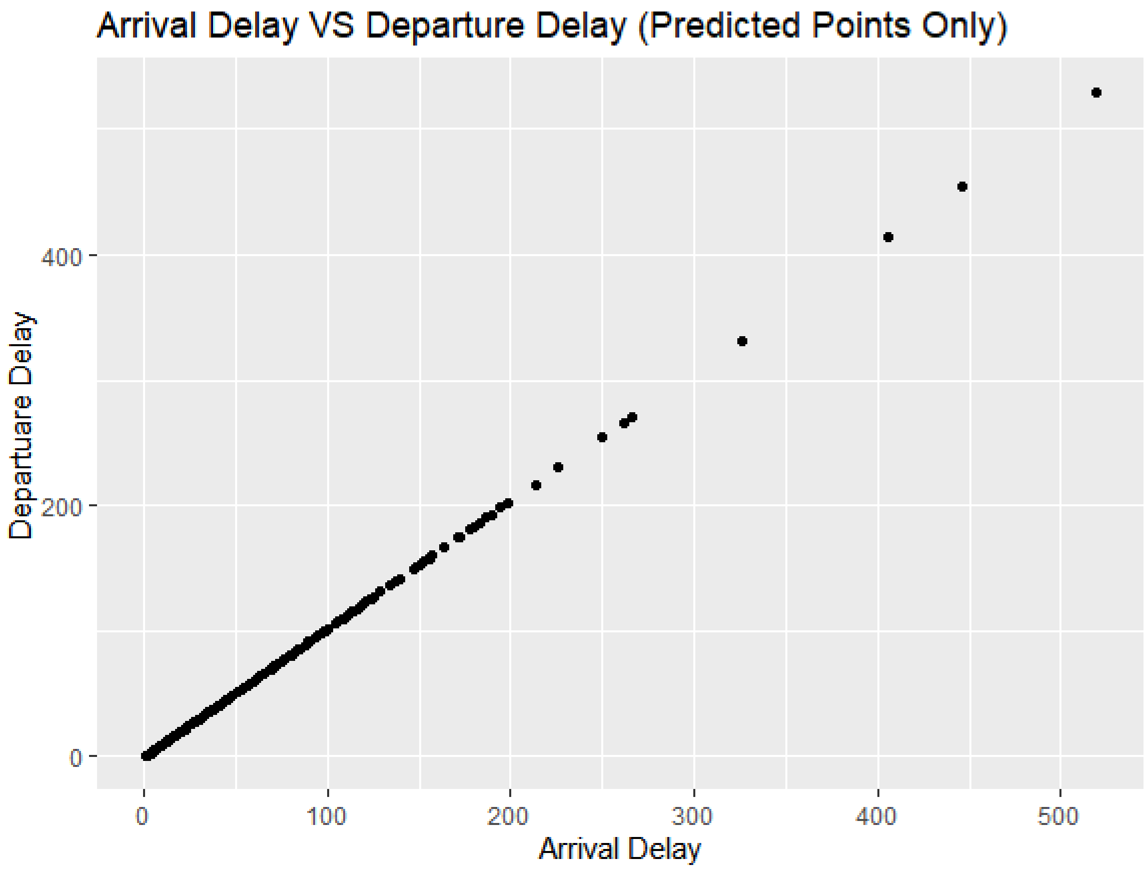


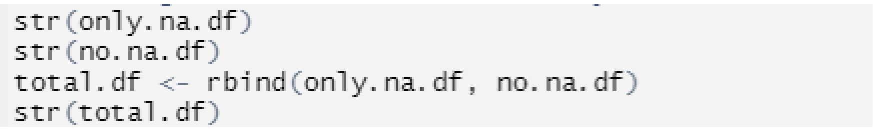
Figure 8 Predicted Arrival Delays Only

After that we merge the two data frames together in order to establish a total data frame that contains all rows ready. We use the below code to perform the merge. We simply just add up the rows.

The number of predicted rows: **393**

The number of other rows (already known): **129487**

Then the total number of rows is: 129880 which is the same as the original number of rows.



```
str(only.na.df)
str(no.na.df)
total.df <- rbind(only.na.df, no.na.df)
str(total.df)
```

Now we plot the Arrival Delay vs Departure Delay in a plot to see the total result and make sure there is no mistakes. Fortunately, we find out that the plot looks identical. This is expected since only 393 points are added to the previous plot, and the predicted points exist on the diagonal line.

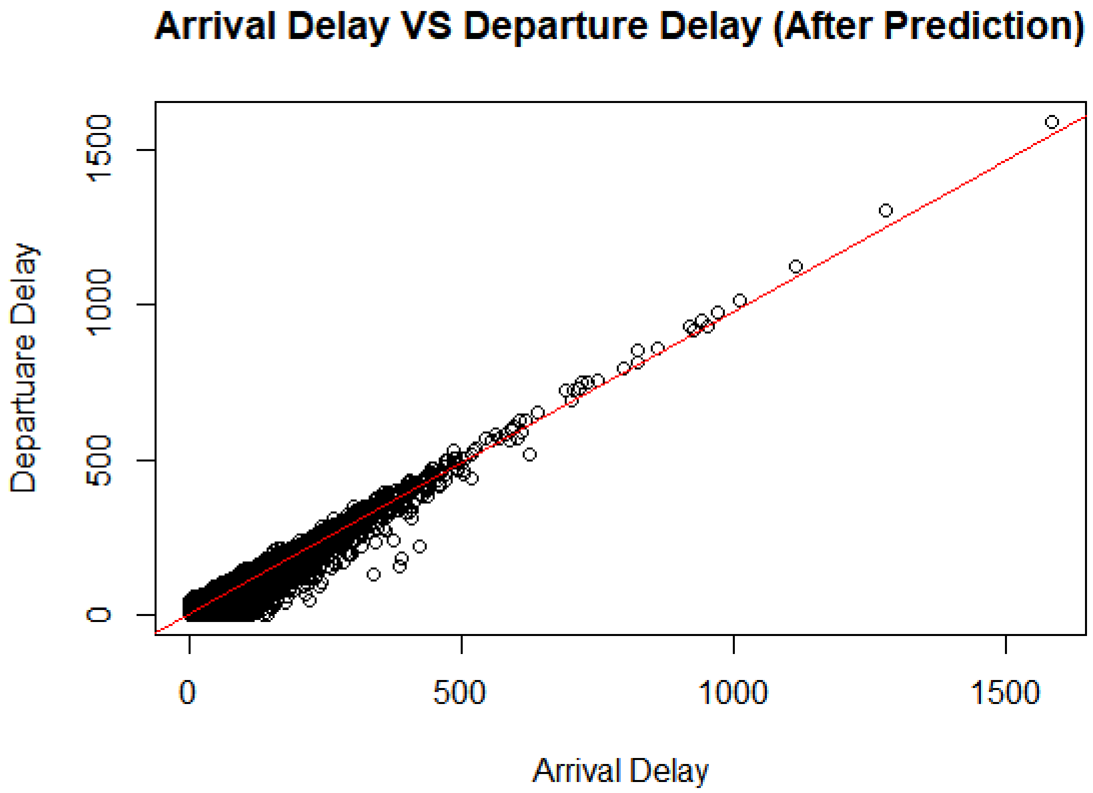


Figure 9 Arrival Delay vs Departure Delay (After Prediction)

4.2.3 Evaluating the model of removing NAs:

Before we leave this section, let’s talk about the error types and how to measure the error of the regression model we have just created:

Let’s explain the errors of this regression model:

ME = Mean Error = $\frac{Sum\ of\ All\ Errors}{Number\ of\ Observations} = \frac{1}{N}\sum_{i=1}^{N} Y_i - \widehat{Y_i}$

- The above equation is the simplest equation to calculate error. It simply sum up the residuals regardless of being positive or negative. The main disadvantage of this equation is that it doesn’t distinguish based on the sign. So, residuals might be neglecting each other’s, and hence the total value of the error might be almost or equal to zero although there might be too many errors.
  As a result, we may choose Mean Absolute Error as explained below:

  MAE = Mean Absolute Error = $\frac{1}{N}\sum_{i=1}^{N} |Y_i - \widehat{Y_i}|$ = 5.28889248527955

- Mean Absolute Error overcomes the problem of neglecting the error values when some are positive, and others are negative because of the absolute values.

- Now, let's examine Mean Square Error. This one is very similar in common to Mean Absolute Error. However, it squares the values of residuals instead of taking the absolute values. The equation is:

MSE = Mean Square Error = $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2$ = 100.927499434029

- We need to take into account that MAE is less biased for large error (& outliers) because it doesn't square the residuals like MSE. But MAE may not be adequately representing large errors. On the other hand, MSE get highly biased for large errors (& outliers).
- Someone might ask, if the MSE, or MAE is 400 or 1200, would that be good or bad for the model. There is no direct answer. This whole thing depends on the model and the magnitude of the datapoints. So MSE & MAE is a metric (loss or error function) and can measure if model A is better that model B and vise-versa.
- People usually use Root Mean Square Error as this is the same as MSE but only take the squared-root of the value.

RMSE = Root Mean Square Error = $\sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}$ = 10.0462679356082

This would be better since it gets the value smaller by taking the root of MSE. In addition, RMSE represents the standard deviation of the residuals (i.e., differences between the model predictions and the true values).

- Also, we may calculate the percentage of MAE using the below formula.

MAPE = Mean Absolute Percentage Error = $\frac{100\%}{n}\sum_{i=1}^{n}|(y_i - \hat{y_i})/y_i|$ =

- There is another formula called R-squared

$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}$ = 0.9318

- Finally, there is the formula of Adjusted R-Squared
R^2 has limitation that when we add an independent variable which has less significance and is not that useful variable, then R^2 get increased.
R^2-adjusted resolves this problem by adding penalty to the formula if we added a new variable that is not significant to the model.
As a result, the R^2-adjusted decreases if we added useless predicators to the model and vice-versa.
The formula is as below, and K is the number of independent variables and n is the number of samples.

$R^2_{adjusted} = 1 - [\frac{(1 - R^2)\,(n-1)}{n-k-1}]$ = 0.9318

The four plots below explain the errors and residuals.

<u>Residual vs Fitted Plot</u>: This plot is used to check linearity of the residuals. In other words, the mean residual for all fitted residuals close to 0. This could be checked if the red line is close to the dashed line.

It could also help to check the outliers. This is checked if some residuals are extremely far from the rest.

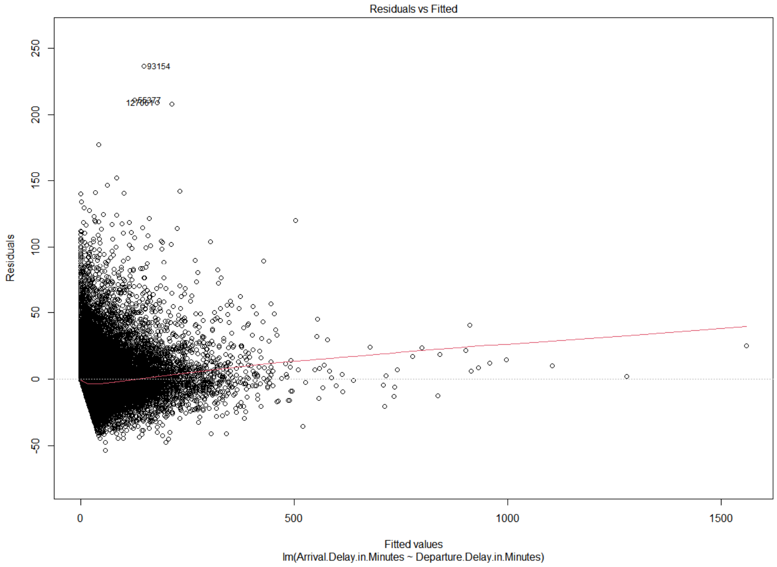


Figure 10 Residual vs Fitted

<u>Normal Q-Q Plot</u>: Q-Q plot is used to check the points are on the dotted line, if not then the points and errors are not Gaussian distribution. In both sides the residuals depart from the diagonal line.

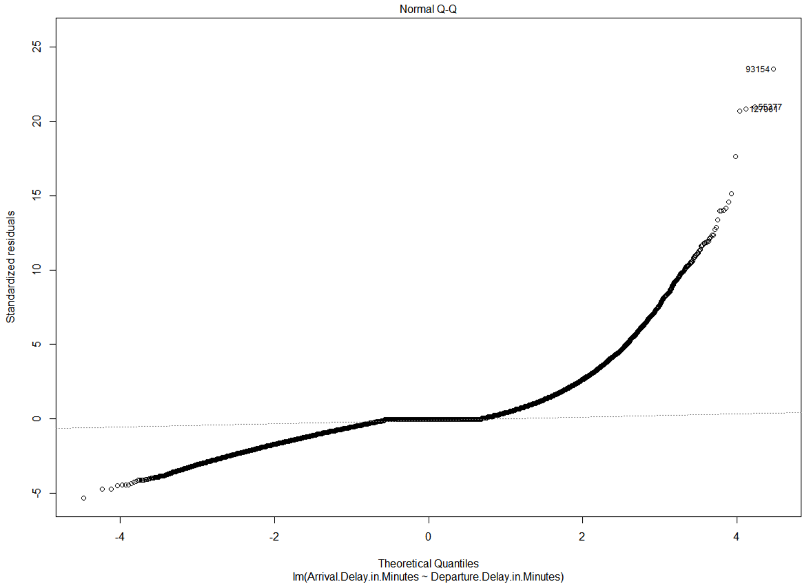


Figure 11 Normal Q-Q

<u>Scale Location Plot</u>: This plot is similar to Residual vs Fitted Plot, where X axis has the fitted values. However, unlike Residual vs Fitted plot, the Y axis we have the root of the standardized value of the residuals.

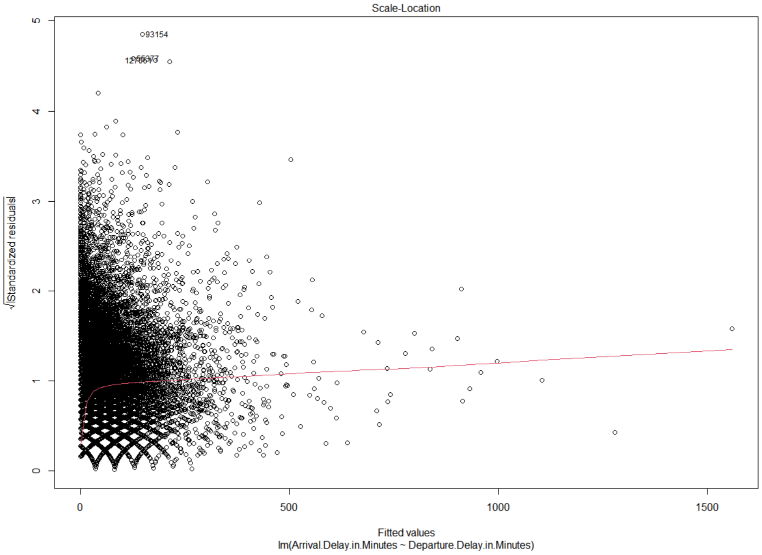


Figure 12 Scale-Location

<u>Residuals vs Leverage</u>: This plot introduces the Leverage concept. It gives an overview between the leverage values and the standardized residuals of the regression line.

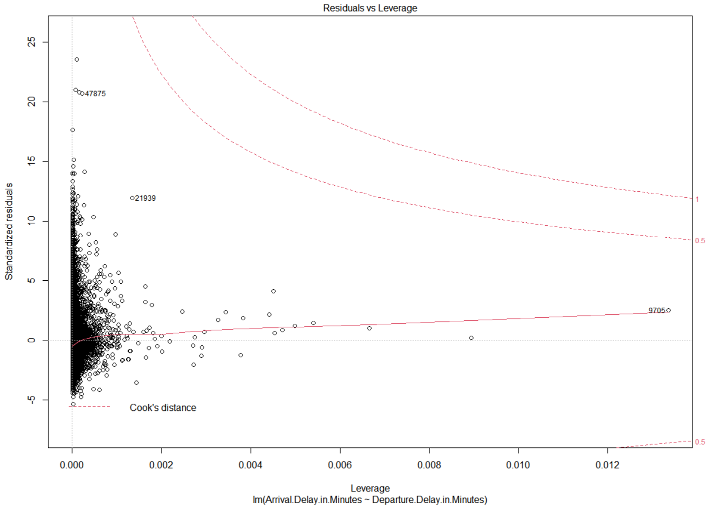


Figure 13 Residual vs Leverage

4.2.4 Checking duplicates:

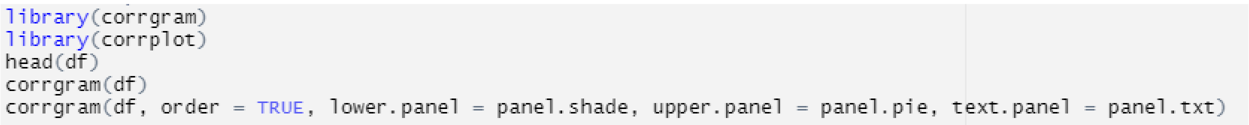
Data represent the feedback of the customers against the services of the trip. The data might be asked for more than one customer on the same trip. However, the identification would be the customer Id. We checked the duplication, and we found no duplication. As a result, we will not delete any row.

The command executed was: **which(duplicated(df))** where df represent the dataframe of the whole dataset.

4.3 Correlation between attributes

Now we have removed the NAs and we are sure that there is no duplicates in the dataset. Let's now check the correlation between the predictors.

In order to check the correlation, we are going to use the following piece of code.



```
library(corrgram)
library(corrplot)
head(df)
corrgram(df)
corrgram(df, order = TRUE, lower.panel = panel.shade, upper.panel = panel.pie, text.panel = panel.txt)
```

The corrgram and corrplot libraries are used to build the correlation plots. The corrgram is newer. The below is the correlation plot for all variables even the factor ones.

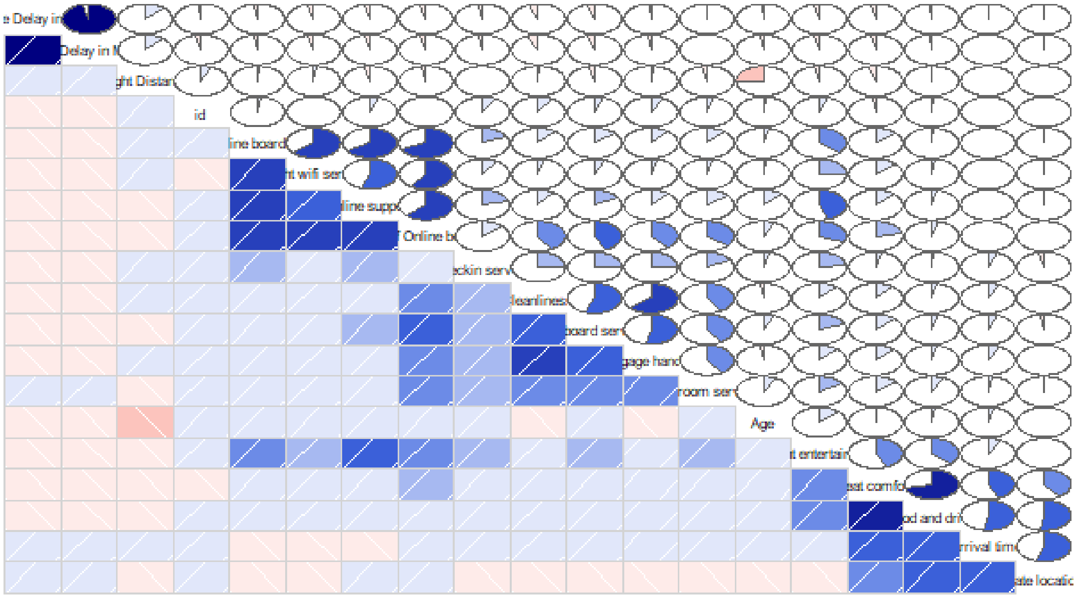


Figure 14 All Attributes Correlation (1)

Another plot is like below which also shows some correlation between categorical variables.

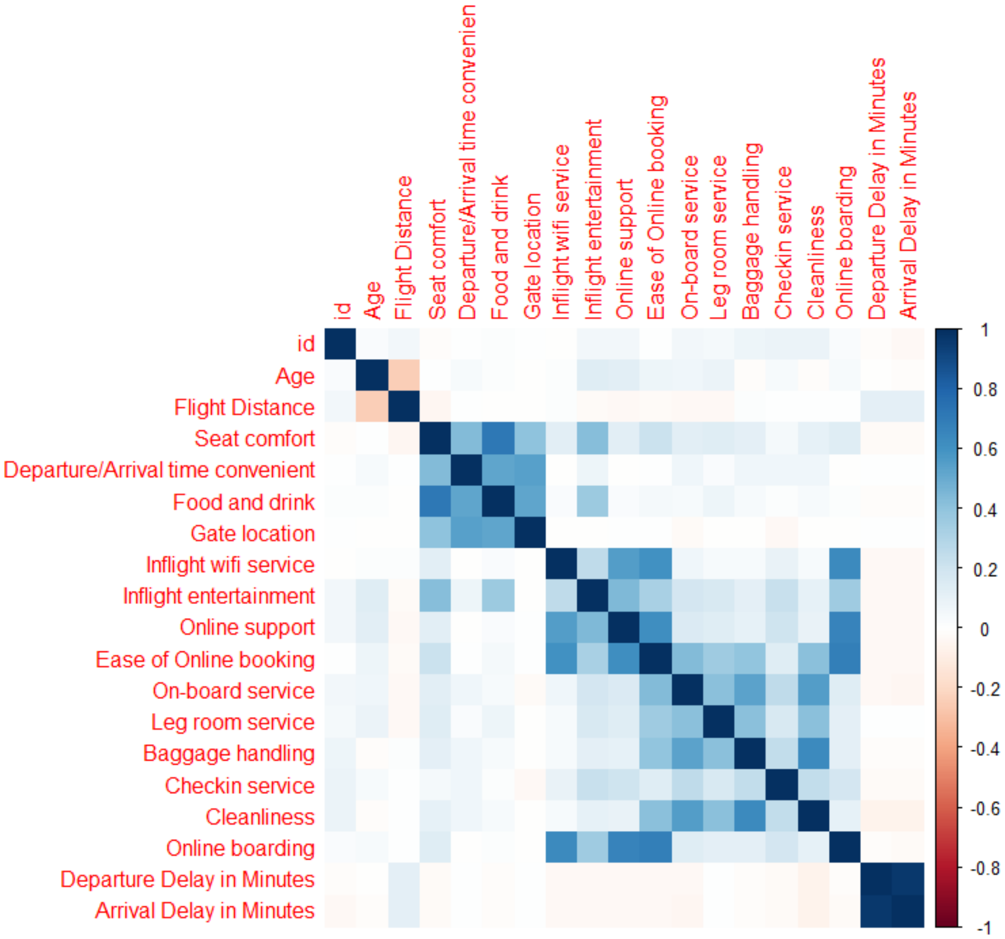


Figure 15 All Attribute Correlation (2)

However, to calculate the correlation we would mainly depend on the continuous attributes such as Age, Flight.Distance, Departure.Delay, Arrival.Delay. so, we will be using the below plot. That plot shows there is a correlation between Departure.Delay and Arrival.Delay. This means that whenever there is a delay in departure then there is a delay in arrival, which is a logic because the flight duration is constant and is already known. However, when you delay the departure, then there is delay for sure in the arrival cause the aircraft is no going to move faster because there is a delay in departure.

In addition, this is the correlation plot of the continuous variables only:

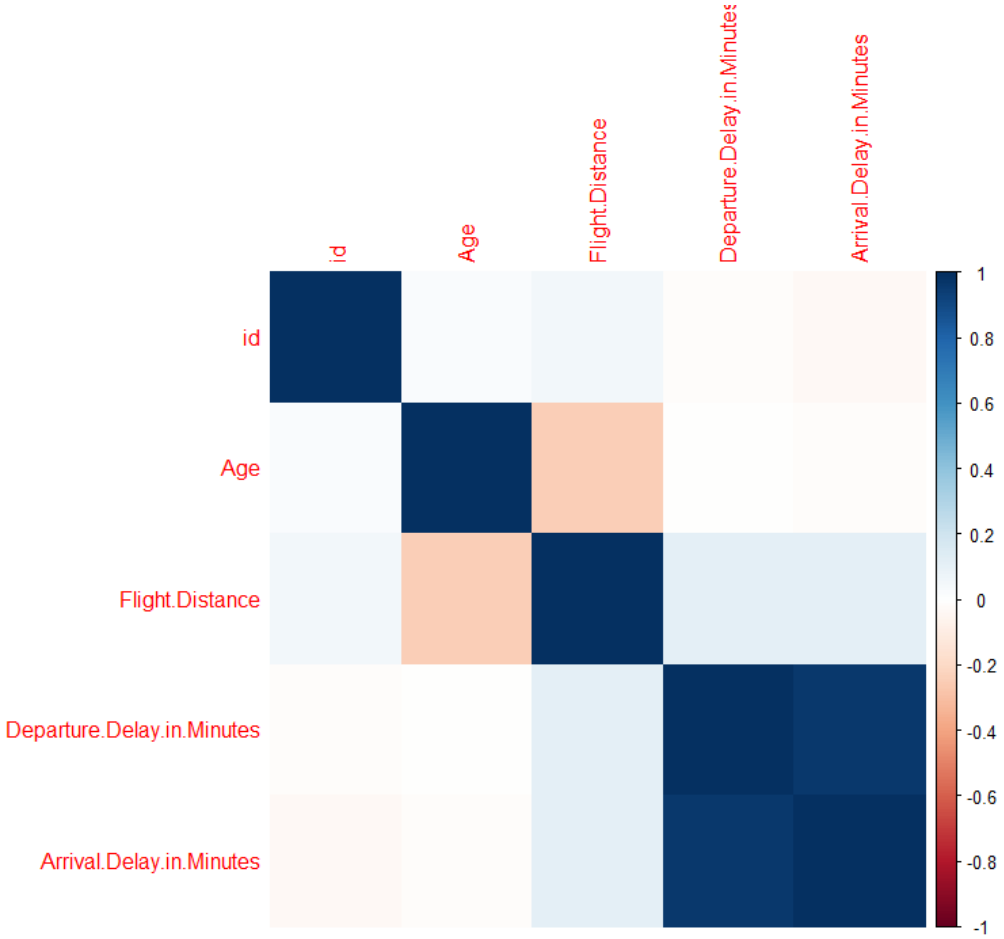
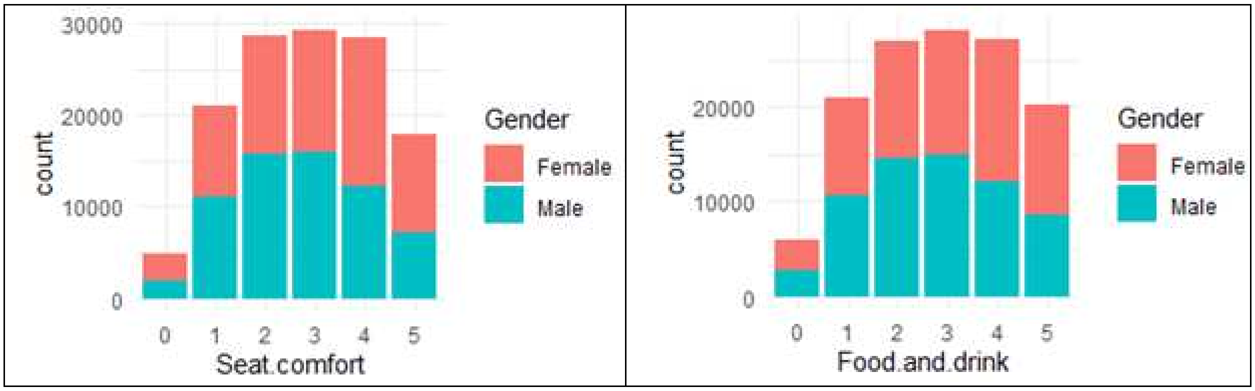


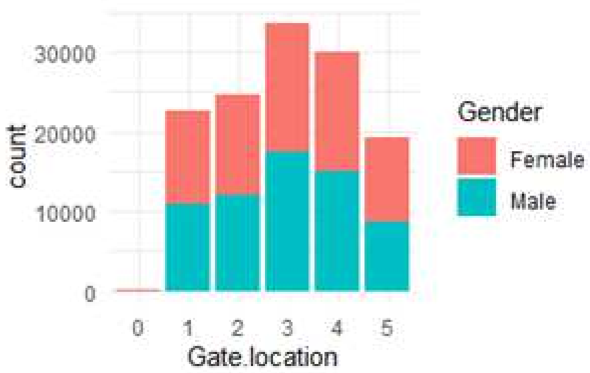
Figure 16 Numeric Features Correlation

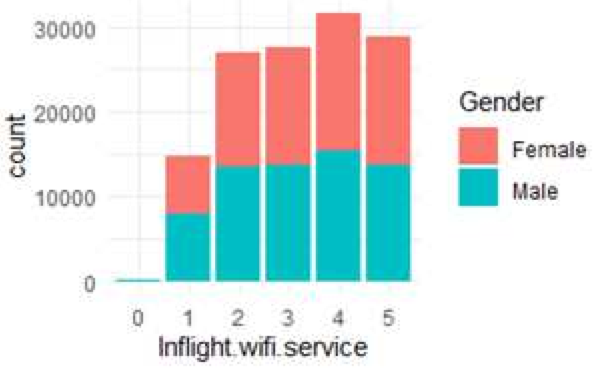
4.4 Data Exploration and Visualization

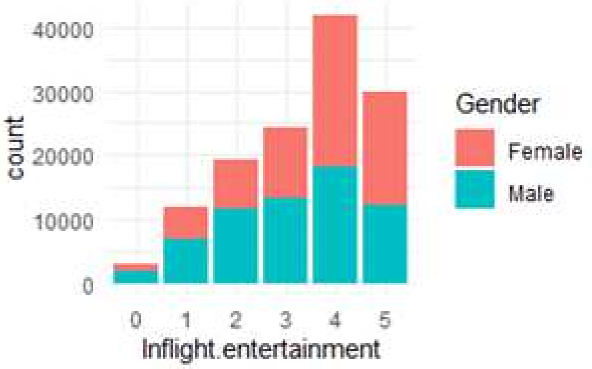
4.4.1 Services evaluation against gender:

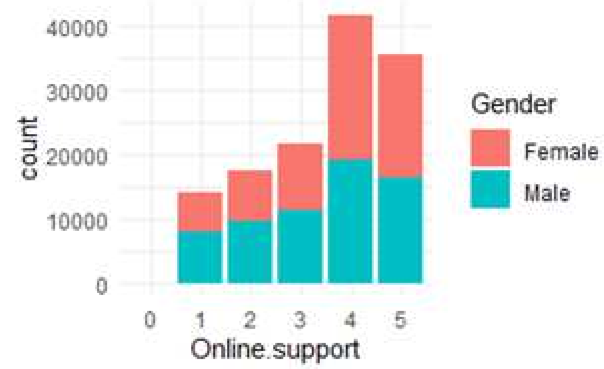
Now we are going to check the values of each of the ground services and on-flight services to check if some of them have values outside the range [0, 5] and see if there is any tendency from one of the genders to answer some specific rating.

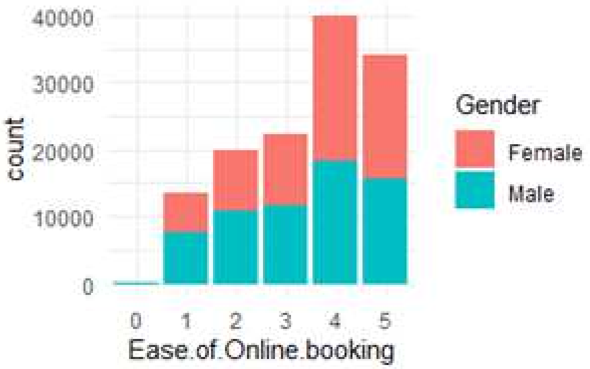


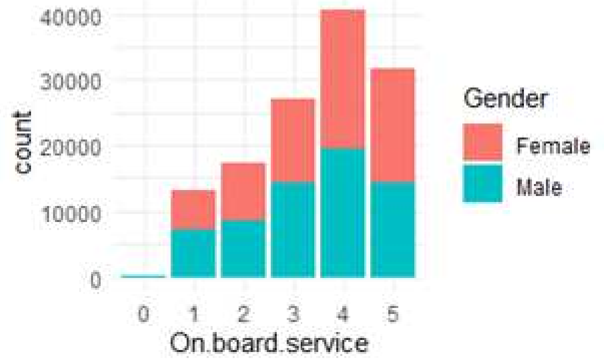

count
30000
20000
10000
0
0 1 2 3 4 5
Gate.location
Gender
Female
Male

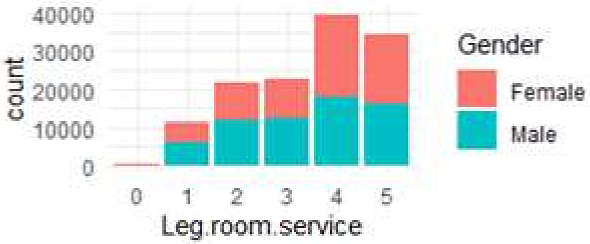

count
30000
20000
10000
0
0 1 2 3 4 5
Inflight.wifi.service
Gender
Female
Male

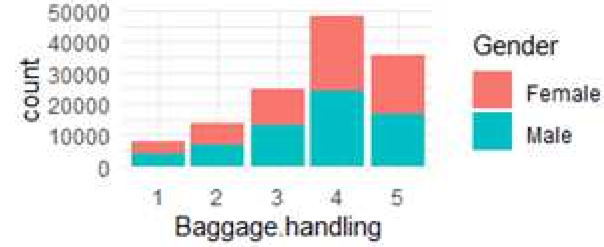

count
40000
30000
20000
10000
0
0 1 2 3 4 5
Inflight.entertainment
Gender
Female
Male

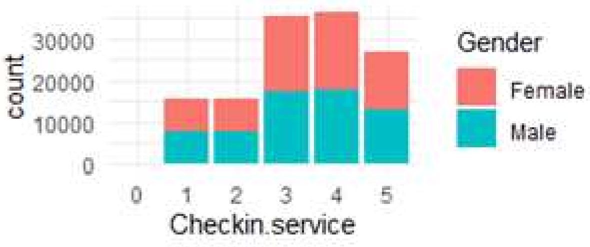

count
40000
30000
20000
10000
0
0 1 2 3 4 5
Online.support
Gender
Female
Male

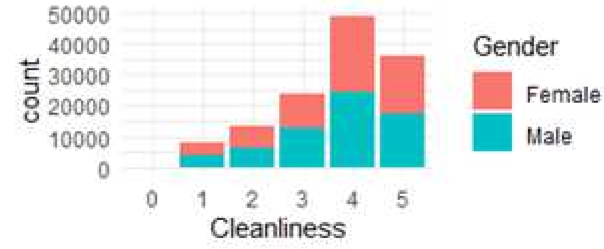

count
40000
30000
20000
10000
0
0 1 2 3 4 5
Ease.of.Online.booking
Gender
Female
Male


count
40000
30000
20000
10000
0
0 1 2 3 4 5
On.board.service
Gender
Female
Male


count
40000
30000
20000
10000
0
0 1 2 3 4 5
Leg.room.service
Gender
Female
Male


count
50000
40000
30000
20000
10000
0
1 2 3 4 5
Baggage.handling
Gender
Female
Male


count
30000
20000
10000
0
0 1 2 3 4 5
Checkin.service
Gender
Female
Male


count
50000
40000
30000
20000
10000
0
0 1 2 3 4 5
Cleanliness
Gender
Female
Male

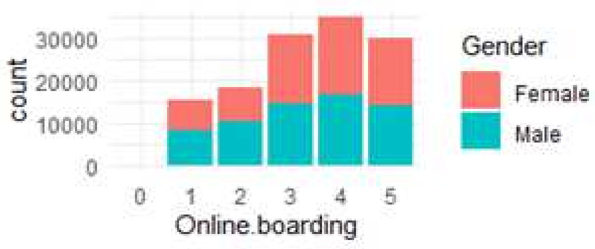
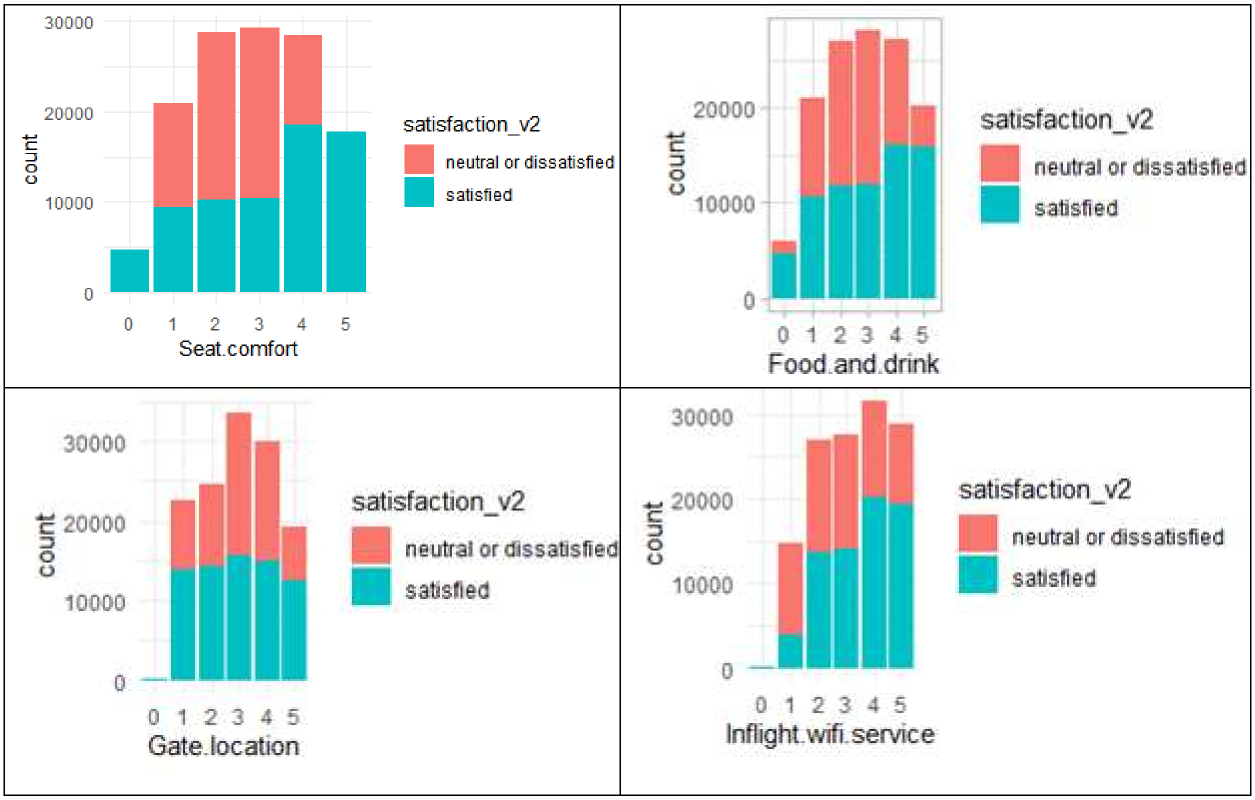


Figure 17 Services Evaluation Against Gender

As we could see for all services (ground & on-flight) there is no tendency for the gender to be aligned to any specific value. All values are balanced between the two genders.

4.4.2 Services evaluation against class:

We will repeat the same checks for all services (ground & on-flight) to see if there is any tendency to of any service correlated with being satisfied or not.



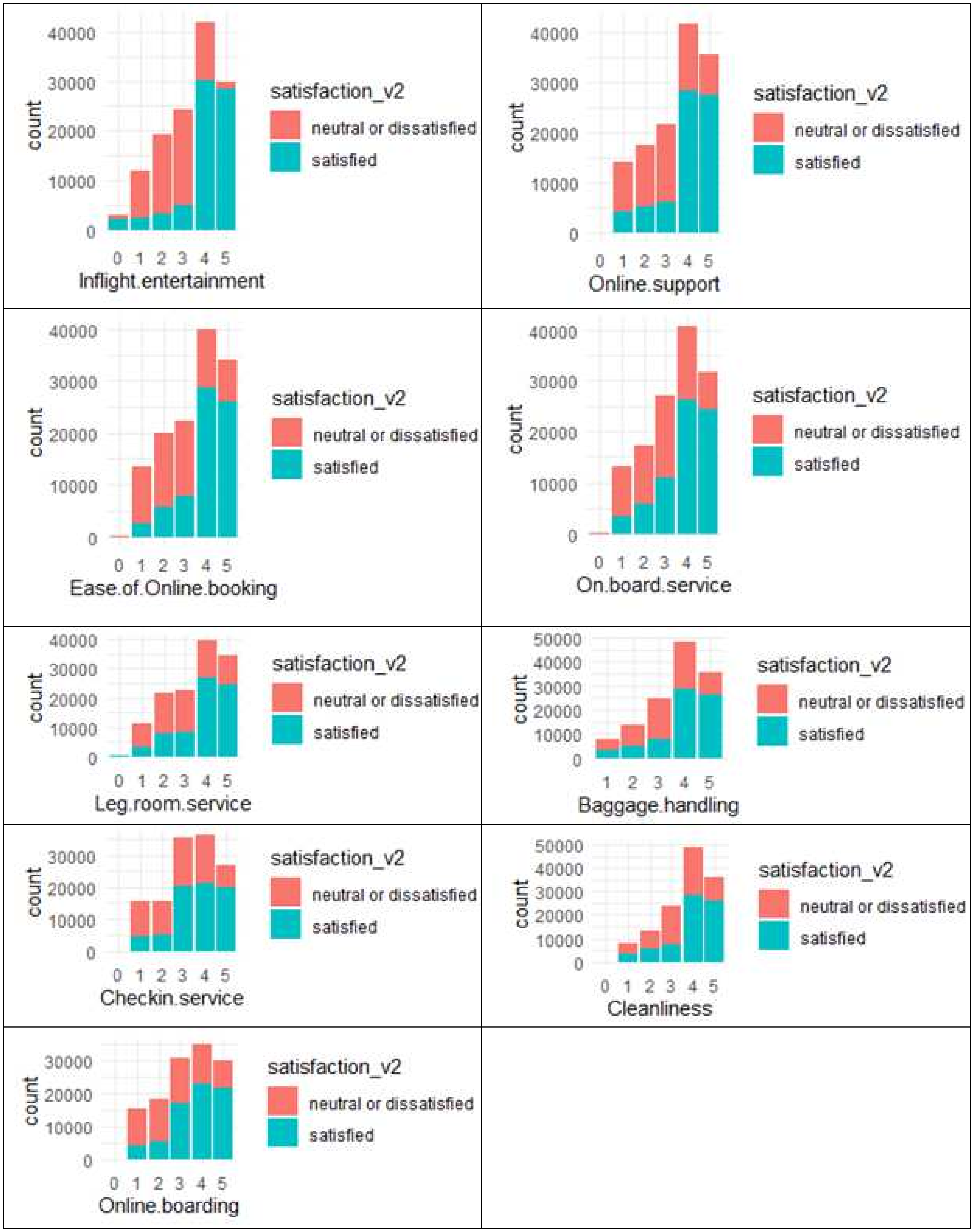


Figure 18 Services Evaluation Against Class

Based on the charts we have we see that there is no relation between being satisfied or not with a steady response of any of the services (ground or on-flight).

4.4.3 Customer-Categorical features evaluation against gender:

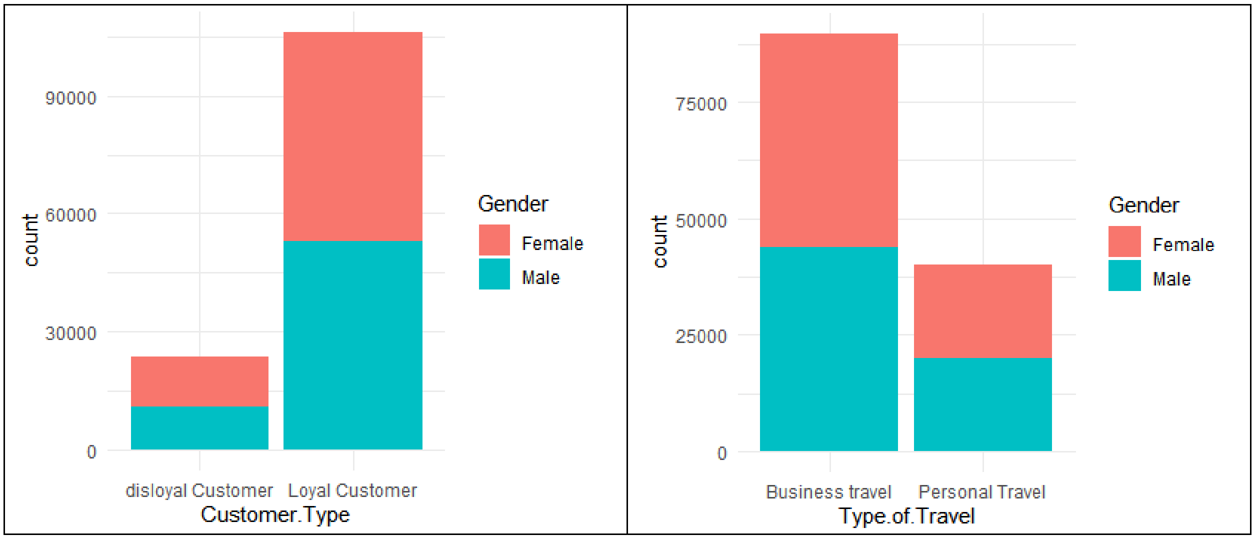


Figure 19 Customer-Categorical Features Evaluation Against Gender

As we could see for customer categorical features there is no tendency for the gender to be aligned to any specific value. All values are balanced between the two genders.

4.4.4 Customer-Categorical features evaluation against class:

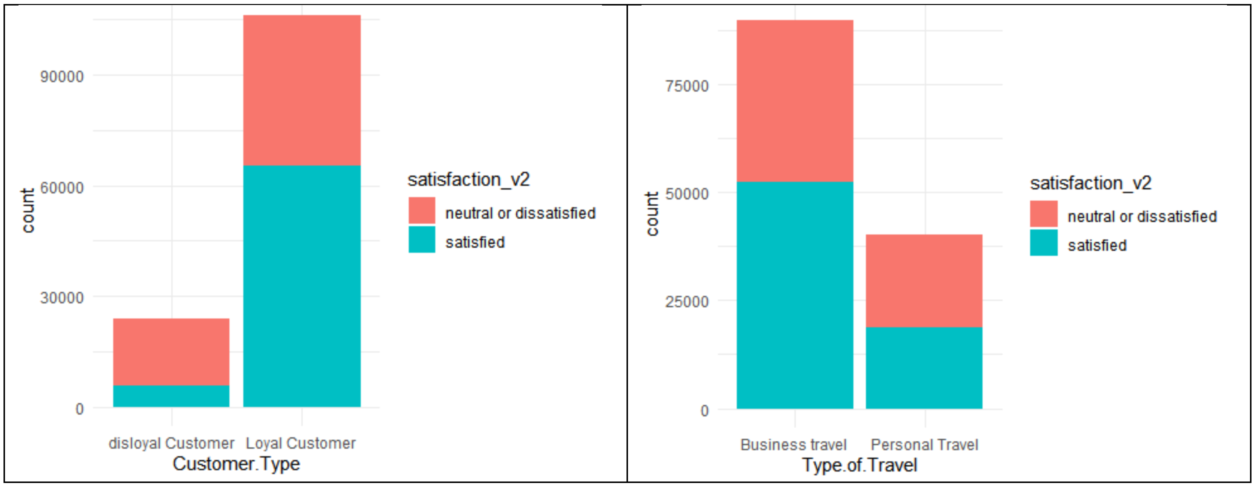


Figure 20 Customer-Categorical Features Evaluation Against Class

Based on the charts we have we see that there is no relation between being satisfied or not with a steady response of any of customer categorical features.

4.4.5 Balance of dependent variable

Below we are checking the class column values and its distribution. Normally in classification problems and especially in binary classification ones, there would be imbalanced between the two classes. This would be in many fields such as:

a. Default Prediction
b. Spam Detection
c. Disease Detection

There would be two types of imbalances: Slight Imbalanced such as 40:60 ratio. And the Severe Imbalanced such as 95:5 percent ration.

The severe imbalance can not be tolerated and must have special techniques to be handled. This is because in such classification problems, the classification of the minority class is the most important above all. So even if the accuracy is high such as 95%, this doesn't mean that the model is accurate. Because if it is just missing the 5% which is the whole cases of the minority class, then the whole model is biased and is not predicting any good results at all.

In such cases there would be other techniques to be followed such as K-Fold validation technique in which we divide the dataset into say 10 subsets, and then perform the training and testing against each 9 subsets against the 10th subset and check the performance. Then we repeat the same for the other 9 subsets and test it for the second 10th subset. We repeat this until we reach the end of the 10 trials then we take the results and aggregate them all together.

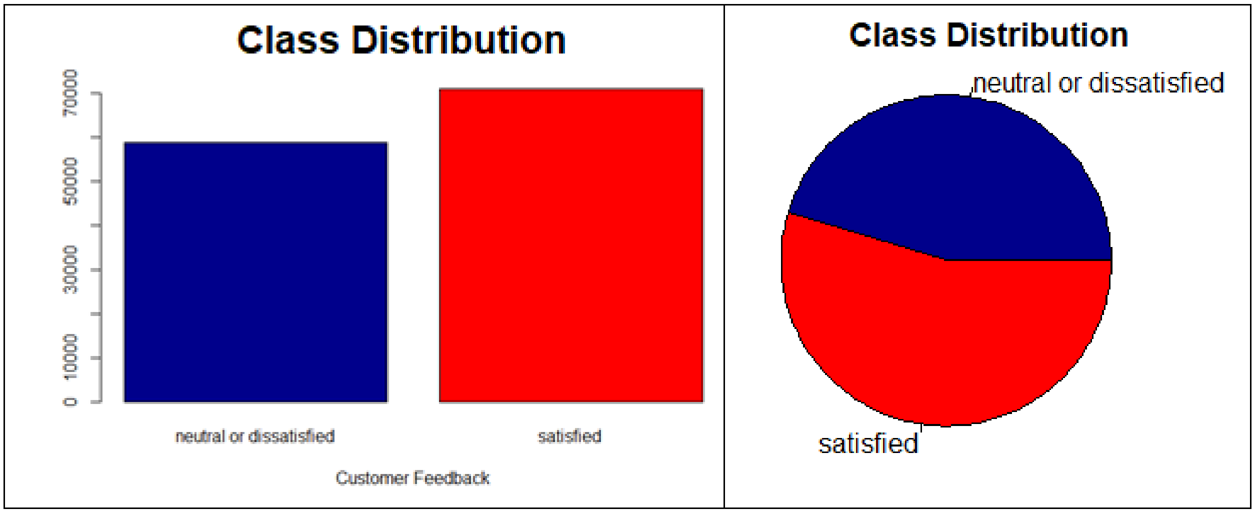


Figure 21 Class Column Distribution

As we could see in the plots below, there is an imbalance in the class column of the studied dataset. However, the imbalance is not that severe one. As a result, we can consider the imbalance as a slight one and can be tolerated and study the complete the classification problem in a normal way.

4.4.6 Age attribute distribution

Now we are going to explore the attribute Age. Using the histogram below, we could see that the values of the age are ranging between the 0 and below 90. This is the normal range of ages that we could possibly see in our daily life. We have splitted data between the two genders we have (Male/Female). We could see that the distribution between the two genders is the same. It seems the sample taken in the questioner has taken with case about the equality between male and females.

We could also check the data using the boxplot and violin plot. We could see that the mean and first quartile and third quartile. The mean is almost 40 for both female and male.

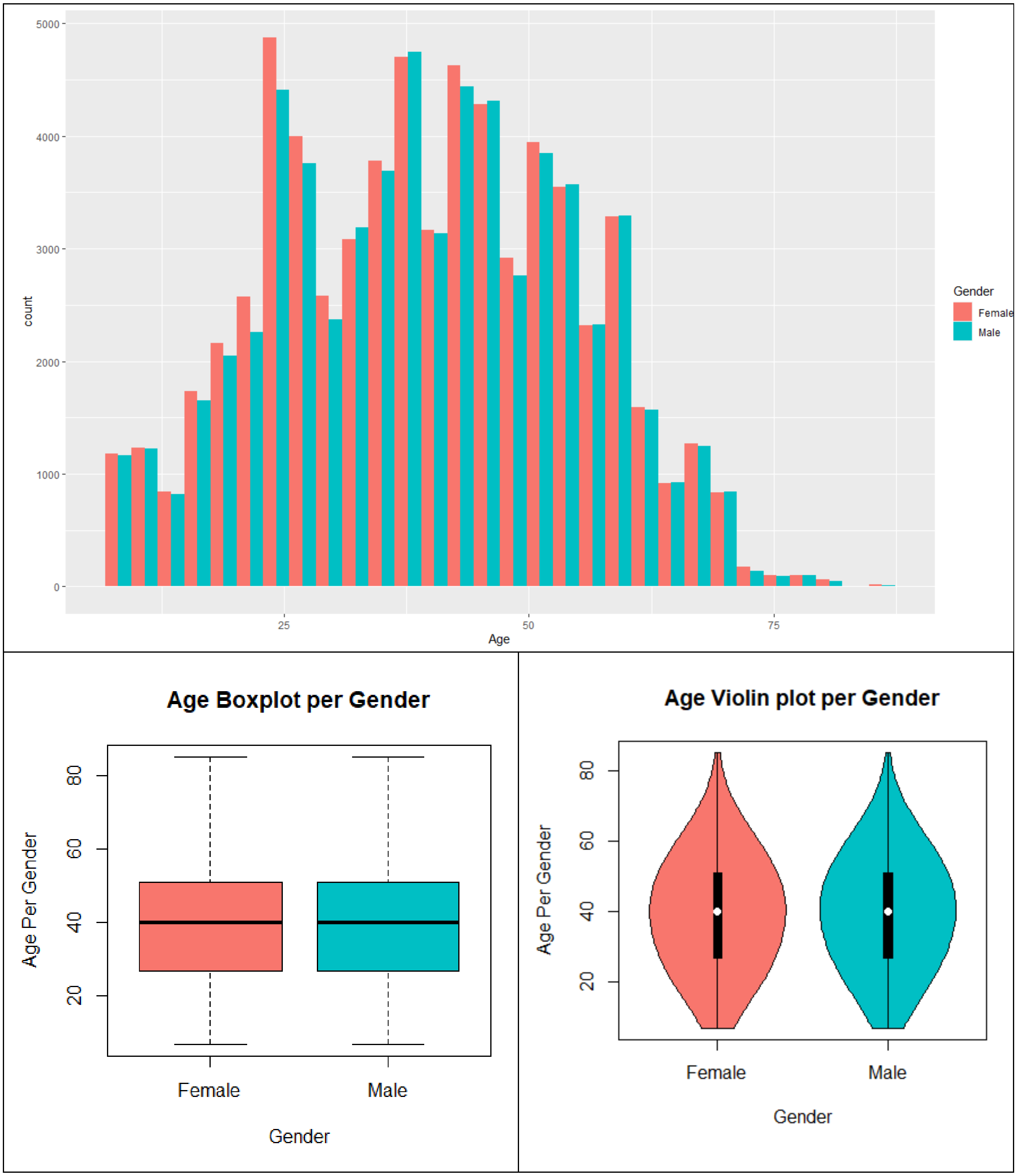
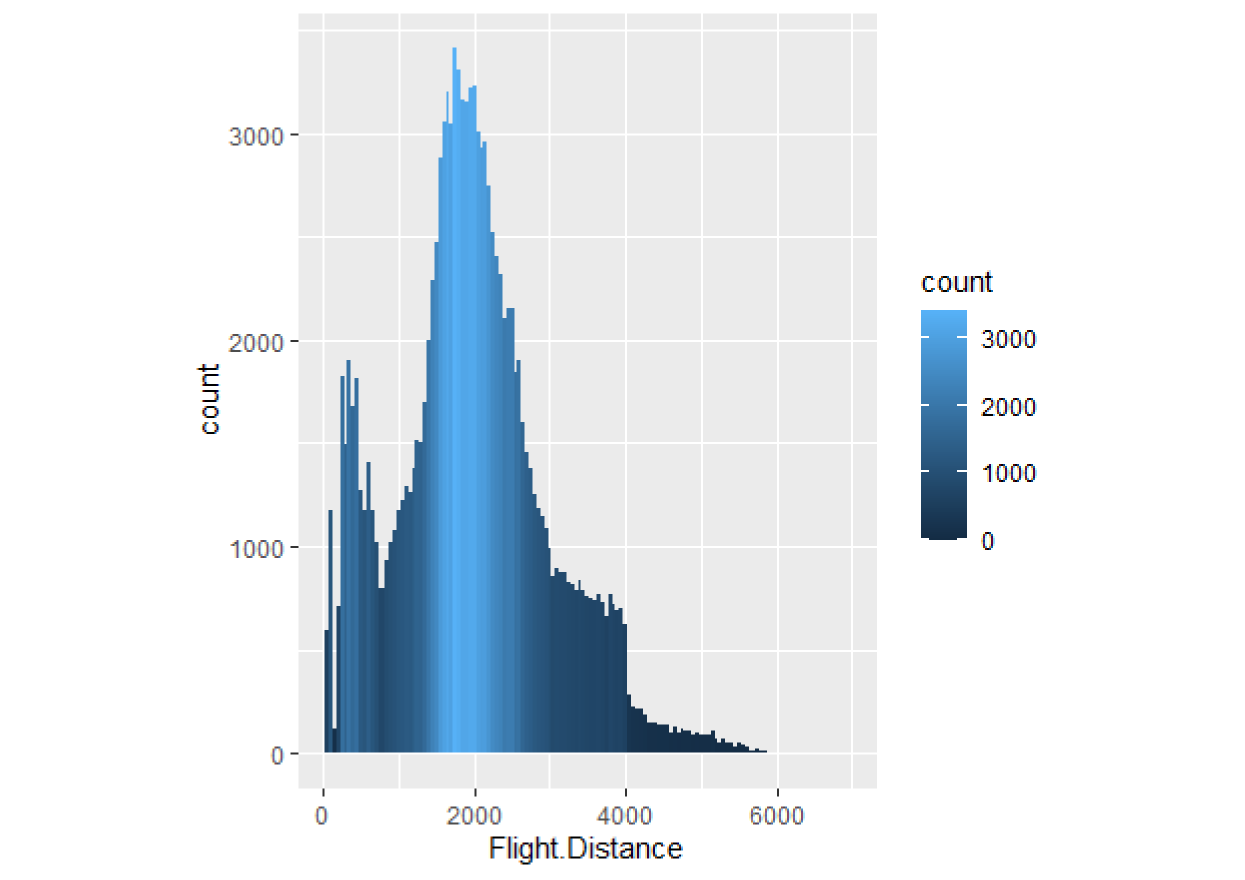


Figure 22 Age Feature Exploration

4.4.7 Flight distance attribute distribution

Now we check the Flight Distance attribute. We could see that that that the flight distance ranges from zero to 6000. The values see to be realistic. I understand that zero is impossible, but we could consider that to be unknown or extremely small. The histogram shows that the values at zero are not that much.

We also examined the distribution between the two genders (male/female) as well as that distribution against the class column. We found that there is no change based on class or change in the flight distance variable.



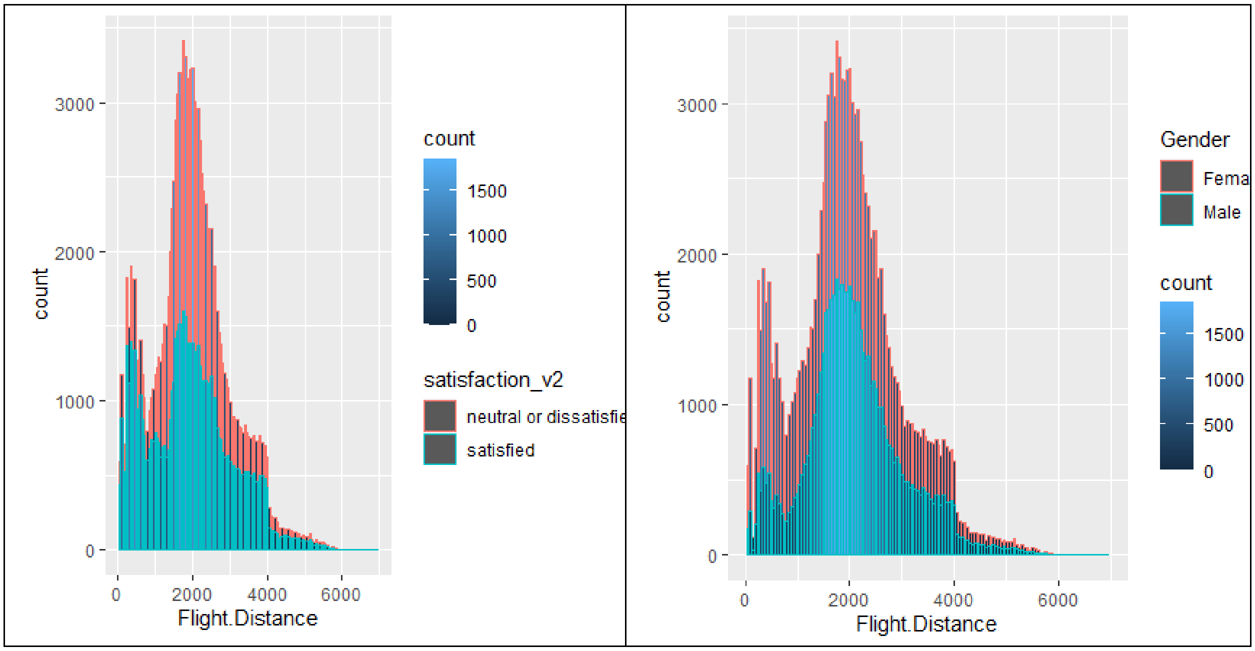
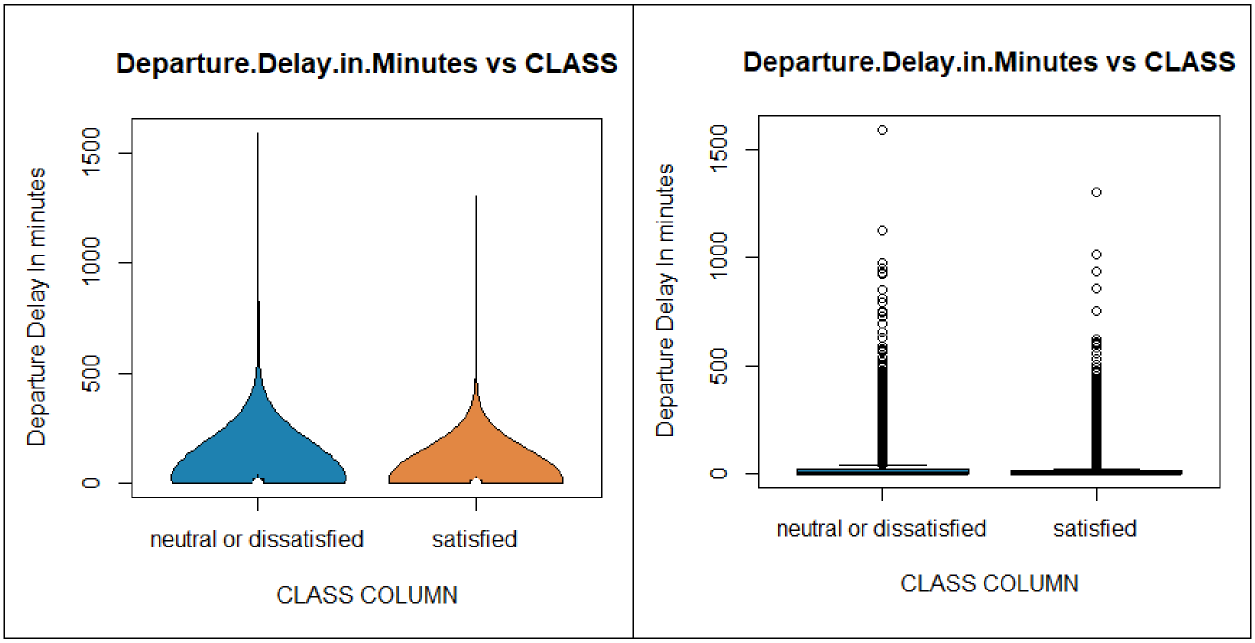


Figure 23 Flight Distance Feature Exploration

4.4.8 Departure Delay Attribute distribution

As for Departure Delay, we could see that the values are not less than 0 at all. The highest normal is about 500 minutes. The delay is not linked to dissatisfied people or satisfied ones. In the histogram we presented only values between [0, 400] to enhance visibility.



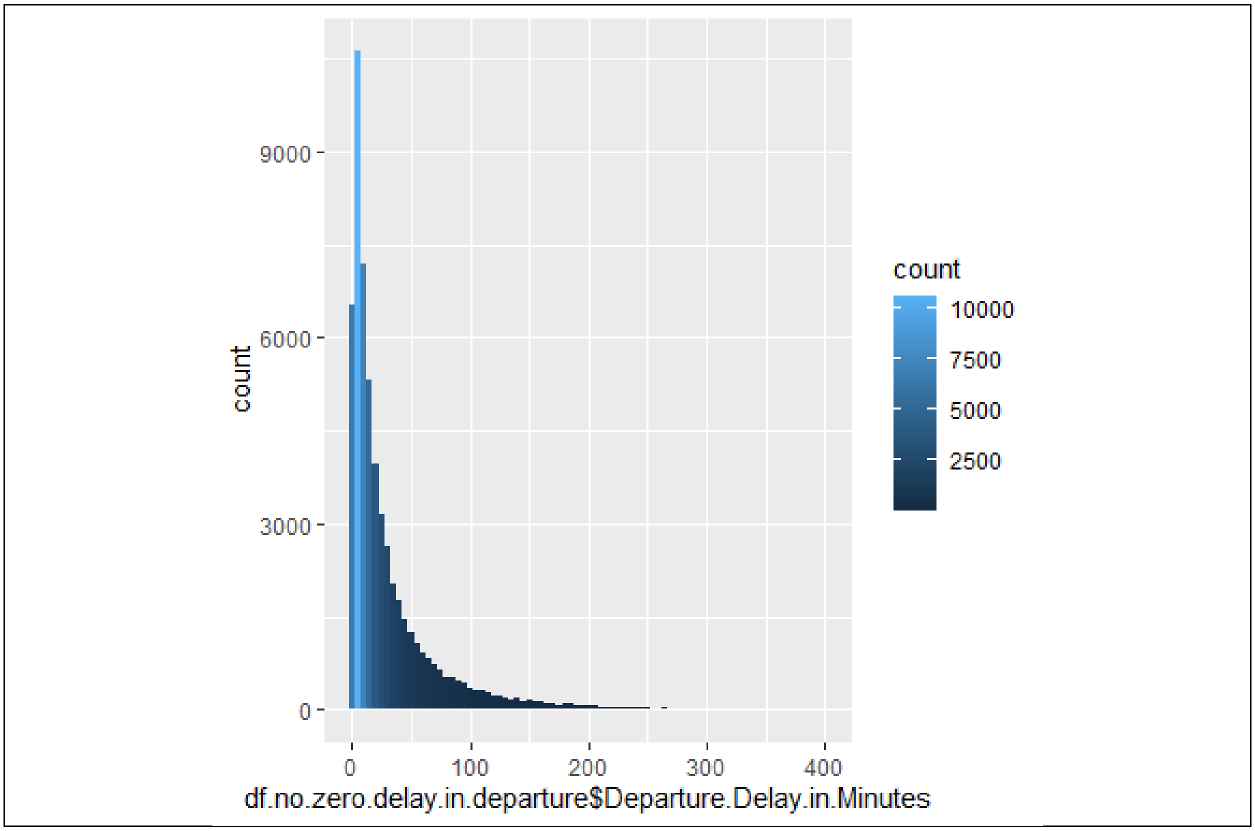


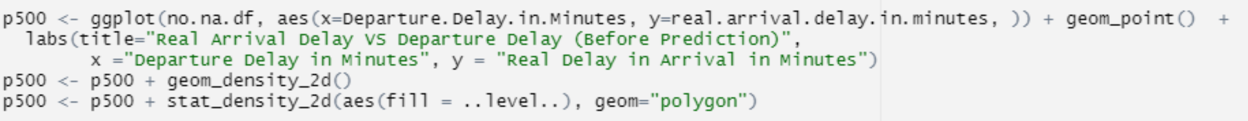
Figure 24 Departure Delay Feature Exploration

4.4.9 Real Delay in Arrival (Before prediction)

Here, we are exploring a new feature that we are generating. It is not predefined. We are going to talk in details in the "Feature Engineering" section. But here we prefer to introduce to show how the feature status before the prediction. The basic idea is: as long as there is a delay in departure, then there is for sure delay in arrival. Hence, the value of delay in arrival is not a good indicator. The value that we have to take into account is the real delay in arrival which is the result of subtraction as follow: Departure delay – Arrival Delay.

The plot below shows how the values are. We could see that although there are some negative values, and the rest are positive. This mean that sometimes the trip arrives earlier than expected.

We will take this plot into account, and we will compare it with the other plot that will hold all points after filling the NA values.



```
p500 <- ggplot(no.na.df, aes(x=Departure.Delay.in.Minutes, y=real.arrival.delay.in.minutes, )) + geom_point()  +
  labs(title="Real Arrival Delay VS Departure Delay (Before Prediction)",
       x ="Departure Delay in Minutes", y = "Real Delay in Arrival in Minutes")
p500 <- p500 + geom_density_2d()
p500 <- p500 + stat_density_2d(aes(fill = ..level..), geom="polygon")
```

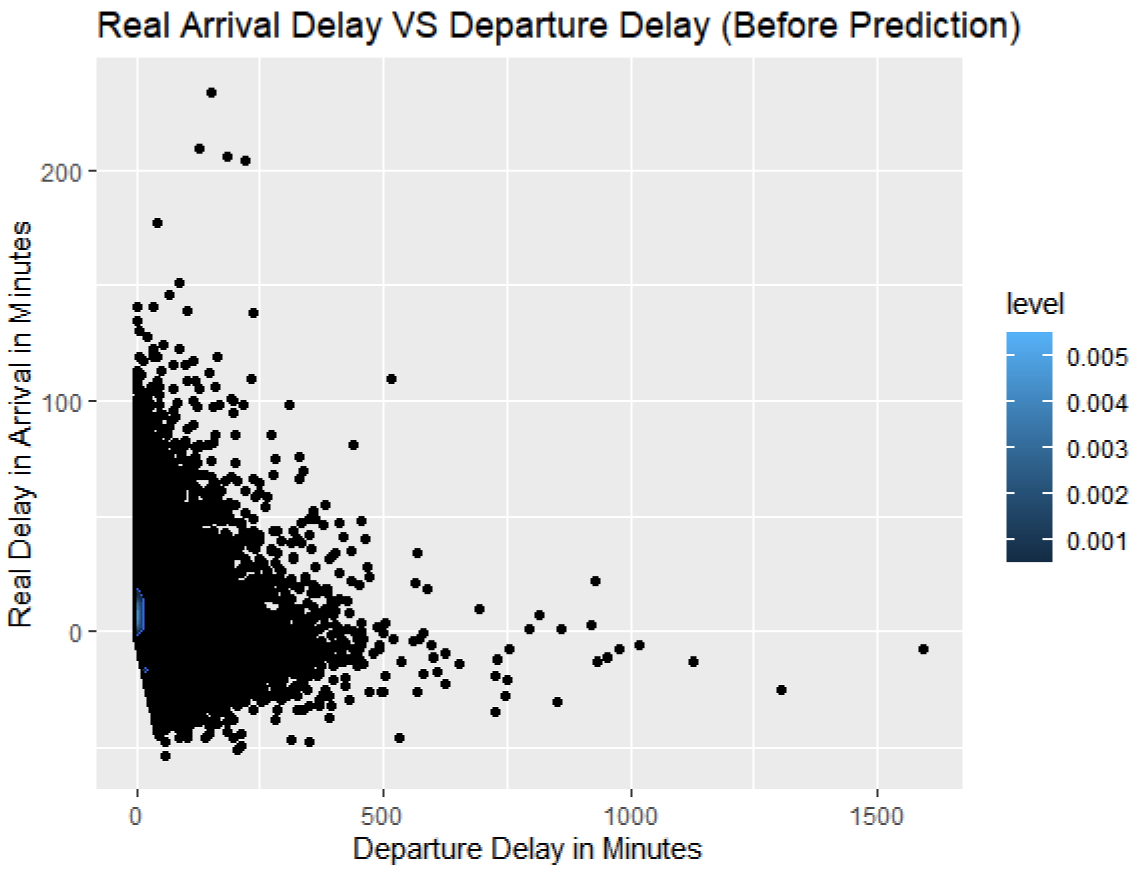


Figure 25 Real Arrival Delay VS Departure Delay (Before Prediction)

4.5 Data Quality Dimensions

By now, we have removed all NAs, checked the correlation between attributes and performed a comprehensive EDA. It is time to check the quality of the quality of the data before we perform any further processing or prediction over the data. There would 6 factors to check for data quality as mentioned below:

1. **Completeness:** We have already checked the whole set of attributes and identified where the NAs exist. We have built a linear regression model to predict the NAs values and filled those values. So, there are no NAs in the dataset, and we have not removed any row.
2. **Conformity:** The categorical variables are defined, and all rows are within that range of the categorical variables. for example, the Gender, Type of Travel, or Class, all have a pre-defined choice list, and we have validated that all cells have values within that range. Either they are strings or integers, they all are ok, and there are no problems in case sensitivity for string values. Even for the star rating of the ground-services or in-flight servicers they are all from 0-5 integers. In addition, the continuous values are also ok, and don't have remarkable outliers.
3. **Consistency:** The attributes are independent from each other's. So, we are having three (3) main categories of the attributes: (1) customer describing, (2) trip describing, and (3) evaluation of services. These are independent even within one category. So, there is no relation between being a male, and the age. Being loyal or traveling on business class. Similarly, each customer may rate

services independently from each other's. Finally, for flight describing, the flight distance might be short or long, but this has nothing to do with late departure and late arrival. So we can consider the dimension of consistency valid for this dataset.

4. **Accuracy:** The data was taken directly from Kaggle, and it has not been manipulated in a way that may affect its accuracy. The accuracy is provided by the publisher of the dataset and hence we can consider the accuracy dimension of data quality is valid.
5. **Duplicates:** We have checked the duplicates by checking if the ID is duplicated or not. We didn't find and column that has duplications
6. **Integrity:** The dataset is not normalized and consist of a single dataset. So, there is no potential missing attributes

4.6 Feature Engineering

In order to have a better machine learning model, we are going to add other features that help to assist the tuples better. The following would be a set of features to add to the dataset.

1. **Real delay in arrival**: The delay in arrival could be caused by several reasons. However, a delay in departure is a direct cause to delay in arrival for obvious reasons. As a result, and based on the study of the dataset, we could see that usually the delay in arrival is almost the same as delay in departure. As a result, we need to have the REAL arrival delay. So, the value would be calculated as fellow: Departure delay – Arrival Delay.

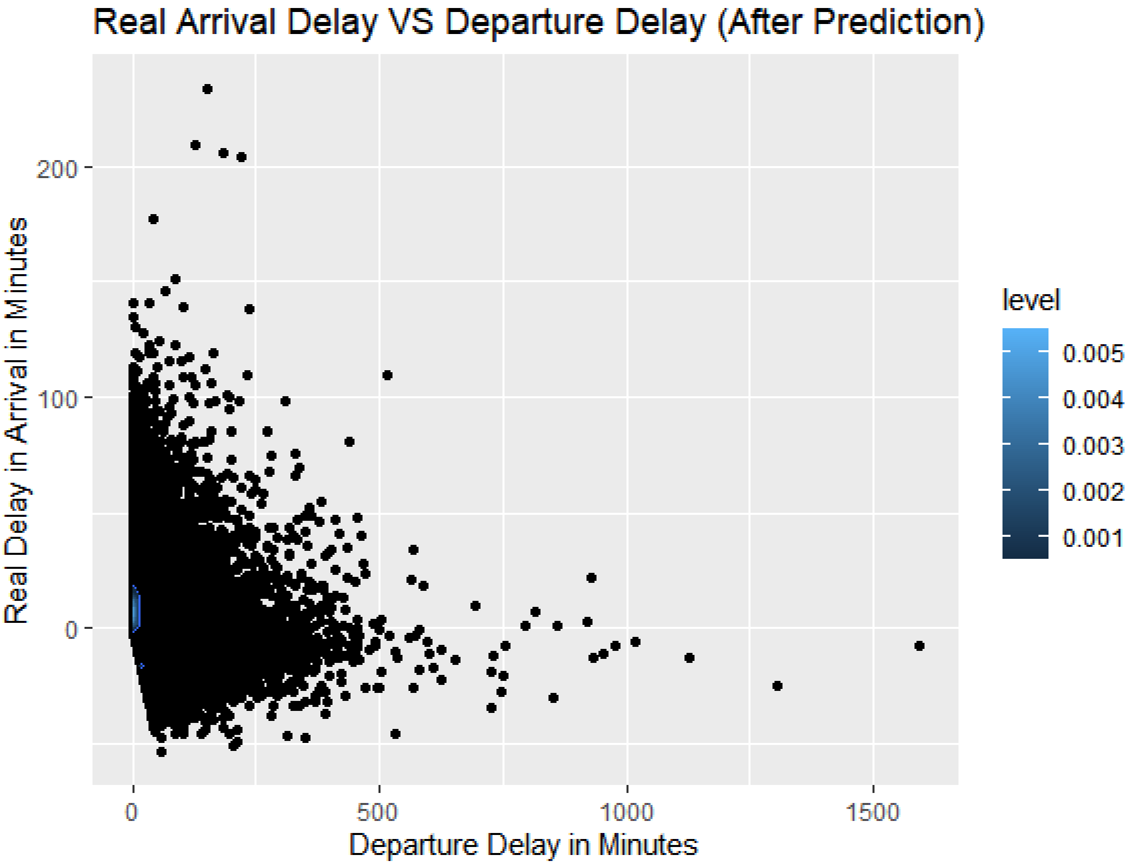


Figure 26 Real Arrival Delay VS Departure Delay (After Prediction)

2. **Flight Distance Range:** Categorize Flight Distance to Short, Moderate, Long

Distance Value	Distance Range
0-2000	Short
2000-4000	Moderate
4000-6000	Long

Table 3 Flight Distance Ranges

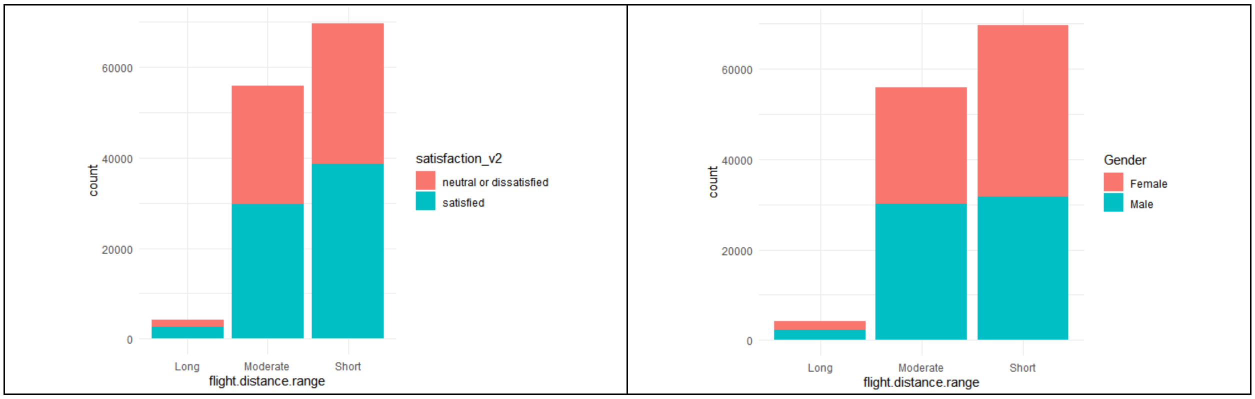


Figure 27 Flight Distance Range Exploration

3. **Age Range:** we may add another feature called Age.Range as the following:

Age Value	Age Range
<20	Under Age
21 - 40	Young
40 - 60	Middle Age
60+	Senior

Table 4 Age Ranges

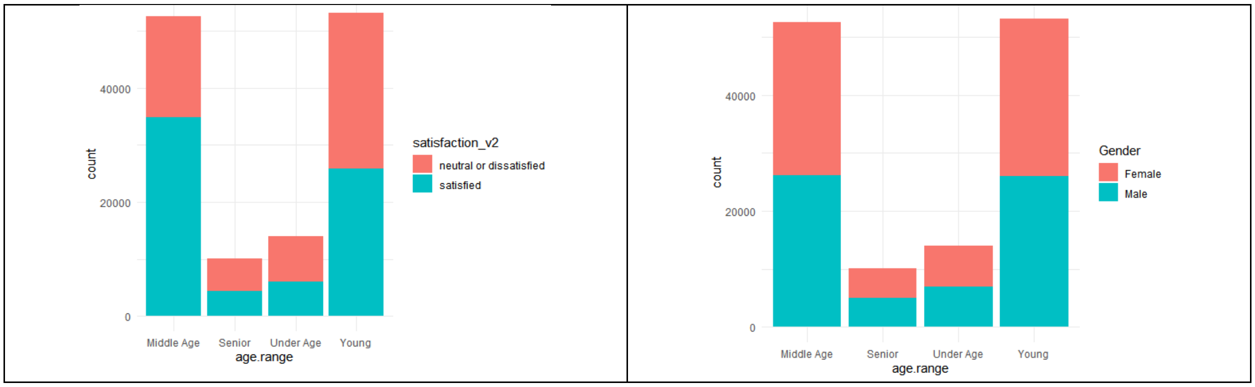


Figure 28 Age Range Exploration

4. **Departure Delay Range**: we may add another feature called Departure.Delay.Range as the following:

Delay Value	Delay Range
< 15 minutes	Trivial
15 - 60	Considerable
60 – 210	Abnormal
210+	Disaster

Table 5 Departure Delay Ranges

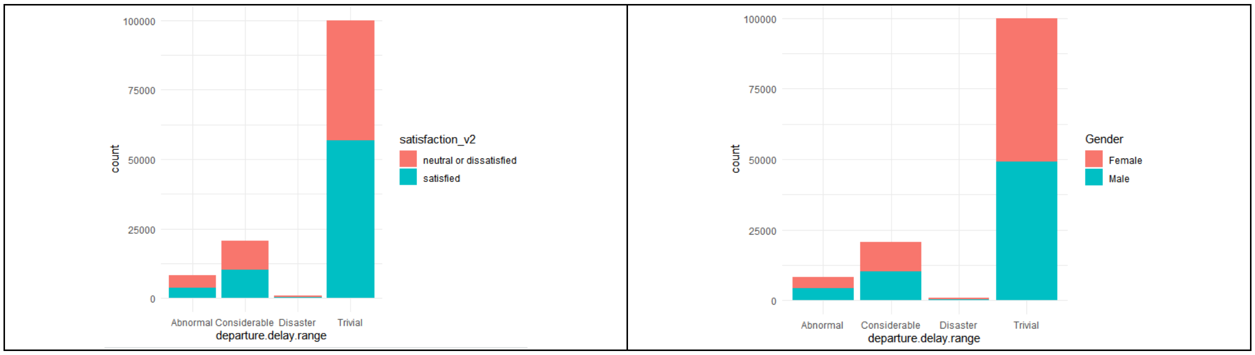


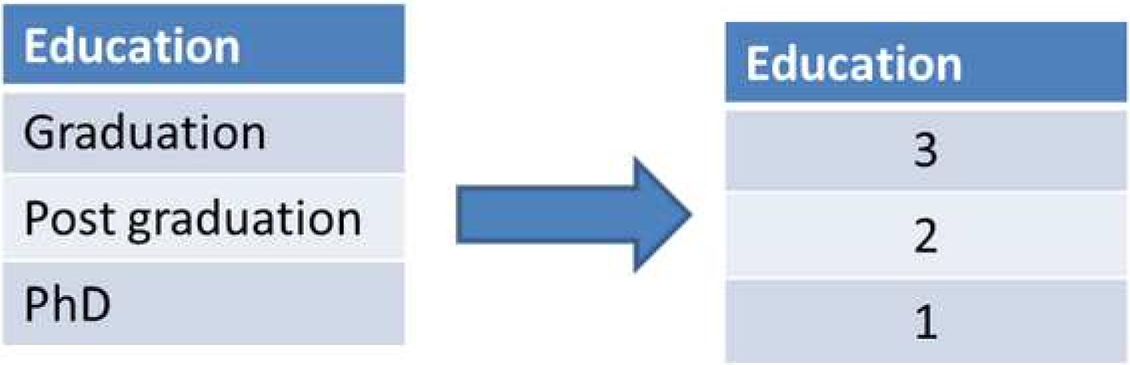
Figure 29 Departure Delay Range

4.7 Feature Scaling (Normalization & Standardization)

In order to perform scaling (either Normalization or Standardization) we need to change the non-numeric values to numeric ones. This is mandatory because many of machine learning models depend of Euclidian Distance to perform.

For categorical values, we can distinguish two types of values: Ordinal and Nominal.

For ordinal, this means having order relationship between them such education level. Let’s say we have (Graduation, Post-Graduation, and PhD) these three values have order between then so we can assign them numbers as illustrated below. This is called **Integer Encoding**. So just mark each unique value as unique integer.



Education
Graduation
Post graduation
PhD

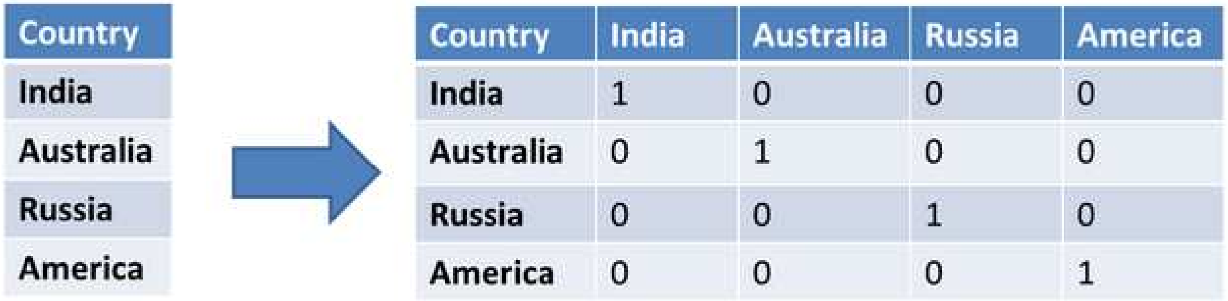
Education
3
2
1

Figure 30 Integer Encoding

However, there are some other types of categorical variables (Nominal values) where that order relationship doesn't exist. So if we encoded the values as 1, 2, 3...etc then we are giving some order information implicitly that doesn't exist. As a result, we need to find another way to do so. The another encoding methodology for such case is called: **One Hot Encoding**.

One hot encoding is used when the order doesn't matter in cases like countries or gender for example.

Let's see how we can encode a list of counties to be used into machine learning models:



Country
India
Australia
Russia
America

Country	India	Australia	Russia	America
India	1	0	0	0
Australia	0	1	0	0
Russia	0	0	1	0
America	0	0	0	1

Figure 31 One Hot Encoding

As we could see, we mapped each value of the countries list into a vector or zeros and single one. That way, the Euclidian-Distance won't be different, and we kept the representation of that different values.

For sure, we won't need the original column any more after performing the encoding.

What we have talked about is the pre-processing needs to be done before performing scaling (Normalization or Standardization). So first convert the non-numeric values to numeric using the two encoding methods as above, and then perform different calculation on these values so the results are:

1. Normalization case: the values would be from [0, 1] because the values are needed in such scale for deep learning models such ANN.
2. Standardization Case: this is needed for algorithms such KNN that uses Euclidian-Distance. If we didn't do so, then the models may yield inaccurate results or funky ones. For standardization, the mean would be 0 and the variance would be 1. This is similar to Gaussian (Normal) Distribution.

Someone might ask, doesn’t Normalization and Standardization change the values of the dataset. The right answer is, they are changing the scale. So, although the values are changed, but together they are kept the same, however with different scale. This what we could see in the below figure:

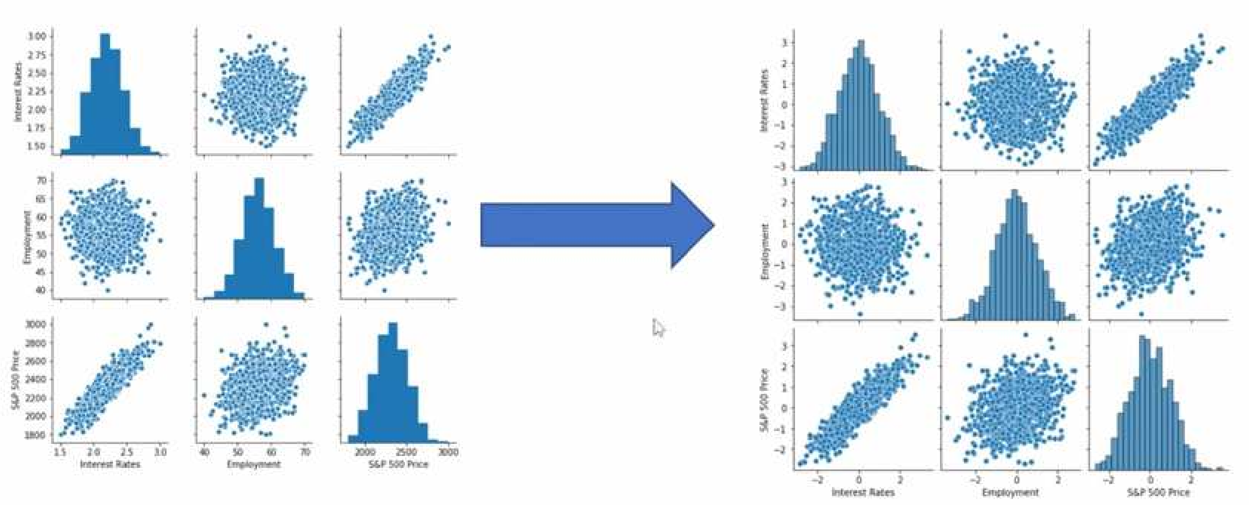


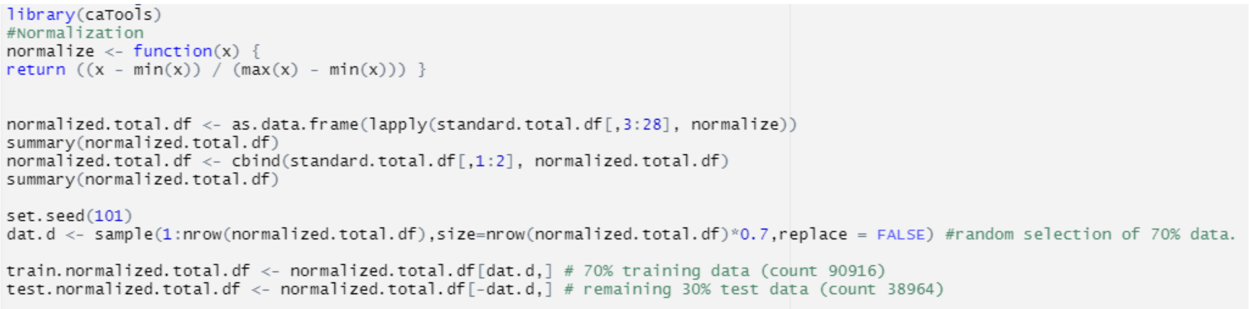
Figure 32 Sample of data Before & After Standardization

Now, in order to perform scaling in R we use the function scale() in R in base package.

However, for normalization, we are going to use the min and max calculation as below:

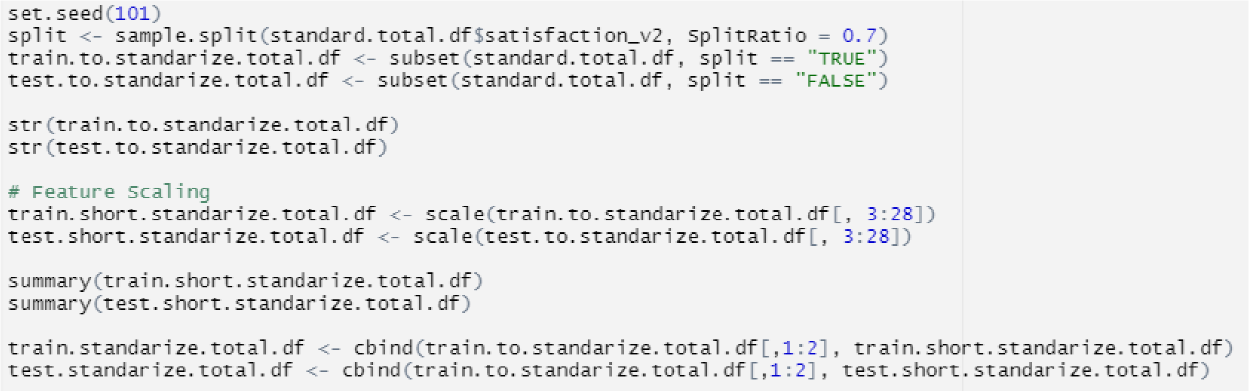
Normalized Value of X = (X – min(X)) / (max(X) – min(X))

Below is the script of R used to do so:



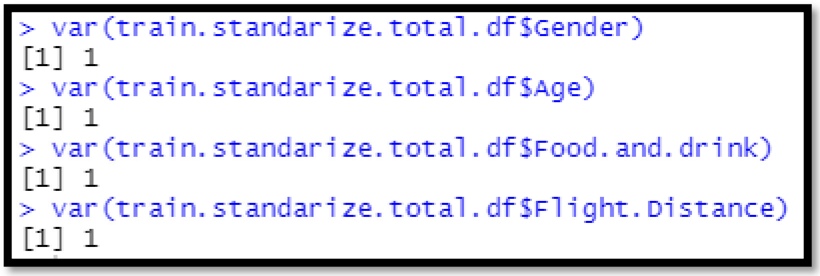
```
library(caTools)
#Normalization
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x))) }


normalized.total.df <- as.data.frame(lapply(standard.total.df[,3:28], normalize))
summary(normalized.total.df)
normalized.total.df <- cbind(standard.total.df[,1:2], normalized.total.df)
summary(normalized.total.df)

set.seed(101)
dat.d <- sample(1:nrow(normalized.total.df),size=nrow(normalized.total.df)*0.7,replace = FALSE) #random selection of 70% data.

train.normalized.total.df <- normalized.total.df[dat.d,] # 70% training data (count 90916)
test.normalized.total.df <- normalized.total.df[-dat.d,] # remaining 30% test data (count 38964)
```

In addition, below is the code we used for Standardization:



```
set.seed(101)
split <- sample.split(standard.total.df$satisfaction_v2, SplitRatio = 0.7)
train.to.standarize.total.df <- subset(standard.total.df, split == "TRUE")
test.to.standarize.total.df <- subset(standard.total.df, split == "FALSE")

str(train.to.standarize.total.df)
str(test.to.standarize.total.df)

# Feature Scaling
train.short.standarize.total.df <- scale(train.to.standarize.total.df[, 3:28])
test.short.standarize.total.df <- scale(test.to.standarize.total.df[, 3:28])

summary(train.short.standarize.total.df)
summary(test.short.standarize.total.df)

train.standarize.total.df <- cbind(train.to.standarize.total.df[,1:2], train.short.standarize.total.df)
test.standarize.total.df <- cbind(train.to.standarize.total.df[,1:2], test.short.standarize.total.df)
```

And here we ensure that the variance for the features became one:



```
> var(train.standarize.total.df$Gender)
[1] 1
> var(train.standarize.total.df$Age)
[1] 1
> var(train.standarize.total.df$Food.and.drink)
[1] 1
> var(train.standarize.total.df$Flight.Distance)
[1] 1
```

4.8 Feature Selection

Feature selection or reducing the number of features in the dataset is essential in machine learning algorithms. This is a matter of fact for several reasons:

1. Removing irrelevant feature would result in a better performance of the model.
2. It makes the model easier to understand and interpret the results.
3. The model with less irrelevant feature would perform faster.

In order to do so, there are lots of techniques and methodologies to do so. In fact, there are books that talk about dimensionality reduction and optimizing feature selection. We will enumerate the most remarkable Feature Selection / Dimensionality Reduction Techniques below:

1. **Percent of Missing Values:**
   If a feature has high percentage of NA values, we could remove the whole feature. This is because the model can not learn from.
2. **Amount of variation:**
   If the values are mostly the same all the time, then the model would not learn from it.
3. **Pairwise correlation**
   If two features are highly correlated, then we can drop one of them, because having them would be redundant. In addition, by dropping one of them, we would not be losing much information.
4. **Multi co-linearity**

This is similar to the pairwise correlation, however, when two or more variables are highly correlated with each other's, then dropping one or more variables should reduce dimensionality without substantial loss of information.

5. **Principal Component Analysis (PCA)**
   PCA is a technique that is considered as statistical procedure that ultimately reduces the number of features of a given dataset, increase interoperability. However, minimize information loss at the same time.
6. **Cluster Analysis**
   It is a dimensionality reduction technique that identify group of features that are correlated among themselves and are uncorrelated with other features in other clusters.
7. **Correlation (with the target)**
   If a variable has low correlation with the target, then we can drop that column. However, we have to be careful when performing such elimination. This is because Column A might not be correlated with the target, and similarly column B might not be correlated with the target so someone might be encouraged to delete both columns. However, Columns A+B together might be correlated with the target. As a result, some other techniques might need to be used.
8. **Forward Selection**
   You select one feature that you think it is the best feature, then you do cross-validation. Then add next best feature into the model, keep going on until you reach some predefined criteria.
9. **Backward elimination (RFE)**
   Unlike Forward selection, start with all variables and try to drop least useful feature. Keep going on until you reach some predefined criteria.
10. **Stepwise Selection**
    This method is similar to Forward Selection; however, the feature may be dropped if it turned out that it is not useful after certain number of steps.
11. **LASSAO**
    An algorithm for creating a regularized linear model. After several adjustment of the reggeization, LASSO drop the coefficient to zero. Then the feature is discarded.
12. **Tree-based Selection**
    In ensembles of trees such Random Forest and other similar ones, it automatically calculates feature importance using Entropy, GINI index, and information Gain.

- Based on the previous demonstrated techniques, we can see that there is no much missing values in our dataset. As a result, the first rule is not eliminating any feature.
- After investigation, it turned out that there are no values that have no variation or very tiny variation in the values. Even the columns with categorical variables have variety of the values. As a result, we will not eliminate any of the features due to small variation issue.
- We found that the (Delay in Departure) and (Delay in Arrival) are highly correlated. As a result, and by applying feature selection, we are going to delete (Delay in Arrival). As a replacement of that feature, we are going to add another feature called (Real Delay in Arrival) which is simply (Delay in Departure) subtracted from (Delay in Arrival).
- We will delete customer Id column by common sense as it won't be related to the learning. We would also delete any other column, if exist, such identity card, name…etc. but we don't have.

- Tree-Based Selection is already applied internally in Decision Trees and in Random Forest Algorithms.

4.9 Variable Dictionary

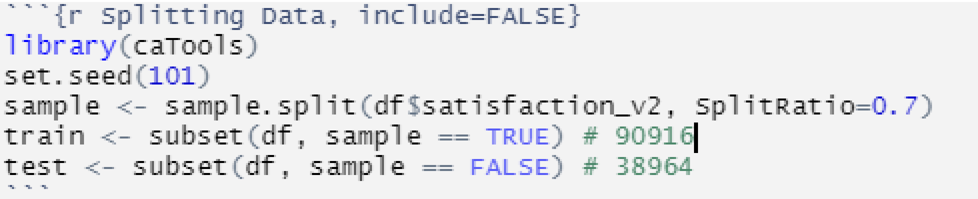
#	Column	Data Type	Explanation
1	Id	Integer	Customer Id
2	Satisfaction	String	Class column
3	Gender	String	Male/Female
4	Customer Type	String	Loyal or not
5	Age	Integer	Age in years
6	Type of Travel	String	Personal/Business
7	Class	String	Eco/Business
8	Flight Distance	Integer	Distance in KM
9	Seat comfort	Integer	Level from 0 to 5
10	Departure/Arrival time convenient	Integer	Level from 0 to 5
11	Food and Drink	Integer	Level from 0 to 5
12	Gate Location	Integer	Level from 0 to 5
13	Inflight Wi-Fi service	Integer	Level from 0 to 5
14	Inflight entertainment	Integer	Level from 0 to 5
15	Online support	Integer	Level from 0 to 5
16	Ease of online booking	Integer	Level from 0 to 5
17	Onboard service	Integer	Level from 0 to 5
18	Leg room service	Integer	Level from 0 to 5
19	Baggage handling	Integer	Level from 0 to 5
20	Check in service	Integer	Level from 0 to 5
21	Cleanness	Integer	Level from 0 to 5
22	Online boarding	Integer	Level from 0 to 5
23	Departure Delay in minutes	Integer	Departure Delay in minutes
24	Arrival Delay in minutes	Integer	Arrival Delay in minutes
25	**Real delay in arrival**	Integer	Departure delay – Arrival Delay
26	**Flight Distance Range**	String	Short, Moderate, Long
27	**Age Range**	String	Under Age, Young, Middle Age, Senior
28	**Departure Delay Range**	String	Trivial, Considerable, Abnormal, Disaster

Table 6 Variable Dictionary of Final Dataset

4.10 Data Separation

In order to perform the modeling and conduct models performance check, we would need to split the dataset we have into two portions. The first portion would be the training data, which will be used to train the model. The second portion would be the testing one. This would be used to evaluate the performance of the model by checking the predicted values against the actual known values.

The code used to perform the split would be as the following:

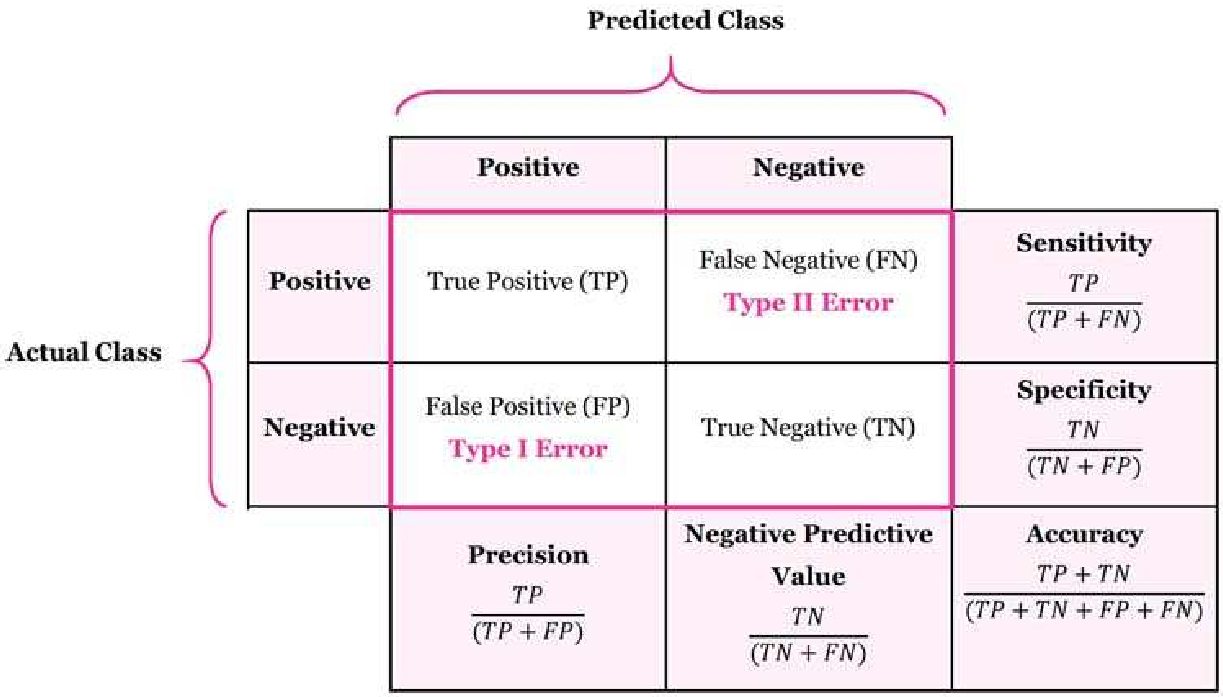


~~~
```{r Splitting Data, include=FALSE}
library(caTools)
set.seed(101)
sample <- sample.split(df$satisfaction_v2, SplitRatio=0.7)
train <- subset(df, sample == TRUE) # 90916
test <- subset(df, sample == FALSE) # 38964
```
~~~

We first import the library caTool which is providing the functionality for splitting data. We use the line set.seed to perform random split of the dataset. We would include any column in the split function, but preferably the class column. The split ratio is subjective. Some would choose 70:30, others would choose 80:20 and rest would elect 90:10 percentage for training:testing respectively.

4.11 Data Modeling

Now we have finished exploring the dataset and we are done with the required data preprocessing. Let's now start modeling our data and create variety of models and check their performance. We are going to create variety of models. Some of them are rule-based such as decision-trees and Random Forest. These ones are not depending on Euclidean-Distance. Others are probabilistic such Naïve-Bayes. We will study SVM and KNN which are dependent on Euclidean-Distance thus would need to have data scaling such as standardization. At the end we will introduce ANN as an example of deep learning which would better to have normalization to make values ranges from [0, 1] to be used in perceptron. We are going to evaluate each model and compare the metrics of evaluation of all models in a radar polygon so comparison of the results would be easy and visually comparable. The evaluation of each model would be through confusion matrix. Confusion Matrix could be defined as follow:



		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure 33 Confusion Matrix illustration

We are going to consider five metrics for the performance of the classification model as follow:

#	Metric Name	Formula
1	**Accuracy**	(TP + TN) / (TP + TN + FP + FN)
2	**Precision**	TP / (TP + FP)
3	**Sensitivity (Recall)**	TP/(TP+FN)
4	**Specificity**	TN/(TN+FP)
5	**F1 score**	2*(Recall * Precision) / (Recall + Precision)

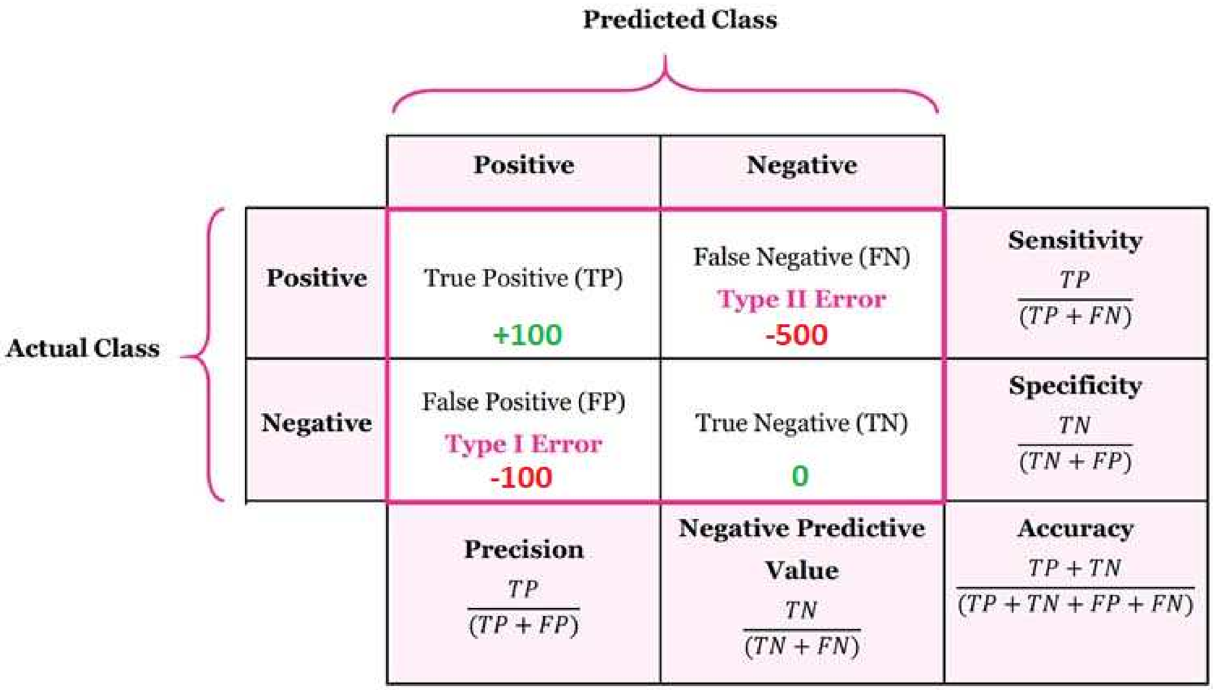
Table 7 Classifiers Performance Metrics Formulas

Below we shortly explain what each metric implies. Say we predict a disease, then positive means infected:

#	Metric Name	Explanation
1	**Accuracy**	How many truly predicted out of all
2	**Precision**	How many of those who we labeled as positive are positive?
3	**Sensitivity (Recall)**	Of all the people who are positive, how many of those we correctly predict?
4	**Specificity**	Of all the people who are negative, how many of those did we correctly predict?
5	**F1 score**	F1 Score is best if there is some sort of balance between precision (p) & recall (r) in the system. Oppositely F1 Score isn't so high if one measure is improved at the expense of the other.

Table 8 Classifiers Performance Metrics Equations

There is a further way to evaluate the performance of classification model. The other method is called the Cost Matrix. The cost matrix is very similar to confusion matrix. However, it concentrates on calculating the cost of wrong prediction. In other words, say that we predict someone is not infected but he is. If the percentage of such cases is pretty low (say 0.1%) then the accuracy would be (say 99.9%) but this gives a misleading understanding. So, we need a method to adjust the weigh assigned to miss-classified cases. This method is cost matrix.



		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP) +100	False Negative (FN) Type II Error -500	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error -100	True Negative (TN) 0	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure 34 Cost Matrix Illustration

In the Cost Matrix Illustration above, we see that for TP (True Positive) we offer +100 points. However, to FP (False Positive) we penalize for -100 points. This could be translated as:

- If we predicted a patient as sick and he really is, then we gratify the model with +100 point.
- If we predicted a patient as sick and he is NOT, then we penalize the model with -100 point.

Similarly, we penalize the model for FN, and be neutral for TN. This could be translated as:

- If we predicted a patient as healthy while he is sick, then we penalize the model with -500 points
- If we predicted a patient as healthy and he is healthy, then we are neutral and add nothing.

That way we lower Accuracy for Type 1&2 errors, because TP & TN are not promoted same for FN & FP

Cost matrix is a good way to give exact performance metrics for classifier models especially when the classes are imbalanced. Since we found that the result class in our case is almost balanced, so we will not use the cost matrix for performance evaluation and will depend only on confusion matrix.

4.11.1 Decision Tree

The first modeling we tackle would be Decision Trees. This machine learning algorithm is famous and has several advantages over other machine learning algorithms. First it is easy to use and doesn't need much preprocessing such as scaling (normalization or standardization). Not sensitive towards NAs and very easy and intuitive to explain to stakeholders. However, it has a lot of disadvantages that must be considered. The most critical disadvantage is that the model generated is adaptive to dataset. It is likely to have the accuracy much lower when it predicts new data. In other words, it is overfitting.

Below we would see that two decision trees built from two subsets of the same dataset. Although we applied the same algorithm. However, since we had two different subsets so we got different decision trees. That mean the result is related to the training dataset, and the resulting tree is not the best for unseen data.

The first sample of decision tree is as follows:

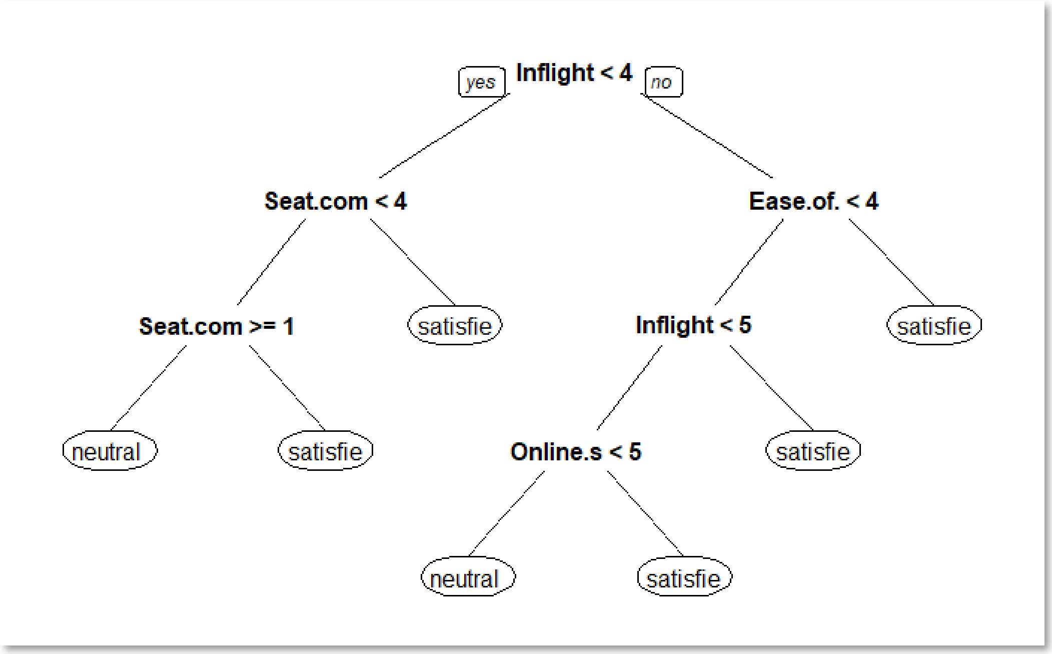


Figure 35 First Sample of Decision Tree

The second sample of decision tree is as follows:

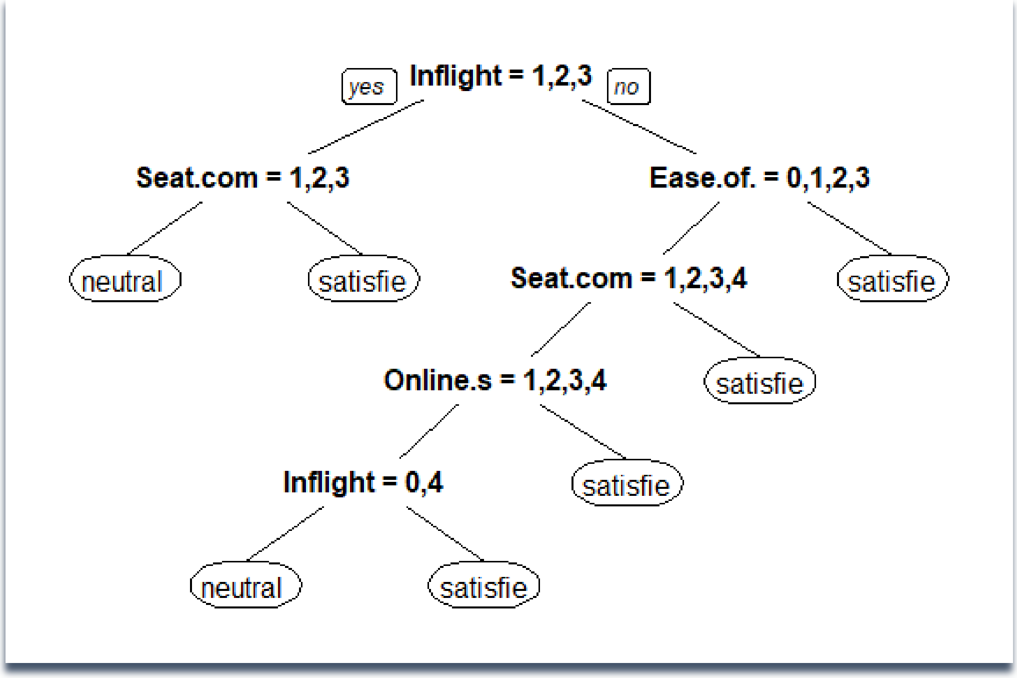
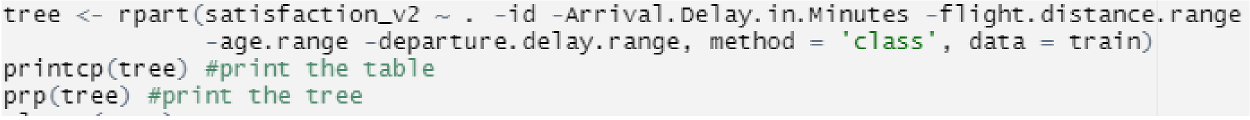


Figure 36 Second Sample of Decision Tree

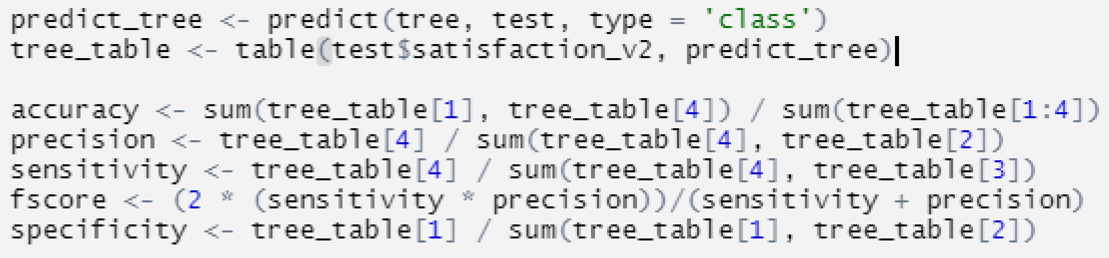
This problem would be overcome using the other algorithm Random Forest which construct many bootstrapped decision trees and take the average of them.

Now let's construct the model and print the tree:



```
tree <- rpart(satisfaction_v2 ~ . -id -Arrival.Delay.in.Minutes -flight.distance.range
              -age.range -departure.delay.range, method = 'class', data = train)
printcp(tree) #print the table
prp(tree) #print the tree
```

The performance metrics of the algorithm Could be calculated using confusion matrix as follows:



```
predict_tree <- predict(tree, test, type = 'class')
tree_table <- table(test$satisfaction_v2, predict_tree)

accuracy <- sum(tree_table[1], tree_table[4]) / sum(tree_table[1:4])
precision <- tree_table[4] / sum(tree_table[4], tree_table[2])
sensitivity <- tree_table[4] / sum(tree_table[4], tree_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- tree_table[1] / sum(tree_table[1], tree_table[2])
```

Now we are going to check the confusion matrix and the performance metrics:

#	Metric Name	Value
1	**Accuracy**	0.869725900831537
2	**Precision**	0.883944480915315
3	**Sensitivity (Recall)**	0.878752563863509
4	**Specificity**	0.858668341708543
5	**F1 score**	0.881340876151293

Table 9 Decision Tree Performance Metrics

4.11.2 Random Forest

Random Forest is invented in response to limitations of Decision Trees algorithm. Data scientists need machine learning algorithm that uses the same basic concepts of Decision Trees but overcome its limitations. Random Forest has several advantages over Decision Trees as follow:

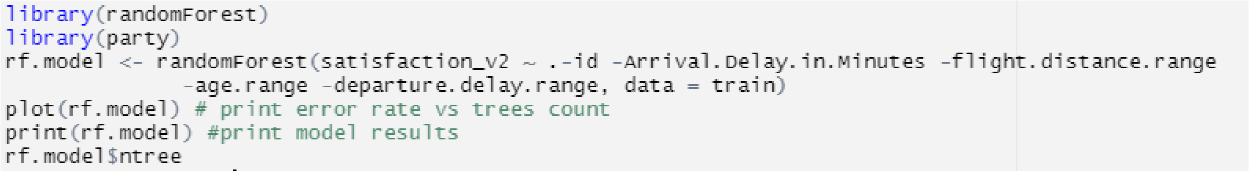
1. The Accuracy provided by Random Forest is high in general.
2. Random Forest is an efficient algorithm even in large datasets.
3. Unlike Decision Tree, it could give an order of importance of model variables.
4. It doesn't overfit and it is not adaptive to training dataset.

Random Forest works differently than Decision Trees. As the name implies, it creates a random set of ensemble trees, and based on max votes of trees it provides the final decision.

Although Random Forest has many advantages, but it also has some disadvantages as follow:

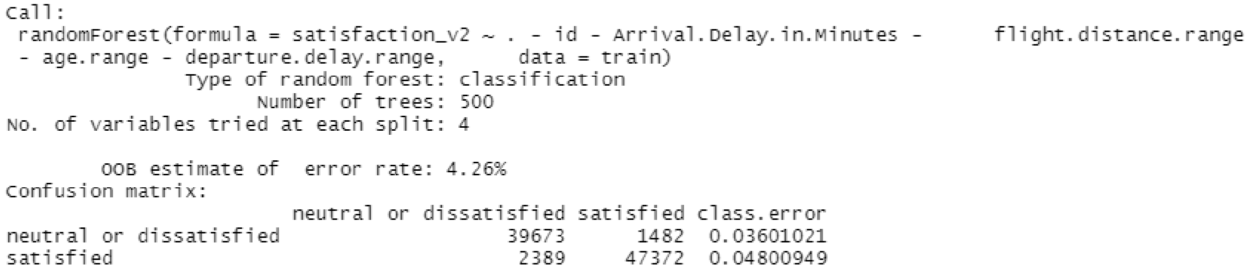
1. High computation power as well as resources are required since it creates an immense number of trees and do combination to result outputs.
2. Unlike Decision Trees, it suffers interpretability because of ensemble of decision trees.
3. The training time is much higher as it needs to combine a lot of decision trees.

Below we are going to create Random Forest model and evaluate it.



```
library(randomForest)
library(party)
rf.model <- randomForest(satisfaction_v2 ~ .-id -Arrival.Delay.in.Minutes -flight.distance.range
              -age.range -departure.delay.range, data = train)
plot(rf.model) # print error rate vs trees count
print(rf.model) #print model results
rf.model$ntree
```

Then we can see model results and it used the default number of trees 500



```
Call:
 randomForest(formula = satisfaction_v2 ~ . - id - Arrival.Delay.in.Minutes -      flight.distance.range
 - age.range - departure.delay.range,      data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 4.26%
Confusion matrix:
                        neutral or dissatisfied satisfied class.error
neutral or dissatisfied                   39673      1482  0.03601021
satisfied                                  2389     47372  0.04800949
```

As we could see from the chart below, it would have been enough if we used only 200 trees. Cause after that the error rate is not going down.

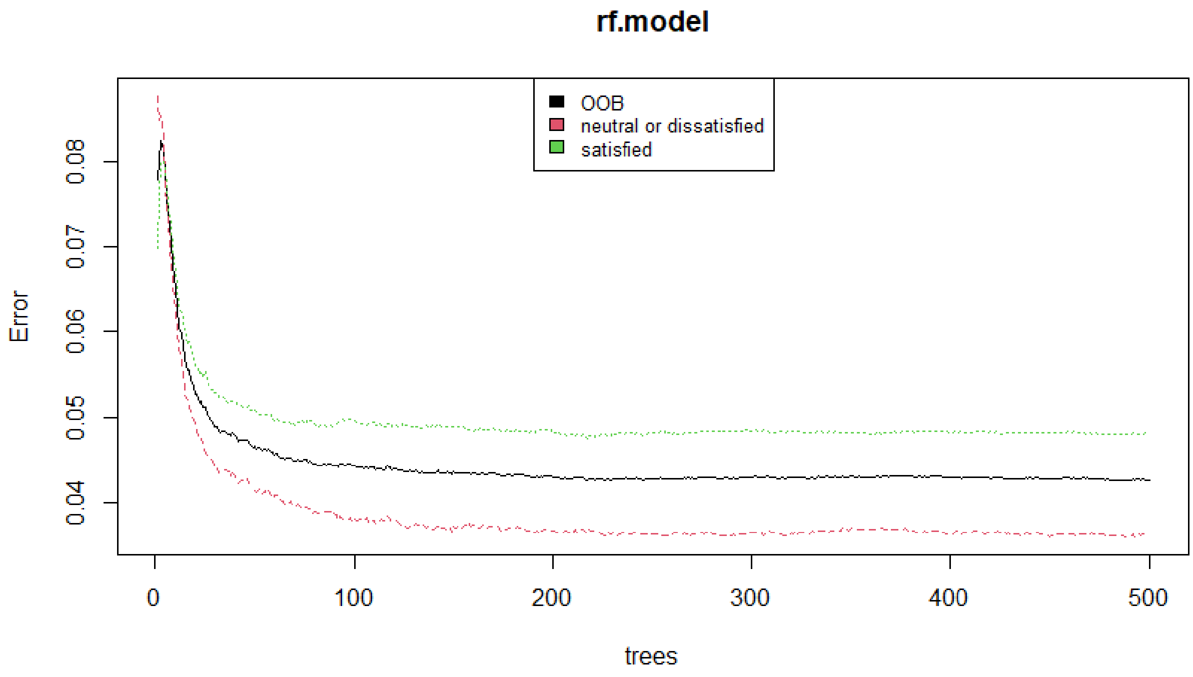
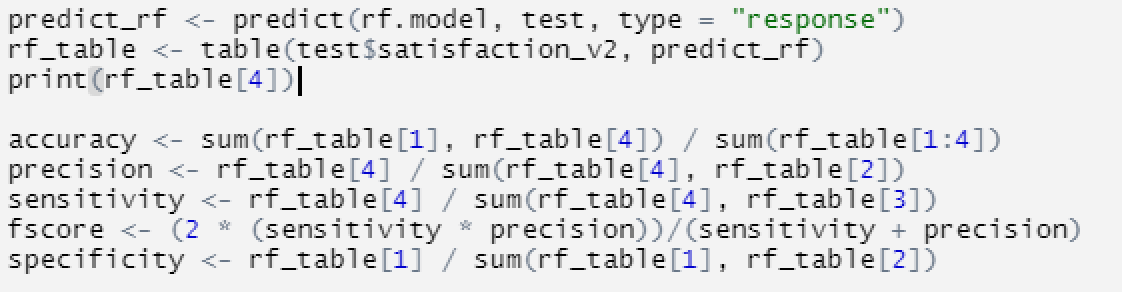


Figure 37 Error vs Trees Count in Random Forest

Now it is time to evaluate the performance of the model. We established the confusion matrix.



```
predict_rf <- predict(rf.model, test, type = "response")
rf_table <- table(test$satisfaction_v2, predict_rf)
print(rf_table[4])

accuracy <- sum(rf_table[1], rf_table[4]) / sum(rf_table[1:4])
precision <- rf_table[4] / sum(rf_table[4], rf_table[2])
sensitivity <- rf_table[4] / sum(rf_table[4], rf_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- rf_table[1] / sum(rf_table[1], rf_table[2])
```

We got the results as follow from confusion matrix:

#	Metric Name	Value
1	**Accuracy**	0.959526742634226
2	**Precision**	0.952593078870862
3	**Sensitivity (Recall)**	0.972894018485705
4	**Specificity**	0.944091135320467
5	**F1 score**	0.96263652948563

Table 10 Random Forest Performance Metrics

We see that we got accuracy pretty similar to the one we got from the model result. We will discuss the importance order of attributes in later section when we discuss the importance of the variables.

4.11.3 Logistic Regression

Logistic Regression is a machine learning algorithm used to resolve binary classification problems. It uses sigmoid function to convert the linear regression to be sigmoid curve:

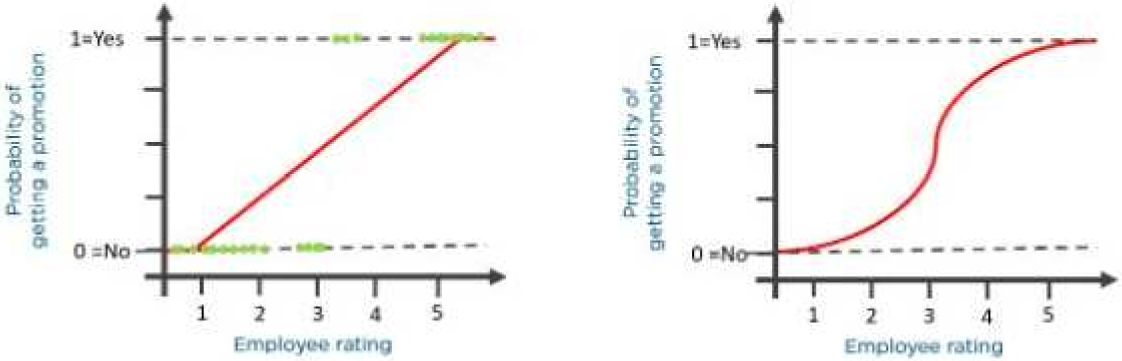


Figure 38 Linear Regression vs Logistic Regression

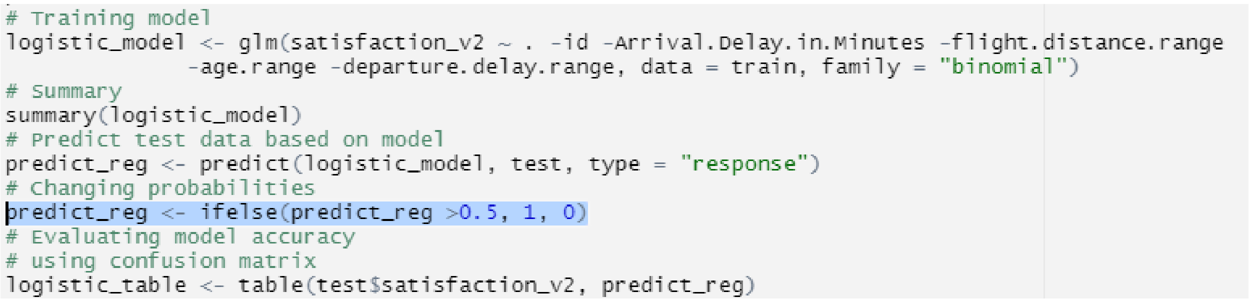
The sigmoid function looks like the following equation:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Figure 39 Sigmoid Function in Logistic Regression

The following lines build the model of logistic regression, and then use that model to predict the values of the class column for the testing dataset.

Main point here, the predicted value is not the class value (in our case satisfied or not, or (1 or 0)) it would be the probability of being either satisfied = 1 or dissatisfied = 0. So, we will consider any probability for example above 0.5 as satisfied, and any probability below that as dissatisfied = 0. In other words, we consider 0.5 as threshold (cutoff).



```
# Training model
logistic_model <- glm(satisfaction_v2 ~ . -id -Arrival.Delay.in.Minutes -flight.distance.range
              -age.range -departure.delay.range, data = train, family = "binomial")
# Summary
summary(logistic_model)
# Predict test data based on model
predict_reg <- predict(logistic_model, test, type = "response")
# Changing probabilities
predict_reg <- ifelse(predict_reg >0.5, 1, 0)
# Evaluating model accuracy
# using confusion matrix
logistic_table <- table(test$satisfaction_v2, predict_reg)
```

After that we will construct the confusion matrix and calculate the performance metrics:

#	Metric Name	Value
1	**Accuracy**	0.907991992608562
2	**Precision**	0.910391071930976
3	**Sensitivity (Recall)**	0.920622125278581
4	**Specificity**	0.893090909090909
5	**F1 score**	0.915478014853236

Table 11 Logistic Regression Performance Metrics

Now let's discuss extra performance topics about Logistic Regression. Let's talk about ROC. The initial plot of ROC looks like this to the left below. ROC is a comparison between Specificity vs Sensitivity. It helps to find the point that has maximum of both. We do some customization, so we find the below chart to the right. The new axis names are False Positive Rate (FPR) and True Positive Rate (TPR):

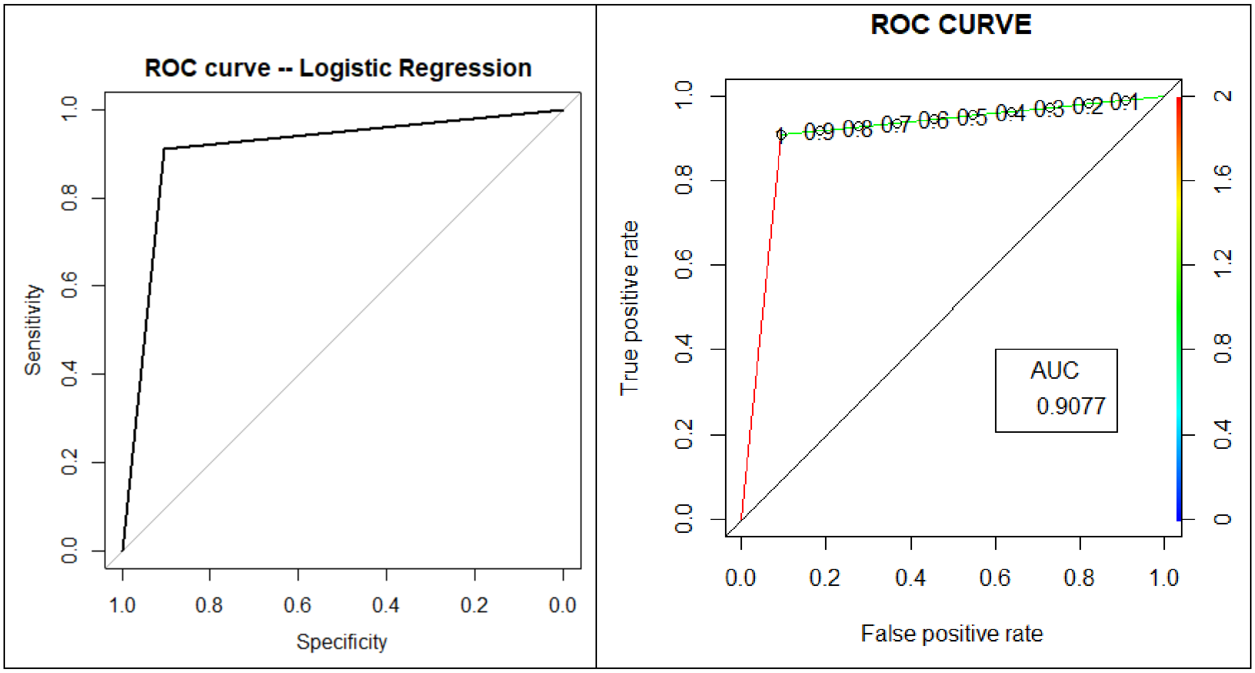
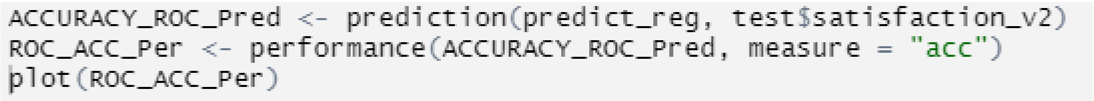


Figure 40 ROC Curve of Logistic Regression

We can be more confident that this model has great performance as the AUC is calculated and it is 0.9077. Let's remember that AUC could be one the value of interval [0.5, 1]. In addition, let's remember that the best point of ROC is the one that hits the top-left corner. Now let's try to make sure that our threshold (cutoff) was correct. Let's plot the cutoff vs accuracy. We obtained that figure using code as below:



```
ACCURACY_ROC_Pred <- prediction(predict_reg, test$satisfaction_v2)
ROC_ACC_Per <- performance(ACCURACY_ROC_Pred, measure = "acc")
plot(ROC_ACC_Per)
```

We could see that our assumption about the threshold was correct. As we get the cutoff at 0.2 or 0.8 the accuracy gets lower.

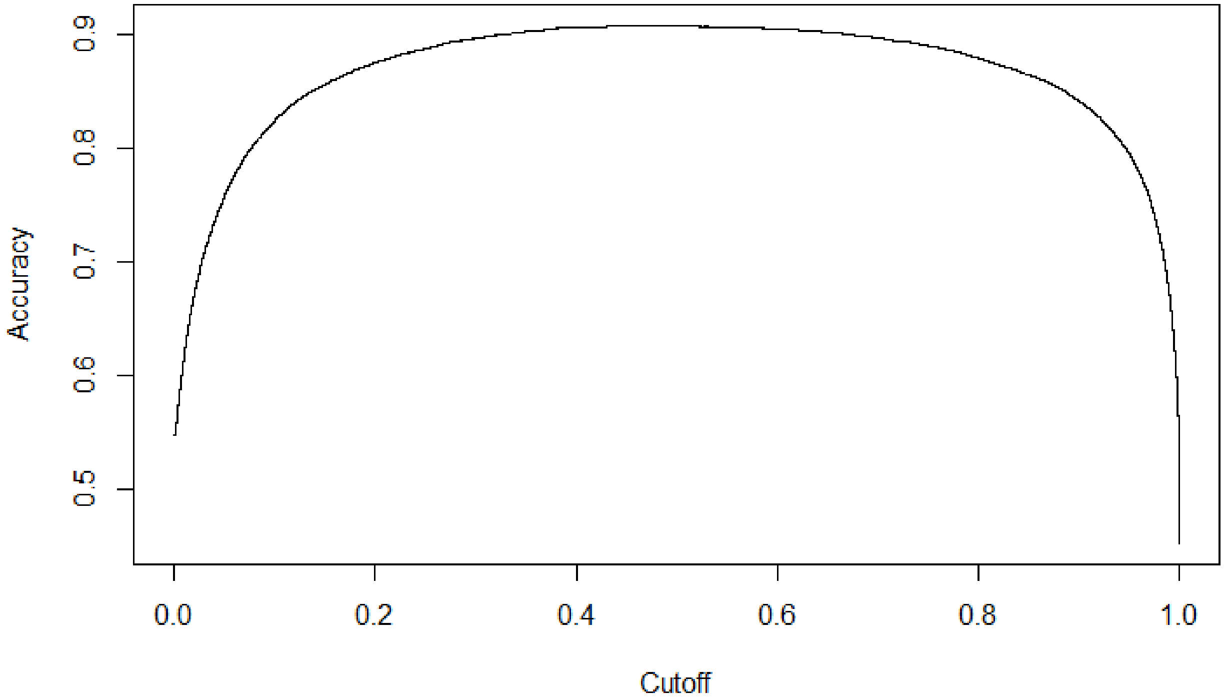


Figure 41 Logistic Regression Cutoff vs Accuracy

4.11.4 Naive Bayes

Naive Bayes is a machine learning algorithm used to solve classification problems. The approach it uses is the *probabilistic approach*. The mathematical background behind this algorithm came from Bayes theorem which could be described in the following conditional probability equation:

$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}$$

Figure 42 Bayes Theorem

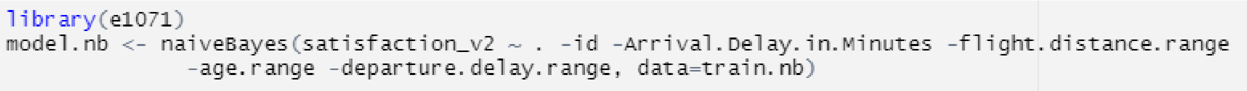
Naive Bayes is called Naïve because it assumes the independence between predictors when it calculates the probability of each. We have to say, in real-world problem, this is not always true. There would be some correlation between two variables or more that affects together the predicted value of the problem.

Let’s have some formal definitions of the components:

#	Expression	Definition
1	A	The proposition and B is the evidence
2	P(A)	The prior probability of the proposition
3	P(B)	The prior probability of evidence
4	P(A|B)	The posterior
5	P(B|A)	The likelihood

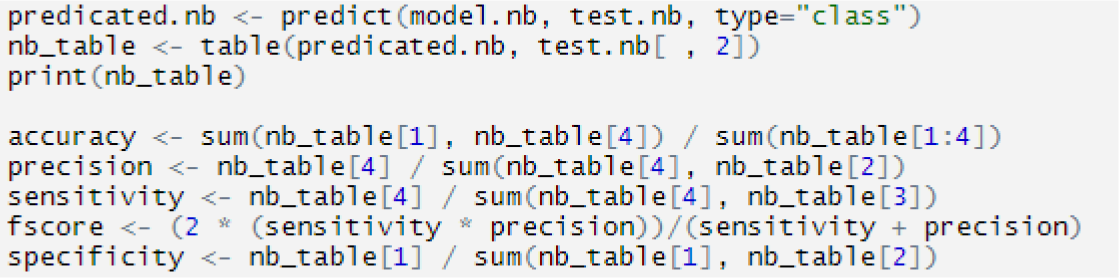
Table 12 Naive Bayes Definitions

Now let’s build the Naïve Bayes model:



```
library(e1071)
model.nb <- naiveBayes(satisfaction_v2 ~ . -id -Arrival.Delay.in.Minutes -flight.distance.range
                -age.range -departure.delay.range, data=train.nb)
```

Now let’s calculate the performance metrics:



```
predicated.nb <- predict(model.nb, test.nb, type="class")
nb_table <- table(predicated.nb, test.nb[ , 2])
print(nb_table)

accuracy <- sum(nb_table[1], nb_table[4]) / sum(nb_table[1:4])
precision <- nb_table[4] / sum(nb_table[4], nb_table[2])
sensitivity <- nb_table[4] / sum(nb_table[4], nb_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- nb_table[1] / sum(nb_table[1], nb_table[2])
```

Below are the metrics values that evaluate the performance of the model we created

#	Metric Name	Value
1	**Accuracy**	0.812339331619537
2	**Precision**	0.824074074074074
3	**Sensitivity (Recall)**	0.835680751173709
4	**Specificity**	0.784090909090909
5	**F1 score**	0.82983682983683

Table 13 Naive Bayes Performance Metrics

4.11.5 KNN Classifier

KNN stands for k-nearest neighbors. As the name implies, the core of the algorithm logic is classifying the new tuple based on its neighborhood.

The algorithm stores the training dataset and when a new tuple came in, it starts to compare it based on other points. Let’s look at the chart below:

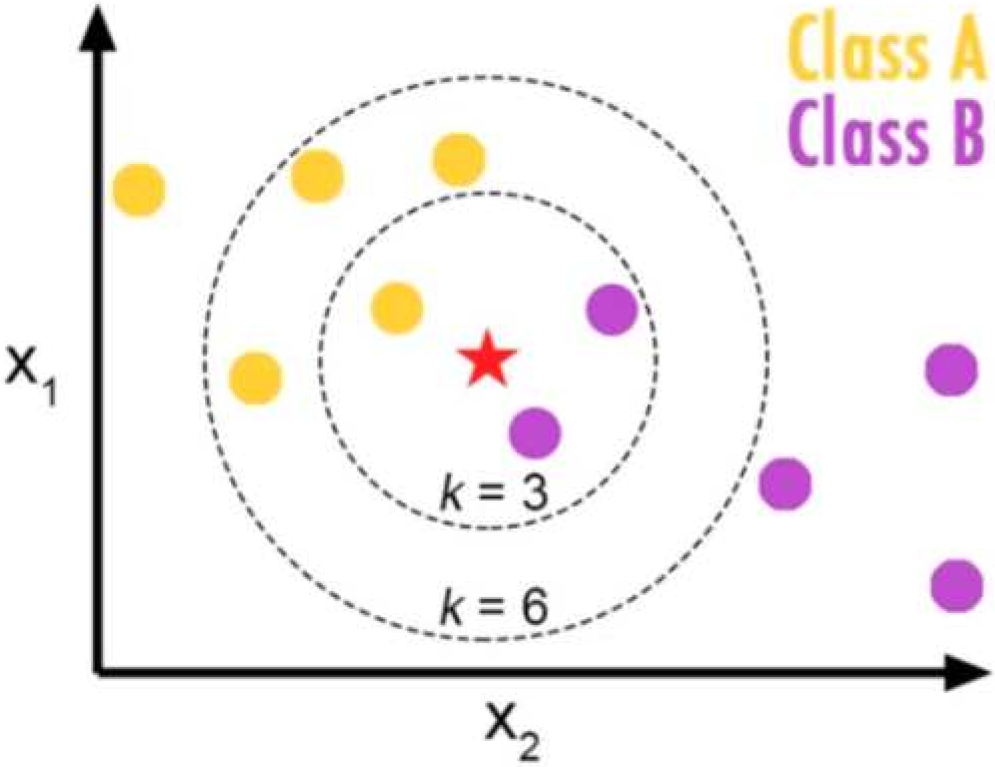
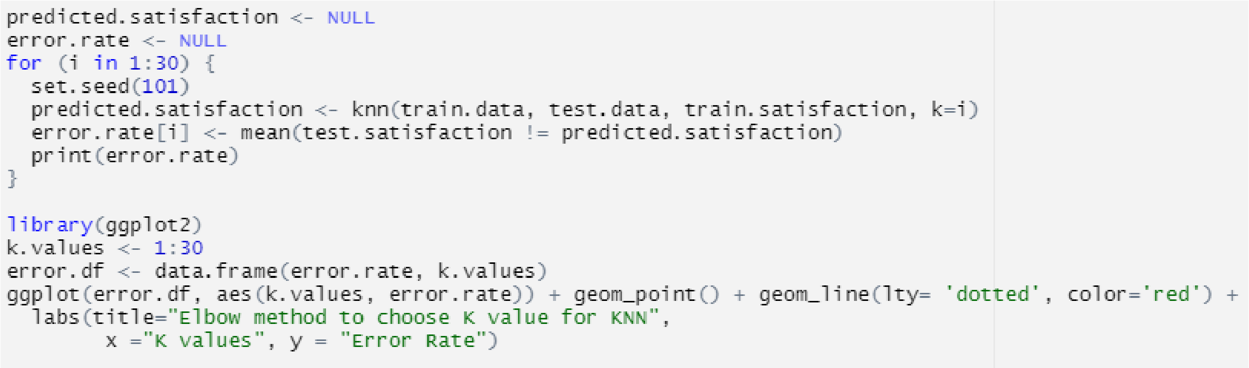


Figure 43 Explanation of K Value in KNN Algorithm

Let's say that the star is the new point that we need to study. If k = 3 then the algorithm would study the three points neighboring the new point. If the majority are of class B then the new point is classified as A. However, that may not be accurate. If we increased K and it became 6. Then it will be of class A. The K is a vital parameter of KNN method and usually we perform several tries and then perform what classed ELBOW plot to see which minimum value of K is efficient and no real need to get it higher. The advantage of the algorithm is that it requires no training (just stores the training data inside it), and it is easy to implement. It mainly needs to specify the k value. The disadvantages of it is that it doesn't work well with large datasets. It would need scaling (normalization & standardization). To prepare for the algorithm we first create a dataframe that holds all features (except class column) as numeric. So, we change the categorical variables represented as string into using encoding methodologies explained in encoding (normalization & standardization). Then we perform the scale function.

One of the most important points of KNN algorithm is to calculate K value. We calculate it by running the same algorithm several times and see which has latest error and error after that doesn't change remarkably.



```
predicted.satisfaction <- NULL
error.rate <- NULL
for (i in 1:30) {
  set.seed(101)
  predicted.satisfaction <- knn(train.data, test.data, train.satisfaction, k=i)
  error.rate[i] <- mean(test.satisfaction != predicted.satisfaction)
  print(error.rate)
}

library(ggplot2)
k.values <- 1:30
error.df <- data.frame(error.rate, k.values)
ggplot(error.df, aes(k.values, error.rate)) + geom_point() + geom_line(lty= 'dotted', color='red') +
  labs(title="Elbow method to choose K value for KNN",
         x ="K values", y = "Error Rate")
```

After creating the model we re-run the algorithm several times to choose which K value suffice to not get high error rate. We use Elbow method. After plotting the error rate for ascending values from 1 to 30 we get the following plot. We could notice that at specific point the error rate doesn't go below remarkably. At that point we choose the K value, and from the chart below we see desired K = 13 as below plot:

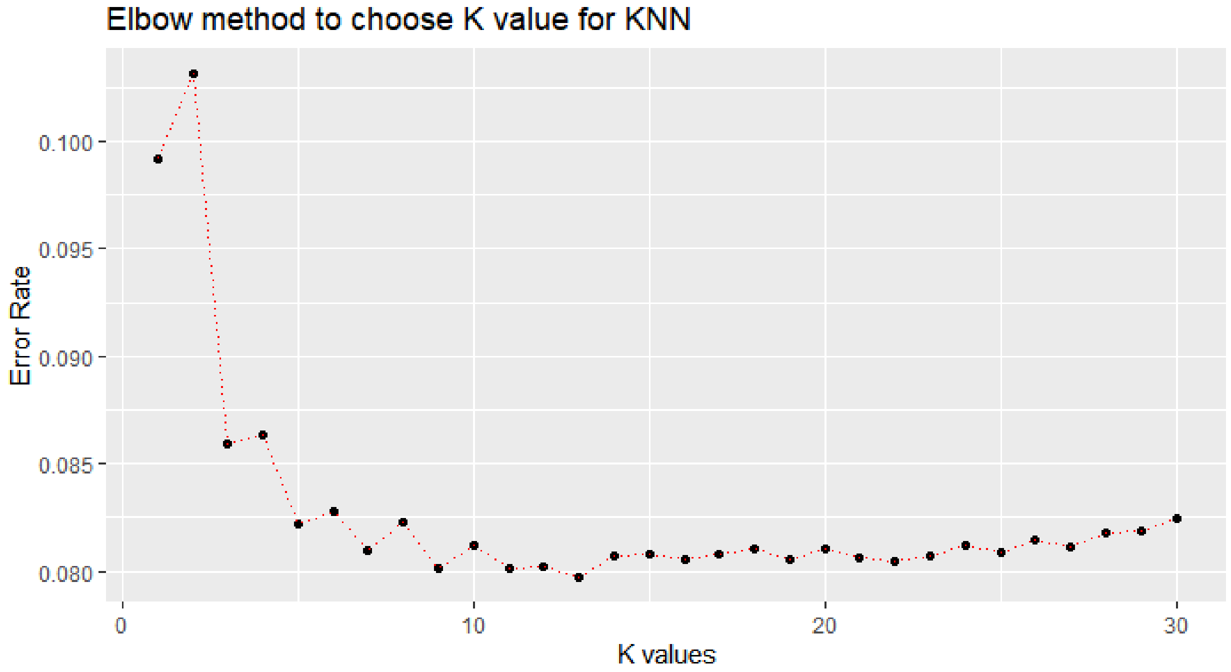
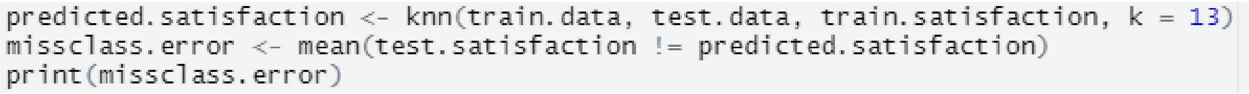


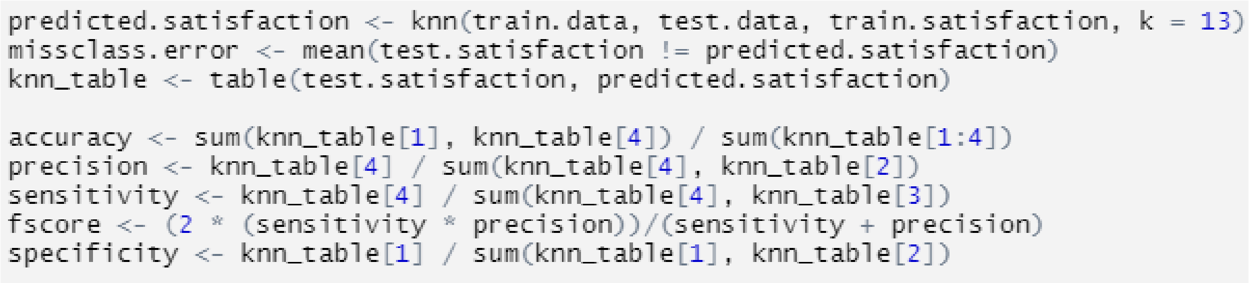
Figure 44 Error Rate of KNN vs K Value

Below is the script used to create the model of KNN for K =13:



```
predicted.satisfaction <- knn(train.data, test.data, train.satisfaction, k = 13)
missclass.error <- mean(test.satisfaction != predicted.satisfaction)
print(missclass.error)
```

Now we calculate the error and other metrics using confusion matrix:



```
predicted.satisfaction <- knn(train.data, test.data, train.satisfaction, k = 13)
missclass.error <- mean(test.satisfaction != predicted.satisfaction)
knn_table <- table(test.satisfaction, predicted.satisfaction)

accuracy <- sum(knn_table[1], knn_table[4]) / sum(knn_table[1:4])
precision <- knn_table[4] / sum(knn_table[4], knn_table[2])
sensitivity <- knn_table[4] / sum(knn_table[4], knn_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- knn_table[1] / sum(knn_table[1], knn_table[2])
```

#	Metric Name	Value
1	**Accuracy**	0.92025972692742
2	**Precision**	0.908796773891025
3	**Sensitivity (Recall)**	0.943435720196661
4	**Specificity**	0.894413984039954
5	**F1 score**	0.925792352337051

Table 14 KNN Performance Metrics

4.11.6 SVM - Support Vector Machines

SVM stands for Support Vector Machine. This algorithm is used when the dataset is separable. It tries to have the hyperplane that separate the dataset the best. It uses the vectors to maximize the margin between the hyperplane and datapoints.

The illustrator below explains the idea as it shows below:

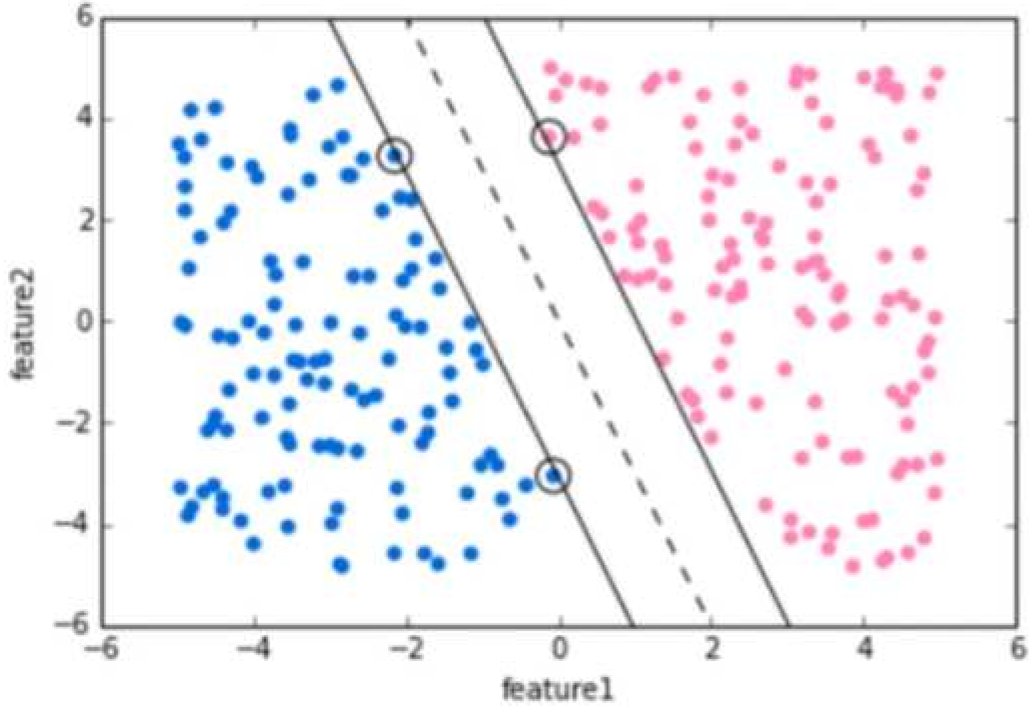


Figure 45 Basic Interpretation of Hyperplane

The solution of hyperplane is not limited to linearly separable dataset. It could be expanded to non-linearly separable datasets as shown below:

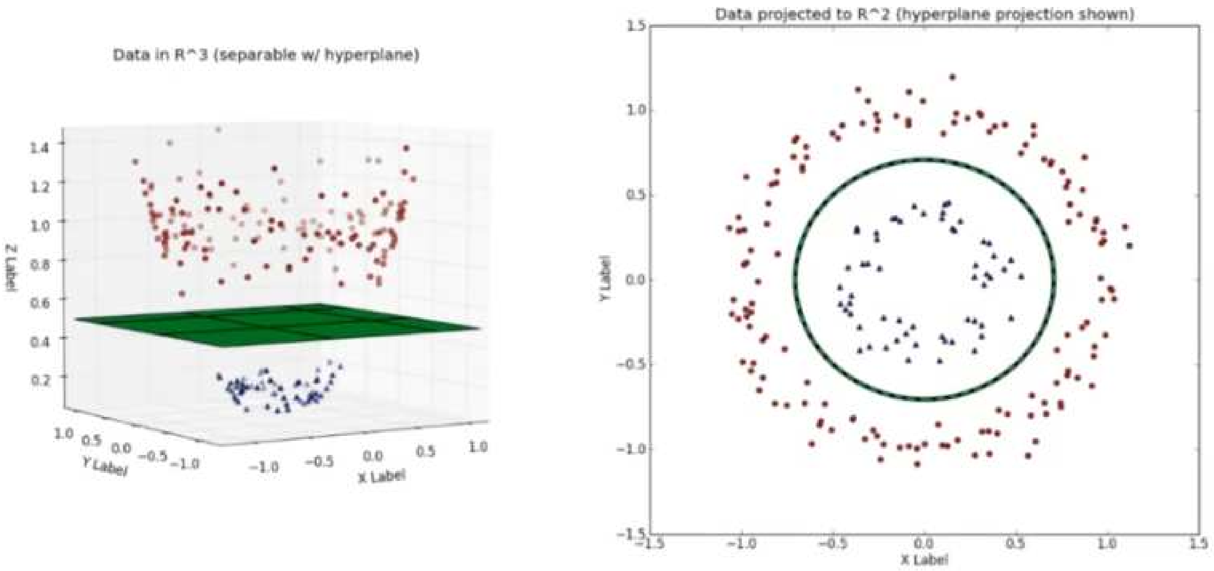
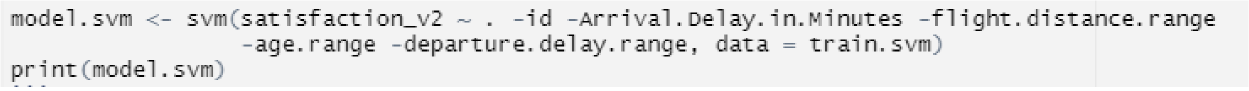


Figure 46 Basic Illustration of Nonlinear Hyperplane

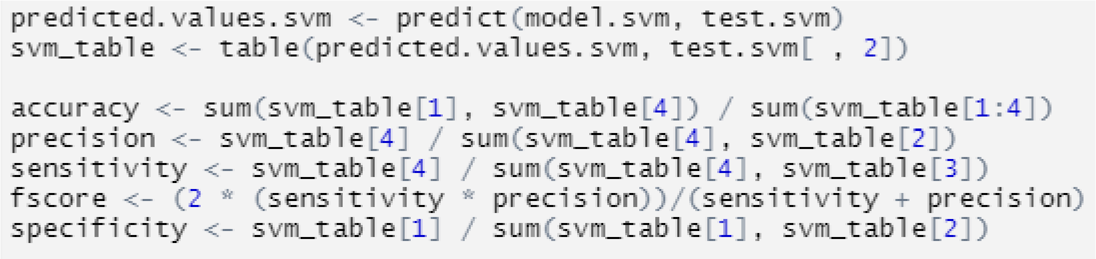
For a new tuple, the classification would be based on which side of the hyperplane the new tuple is located.

To build the model, we use the standardized dataset. We have the mean = 0 and variance = 1 for all features. Below is the script that create the model:



```
model.svm <- svm(satisfaction_v2 ~ . -id -Arrival.Delay.in.Minutes -flight.distance.range
                 -age.range -departure.delay.range, data = train.svm)
print(model.svm)
```

And once it is done, we would calculate the metrics of performance as follow:



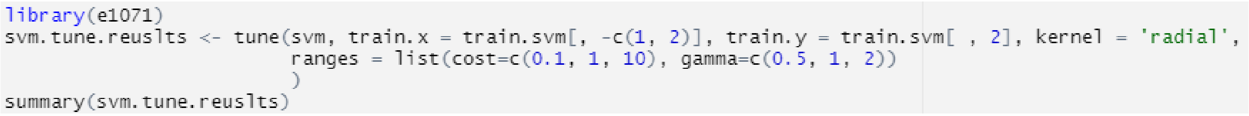
```
predicted.values.svm <- predict(model.svm, test.svm)
svm_table <- table(predicted.values.svm, test.svm[ , 2])

accuracy <- sum(svm_table[1], svm_table[4]) / sum(svm_table[1:4])
precision <- svm_table[4] / sum(svm_table[4], svm_table[2])
sensitivity <- svm_table[4] / sum(svm_table[4], svm_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- svm_table[1] / sum(svm_table[1], svm_table[2])
```

The results are as follows:

#	Metric Name	Value
1	**Accuracy**	0.884318766066838
2	**Precision**	0.903846153846154
3	**Sensitivity (Recall)**	0.882629107981221
4	**Specificity**	0.886363636363636
5	**F1 score**	0.893111638954869

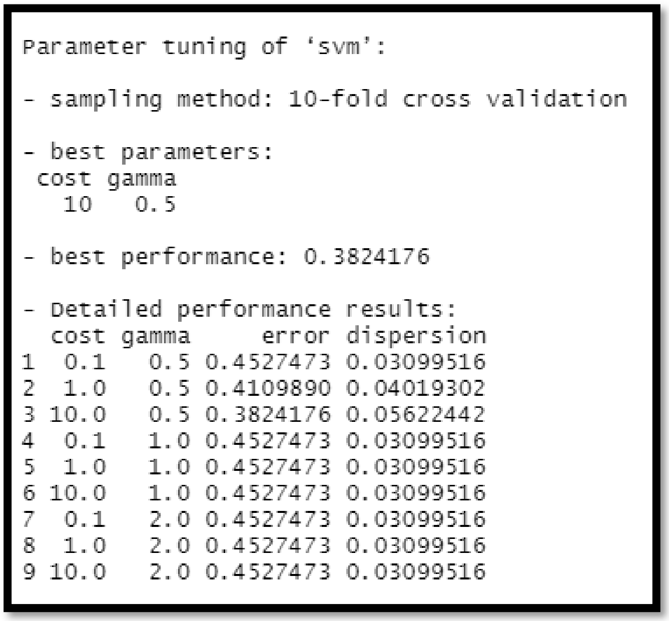
Table 15 SVM Performance Metrics

To tune the results, we may adjust the values of “gamma”, and “cost”. To find the best values of these two parameters, we may pass several values of “gamma”, and “cost” and let the computer choose the best:



```
library(e1071)
svm.tune.reuslts <- tune(svm, train.x = train.svm[, -c(1, 2)], train.y = train.svm[ , 2], kernel = 'radial',
                         ranges = list(cost=c(0.1, 1, 10), gamma=c(0.5, 1, 2))
                         )
summary(svm.tune.reuslts)
```

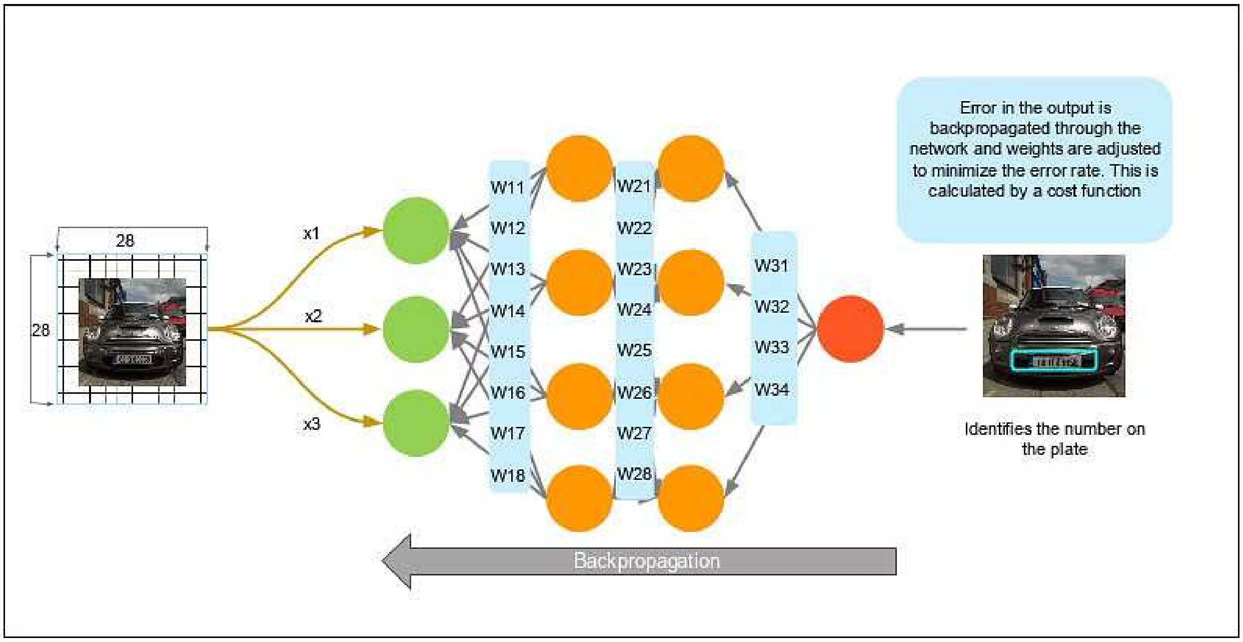
The results may look like the following:



```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10   0.5

- best performance: 0.3824176

- Detailed performance results:
  cost gamma     error dispersion
1  0.1   0.5 0.4527473 0.03099516
2  1.0   0.5 0.4109890 0.04019302
3 10.0   0.5 0.3824176 0.05622442
4  0.1   1.0 0.4527473 0.03099516
5  1.0   1.0 0.4527473 0.03099516
6 10.0   1.0 0.4527473 0.03099516
7  0.1   2.0 0.4527473 0.03099516
8  1.0   2.0 0.4527473 0.03099516
9 10.0   2.0 0.4527473 0.03099516
```

4.11.8 ANN – Artificial Neural Networks

The models introduced so far are used in regular machine learning. There is another category of machine learning called deep learning. Neural Network algorithms has different approach that tend to process data in a way that simulate how the human brains work. Neural Networks algorithms tend to resolve problems of unstructured data, however they are not limited to. Neural Networks (NN) use layers to process data. There would be an input layer, and out layer. Between them, there would be N (1 to n) number of hidden layers. Each hidden layers have number of perceptron that receive inputs, weight inputs, sum inputs, and generate outputs. There is two ways of handling: either Feed-Forward, or back-propagated. The figure below explains the NN and back-propagated quickly and easily

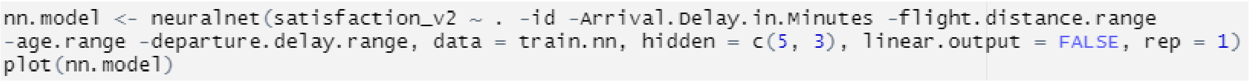


Error in the output is backpropagated through the network and weights are adjusted to minimize the error rate. This is calculated by a cost function

Identifies the number on the plate

Figure 47 Quick illustration of Hidden Layers of NN

The perceptron inside the hidden layers uses activation function. The values are [0, 1] hence the input values of the model have to be normalized to be [0, 1] range. The code below explains how to call the ANN model. The function is called neuralnet, and the dependent variable is satisfaction_v2. Then we insert the training dataset, then we define a vector that contains the number of hidden layers for each layer of the hidden ones.



```
nn.model <- neuralnet(satisfaction_v2 ~ . -id -Arrival.Delay.in.Minutes -flight.distance.range
-age.range -departure.delay.range, data = train.nn, hidden = c(5, 3), linear.output = FALSE, rep = 1)
plot(nn.model)
```

The figure below explains how the neural net constructed with hidden layers:

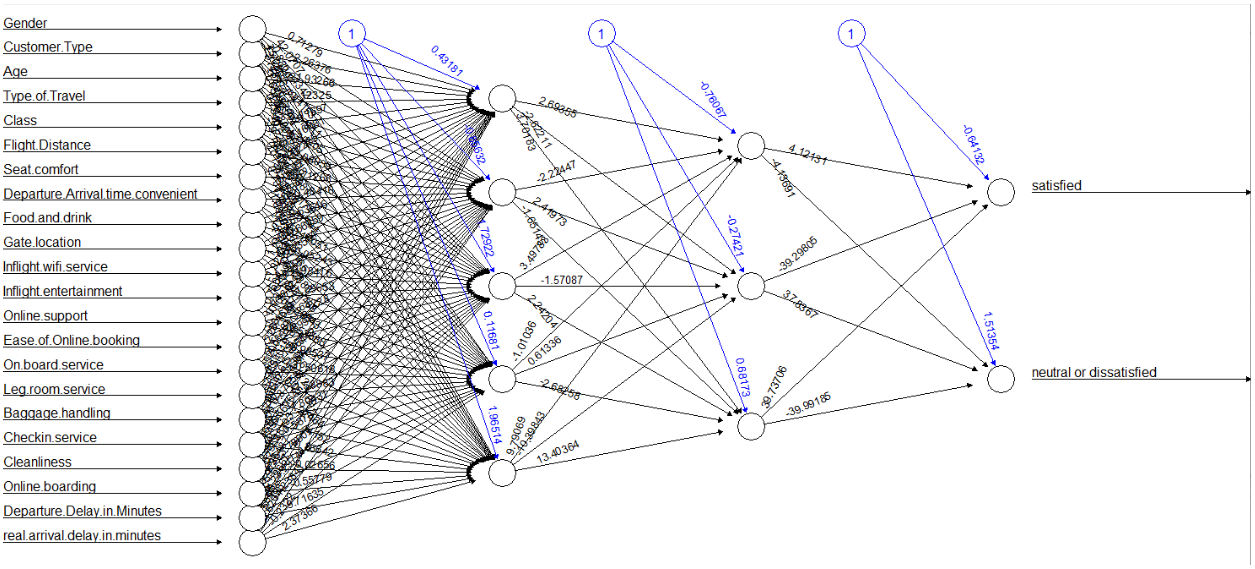


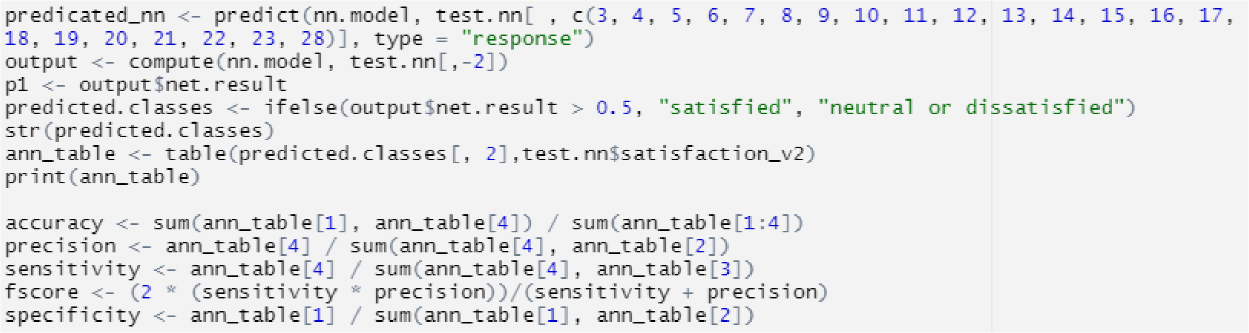
Figure 48 Presentation of ANN Model

Below is a short explanation about the parameters passed to neuralnet function:

#	Parameter Name	Explanation
1	**rep**	how many times you train your neural network
2	**stepmax**	give your model more chances to learn/converge
3	**threshold**	to allow an earlier stop for convergence.
4	**hidden**	Vector to specify the number of layers of each hidden layers.

Table 16 ANN formula parameters

Now let’s evaluate the performance of model:



```
predicated_nn <- predict(nn.model, test.nn[ , c(3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 28)], type = "response")
output <- compute(nn.model, test.nn[,-2])
p1 <- output$net.result
predicted.classes <- ifelse(output$net.result > 0.5, "satisfied", "neutral or dissatisfied")
str(predicted.classes)
ann_table <- table(predicted.classes[, 2],test.nn$satisfaction_v2)
print(ann_table)

accuracy <- sum(ann_table[1], ann_table[4]) / sum(ann_table[1:4])
precision <- ann_table[4] / sum(ann_table[4], ann_table[2])
sensitivity <- ann_table[4] / sum(ann_table[4], ann_table[3])
fscore <- (2 * (sensitivity * precision))/(sensitivity + precision)
specificity <- ann_table[1] / sum(ann_table[1], ann_table[2])
```

Below is the modeling code and the performance evaluation.

#	Metric Name	Value
1	**Accuracy**	0.858974358974359
2	**Precision**	0.866359447004608
3	**Sensitivity (Recall)**	0.878504672897196
4	**Specificity**	0.835227272727273
5	**F1 score**	0.872389791183295

Table 17 ANN Performance Metrics

4.11.9 Comparing Results of Multiple Classifiers vis Radar Chart

Below is a polygon that gives us an easy way to compare the performance metrics of all machine learning models that we used to resolve our classification problem. Although the numbers of the metrics such (Accuracy, Precision, Sensitivity…etc) are represented in each section, however, comparing the numbers would be easier to digest by looking into such visualizations. In order to do so, we have represented all numbers in the polygon below. The legend represents the color of the line verses the algorithm shortcut. The shortcuts represented as below table:

Shortcut	Algorithm Name
DT	Decision Trees
RF	Random Forest
LR	Linear Regression
NB	Naïve Bayes
KNN	K-Nearest Neighbor
SVM	Support Vector Machines
ANN	Artificial Neural Networks

Table 18 Algorithms Shortcuts

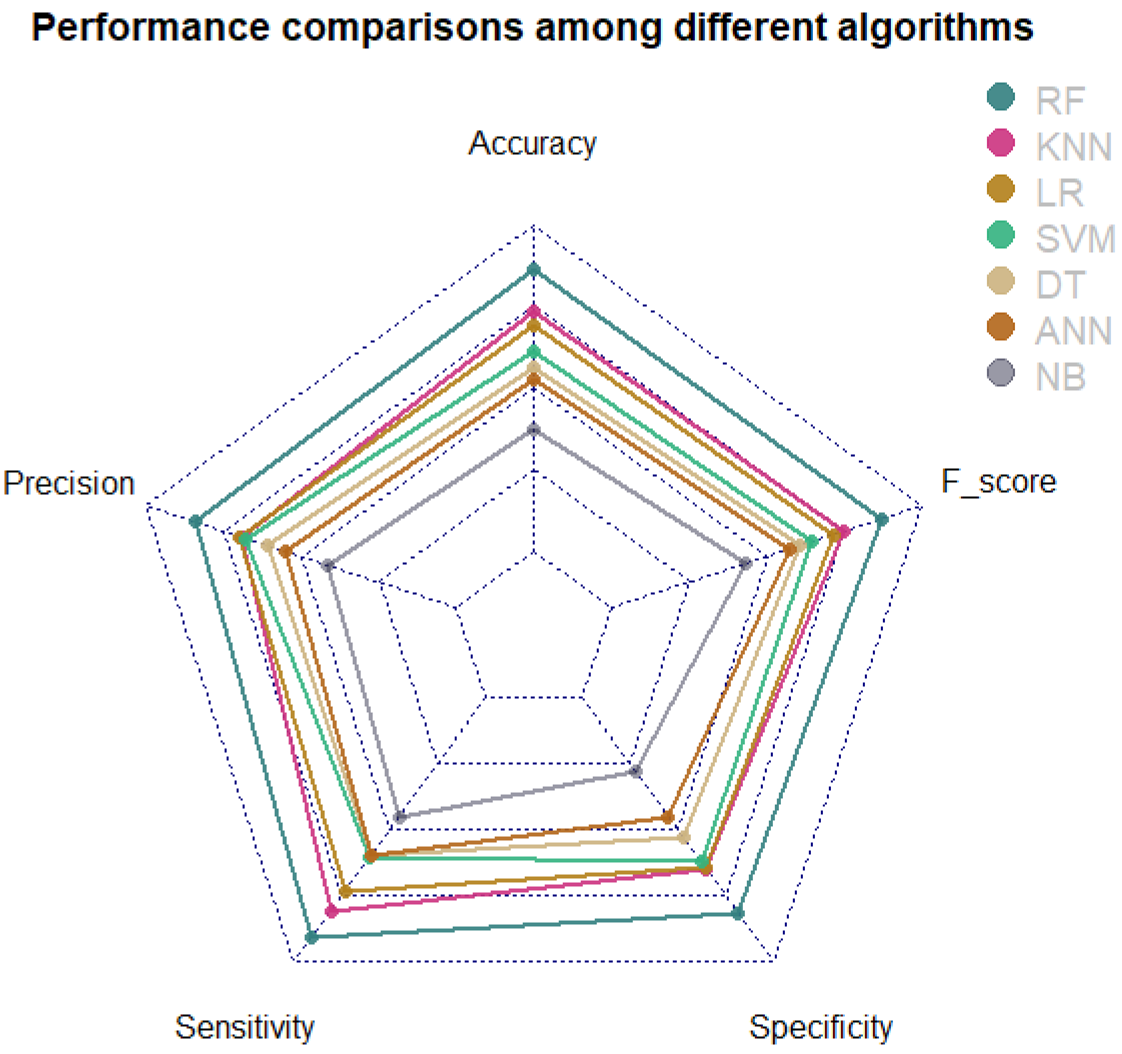


Figure 49 Metrics Comparison of Different Classifiers

4.11.10 Result of Predictors (Which predictor is more important)

When we tried to build all models above, we tried to include all features with exception of some obviously undesired features that would not help the model such as id or the target class. In addition, we omitted some features that are highly correlated with another like Arrival Delay and Departure Delay. But how we could really know which feature is more important than others in the model. To do so, there are some techniques that may help doing that. We already highlighted some of the techniques in (Feature Selection) section. We will consider Forward Feature Selection approach in selecting the remarkable features. Below I am going to summarize how this technique is performed and then we will apply to find out which features were the most remarkable to some of our models. The ultimate target is to show how feature selection could improve the performance of the model.

To perform Feature Selection techniques, we do several iterations. The first iteration in Forward Feature Selection is to create a model for each feature individually. Then we measure the performance of that model using some metric. There are some metrics to do so, and in our case, we will use the Accuracy metric to evaluate the performance of the model. Once we build all single-feature-models and evaluate all of them, we elect the model with the highest metric score. The feature used for that winning model is considered the first feature.

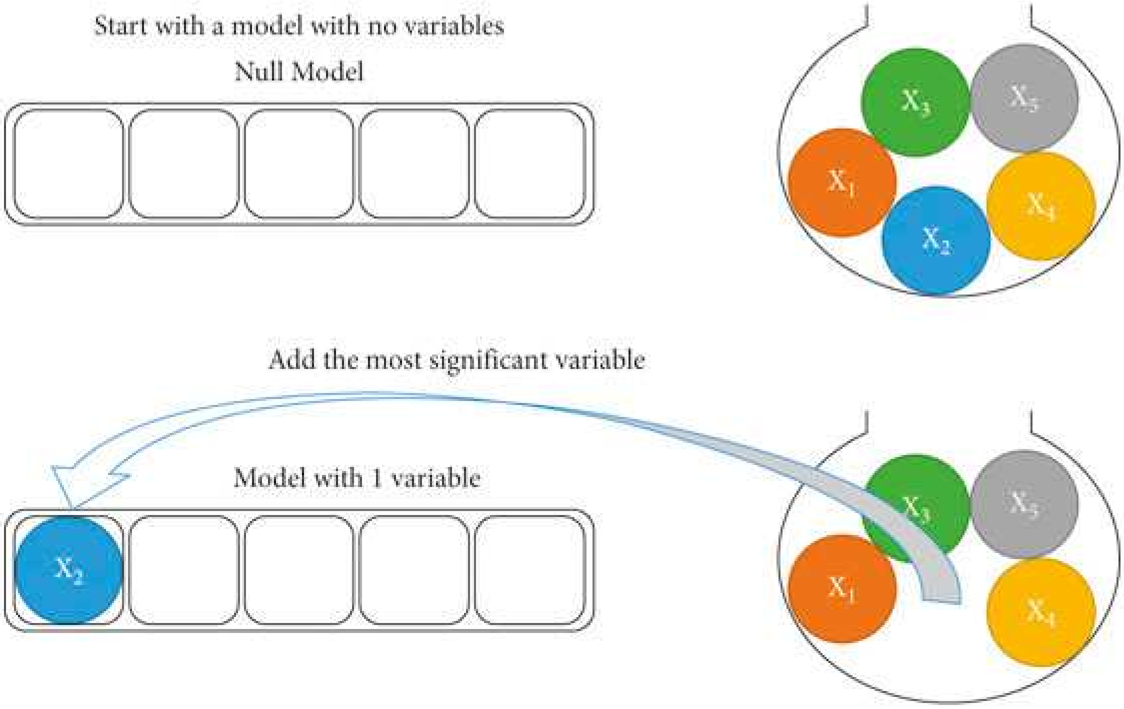


Figure 50 Illustration of Forward Feature Selection - First Selection

Now the second iteration, we will create a model for each feature plus the wining feature in iteration number one. As a result, the result of models in the second iteration is containing two features, and the number of the models is less by 1 of the number of models generated in the first iteration because we skip the wining feature in iteration 1.

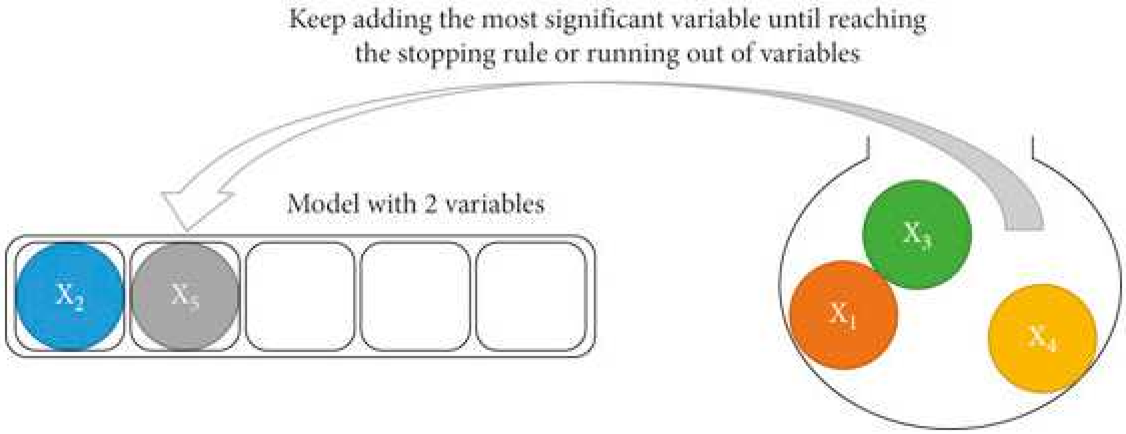

Keep adding the most significant variable until reaching
the stopping rule or running out of variables
Model with 2 variables
X_2
X_5
X_3
X_1
X_4

Figure 51 Illustration of Forward Feature Selection - Further Selections

We keep continuing until we run out of features, or there is no remarkable improvement in the metric. At that point we have a model with all features that are considered as selected for best model. We can conclude that all features participate in improving the model, or we find that there is no improvement in the performance of the model by adding extra features. At that point we know that only added features are the ones that play major role in that model, so our model would not be cumbersome with too much unneeded features.

The results of our Forward Feature Selection are written in MS-Excel file. This file has several sheets. Each sheet is dedicated for a specific model algorithm (Decision Trees, Naïve Bayes…etc). The first column is presenting the features names. The later columns represent the metric score. We will denote the name of columns as (F1, F2, F3…etc). Consequently, the second column represent the score of single-feature-models (called F1). The third column represent the score of two-feature-models (called F2). The fourth column represent the score of three-feature-models (called F3). When we find that resulted accuracy is exactly the same as best feature of last iteration, we add the word same. NA is added when the feature is already elected before as winning feature. We count up to 10 features. Let’s see the results for Decision Trees model:

Predictor Name	F1	F2	F3	F4	F5
Gender	60.61	**Same**	**Same**	**Same**	**Same**
Customer.Type	64.48	**Same**	**Same**	**Same**	**Same**
Age	59.56	**Same**	**Same**	**Same**	**Same**
Type.of.Travel	56.68	**Same**	**Same**	**Same**	**Same**
Class	65.12	**Same**	**Same**	**Same**	**Same**
Flight.Distance	58.77	**Same**	**Same**	**Same**	**Same**
Seat.comfort	69.31	84.02	**NA**	**NA**	**NA**
Departure.Arrival.time.convenient	54.73	**Same**	**Same**	**Same**	**Same**
Food.and.drink	60.39	**Same**	**Same**	**Same**	**Same**
Gate.location	57.01	**Same**	**Same**	**Same**	**Same**
Inflight.wifi.service	59.76	**Same**	**Same**	**Same**	**Same**
Inflight.entertainment	81.01	**NA**	**NA**	**NA**	**NA**
Online.support	71.79	82.13	86.06	86.97	**NA**
Ease.of.Online.booking	72.48	**Same**	86.42	**NA**	**NA**
On.board.service	67.36	**Same**	**Same**	**Same**	**Same**
Leg.room.service	66.88	**Same**	**Same**	**Same**	**Same**
Baggage.handling	64.82	**Same**	**Same**	**Same**	**Same**
Checkin.service	63.01	**Same**	**Same**	**Same**	**Same**
Cleanliness	64.48	**Same**	**Same**	**Same**	**Same**
Online.boarding	66.47	82.04	85.91	**Same**	**Same**
Departure.Delay.in.Minutes	55.55	**Same**	**Same**	**Same**	**Same**
real.arrival.delay.in.minutes	55.51	**Same**	**Same**	**Same**	**Same**

Table 19 Decision Tree Feature Selection

We see that the best accuracy we got is: 86.97 and this is actually exactly the same result we got when we discussed Decision Trees accuracy using confusion matrix. This is because Decision Trees algorithm implicitly performs feature selection, because it is Tree-based algorithm. What’s more, we find out that the selected features using Forward Feature Selection are exactly the features selected in Decision Tree.

Now let’s do the same technique using Logistic regression. We repeated the selection of features for 10 iterations. As the results shows below, there is no point that we reached that we found no improvement in performance. In fact, we noticed a remarkable slowness in performance improvement. However, the first features we elected are the most remarkable features in the model. We would find some of the attributes are meaningless to be included in the model or would be lowering the performance if the model if not removed. This is what we could detect in the Naïve Bayes model in a smaller number of iterations.

Predictor Name	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Gender	60.61	Same	Same	86.54	86.63	86.82	88.38	89.10	89.62	89.83
Customer.Type	64.48	80.87	84.78	86.71	87.23	88.00	NA	NA	NA	NA
Age	57.89	Same	84.23	86.48	86.92	87.14	88.02	88.71	89.38	89.82
Type.of.Travel	56.68	Same	84.10	86.99	87.28	NA	NA	NA	NA	NA
Class	65.12	Same	83.95	86.61	87.01	87.01	88.36	88.77	89.59	89.83
Flight.Distance	54.18	Same	Same	86.63	87.07	87.19	88.08	88.67	89.36	89.77
Seat.comfort	69.31	84.25	NA	NA	NA	NA	NA	NA	NA	NA
Departure.Arrival.time.convenient	54.73	Same	85.43	86.67	86.99	87.12	88.68	NA	NA	NA
Food.and.drink	60.89	81.35	85.16	86.75	87.10	87.35	88.22	88.74	89.42	89.87
Gate.location	57.01	Same	84.62	86.52	86.93	87.15	88.35	88.73	89.47	89.83
Inflight.wifi.service	59.78	80.88	84.88	86.32	86.61	87.05	87.96	88.83	89.44	89.76
Inflight.entertainment	81.01	NA	NA	NA	NA	NA	NA	NA	NA	NA
Online.support	71.79	Same	85.30	86.96	86.98	87.34	88.20	88.96	89.79	NA
Ease.of.Online.booking	72.48	82.03	86.69	NA	NA	NA	NA	NA	NA	NA
On.board.service	67.36	81.14	85.93	86.38	86.58	87.22	88.57	89.21	89.54	89.84
Leg.room.service	66.88	81.18	85.74	86.71	86.96	87.42	88.54	89.26	89.74	89.97
Baggage.handling	64.82	80.96	85.79	86.25	86.64	87.34	88.38	89.35	NA	NA
Checkin.service	63.01	Same	84.96	87.09	NA	NA	NA	NA	NA	NA
Cleanliness	64.48	80.91	85.72	86.30	86.43	87.36	88.41	89.31	89.54	89.96
Online.boarding	66.47	Same	85.48	86.16	86.62	86.88	88.00	88.62	89.37	89.74
Departure.Delay.in.Minutes	55.62	80.97	84.28	86.67	87.07	87.17	88.10	88.72	89.50	89.83
real.arrival.delay.in.minutes	54.87	Same	Same	86.65	87.05	87.32	87.98	88.70	89.39	89.79

Table 20 Logistic Regression Feature Selection

The table below describes the iterations for the Naïve Bayes model. As we remember that when we included all features that we intuitively selected, we got a score of accuracy equals to 81.23. However, using Forward Feature Selection we got a score of 86.37. We repeated the process of electing features for five times. We could find five features that when we add to Naive Bayes algorithm, we got a higher score of accuracy equals to 86.37. when we tried to add any extra feature, we found out that the accuracy is lowered. Which means that the set of selected features till iteration number 5 are the only desired features. Any other features would distract the model performance.

Predictor Name	F1	F2	F3	F4	F5	F6
Gender	54.24	Same	82.00	83.03	85.34	85.60
Customer.Type	65.80	80.97	84.06	86.11	NA	NA
Age	54.49	Same	84.06	Same	86.37	NA

Type.of.Travel	54.49	**Same**	83.03	**Same**	**Same**	85.34
Class	64.01	79.69	84.31	84.31	85.60	85.86
Flight.Distance	53.72	**Same**	83.03	**Same**	**Same**	86.11
Seat.comfort	70.69	83.29	NA	NA	NA	NA
Departure.Arrival.time.convenient	52.95	**Same**	84.83	84.31	86.37	86.11
Food.and.drink	55.52	**Same**	77.12	82.51	85.60	85.34
Gate.location	56.04	**Same**	81.74	83.29	**Same**	85.60
Inflight.wifi.service	57.32	80.46	81.49	84.31	85.60	85.60
Inflight.entertainment	80.20	NA	NA	NA	NA	NA
Online.support	70.17	80.97	84.31	83.54	85.34	85.60
Ease.of.Online.booking	71.72	79.43	84.57	82.26	84.57	84.57
On.board.service	67.86	79.94	84.31	82.77	84.57	85.60
Leg.room.service	69.40	**Same**	86.11	84.06	84.83	84.83
Baggage.handling	66.32	78.92	84.83	NA	NA	NA
Checkin.service	65.03	79.69	83.54	84.57	84.06	84.06
Cleanliness	65.03	78.92	84.06	84.06	85.86	85.34
Online.boarding	63.23	78.92	83.54	83.80	83.80	83.80
Departure.Delay.in.Minutes	54.49	80.46	82.77	84.06	85.60	85.60
real.arrival.delay.in.minutes	53.47	78.40	82.77	82.77	83.80	83.80

Table 21 Naive Bayes Feature Selection

If we examined the importance of the features using Random Forest model we created before using importance(rf.model) we would find the following results:

#	Predictor Name	MeanDecreaseGini
1	**Inflight.entertainment**	9202.05570
2	**Seat.comfort**	6154.89200
3	**Ease.of.Online.booking**	3525.42550
4	**Online.support**	2862.62280
5	**Customer.Type**	1909.64830
6	**Leg.room.service**	1895.26140
7	**On.board.service**	1717.15970
8	**Food.and.drink**	1655.04340
9	**Class**	1545.02760
10	**Online.boarding**	1442.06880
11	**Flight.Distance**	1372.83880
12	**Departure.Arrival.time.convenient**	1357.09550
13	**Cleanliness**	1256.50680
14	**Baggage.handling**	1248.28240
15	**Age**	1219.68490
16	**Checkin.service**	1133.66670
17	**Gate.location**	1124.49040
18	**Gender**	1105.07200
19	**Type.of.Travel**	1050.74250
20	**Inflight.wifi.service**	0776.24030

21	**Departure.Delay.in.Minutes**	0633.92000
22	**real.arrival.delay.in.minutes**	0586.73920

Table 22 Random Forest Feature Importance

For RF, the top two features are the same as other models (Inflight.entertainment, Seat.comfort)

As we could see, there is a no unique set of features that are identical across the different models. However, the first two features are repeated in the three modes we studied. The two features are (Inflight.entertainment, Seat.comfort), the other features may re-appear but not necessarily in the same order. The process could be repeated for any other model using other algorithm.

Chapter 5 - Conclusion

5.1 Summary

As we could see, we had the initial dataset as feedback from the customers about in-flight and ground services. So, we explored the dataset, removed NAs, created new features, omitted useless ones, and performed scaling (normalization & standardization) where it needs. Once we are done with data pre-processing and EDA we performed prediction using Decision Trees, Random Forest, Logistic Regression, Naïve Bayes, KNN, SVM, and finally applied deep learning algorithm based on neuralnet called ANN. We performed confusion matrix one all and calculated the classification metrics such accuracy, specificity, and others. Then finally compared the results using Radar chart for easy comparison. After we finished all of that work, we compared the performance of the features using forward-feature selection.

5.2 Conclusion

To conclude, we see that we collected the data from open-source website (Kaggle), then performed EDA and demonstrated the relationships between the features that have and used several techniques to clean the data. We created extra features that may help in our machine learning study. The main target of performing this project was to be able to establish a good predictor whether the airline customer is satisfied or not. After performing the EDA and feature engineering, we did features scaling for normalization and standardization. Once we got done of all that work, we demonstrated data quality dimensions, and showed how our dataset is ready for creating the models and using them in prediction. We created seven models using the optimized dataset we collected. For some models we applied a sample of the dataset to avoid the limitation of computational power. We compared between the various algorithms of our problem that we had used and explained the pros and cons of each. This discussion was really helpful in this study, because there is no best algorithm all the time. It varies from case to another. As a result, we found that Random Forest algorithm had the best performance using the performance metric. In addition, we tried to find out the most important predictor among other predictors.

5.3 Recommendations

We would recommend that the airlines adopt the usage of such systems. A passenger might be unsatisfied for a minor issue that could be avoided at no additional cost, and it would be easy to customize the service to best suite that customers and make him come repeatedly to the same airline.

5.4 Future Works

The study shows that a customer satisfaction may vary from person to another. This could be affected by different factors and those factors may be different from person to another. Further studies may have additional factors that describe the personality of the traveler such as nationality, income, frequency of travel, etc. In addition, the metrics of trip may be more detailed such as external weather (having Aerobic pitfalls) or the trip was during the night or day. Additional point that we may elaborate in in future studies is to reduce the number of attributes by using PCA (Principal Component Analysis), and Lasso. In addition, we may use Linear Discriminant Analysis (LDA). In addition, we may also include BigData pipeline to allow automatic analysis from the passengers and react to that feedback immediately once the system realizes that the customer might be unsatisfied.

References

1. Hwang, S. Kim, J. Park, E. Kwon, S., 2020. Who will be your next customer: A machine learning approach to customer return visits in airline services. *Journal of Business Research*, 121, pp. 121 – 126

2. Ulkhaq, M., Adyatama, A., Fidiyanti, F., Rozaq, R., Raharjo, F., 2020. An Artificial Neural Network Approach for Predicting Customer Loyalty: A Case Study in an Online Travel Agency. *International Journal of Machine Learning and Computing,* 10, pp. 283-289

3. Overtveld, C. & Balsingh, V., 2019. Flight Delay Error Analysis for Amsterdam Airport Schiphol Using a Deep Neural Network (DNN).

4. Kaur, G. & Malik, K. 2021. A Sentiment Analysis of Airline System using Machine Learning Algorithms. *International Journal of Advanced Research in Engineering*, 12, pp. 731-742.

5. Kumar, S., Zymbler, M., 2019. A machine learning approach to analyze customer satisfaction from airline tweets. *Journal Big Data* 6, 62

6. Park, S.-H., Kim, M.-Y., Kim, Y.-J., & Park, Y.-H., 2022. A Deep Learning Approach to Analyze Airline Customer Propensities: The Case of South Korea. *Applied Sciences*, *12*(4), 1916.

7. García, V. & Florencia, R. & Sánchez J. & Zarate, G. & Contreras-Masse, R., (2019). Predicting Airline Customer Satisfaction using k-NN Ensemble Regression Models. *Research in Computing Science*. P. 148

8. Filipe R. Lucini, Leandro M. Tonetto, Flavio S. Fogliatto, Michel J. Anzanello, 2020, Text mining approach to explore dimensions of airline customer satisfaction using online customer reviews. Journal of Air Transport Management, 83.

9. Lhéritier, A., Bocamazo, M., Delahaye, T., Acuna-Agost, R, 2019. Airline itinerary choice modeling using machine learning, *Journal of Choice Modelling*, 31, pp. 198 – 209.

10. Ustebay, S., Yelmen, I., Zontul, M. (2019). Airline customer purchase segmentation using machine learning techniques.

11. Baswardono, W., Kurniadi, D., Mulyani, A. and Arifin, D. (2019). Comparative analysis of decision tree algorithms: Random Forest and C4.5 for airlines customer satisfaction classification, *Journal of Physics: Conference Series*, 1402

12. Wong, A., Marikannan, B. (2020) Optimising e-commerce customer satisfaction with machine learning, *Journal of Physics Conference Series*.

13. Bouzakraoui, M., Sadiq, A., Alaoui, A. (2020) Customer Satisfaction Recognition Based on Facial Expression and MachineLearning Techniques, *Advances in Science, Technology and Engineering Systems Journal* Vol. 5, No. 4, 594-599

14. PANDEY, H., Tiwari, P., Khamparia, A, Kumar, S. (2019) Twitter-based Opinion Mining for Flight Service Utilizing Machine Learning. *Journal: Informatica (Slovenia)*, P. 381-386.

15. Park, S., Kim, M., Kim, Y., and Park, Y. (2022) A Deep Learning Approach to Analyze Airline Customer Propensities: The Case of South Korea. *Applied Sciences* 12(4):1916

16. Sabapathi, P., Kaliyamurthie, K.P. (2022) Review: Analysis of Customer Review and Predicting Future Release of the Product using machine learning concepts. *International Conference on Communication, Computing and Internet of Things (IC3IoT)*.

17. Chang, Y., Ku, C., Nguyen, D. (2022) Predicting aspect-based sentiment using deep learning and information visualization: The impact of COVID-19 on the airline industry. *Information & Management* Volume 59, Issue 2

18. Samunderu, E., Farrugia, M. (2022) Predicting customer purpose of travel in a low-cost travel environment—A Machine Learning Approach. *Machine Learning with Applications* Volume 9

19. Gao, K., Yang, Y., Qu, X. (2021) Examining nonlinear and interaction effects of multiple determinants on airline travel satisfaction. *Transportation Research Part D: Transport and Environment*. Volume 97

20. Aljedaani, W., Rustam, F., Mkaouer, M., Ghallab, A., Rupapara, V., Washington, P., Lee, E., Ashraf, I. (2022) Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. *Knowledge-Based Systems*. Volume 255

21. Thiagarajan, B., Srinivasan, L., Sharma, A., Sreekanthan, D., Vijayaraghavan, V. (2017) A machine learning approach for prediction of on-time performance of flights. *36th Digital Avionics Systems Conference (DASC)*

22. Farzadnia, S., Vanani, I. (2022) Identification of opinion trends using sentiment analysis of airlines passengers' reviews. *Journal of Air Transport Management*. Volume 103

23. Baydoğan, C. & Alatas, B. (2019) Detection of Customer Satisfaction on Unbalanced and Multi-Class Data Using Machine Learning Algorithms. *1st International Informatics and Software Engineering Conference (UBMYK).*

24. Tusar, T., Islam, T (2021) A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data.

25. Demircana, M., Seller, S., Abut, F, Akay, M. (2021) Developing Turkish sentiment analysis models using machine learning and e-commerce data. *International Journal of Cognitive Computing in Engineering* Volume 2, P 202-207

26. CRISP-*DM*. (2022, February 8). Data Science Process Alliance. https://www.datascience-pm.com/crisp-dm-2/