

**Comparison of Machine Learning Algorithms with Monte Carlo
Stimulation on Employee Attrition Dataset and suggesting an
optimal algorithm for Employee Attrition tool**

Dissertation submitted in part fulfilment of the requirements for the
degree of

Masters in Data Analytics

At Dublin Business School

Keerthy Krishnan

Acknowledgment

I would like to thank everyone who has been a reason that I am doing my Masters Degree and Dissertation. I extend my gratitude to Dr Shahram Azizi for accepting to be my supervisor, in helping me out throughout the dissertation and also guiding me in the right path. I am grateful to have had the opportunity to work with Dr Shahram and to have learnt a lot during these few months.

I extend my thanks to all other staff members in Dublin Business School including John O'Sullivan, Terri Hoare and John Honan for having played a very important role in teaching and making us understand various topics and programming languages that I was able to make use in my dissertation work. I must also thank Dr Andrew Browne from whose lectures I had a better picture about report writing and how to do research.

It is important to thank all of them who have been constant support and a reason that I got an opportunity to do my Masters. My amazing family members have all been very supportive right from the beginning when I decided to do my Masters, right till the end of submitting my report. I cannot thank them enough for the kind of moral support and encouragement. The confidence they had on me that was a driving force throughout the dissertation because of which I have reached so far.

I would like to thank my classmates from whom I was able to learn a lot of things during group projects and presentation. I also should thank Leo and others in IT support who were diligent in getting our logins created and also for the kind of support. I thank everyone in the reception who with diligence were able to help us with the questions and requests that we had. I almost felt that I was part of the institution on the very first day in DBS because of them. I would like to thank the Student Union for all the activities that they had conducted for students.

Declaration

I, Keerthy Krishnan, declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School's academic honesty policy.

Signed: Keerthy Krishnan

Date: 16/12/2018

Abstract

The aim of the dissertation is to compare different models on the employee attrition dataset (IBM website, Sample Data: HR Employee Attrition and Performance) and make suggestions based on the result by comparing the accuracy of the models. This will be carried out in two different steps. First one will be in R programming language and the second one will be done in Rapid Miner. Tableau has been used to visualize the dataset and get an insight into the attributes.

Four models Naïve Bayes, Decision Tree, Random Forest and Support Vector Machines are run together in a Monte Carlo simulation and their probabilities are calculated and plotted. With the help of the Rattle feature in R, the performance and the accuracy of models like Neural Networks and Logistics regression will be calculated and comparisons will be made. Then the Auto model feature is used in Rapid miner to make another comparison of different models and a general study is done with the dataset. At the end of dissertation, a recommendation will be made based on the comparison with which an application for Human Resource Department can be built. This is one of the reasons why an employee attrition dataset is used. Visualization of the dataset is done with the most popular visualization tool Tableau

A comparison of models such as Naïve Bayes, Logistic Regression, Deep Learning, Decision Tree, Random Forest and Gradient Boosted tree will be made on Rapid Miner. A brief discussion on the performance of these models will be done. Recommendations will be based on the results of coding in R programming language and Rapid Miner.

Table of Contents

| | |
|---|-----------|
| List of Tables and Figures..... | 7 |
| Chapter 1 Introduction..... | 12 |
| 1.1 The Problem..... | 13 |
| Chapter 2 Literature Review..... | 14 |
| 2.2 Methodologies adopted..... | 14 |
| 2.3 Rapid Miner..... | 14 |
| 2.3.1 Industrial Use of Rapid Miner..... | 14 |
| 2.3.2 Features in Rapid Miner..... | 15 |
| 2.3.3 Guided Data Preparation..... | 15 |
| 2.3.4 Automated Model Selection and Optimization..... | 15 |
| 2.3.5 Turn Predictive Models into Prescriptive Actions..... | 15 |
| 2.4 Machine Learning Algorithms..... | 10 |
| 2.5 Choice Of Algorithm..... | 16 |
| 2.6 Tableau..... | 17 |
| Chapter 3 Research Methodology and Methods..... | 18 |
| 3.1 Naïve Byes..... | 18 |
| 3.2 Decision Tree..... | 21 |
| 3.3 Random Forest..... | 27 |
| 3.4 Logistic Regression..... | 34 |
| 3.5 Support Vector Machines..... | 42 |
| 3.6 Monte Carlo Simulations..... | 42 |
| 3.6.1 Libraries Needed..... | 46 |

| | |
|---|-----------|
| 3.6.2 Arguments in Monte Carlo..... | 47 |
| 3.7 Working on RapidMiner..... | 48 |
| Chapter 4 Data Analysis and Finding..... | 54 |
| Chapter 5 Discussion..... | 60 |
| 5.1 Comparison with Monte Carlo Simulation..... | 60 |
| 5.2 Monte Carlo Simulation With Different Iterations..... | 60 |
| 5.2.1 Monte Carlo with nrep = 5..... | 60 |
| 5.2.2 Monte Carlo with nrep = 10..... | 62 |
| 5.2.3 Monte Carlo with nrep = 50..... | 64 |
| 5.2.4 Monte Carlo with nrep = 100..... | 66 |
| 5.2.5 Monte Carlo with nrep = 200..... | 68 |
| 5.3 Comparison with RapidMiner..... | 70 |
| 5.3.1 Naïve Bayes..... | 71 |
| 5.3.2 Logistic Regression..... | 72 |
| 5.3.3 Deep Learning..... | 73 |
| 5.3.4 Decision Tree..... | 74 |
| 5.3.4 Random Forest..... | 75 |
| 5.3.6 Gradient Boosted Trees..... | 76 |
| Chapter 6 Conclusion and Recommendations..... | 77 |
| Chapter 7 Reflection..... | 81 |
| Appendix..... | 83 |
| Bibliography..... | 84 |

List of Table and Figures

Tables

- Table 1 - Confusion Matrix Naïve Bayes
- Table 2 – Confusion Matrix Decision Tree
- Table 3 – Confusion Matrix Random Forest
- Table 4 – Confusion Matrix Random Forest
- Table 5 – Confusion Matrix Logistic Regression
- Table 6 – Confusion Matrix Support Vector Machines 'Linear Kernel'
- Table 7 – Output of accuracy for 5 Monte Carlo simulation
- Table 8 - Output of accuracy for 10 Monte Carlo simulation
- Table 9 - Output of accuracy for 50 Monte Carlo simulation
- Table 10 - Output of accuracy for 100 Monte Carlo simulation
- Table 11 - Output of accuracy for 200 Monte Carlo simulation
- Table 12 - Confusion Matrix Naive Bayes (RapidMiner)
- Table 13 - Confusion Matrix Logistic Regression (RapidMiner)
- Table 14 – Confusion Matrix Deep Learning (RapidMiner)
- Table 15 - Confusion Matrix Decision Tree (RapidMiner)
- Table 16 - Confusion Matrix Random Forest (RapidMiner)
- Table 17 - Confusion Matrix Gradient Boosted Trees (RapidMiner)
- Table 18 - Comparison of Outputs of Algorithms that were run separately in R
- Table 19 - Comparison of Performance in RapidMiner

Figures

- Figure 1 - Recruitment Process

Figure 2 – Naïve Bayes - Model Output

Figure 3 – Naïve Bayes – Predicted Values

Figure 4 - Naïve Bayes (Naïve Bayes Theorem, June 30, no year)

Figure 5 - Decision Tree (Things to keep in mind while working with Decision Trees, Prakash Saxena, August 6, 2017)

Figure 6 – Decision Tree – Model Output

Figure 7 – Decision Tree – Predicted Values

Figure 8 – Decision Tree – Model Output – Gini

Figure 9 - Decision Tree – Predicted Values – Gini

Figure 10 - Decision Tree – Tree plot

Figure 11 - Decision Tree – Tree plot – Gini

Figure 12 - Decision Tree – Tree plot – Information Gain

Figure 13 – Random Forest Output

Figure 14 - Logistic Regression Model (Vijay Kotu, Bala Deshpande Phd, 2015, chapter 5, pg 183)

Figure 15 – GUI of Rattle

Figure 16 – Variables

Figure 17 – Model Tab in Rattle

Figure 18 – Model Summary Logistic Regression

Figure 19 – Logistic Regression Model Plot

Figure 20 – Evaluate Tab in Rattle

Figure 21 – Cost Curve Plot – Logistic Regression

Figure 22 – Plot Precision Vs Recall

Figure 23 – Plot Sensitivity Vs Specificity

Figure 24 - Support Vector Machine (Vijay Kotu, Bala Deshpande Phd, 2015, Pg 64)

Figure 25 – Support Vector Machine – Linear Kernel – Output Summary

Figure 26 - Support Vector Machine – Polynomial Kernel – Output Summary

Figure 27 - Support Vector Machine – Sigmoid Kernel – Output Summary

Figure 28 - Support Vector Machine – Radial Kernel – Output Summary

Figure 29 - First Screen in RapidMiner

Figure 30 - Loading Data in Auto Model

Figure 31 – Selecting the data from My Computer or Database

Figure 32 - Selecting Target Values

Figure 33 - Selecting Inputs

Figure 34 - Model Types

Figure 35 - Model Types Continued

Figure 36 – Results

Figure 37 – ROC Comparison

Figure 38 - Attrition Vs Department

Figure 39 - Attrition Vs Education Field

Figure 40 - Attrition Vs Education Field 2

Figure 41 - Job Role Vs Environment Satisfaction

Figure 42 - Business Travel Categories

Figure 43 - Different Departments

Figure 44 - Different Education Field employees belong to

Figure 45 - Male Female Ratio

Figure 46 - Categories of Job Role

Figure 47 - Marital Status of employees

Figure 48 – Overtime

Figure 49 - Overtime Vs Attrition

Figure 50 - Plot with 5 Monte Carlo repetition

Figure 51 – Scatter plot (5 Monte Carlo repetition)

Figure 52 – Scatter plot with elliptical (5 Monte Carlo repetition)

Figure 53 – Plot with 10 Monte Carlo repetition

Figure 54 – Scatter plot (10 Monte Carlo repetition)

Figure 55 – Scatter plot with elliptical (10 Monte Carlo repetition)

Figure 56 – Plot with 50 Monte Carlo repetition

Figure 57 – Scatter plot (50 Monte Carlo repetition)

Figure 58 – Scatter plot with elliptical (50 Monte Carlo repetition)

Figure 59 - Plot with 100 Monte Carlo repetition

Figure 60 – Scatter plot (100 Monte Carlo repetition)

Figure 61 – Scatter plot with elliptical (100 Monte Carlo repetition)

Figure 62 – Plot with 200 Monte Carlo repetition

Figure 63 – Scatter plot (200 Monte Carlo repetition)

Figure 64 – Scatter plot with elliptical (200 Monte Carlo repetition)

Figure 65 – ROC Comparison in RapidMiner

Figure 66 – Performance

Figure 67 – AUC Plot for Naïve Bayes

Figure 68 – AUC Plot for Logistic Regression

Figure 69 – AUC Plot for Deep Learning

Figure 70 – AUC Plot for Decision Tree

Figure 71 – AUC Plot for Random Forest

Figure 72 – AUC Plot for Gradient Boosted Trees

Figure 73 - Viz of Accuracy and Runtime

Chapter 1 Introduction

The Human Resource Department plays a vital role in every company. They take care of the entire hiring process right from posting about vacancies, screening various applications and applicants, conducting interviews to onboarding employee. A huge amount of work and time is spent during the hiring process. First it begins with posting about the job opportunities. The recruitment process (Typical recruitment process, no date) is like the Figure 1.

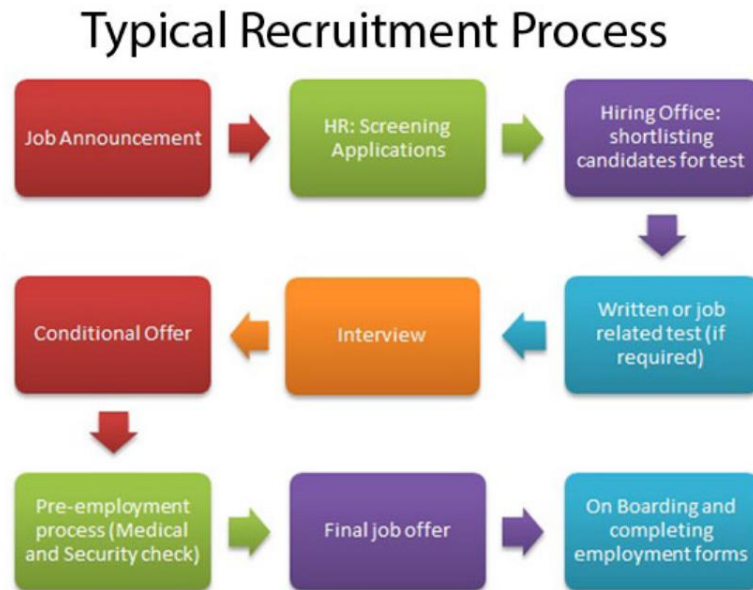


Figure 1

Employee are one of the biggest assets in any organization. Let's face it is hard for any business or organization to function effectively if it doesn't have the right kind of people to work. Hence, it is a huge responsibility for a company to take care of the employee and also make sure they are being productive. The organization also has to make sure that they hire the right candidate too. Hence, they ensure in taking the right steps while hiring. Later on, after hiring they also take care of the performance of the employee and help them accordingly.

There are several factors which decides whether a person stays or not in an organization. Some of them includes salary structure, job satisfaction, equal opportunities for all, how challenging the job role is and many more. Here the monthly salary is one of the most important reason for an employee to stay or leave. If as a person we don't get paid fairly well for the work done, the next thing will be to start

thinking about another job or moving ahead career wise. Some might opt to discuss about the pay to the manager as well.

After monthly salary comes the job satisfaction. If the employee is not happy with the work, the coworkers, the environment and the place, they would start thinking about looking for another job. Job satisfaction is actually a collection of several factors. So, there are several factors that the HR department has to look at. They might tie up with the talent management or the learning and development team to discuss on how to do this.

1.1 The Problem

One of the things that costs organizations much is from employee turnover. While some of the costs are tangible such as the time taken by an employee to become a productive member and the training that was given to the employee. But, most of the costs are intangible like relationship with the customer created by that employee, good management skills and many more. With the help of machine learning algorithm, it is not only possible to predict employee attrition, but it is also possible decipher the key factors that affect employee attrition. (Matt Dancho, BusinessScience.io kdnuggets). Hence the aim of the dissertation is to suggest a model or two that is useful for predicting Employee attrition.

Chapter 2 – Literature Review

2.1 Insight on Previous works

There are quite a few works done on Employee attrition previously like “Employee Attrition Prediction” - by Rahul Yedida, Rakshit Vahi, Abhilash, Rahul Reddy, Rahul j and Deepti Kulkarni, PESIT Bangalore. The project was on predicting Employee turnover with the help of KNN Algorithm. The project also has a comparison of other algorithms such as Naïve Bayes, Logistic Regression and MLP Classifier (Artificial Neural Networks). The conclusion of the paper was that KNN algorithm was the best in terms of predicting employee turnover.

This dissertation is about comparing different machine learning algorithms such as Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine. Comparison is done with the help of Monte Carlo Simulation and RapidMiner. Data Visualization is done with the help of Tableau which is a powerful visualization tool used by any organizations. Let’s look at the methodologies, tools that are being used in this dissertation.

2.2 Methodologies Adopted

Monte Carlo simulation is adopted to run the models several times and then visualize the accuracy of the model in a density plot. Monte Carlo simulation is used so that the test set and trainset split will be random. The simulation does not have two iterations that are similar. Hence, the output would be more efficient. The feature of Auto Model in RapidMiner is used for comparison of different models. As the Auto Model feature in RapidMiner automatically suggests different models based on the dataset, the comparison will be done among Logistic Regression, Deep Learning, Decision Tree, Random Forest and Gradient Boosted Trees.

2.3 RapidMiner

Wikipedia says that “Rapid Miner is a data science platform that provides an integrated environment for data preparation, machine learning, deep learning, text mining and predictive analysis. This is used for both business and commercial purposes.” (Rapid Miner, Wikipedia, no date)

2.3.1 Industrial Use Of RapidMiner

Some of the companies that make use of rapid miner are LIAT (Leeward Island Air Transport), Mobilkom Austria, Lufthansa, MMC (Modern Marketing Concepts, based in NY) and many more. Companies like

PayPal and a pharmaceutical company in Europe use RapidMiner for Sentiment analysis. With the help of RapidMiner's Predictive Analysis, Lufthansa profited with preventive maintenance and reduction downtime and failure. Lufthansa was also able to increase their operational probability with the help of better flight arrival time predictions. (Data Science Case Studies, Rapid Miner, no date)

2.3.2 Features in RapidMiner

Rapid Miner has a feature called the Auto Model. This improves the productivity of a data scientist and also helps them to know better of why and how a model works. In other words, a user can upload a dataset and can prepare data, model them in a guided manner. Finally, we get to compare results and suggestions will be given by Rapid Miner. Some of the features that we can find in Auto model are (RapidMiner Auto Model, RapidMiner, no date)

2.3.3 Guided Data Preparation

First, we have to upload the data for which we are about to build a model. We can select an attribute for which a prediction, classification or finding outliers can be done. As the name says, RapidMiner automatically does a data analysis and finds out things like missing values, outliers, stability, correlation etc.

2.3.4 Automated Model Selection and Optimization

Once the data is loaded and the attributes are selected for which prediction or classification is selected, next is modelling. RapidMiner helps a user by suggesting the best models that can be applied to the data. We can either selected the desired models or can choose all of them which are Naïve Bayes, Logistic Regression, Deep Learning, Random Forest and Gradient Boost Trees.

2.3.5 Turn Predictive Models into Prescriptive Actions

The feature lets you explain each prediction individually. It also lets the user to optimize by making changes in the parameters to build the model that they desire. The results are in depth that you get to see the visualization, values, predictions etc. The best thing about the Auto Model feature in RapidMiner Studio is that, the process is visible to the analyst. This lets them make changes at different steps if required.

2.4 Machine Learning Algorithms

Machine learning is a part of Artificial Intelligence (AI). Some of the applications of machine learning are virtual personal assistants like Siri and Alexa. Another example is the suggestions that we get on shopping websites based on the products that we have purchased and the items in the shopping cart. The map that we use while commuting to find the routes.

Machine learning as the name suggests is the capability of a machine to learn without the intervention of human beings. That is, it learns on its own from the data that is fed. It can learn in different ways; one way is machine learns with the already available data. Another way is to learn from data with similar patterns. Hence machine learning is of two types.

The two types of machine learning algorithms are supervised and unsupervised. Supervised learning models have the attributes known. Supervised Learning algorithms are again divided into prediction and classification. There are different algorithms for predictions such as regression and the algorithms for classification such as K-Nearest Neighbor. Then there are algorithms like time series forecasting that are unsupervised learning algorithm which are mainly used for forecasting.

The reason for choosing algorithms like Naïve Bayes, Decision Tree, Random Forest, Logistic Regression and Support Vector Machines. All these algorithms are suitable for prediction of a binary variable. Though there are other algorithms that are suitable for predicting binary variables, only the five said algorithms were chosen to make sure that there is enough time for research.

2.5 Choice of Algorithms

Choosing algorithms for a set of data is very important and the one of the initial steps in data analysis. Since the target value (Attrition) is binary the algorithms Naïve Bayes, Decision Tree, Random Forest and Support Vector Machines chosen to run in Monte Carlo simulation for comparison. The dissertation aims at comparing the algorithms and suggest the optimal one for prediction of employee attrition. This will be helpful for the HR department to lookout for people who are most likely to leave the organization. When this is known in advance, they can do something proactively for the employee to stay. When an employee leaves a company, the company also losses the time invested in training the employee and the unique talents of that particular person that is essential for running business. The rapport with the client also is lost as it is different with different people. Hence, it is essential for the organization to look into the employee turnover and plan accordingly.

2.6 Tableau

Tableau is one of the popular visualization tools that has been used for visualization in this dissertation.

Tableau is one of those data visualization tools that are used by most of the Industries, where data visualization is important. Organizations like Lufthansa and PepsiCo and universities like University of Michigan Medical Center use tableau which has helped in better visualization of their data.

Chapter 3 - Research Methodology and Methods

The methodology chapter deals with different algorithms that will be applied on the selected dataset such as Decision Tree, Random Forest and Naïve Bayes and Support Vector Machines. Then a comparison is made among the performances of the algorithms chosen. Based on the results a conclusion is drawn as to how to solve the research question is the most efficient way. First the four algorithms are run in a Monte Carlo Simulator and then the auto model feature is chosen in RapidMiner to make further comparison.

3.1 Naïve Bayes

Empirical models consist of explanatory and predictive algorithms which are further classified into supervised and unsupervised learning. Supervised learning has two categories such as regression and classification. Naïve Bayes is one of the categorical algorithms that are used for classification. In this classification algorithm the output class is found by finding the prior probability and class conditional property which is based on Bayes theorem.

The Naïve Bayes Algorithm is derived from probability and statistics theory. It leverages the probabilistic relationship between the factors/attributes and the class label/outcome. As and when we get the evidence of the factors affecting the outcome, we can make quality guesses on the probability of the outcome. As the name suggest, the algorithm makes a naïve assumption on the independence between attributes, which need not be true at all times. Naïve Bayes is named after the Reverend Thomas Bayes.

Applying Naïve Bayes algorithm on 80% of the dataset in R we get the model as below. (Appendix B)

```
model<- naiveBayes(Attrition~., data = trainset)
model
```

```

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      No      Yes
0.837585 0.162415

Conditional probabilities:

Age
Y      [,1]      [,2]
No 37.39086 8.902162
Yes 33.51832 9.519422

BusinessTravel
Y      Non-Travel Frequently Travel_Rarely
No 0.11065990 0.16243655 0.72690355
Yes 0.03664921 0.30366492 0.65968586

DailyRate
Y      [,1]      [,2]
No 813.4447 403.2052
Yes 730.0576 393.2612

Department
Y      Human Resources Research & Development Sales
No 0.03959391 0.67309645 0.28730964
Yes 0.03664921 0.56544503 0.39790576

DistanceFromHome
Y      [,1]      [,2]
No 8.93401 8.070995
Yes 10.70681 8.421020

Education
Y      [,1]      [,2]
No 2.921827 1.0235940
Yes 2.853403 0.9891391

```

Figure 2

After modeling, the next step is to make predictions based on the testset, which is as per the below

```

> n_predict<- predict(model, testset)
> n_predict
 [1] No No No No No No Yes No Yes No No No No No No No Yes No No No Yes Yes
[22] Yes No No No No No No No Yes No No No Yes No No No No No Yes Yes No No No No
[43] No No No No No Yes No Yes No No No No No No No No No No No No No No No Yes
[64] No Yes No Yes No Yes Yes No No No No Yes No Yes No No No No No No No Yes No
[85] No No No No No No Yes Yes No Yes No Yes Yes No No No No Yes Yes No No Yes
[106] Yes No No No No No Yes No No Yes No No No No No No Yes No No No Yes Yes
[127] No No No Yes No No No No Yes Yes Yes No No No No Yes No No No Yes Yes No
[148] No No No Yes Yes Yes Yes No No Yes No No No No No Yes No No No No No No
[169] No Yes Yes No No Yes Yes No No No No No Yes No No No No No No Yes No
[190] No No No No Yes No No No Yes No No No No No No Yes No Yes No No No No
[211] No No No Yes No Yes No Yes No No Yes Yes No No Yes No No No No Yes No
[232] No No No No No No No Yes Yes No No No No Yes No Yes No No Yes No No
[253] No Yes No No No No No No Yes No No Yes Yes No No Yes No No Yes No Yes
[274] No No No No No No Yes No No Yes Yes No No No Yes Yes No Yes No No
Levels: No Yes
>

```

Figure 3

Let's assume that Y is the target variable, or the outcome and X is the evidence or factor. X is a set and not an individual attribute hence we consider $X = \{X_1, X_2, X_3, X_4, \dots, X_n\}$. Then the prior probability would be the probability of outcome that is $P(Y)$, which shows the likelihood of an outcome in the taken data set. Here $P(Y)$ is the probability of a person leaving the organization or not. $P(Y|X)$ is called the conditional probability. Conditional probability is the outcome of the target variable given an evidence, which is attrition given the evidence monthly income or promotion. Conditional probability is also known as the p

posterior probability. The objective of Naïve Bayes is to calculate the posterior probability which can be calculated as

$$P(Y|X) = (P(Y) * \frac{P(X|Y)}{P(X)})$$

Here $P(X|Y)$ is called the class conditional probability.



Figure 4

The confusion matrix can be calculated with the code as below

```
confusion_nb<- table(n_predict, testset$Attrition)
```

```
confusion_nb
```

which gives us the confusion matrix as below

Table 1

| n = 294 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 205 | FP = 16 | 221 |
| Actual: yes | FN = 44 | TP = 29 | 73 |
| | 249 | 45 | |

From the confusion matrix we can calculate the following

- Accuracy: $(TP+TN)/\text{Total} = 0.79591$, says how often the classifier is correct
- Misclassification: $(FP+FN)/\text{Total} = 0.2040816$, says how often it is wrong.

- True Positive Rate also called Sensitivity: $TP/\text{actual yes} = 0.3972$, how often does it predict yes when it actually yes
- False Positive Rate: $FP/\text{actual no} = 0.072398$, how often does it predict yes when it actually is no
- Specificity: $TN/\text{actual no} = 0.9276$, how often does it predict no when it actually no.
- Precision: $TP/\text{predicted yes} = 0.64444$, how often does it predict yes, and it is right.
- Prevalence: $\text{Actual Yes}/\text{Total} = 0.24829$, How often does the yes condition actually occur

Bayesian models are relatively easy to understand and practically implement in a programming language. The model is robust and can handle missing values without any hassle. When there is a missing value, it simply omits that particular class conditional probability which cannot be done in Random Forest and Decision Tree. In spite of its robustness, Naïve Bayes has quite few limitations. Some of them are:

- Incomplete training set: When the test set does not contain examples similar to the ones in trainset, then it is difficult to model.
- Continuous Attribute: If the attributes are continuous numerical values instead of nominal, then we will have to convert the attributes. This is because it is not possible to compute the possibility of continuous variable in the method used for counts.
- Attribute Independence: The most important assumption that is made in the algorithm is that the variables are independent. This condition is a stringent as it might not be the case in real life. Anyway, in real life we can sort this issue with a set of modified correlated features. Or a test for independence can be done with the help of chi-square test.

3.2 Decision Tree

Decision Trees are supervised, non-parametric learning method, which is used Regression and Classification. It learns different decision rules from the given data that is the aim of Decision Tree. This is one of the algorithms which are easy to build for an analyst at the same time very easy to understand from a customer's perspective. Let's look at some of the advantages and disadvantages of decision tree. Decision Tree are used to predict or classify the target values which are categorical in nature. It is best suited when the target variable is binary in nature. They are also called classification trees.

While regression trees are used for target variables that are continuous or numerical. Hence for prediction regression trees are used while to classify, classification trees are used. Therefore, the decision to use regression or classification lies on the target variable.

A Decision Tree

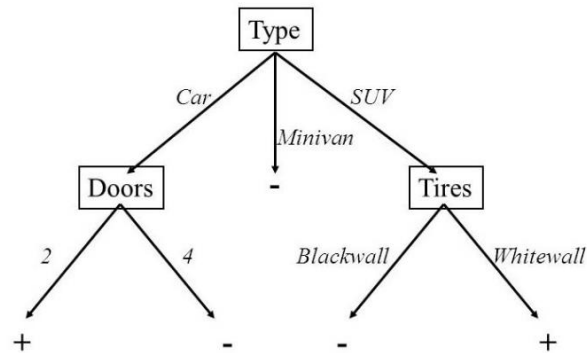


Figure 5

In a Decision Trees Algorithm every attribute is tested at a node like in a flowchart. The prediction is made at the leaf node which is end of every decision tree path. At every node the data is split into different subsets. The data in Decision Tree is split based on the homogeneity of the data. The criteria for Decision Tree according to (Vijay Kotu, Bala Deshpande Phd, 2015, Pg 64) is as follows:

- “The measure of impurity of a data set must be at maximum when all possible classes are equally represented
- The measure of impurity of the data set must be zero when only one class is represented. “

Now, decision tree as the name suggests has a decision to be answered at every Node and then based on a Yes and No, it splits up further. We should know when to stop splitting the data. When the algorithm comes to a situation where it cannot the minimum information gain is not satisfactory then it is time to stop splitting data. The issue when a tree becomes big are there will be many more layers and we will not be able to interpret as well. This situation is otherwise called overfitting. (Vijay Kotu, Bala Deshpande Phd, 2015, Pg 70). A solution for the problem of overfitting are pre-pruning and post-pruning. Modeling with Decision Tree is as below (Appendix C)

```
dt_model<- rpart(Attrition~., data = trainset)
dt_model
```

```

> dt_model
n= 1176

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 1176 177 No (0.84948980 0.15051020)
2) TotalWorkingYears>=2.5 1083 136 No (0.87442290 0.12557710)
4) OverTime=No 774 59 No (0.92377261 0.07622739) *
5) OverTime=Yes 309 77 No (0.75080906 0.24919094)
10) JobRole=Healthcare Representative,Manager,Manufacturing Director,Research Director 104 5 No (0.95192
308 0.04807692) *
11) JobRole=Human Resources,Laboratory Technician,Research Scientist,Sales Executive,Sales Representative
205 72 No (0.64878049 0.35121951)
22) StockOptionLevel>=0.5 121 29 No (0.76033058 0.23966942)
44) BusinessTravel=Non-Travel,Travel_Rarely 99 16 No (0.83838384 0.16161616) *
45) BusinessTravel=Travel_Frequently 22 9 Yes (0.40909091 0.59090909)
90) EducationField=Medical,Other 11 3 No (0.72727273 0.27272727) *
91) EducationField=Human Resources,Life Sciences,Marketing,Technical Degree 11 1 Yes (0.09090909 0
.90909091) *
23) StockOptionLevel< 0.5 84 41 Yes (0.48809524 0.51190476)
46) JobSatisfaction>=3.5 25 5 No (0.80000000 0.20000000) *
47) JobSatisfaction< 3.5 59 21 Yes (0.35593220 0.64406780)
94) JobRole=Research Scientist 20 8 No (0.60000000 0.40000000)
188) MonthlyRate< 8053.5 9 1 No (0.88888889 0.11111111) *
189) MonthlyRate>=8053.5 11 4 Yes (0.36363636 0.63636364) *
95) JobRole=Human Resources,Laboratory Technician,Sales Executive,Sales Representative 39 9 Yes (0
.23076923 0.76923077)
190) YearsAtCompany>=8.5 7 2 No (0.71428571 0.28571429) *
191) YearsAtCompany< 8.5 32 4 Yes (0.12500000 0.87500000) *
3) TotalWorkingYears< 2.5 93 41 No (0.55913978 0.44086022)
6) JobRole=Research Scientist 30 5 No (0.83333333 0.16666667) *
7) JobRole=Human Resources,Laboratory Technician,Sales Representative 63 27 Yes (0.42857143 0.57142857)
14) MaritalStatus=Divorced 11 2 No (0.81818182 0.18181818) *
15) MaritalStatus=Married,Single 52 18 Yes (0.34615385 0.65384615)
30) Age>=33.5 7 1 No (0.85714286 0.14285714) *
31) Age< 33.5 45 12 Yes (0.26666667 0.73333333) *

```

Figure 6

The predicted values are as

```

> pred_att
8 13 14 24 27 34 35 41 46 49 50 55 57 58 70 82 87
No No No No No No No No No No No No No No No No
88 90 94 99 105 108 124 132 138 139 142 145 146 151 153 163 166
No No No No No Yes No No No No No No No No No No
170 177 180 182 196 211 219 222 224 226 230 241 247 255 263 269 272
No No No No No No No No No No No No No No No No
273 283 286 297 304 310 317 318 323 324 326 327 331 332 340 348 349
No No No Yes No No No No No No No No No No No No
356 360 366 371 373 376 377 378 379 383 386 396 404 418 421 428 452
No No No Yes No No No No No No No No No No No No
462 463 469 478 491 496 502 503 505 508 516 521 524 531 532 541 543
No No No No No No No No No No No No No No No No
545 552 563 565 584 586 591 592 596 604 612 613 621 624 635 643 644
No No No No No Yes No No No No No No Yes No No Yes
650 653 654 656 659 660 661 664 665 668 673 678 689 695 703 708 710
No No No No No No No Yes No No No No No Yes No No Yes
711 718 735 736 739 743 744 746 749 752 755 756 758 765 772 776 777
No No No No No No No No No No No No No No Yes No Yes
785 786 787 792 794 798 799 800 805 806 807 809 810 828 839 851 852
No No No Yes No No No No No No No No No Yes No No No
865 873 879 881 893 895 897 900 901 913 914 929 930 944 945 954 960
No No No Yes No No No No No No No No Yes No No No No
962 963 965 967 969 970 978 981 986 989 996 1001 1004 1008 1016 1024 1025
No No No No No No No No No Yes Yes No No No No No No
1027 1033 1041 1044 1052 1056 1059 1072 1076 1078 1079 1087 1094 1107 1114 1119 1126
No Yes No No No No No No No No No No No No No No No
1129 1143 1152 1153 1154 1156 1163 1167 1168 1183 1186 1188 1194 1197 1198 1199 1200
Yes No No No Yes No No No No No No No No Yes Yes No No
1206 1210 1216 1217 1218 1219 1221 1230 1231 1238 1246 1252 1253 1257 1258 1264 1269
Yes No No No No No No No No No No No No No No No No
1271 1273 1276 1277 1279 1281 1292 1294 1296 1328 1334 1338 1339 1340 1344 1353 1355
No No No No No No No No No No No No No Yes Yes No No No
1358 1365 1369 1371 1373 1377 1386 1388 1392 1406 1407 1412 1418 1419 1437 1446 1448
No Yes No No No No No No No No No No No No Yes No No
1454 1459 1462 1464 1467
No No No No No
Levels: No Yes

```

Figure 7

We can also model the data based on Gini index and Information gain. But the output seems to be the same without any changes.

```
> dt_model_gini
n= 1176

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 1176 177 No (0.84948980 0.15051020)
  2) TotalWorkingYears>=2.5 1083 136 No (0.87442290 0.12557710)
    4) OverTime=No 774 59 No (0.92377261 0.07622739) *
    5) OverTime=Yes 309 77 No (0.75080906 0.24919094)
      10) JobRole=Healthcare Representative,Manager,Manufacturing Director,Research Director 104 5 No (0.95192308 0.04807692) *
      11) JobRole=Human Resources,Laboratory Technician,Research Scientist,Sales Executive,Sales Representative 205 72 No (0.64878049 0.35121951)
        22) StockOptionLevel>=0.5 121 29 No (0.76033058 0.23966942)
          44) BusinessTravel=Non-Travel,Travel_Rarely 99 16 No (0.83838384 0.16161616) *
          45) BusinessTravel=Travel_Frequently 22 9 Yes (0.40909091 0.59090909)
            90) EducationField=Medical,Other 11 3 No (0.72727273 0.27272727) *
            91) EducationField=Human Resources,Life Sciences,Marketing,Technical Degree 11 1 Yes (0.09090909 0.90909091) *
              23) StockOptionLevel< 0.5 84 41 Yes (0.48809524 0.51190476)
                46) JobSatisfaction>=3.5 25 5 No (0.80000000 0.20000000) *
                47) JobSatisfaction< 3.5 59 21 Yes (0.35593220 0.64406780)
                  94) JobRole=Research Scientist 20 8 No (0.60000000 0.40000000)
                    188) MonthlyRate< 8053.5 9 1 No (0.88888889 0.11111111) *
                    189) MonthlyRate>=8053.5 11 4 Yes (0.36363636 0.63636364) *
                  95) JobRole=Human Resources,Laboratory Technician,Sales Executive,Sales Representative 39 9 Yes (0.23076923 0.76923077)
                    190) YearsAtCompany>=8.5 7 2 No (0.71428571 0.28571429) *
                    191) YearsAtCompany< 8.5 32 4 Yes (0.12500000 0.87500000) *
                3) TotalWorkingYears< 2.5 93 41 No (0.55913978 0.44086022)
                  6) JobRole=Research Scientist 30 5 No (0.83333333 0.16666667) *
                  7) JobRole=Human Resources,Laboratory Technician,Sales Representative 63 27 Yes (0.42857143 0.57142857)
                    14) MaritalStatus=Divorced 11 2 No (0.81818182 0.18181818) *
                    15) MaritalStatus=Married,Single 52 18 Yes (0.34615385 0.65384615)
                      30) Age>=33.5 7 1 No (0.85714286 0.14285714) *
                      31) Age< 33.5 45 12 Yes (0.26666667 0.73333333) *
```

Figure 8

The predicted value remains the same

```
> pred_att_gini
      8  13  14  24  27  34  35  41  46  49  50  55  57  58  70  82  87
No No No No No No No No No No No No No No No No No
88 90 94 99 105 108 124 132 138 139 142 145 146 151 153 163 166
No No No No No Yes No No No No No No No No No No No No
170 177 180 182 196 211 219 222 224 226 230 241 247 255 263 269 272
No No No No No No No No No No No No No No No No No No
273 283 286 297 304 310 317 318 323 324 326 327 331 332 340 348 349
No No No Yes No No No No No No No No No No No No No No
356 360 366 371 373 376 377 378 379 383 386 396 404 418 421 428 452
No No No Yes No No No No No No No No No No No No No No
462 463 469 478 491 496 502 503 505 508 516 521 524 531 532 541 543
No No No No No No No No No No No No No No No No No No
545 552 563 565 584 586 591 592 596 604 612 613 621 624 635 643 644
No No No No No Yes No No No No No No No Yes No Yes No No
650 653 654 656 659 660 661 664 665 668 673 678 689 695 703 708 710
No No No No No No Yes No No No No No No Yes No No No Yes
711 718 735 736 739 743 744 746 749 752 755 756 758 765 772 776 777
No No No No No No No No No No No No No Yes No No No Yes
785 786 787 792 794 798 799 800 805 806 807 809 810 828 839 851 852
No No No Yes No No No No No No No No No No Yes No No No
865 873 879 881 893 895 897 900 901 913 914 929 930 944 945 954 960
No No No Yes No No No No No No No No No Yes No No No No
962 963 965 967 969 970 978 981 986 989 996 1001 1004 1008 1016 1024 1025
No No No No No No No No No No Yes Yes No No No No No No
1027 1033 1041 1044 1052 1056 1059 1072 1076 1078 1079 1087 1094 1107 1114 1119 1126
No Yes No No No No No No No No No No No No No No No No
1129 1143 1152 1153 1154 1156 1163 1167 1168 1183 1186 1188 1194 1197 1198 1199 1200
Yes No No No Yes No No No No No No No No No Yes Yes No No
1206 1210 1216 1217 1218 1219 1221 1230 1231 1238 1246 1252 1253 1257 1258 1264 1269
Yes No No No No No No No No No No No No No No No No No
1271 1273 1276 1277 1279 1281 1292 1294 1296 1328 1334 1338 1339 1340 1344 1353 1355
No No No No No No No No No No No No No Yes Yes No No No
1358 1365 1369 1371 1373 1377 1386 1388 1392 1406 1407 1412 1418 1419 1437 1446 1448
No Yes No No No No No No No No No No No No Yes No No No
1454 1459 1462 1464 1467
No No No No No
```

Figure 9


```
plot(dt_model, margin = 0.1)
```

```
text(dt_model, use.n = TRUE, pretty = TRUE, cex = 0.8)
```

We get the decision tree as below

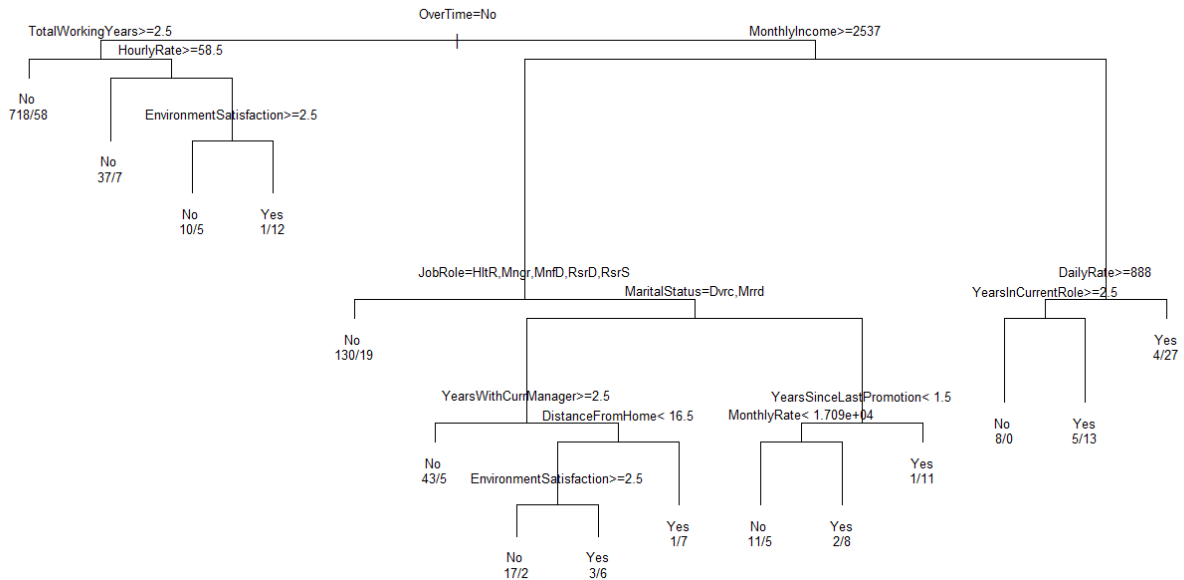


Figure 10

which gives us the confusion matrix as below

Table 2

| n = 294 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 229 | FP = 37 | 266 |
| Actual: yes | FN = 13 | TP = 15 | 28 |
| | 242 | 52 | |

From the confusion matrix we can calculate the following

- Accuracy: $(TP+TN)/Total = 0.82993$, says how often the classifier is correct
- Misclassification: $(FP+FN)/Total = 0.17006$, says how often it is wrong.
- Sensitivity: $TP/actual\ yes = 0.53571$, how often does it predict yes when it actually yes
- False Positive Rate: $FP/actual\ no = 0.139097$, how often does it predict yes when it actually is no
- Specificity: $TN/actual\ no = 0.86090$, how often does it predict no when it actually no.

- Precision: $TP / \text{predicted yes} = 0.22846$, how often does it predict yes, and it is right.
- Prevalence: $\text{Actual Yes} / \text{Total} = 0.09524$, How often does the yes condition actually occurs

We can also model the dataset based on Gini or Information and the output is found to be the same including the confusion matrix and decision tree.

```
plot(dt_model_gini, margin = 0.1)
```

```
text(dt_model_gini, use.n = TRUE, pretty = TRUE, cex = 0.8)
```

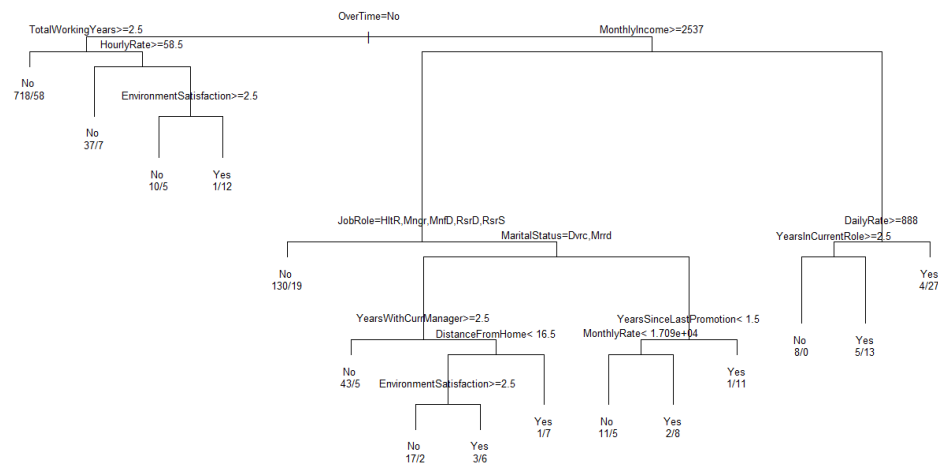


Figure 11

```
plot(dt_model_info, margin = 0.1)
```

```
text(dt_model_info, use.n = TRUE, pretty = TRUE, cex = 0.8)
```

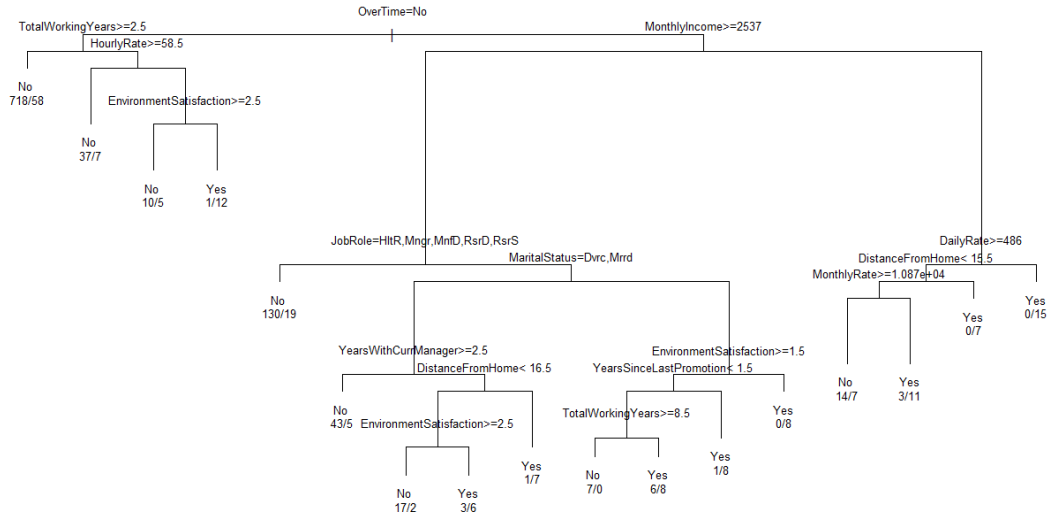


Figure 12

The decision tree model has an accuracy of about 82.99 %. And it predicts the target value No which has a probability of 86.00% accuracy.

3.3 Random Forest

Random Forest Algorithm is one of those algorithms that can do both regression and classification. It has the ability to handle missing values, outliers and makes use of the dimensional reduction method. It is an ensemble machine learning method that brings weaker models together to form one single powerful model. This algorithm works as follows

- Let's say N is the number of cases from the training set. While sampling these N cases are considered with replacement, which is the trainset.
- Consider there are M input variables, m is specified at each node which is lesser than M. m variables are chosen from M input variables randomly. The value m is held constant and the best out of m is considered at each split.
- The trees grow to the maximum extent.
- New data is predicted by calculating the aggregate of the n number of tree samples.

Let's look into the advantages and disadvantages of Random Forest Algorithm

The algorithm can be used for both classification and regression is one of the most important random forest. With the help of importance, it is easy to view which of the input variables matter. Another reason is that the default hyperparameters makes good prediction. Usually overfitting is a problem in

machine learning algorithm. But this isn't an issue because there are enough trees that the overfitting is avoided.

Though the algorithm is effective in every way, there is one limitation. The algorithm slows down as there are many trees for a real-life prediction. In real life applications though the classifier is effective, slows down, which leads to preferring other algorithms over random forest. (The Random Forest Algorithm, Niklas Donges, February 22, no year)

Let's model the dataset with random forest algorithm (Appendix D)

```
rf_model<- randomForest(Attrition ~ ., data = trainset)
```

We get the output as

Call:

```
randomForest(formula = Attrition ~ ., data = trainset)
```

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 5

OOB estimate of error rate: 14.25%

Confusion matrix:

No Yes class.error

No 915 11 0.01187905

Yes 146 30 0.82954545

Calculating the confusion matrix

```
conf_rf<- table(pred_RF,testset$Attrition)
```

we get the output as

```
pred_RF 0 1
```

```
0 304 49
```

```
1 3 12
```

Calculating the importance, we use the code below

```
imp<- importance(rf_model)
```

we get the output as

| | MeanDecreaseGini |
|--------------------------|------------------|
| Age | 16.805898 |
| BusinessTravel | 6.344316 |
| DailyRate | 15.900265 |
| Department | 4.035791 |
| DistanceFromHome | 15.045383 |
| Education | 5.769159 |
| EducationField | 10.613237 |
| EnvironmentSatisfaction | 6.708637 |
| Gender | 2.594661 |
| HourlyRate | 13.090123 |
| JobInvolvement | 6.017912 |
| JobLevel | 5.593018 |
| JobRole | 17.158654 |
| JobSatisfaction | 8.306925 |
| MaritalStatus | 7.503048 |
| MonthlyIncome | 21.523491 |
| MonthlyRate | 14.427617 |
| NumCompaniesWorked | 9.520495 |
| OverTime | 14.450129 |
| PercentSalaryHike | 9.176017 |
| PerformanceRating | 1.166639 |
| RelationshipSatisfaction | 6.289481 |
| StockOptionLevel | 8.414169 |
| TotalWorkingYears | 15.252646 |
| TrainingTimesLastYear | 7.126603 |
| WorkLifeBalance | 9.507054 |
| YearsAtCompany | 12.184399 |

| | |
|-------------------------|-----------|
| YearsInCurrentRole | 7.988544 |
| YearsSinceLastPromotion | 6.777767 |
| YearsWithCurrManager | 10.578628 |

As the name suggests, it tells us the most important values that are significant to the target variable Attrition. With the table it is bit difficult to find out which of the variables are important. To view this slightly easier we can use the code below

`importance(rf_model)[order(importance(rf_model)),]` (Variable Importance for Random Forest Models, Dragonfly Statistics, 31/12/2017)

which gives

| | | |
|--------------------------|-----------------------|-------------------------|
| PerformanceRating | Gender | Department |
| 1.166639 | 2.594661 | 4.035791 |
| JobLevel | Education | JobInvolvement |
| 5.593018 | 5.769159 | 6.017912 |
| RelationshipSatisfaction | BusinessTravel | EnvironmentSatisfaction |
| 6.289481 | 6.344316 | 6.708637 |
| YearsSinceLastPromotion | TrainingTimesLastYear | MaritalStatus |
| 6.777767 | 7.126603 | 7.503048 |
| YearsInCurrentRole | JobSatisfaction | StockOptionLevel |
| 7.988544 | 8.306925 | 8.414169 |
| PercentSalaryHike | WorkLifeBalance | NumCompaniesWorked |
| 9.176017 | 9.507054 | 9.520495 |
| YearsWithCurrManager | EducationField | YearsAtCompany |
| 10.578628 | 10.613237 | 12.184399 |
| HourlyRate | MonthlyRate | OverTime |
| 13.090123 | 14.427617 | 14.450129 |
| DistanceFromHome | TotalWorkingYears | DailyRate |
| 15.045383 | 15.252646 | 15.900265 |
| Age | JobRole | MonthlyIncome |
| 16.805898 | 17.158654 | 21.523491 |

The order starts from the least important to the most important variables.

Plotting the model with the code below

```
plot(rf_model)
```

we get

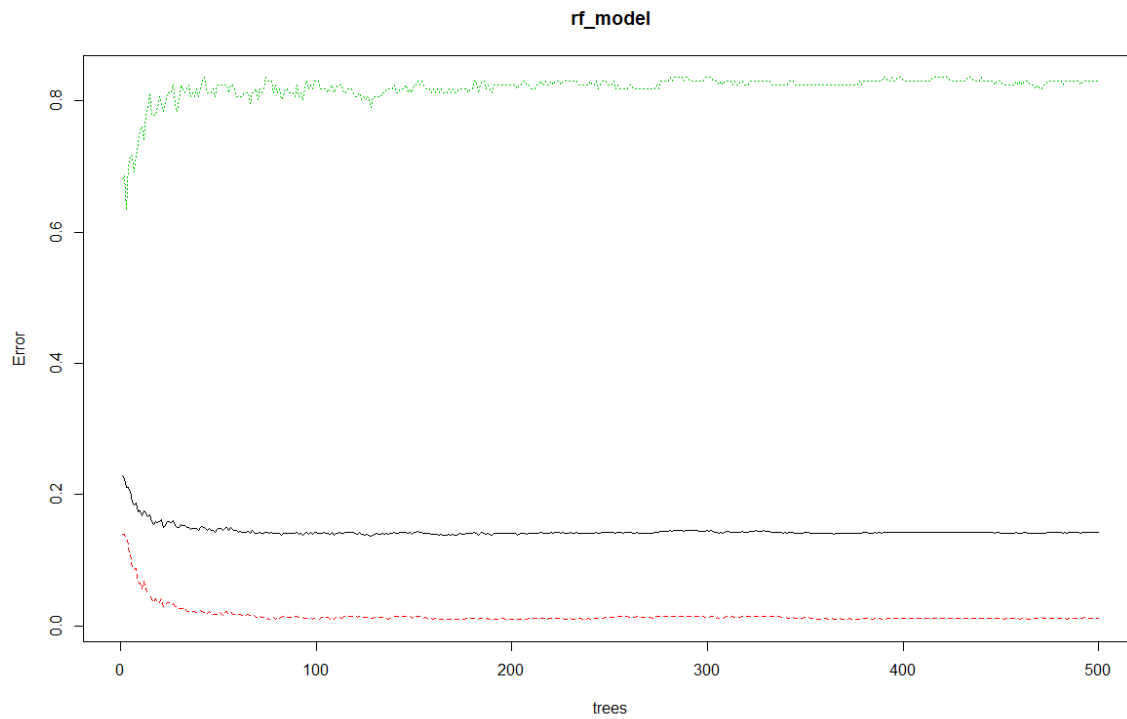


Figure 13

which gives us the confusion matrix as below

Table 3

| n = 368 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 304 | FP = 49 | 353 |
| Actual: yes | FN = 3 | TP = 12 | 15 |
| | 307 | 16 | |

From the confusion matrix we can calculate the following

- Accuracy: $(TP+TN)/Total = 0.8587$, says how often the classifier is correct
- Misclassification: $(FP+FN)/Total = 0.1413$, says how often it is wrong.
- True Positive Rate also called Sensitivity: $TP/actual\ yes = 0.8$, how often does it predict yes when it actually yes
- False Positive Rate: $FP/actual\ no = 0.13881$, how often does it predict yes when it actually is no
- Specificity: $TN/actual\ no = 0.86119$, how often does it predict no when it actually no.
- Precision: $TP/predicted\ yes = 0.75$, how often does it predict yes, and it is right.
- Prevalence: $Actual\ Yes/Total = 0.04076$, How often does the yes condition actually occurs

From the above, we can say that the model is 85.87% accurate. From the output for importance, it is obvious that the monthly income of an employee decides whether they stay in the company or not.

Further, let's select some of the features only and then model the data set. We are only going to retain some of the variables like Age, Monthly income, Number of companies worked, Overtime, Total number of working years, Number of times trained last year, Work life balance, Years at the company, Years at the current role, Years since the last promotion and Years with the manager. We are not considering Daily rate and hourly rate as we have retained monthly rate.

The model was developed with 75% of the data(modified) and the results were as below

Call:

```
randomForest(formula = Attrition ~ ., data = trainset)
```

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 3

OOB estimate of error rate: 14.52%

Confusion matrix:

No Yes class.error

No 904 24 0.02586207

Yes 136 38 0.78160920

And the importance has not changed after feature selection and the results are as below

| | | |
|-------------------------|--------------------|-----------------------|
| YearsInCurrentRole | WorkLifeBalance | YearsWithCurrManager |
| 16.32875 | 16.82707 | 16.97892 |
| YearsSinceLastPromotion | OverTime | TrainingTimesLastYear |
| 18.90823 | 19.95410 | 20.26499 |
| YearsAtCompany | NumCompaniesWorked | TotalWorkingYears |
| 21.20260 | 22.01698 | 29.74415 |
| Age | MonthlyIncome | |
| 45.79580 | 57.25628 | |

The confusion matrix as below

Table 4

| n = 368 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 296 | FP = 52 | 348 |
| Actual: yes | FN = 9 | TP = 11 | 18 |
| | 305 | 63 | |

From the confusion matrix we can calculate the following

- **Accuracy:** $(TP+TN)/Total = 0.83424$.
- **Misclassification:** $(FP+FN)/Total = 0.16576$.
- **Sensitivity:** $TP/actual\ yes = 0.61111$.
- **False Positive Rate:** $FP/actual\ no = 0.14942$.
- **Specificity:** $TN/actual\ no = 0.85057$.
- **Precision:** $TP/predicted\ yes = 0.17460$.
- **Prevalence:** $Actual\ Yes/Total = 0.04891$.

3.4 Logistics Regression

One of the most common predictive analysis techniques is fitting data with a function, which is the main function of regression (Vijay Kotu, Bala Deshpande Phd, 2015, chapter 5, pg 165) Function fitting is actually predicting the value of the dependent variable Y and combining the output variable X , to form a function $y = f(X)$. Two different regression techniques are logistics and linear regression. It is said according to one of the annual surveys in data science, the three most commonly used tools are regression, clustering and decision tree. (Data Science Plus, Logistic Regression, no date) One essential difference between logistic and linear regression is that the dependent variable is binary in nature for logistic while it is continuous for linear regression. One of the famous examples for logistics regression is finding if an email is spam (1) or not (0).

There is a limitation to that we have to deal with while developing the model which is also called “curse of dimensionality” (Vijay Kotu, Bala Deshpande Phd, 2015, chapter 5, pg 165). That is as the predictors X increases, we have to deal with interpretational and computational errors. Also, we will not be able to bring up a good model. Feature selection is a solution to this limitation in regression. Logit is a function that was developed in the mid-twentieth century in biometrics field, which made computation very easy.

Let's consider that both the target and predictor variable are continuous. In case of linear regression x increases while y increases. A line is fit, and it is fine as the data is continuous. But, when it comes to a target variable that is not continuous and has only two values like 0 and 1, then function fitting of a line would not be the right thing. The points on the graph will be pointed only in two parts since we have two values. For a logistic regression a curved line would be appropriate for function fitting like the figure below.

Table 5 (Vijay Kotu and Bala Deshpande, Chapter 5, Pg 183, Figure 5.14)

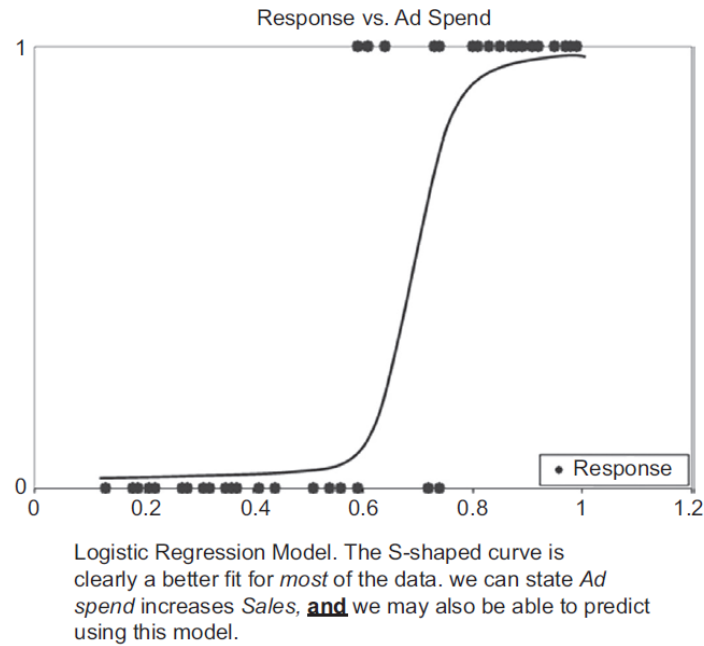


Figure 14

Hence, we can say that the method of function fitting of a non-linear curve to the data (target variable is discrete) is what is Logistic Regression.

The dependent variable y is binomial and has only two values Yes and No or 1 and 0. But our predictors need not be binomial as well. They can be binomial, discrete or continuous. There are no restrictions on the data type for predictors as they can have data of any range. So, the challenge here is to find a sigmoid function that fits the dependent variable y . A logit function helps in finding a solution to the challenge of mapping a continuous function to a discrete function. We are not going to look at the logit function as the dissertation is not about the logit function.

Let's model the data with logistic regression with the help of rattle function in R. This is a GUI (graphical user interface) in R which looks like the below.

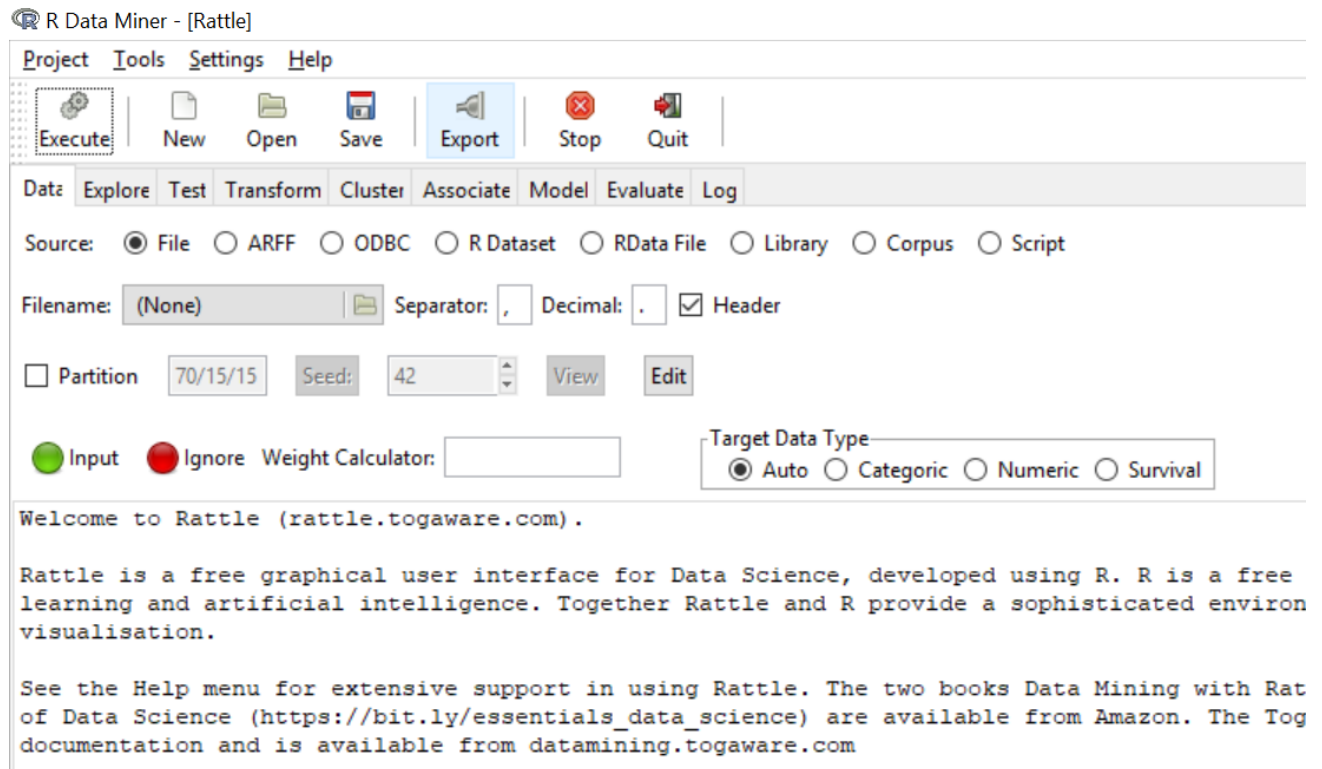


Figure 15

We can load the data with the help of the tab that says filename. After choosing the file we can select on partition, which splits the data into training and test set. Once we choose the partition ratio, we have to click on execute so that the data loads and the partition is done. Once the data is uploaded, all the variables are listed along with the data type. This shows which is the target variable and the variables which are identical, variables which can be ignored and the ones that are unique (employee number). All this is available in the Data tab.

| No. | Variable | Data Type | Input | Target | Risk | Ident | Ignore | Weight | Comment |
|-----|-------------------------|-------------|----------------------------------|----------------------------------|-----------------------|----------------------------------|----------------------------------|-----------------------|---------------|
| 1 | Age | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 43 |
| 2 | Attrition | Categorical | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 2 |
| 3 | BusinessTravel | Categorical | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 3 |
| 4 | DailyRate | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 886 |
| 5 | Department | Categorical | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 3 |
| 6 | DistanceFromHome | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 29 |
| 7 | Education | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 5 |
| 8 | EducationField | Categorical | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 6 |
| 9 | EmployeeCount | Constant | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | Unique: 1 |
| 10 | EmployeeNumber | Ident | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 1,470 |
| 11 | EnvironmentSatisfaction | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 4 |
| 12 | Gender | Categorical | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 2 |
| 13 | HourlyRate | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 71 |
| 14 | JobInvolvement | Numeric | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Unique: 4 |

Figure 16

Let's get to modelling the data, we need to click on the Data tab, which takes us to window with the options as below.

☒ Data
 ☐ Explore
 ☐ Test
 ☐ Transform
 ☐ Cluster
 ☐ Associate
 ☒ Model
 ☐ Evaluate
 ☐ Log

Type:
 ☐ Tree
 ☐ Forest
 ☐ Boost
 ☐ SVM
 ☒ Linear
 ☐ Neural Net
 ☐ Survival
 ☐ All

☐ Numeric
 ☐ Generalized
 ☐ Poisson
 ☒ Logistic
 ☐ Probit
 ☐ Multinomial

Figure 17

We can choose linear here and then we can choose the logistic option. Since the target variable is binary, the option logistic is chosen by default. After this we need to click the Execute button to run the model. We get the summary as below.

Plot

Summary of the Logistic Regression model (built using glm):

Call:

```
glm(formula = Attrition ~ ., family = binomial(link = "logit"),
     data = crs$dataset[crs$strain, c(crs$input, crs$target)])
```

Deviance Residuals:

| Min | 1Q | Median | 3Q | Max |
|---------|---------|---------|---------|--------|
| -1.7074 | -0.4803 | -0.2311 | -0.0749 | 3.3812 |

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) |
|----------------------------------|----------------|----------------|---------|----------------|
| (Intercept) | -11.3798179268 | 642.0937518200 | -0.018 | 0.98586 |
| Age | -0.0283813613 | 0.0168289576 | -1.686 | 0.09171 . |
| BusinessTravelTravel_Frequently | 2.0521444476 | 0.5074058815 | 4.044 | 0.00005246 *** |
| BusinessTravelTravel_Rarely | 1.1969483560 | 0.4660424674 | 2.568 | 0.01022 * |
| DailyRate | -0.0002085492 | 0.0002714038 | -0.768 | 0.44224 |
| DepartmentResearch & Development | 13.0485800205 | 642.0913894418 | 0.020 | 0.98379 |
| DepartmentSales | 12.3198564470 | 642.0917938599 | 0.019 | 0.98469 |
| DistanceFromHome | 0.0411347631 | 0.0134985817 | 3.047 | 0.00231 ** |
| Education | -0.0519463077 | 0.1107946352 | -0.469 | 0.63918 |
| EducationFieldLife Sciences | -0.4284762068 | 0.9887016159 | -0.433 | 0.66474 |
| EducationFieldMarketing | 0.1363672165 | 1.0460366535 | 0.130 | 0.89628 |
| EducationFieldMedical | -0.4500954356 | 0.9799588039 | -0.459 | 0.64602 |
| EducationFieldOther | -0.5457788696 | 1.0763981716 | -0.507 | 0.61213 |
| EducationFieldTechnical Degree | 0.4813800533 | 1.0071235421 | 0.478 | 0.63267 |
| EnvironmentSatisfaction | -0.4541675245 | 0.1011139359 | -4.492 | 0.00000707 *** |

Figure 18

We can plot the model easily by clicking on the plot button to the top left corner of the above picture.

We get the plots as below.

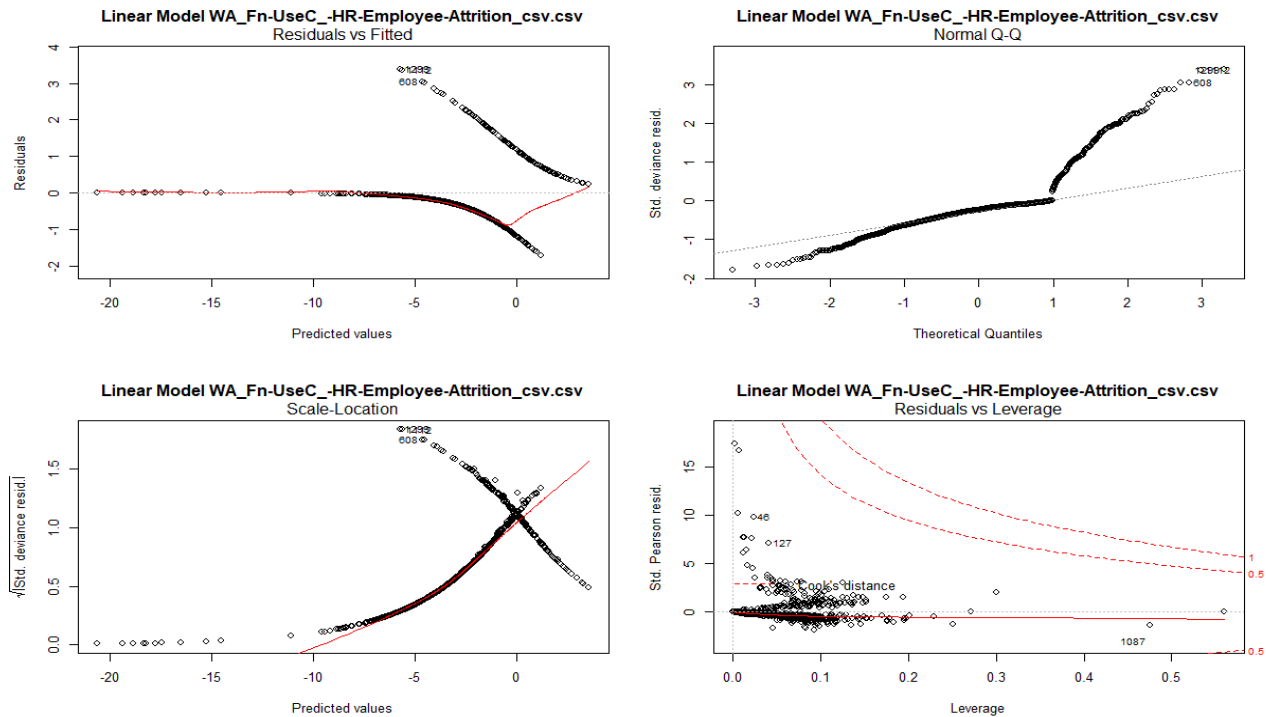


Figure 19

Next, we can compute the confusion matrix or error matrix as mentioned in under the tab evaluation.

Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log

Type:
☒ Error Matrix
☐ Risk
☐ Cost Curve
☐ Hand
☐ Lift
☐ ROC
☐ Precision
☐ Sensitivity
☐ Prv Ob
☐ Score

Model:
☐ Tree
☐ Boost
☐ Forest
☐ SVM
☒ Linear
☐ Neural Net
☐ Survival
☐ KMeans
☐ HClust

Data:
☐ Training
☒ Validation
☐ Testing
☐ Full
☐ Enter
☐ CSV File
☐ employ...
☐ R Dataset

Risk Variable:
Report:
☐ Class
☒ Probability
Include:
☒ Identifiers
☐ All

Error Matrix

An error matrix shows the true outcomes against the predicted outcomes. Two tables will be presented here. The first will be the count of observations and the second will be the proportions.

For a binary classification model the cells of the error matrix are referred to, from the top left going clockwise, as the True Negatives, False Positives, True Positives, and False Negatives.

An error matrix is also known as a confusion matrix.

Figure 20

Once the option is chosen, we need to execute again. Which gives us a matrix like below

Table 5

| n = 220 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 182 | FP = 1 | 183 |
| Actual: yes | FN = 25 | TP = 12 | 37 |
| | 207 | 13 | |

Let's calculate the accuracy, sensitivity, precision, specificity, misclassification error rate and precision from the confusion matrix.

- Accuracy: $(TP+TN)/Total = 0.8818$, says how often the classifier is correct
- Misclassification: $(FP+FN)/Total = 0.11818$, says how often it is wrong.
- True Positive Rate also called Sensitivity: $TP/actual\ yes = 0.32432$, how often does it predict yes when it actually is yes
- False Positive Rate: $FP/actual\ no = 0.005464$, how often does it predict yes when it actually is no
- Specificity: $TN/actual\ no = 0.99453$, how often does it predict no when it actually is no.
- Precision: $TP/predicted\ yes = 0.92308$, how often does it predict yes, and it is right.
- Prevalence: $Actual\ Yes/Total = 0.168118$, How often does the yes condition actually occurs

Calculating the cost curve, we get a plot as below

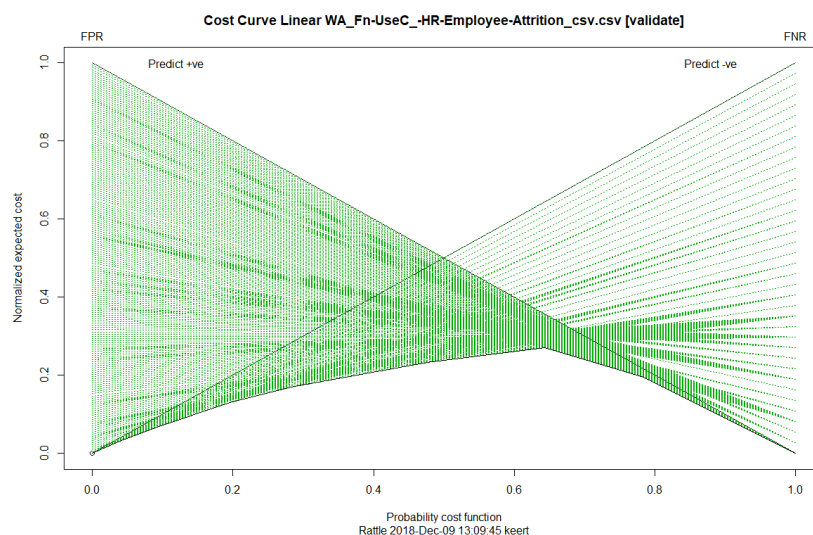


Figure 21

Calculating the Precision and Sensitivity charts we get the plots as below.

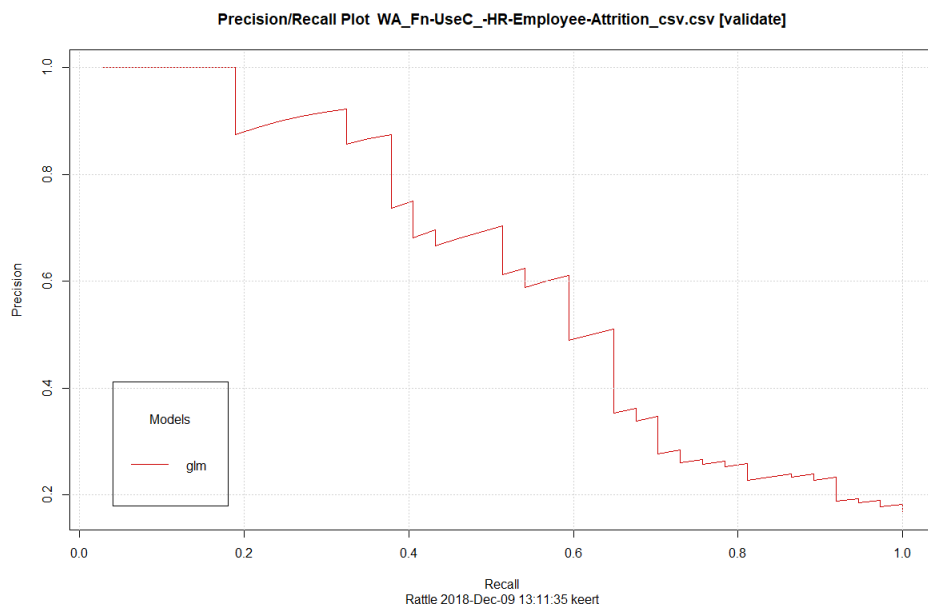


Figure 22

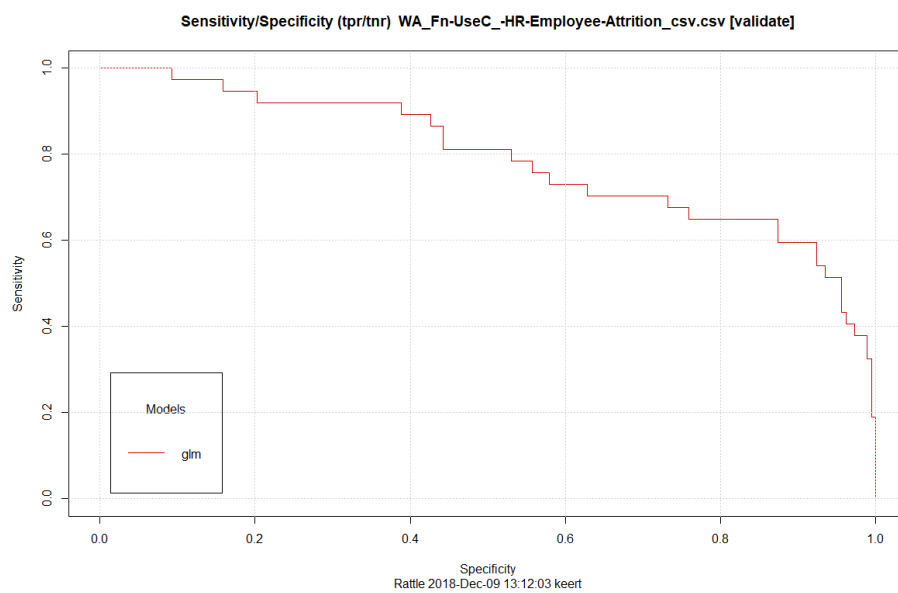


Figure 23

3.5 Support Vector Machines

Support vector machine is a classification method, which works by fitting a boundary to points that are similar. Setting a boundary is done on the train set, once this is done, we have to check if the points in the test set lie in this boundary. This is the basically how classification works in Support vector machines. The training data does not move once the boundary/hyperplane is set. As these points support the boundary as vectors.

The boundary varies with the number of attributes changes. That is when there are two attributes the boundary is a line or curve. The boundary is a complex irregular shape or a plane when it is a three dimensional. Since, it varies with the number of attributes considered, the boundary is generically called a hyperplane.

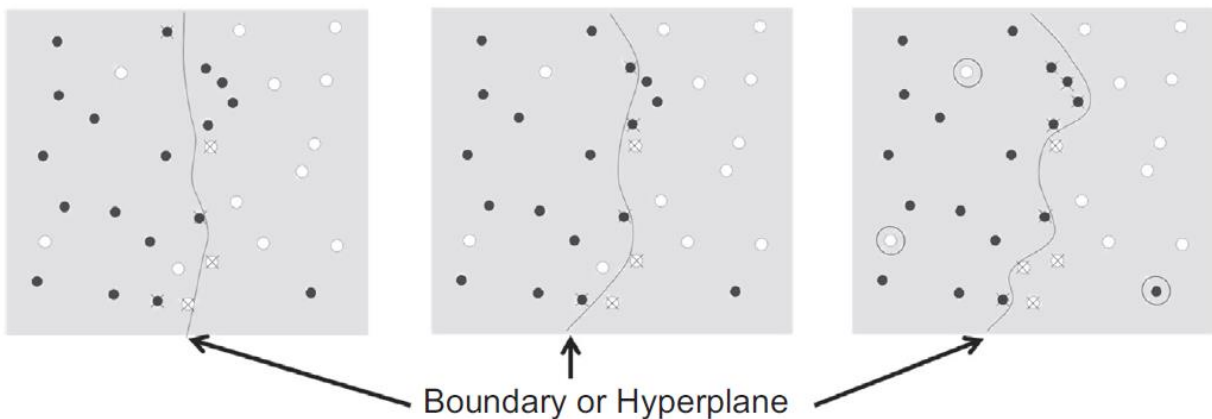


Figure 24

Let's look at the models, to which the algorithm is applied on three different set of data. If we are to pick the best one, it needs to be the third one. This is because, the misclassification error on the third one is zero as the boundary line has separated the classes perfectly, compared to the first and the second one.

After cleaning data, Support Vector Machine algorithm is applied on the dataset and the code is as below

```
svm_lin <- svm(Attrition~ ., data=trainset, method='C-classification', kernel='linear')
```

Support Vector Machines have four different kernels such as linear, polynomial, sigmoid and radial. Data Flair states that "Support Vector Machines use a set of mathematical functions that are defined as kernel" (Kernel Functions-Introduction to SVM Kernel & Examples, 2018). The kernel which is a function

take in the input data and produces an output in the desired form. Linear kernel is used for functions that are linear and the other kernels are used for non-linear relations.

Kernels that are most commonly used are Radial and Polynomial. It is best practice to try out all the basic models when the dataset is huge and work our way up to find the appropriate kernel. There are three tasks in the working of support vector machines. First is to find the boundary, next is to choose the best hyperplane and the third is to see where the test set data lies to go ahead with classification.

Every point in a class is connected. The boundary that results out of connecting the data point is called the hyperplane. The boundary is otherwise called a convex hull. This differs with every class as each one of them have their own convex hull. Since the classes are linear, they do not cross each other. New test example can be found once the hyperplane and the boundary are set. The data in the test set are applied to the hyperplane formula and if the result is +1 then it belongs to the positive class if the result is -1 then it belongs to the negative class.

A major disadvantage of support vector machines is the computational cost since the dot product for every classification. Though this is a disadvantage, the best thing about support vector machine is that once the model is set, even if there are small changes in the training dataset, the coefficients do not change unless there are changes in the support vector.

Advantages of Support Vector Machines are: (Vijay Kotu, Bala Deshpande Phd, 2015, Pg 64)

- “Flexibility is application
- Robustness
- Overfitting resistance”

Let's get into modeling the dataset in SVM with different kernels that is available. The function used is 'SVM' and it available in the package 'e1071'. Hence, the library 'e1071' has to be installed and included before modelling.

We have the data split up as trainset (80%) and testset (20%). The code to classify the data with linear kernel is as below. (Appendix E)

```
svm_lin <- svm(Attrition~ ., data=trainset, method='C-classification', kernel='linear')
```

```
svm_lin
```

We get the output as

```
Call:
svm(formula = Attrition ~ ., data = trainset, method = "C-classification",
     kernel = "linear")
```

```
Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  linear
    cost:    1
   gamma:   0.02222222
```

```
Number of Support Vectors:  346
```

Figure 25

The code to model the dataset with polynomial as kernel is as below

```
svm_pol <- svm(Attrition~ ., data=trainset, method='C-classification', kernel='polynomial')
```

```
svm_pol
```

and the output is as below

```
Call:
svm(formula = Attrition ~ ., data = trainset, method = "C-classification",
     kernel = "polynomial")
```

```
Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  polynomial
    cost:    1
   degree:   3
   gamma:   0.02222222
coef.0:     0
```

```
Number of Support Vectors:  563
```

Figure 26

The code to model the dataset with sigmoid function is as below

```
svm_sig <- svm(Attrition~ ., data=trainset, method='C-classification', kernel='sigmoid')
```

```
svm_sig
```

The output is as below

```
Call:
svm(formula = Attrition ~ ., data = trainset, method = "C-classification",
     kernel = "sigmoid")
```

```
Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: sigmoid
    cost:    1
   gamma:   0.02222222
coef.0:     0
```

```
Number of Support Vectors: 379
```

Figure 27

The code to model the dataset with Radial as the kernel is as below

```
svm_rad <- svm(Attrition~ ., data=trainset, method='C-classification', kernel='radial')
```

```
svm_rad
```

And the output is as below

```
Call:
svm(formula = Attrition ~ ., data = trainset, method = "C-classification",
     kernel = "radial")
```

```
Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: radial
    cost:    1
   gamma:   0.02222222
```

```
Number of Support Vectors: 474
```

Figure 28

Let's construct the confusion matrix for kernel function linear

Table 6

| n = 294 | predicted: no | predicted: yes | |
|-------------|---------------|----------------|-----|
| Actual: no | TN = 236 | FP = 34 | 270 |
| Actual: yes | FN = 16 | TP = 16 | 32 |
| | 242 | 50 | |

- **Accuracy:** $(TP+TN)/Total = 0.85714$, says how often the classifier is correct
- **Misclassification or Error Rate:** $(FP+FN)/Total = 0.17007$, says how often it is wrong.
- **Sensitivity:** $TP/actual\ yes = 0.5$, how often does it predict yes when it actually yes
- **False Positive Rate:** $FP/actual\ no = 0.12593$, how often does it predict yes when it actually is no
- **Specificity:** $TN/actual\ no = 0.87407$, how often does it predict no when it actually no.
- **Precision:** $TP/predicted\ yes = 0.32$, how often does it predict yes, and it is right.
- **Prevalence:** $Actual\ Yes/Total = 0.10884$, How often does the yes condition actually occurs

3.6 Monte Carlo Simulation

There are two types of randomized algorithms one is Las Vegas and Monte Carlo algorithms. The output of a Las Vegas Algorithm is always exact. It either gives the correct answer or it produces a report that it has failed. Such algorithms come with a cost, they either consume a large amount of memory or time. However, with Monte Carlo simulation, we get the results with a random amount of error. By including more time or memory into the simulations, the error can be reduced. Monte Carlo is able to give an approximate result with a given number of computations. (Ian Goodfellow, Yoshua Bengio, Aaron Courville, page 592)

“Monte Carlo simulation is a statistical technique that is used to model probabilistic systems and establish the odds of a variety of incomes” (Peter Dizikes, 17/05/2017). If we look back, Monte Carlo has been there since World War II. This simulation was used to study nuclear fission. Monte Carlo is the type of simulation which generates samples randomly. The reason for choosing Monte Carlo simulation is to increase the randomness of the trainset and testset split and calculate their accuracies.

Monte Carlo is also used in risk analysis by an Environment Protection Agency. For example, if we are to calculate the health risk involved with smog in a city. The smog levels among different places and neighborhood changes. The time that people spend outside also changes from person to person. A Monte Carlo simulation runs in such a way that it chooses a random set of numbers from a given range of values. We will not find two iterations similar in a Monte Carlo simulation. In the long run, the simulation would give a realistic picture of the computation.

3.6.1 Libraries necessary

To run Monte Carlo simulations, there are certain libraries to be installed. One is “montecarlo” and the other is “snowfall”. Monte Carlo function has an argument “ncpu”, which works only when the library “snowfall” is used. The argument “ncpu” is used to increase the core processor capacity so that the iteration can happen even more faster. For huge number of iterations, we can increase “ncpu”. However, we have to make sure that the device that we are working on is able to handle this.


3.6.2 Arguments in Monte Carlo


While coding first we will define the function as it is user defined. The function will contain the method in which the input data are to be computed. In this dissertation it is the different machine learning algorithms. Once the function is defined, the first argument in “montecarlo” will be “func = ‘user defined function’”. There is another argument to be added to Monte Carlo simulation called the “param_list”. There are two things to be defined in “param_list”, which contains a list of grid values. Since we are going to work on five trainset percentages, we will have to define what “n” is. Then we have to include this to param_list. Then we have a model grid to be included, where we have the algorithms’ name mentioned and this also is added to the param_list. The most important argument that defines how many iterations is “nrep”.


With the help of the function summary we will be able to see the summary of the user defined function. Another important code to view the results is “MakeTable”. The output of the Monte Carlo simulation is produced in the form of tex, to view the output in a PDF format we need a tex file viewer. In order to plot the output, we have to convert the output into a data frame and then we can plot them with the ggplot.

3.7 Working on RapidMiner

Start a new project

**Blank**
Start a new process from scratch in the design view.

**Turbo Prep**
Prepare your data interactively: transform, clean and combine data sets.

**Auto Model**
Build and optimize models using automated machine learning.

**Churn Modeling**
Predict which of your customers will churn and why with a decision tree.

**Direct Marketing**
Predict response to campaigns and increase the conversion rate of your campaign.

**Credit Risk Modeling**
Model credit default risk by training an optimized Support Vector Machine (SVM) model.

**Market Basket Analysis**
Find products frequently purchased together and turn them into rules for recommendations.

**Predictive Maintenance**
Model equipment failures to schedule maintenance pre-emptively

**Price Risk Clustering**
Cluster price developments using X-Means to unveil price-risk-relationships.

**Lift Chart**
Create a lift chart to visualize the improvement that a model provides compared to guessing.

**Operationalization**
Embed predictive models into business processes to trigger the right actions automatically.

**Outlier Detection**
Detect anomalies in data resulting from a chemical analysis of wines.

Figure 29

The first screen that appears in RapidMiner is the same as the Figure 29. We get to choose a new project or a template to start with. Click on Auto Model, which will lead to the next page as Figure 30.

Load Data

Select Task

Prepare Target

Select Inputs

Model Types

Results

« RESTART < BACK > NEXT

▶ Training Resources (connected)

▶ Samples

▶ Community Samples (connected)

▶ Time Series Extension Samples

▶ DB

▶ Local Repository (keert)

▶ Cloud Repository (disconnected)

IMPORT NEW DATA

Information
Please select a data set from your repositories.

Figure 30

Figure 30 shows the page where data is loaded. We can choose to store the dataset file in the local repository or we can choose to directly go to the next step. If the file is available in the repository, we just have to choose it or else we can click on the IMPORT NEW DATA button which go to the next screen as in Figure 31.

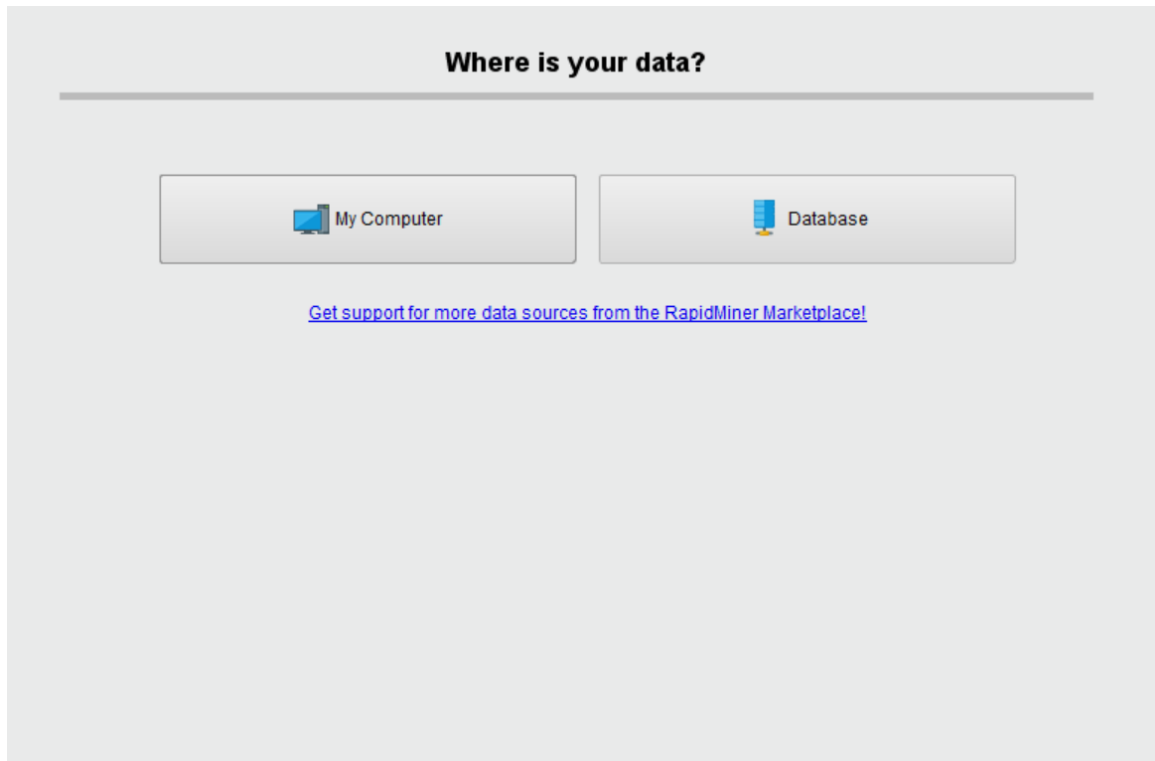


Figure 31

If the dataset is on the computer, the option My Computer can be selected. We can also import from databases, by choosing the Database tab. Once the dataset is chosen, we get the next page as in the Figure 32. Here we can choose if we want to predict, cluster or detect the outliers. We are going to predict Attrition rate, hence choose Predict and select the column Attrition.

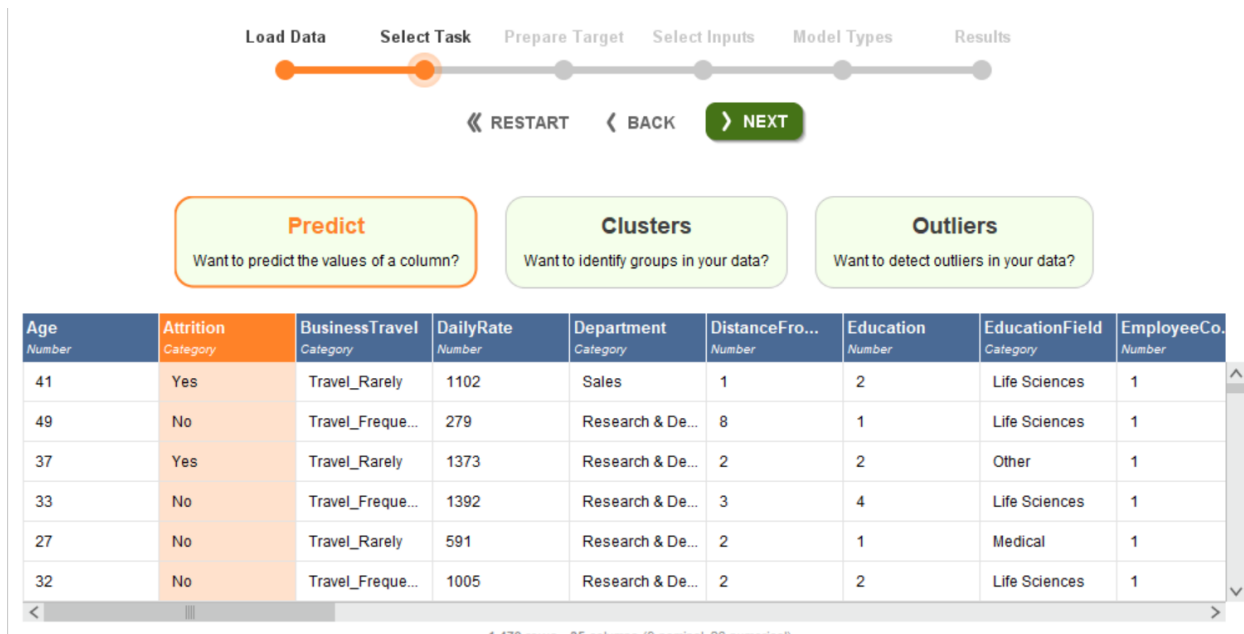


Figure 32

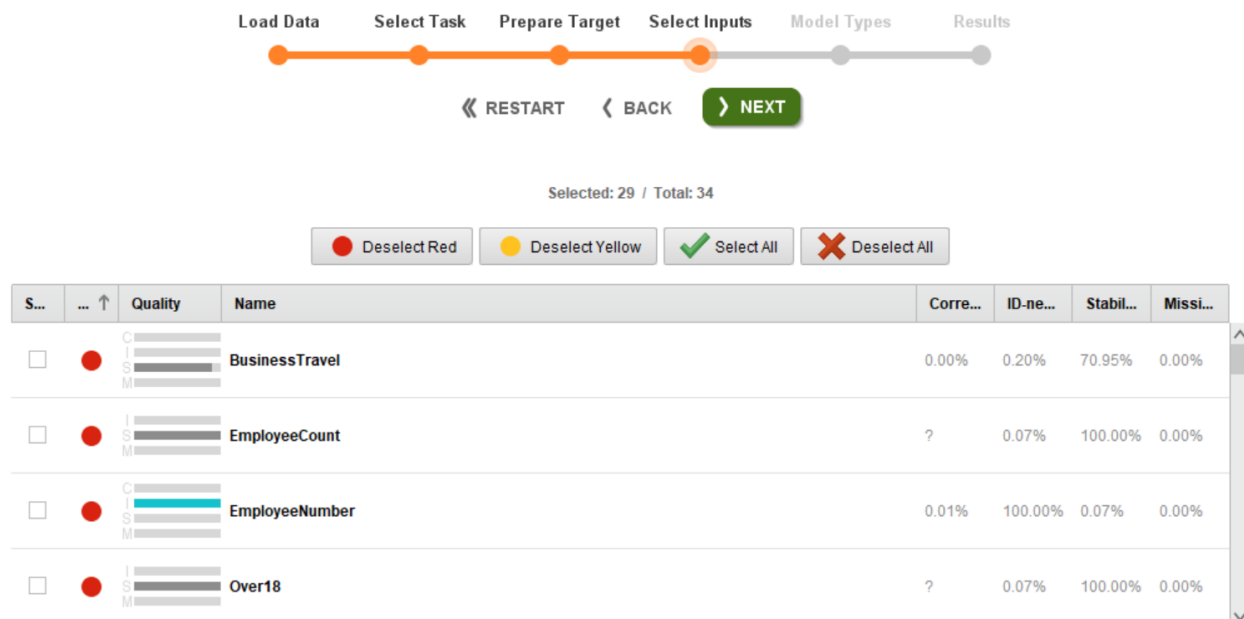


Figure 33

Figure 33 shows the window where the input variables or the attributes can be selected. RapidMiner not only selects the variables but it also tells us which the most important input variable and the least important ones are. Variables with a red dot means that they are not important, yellow dots means in

between important and not very important. Finally, the green tick says that the input variable is the most important one.

The figure shows a workflow diagram at the top with steps: Load Data, Select Task, Prepare Target, Select Inputs, Model Types, and Results. The 'Model Types' step is highlighted with an orange circle. Below the diagram are navigation buttons: 'RESTART', 'BACK', and 'RUN'.

General

- ☒ Correlations between Columns
- ☒ Importance of Columns

Models

- ☒ Naive Bayes
- ☒ Generalized Linear Model (GLM)
 - ☒ Use Regularization
 - ☐ Calculate p-Values
- ☒ Logistic Regression
- ☒ Deep Learning
- ☒ Decision Tree
 - ☒ Automatically Optimize
 - Maximal Depth: 20

Figure 34

In Figure 34, the various models are displayed, to which there are options to optimize. To the left-hand side there is a General tab, where we can choose various options like Correlation between Columns and importance of each columns.

The figure shows a workflow diagram at the top with steps: Load Data, Select Task, Prepare Target, Select Inputs, Model Types, and Results. The 'Model Types' step is highlighted with an orange circle. Below the diagram are navigation buttons: 'RESTART', 'BACK', and 'RUN'.

Logistic Regression

Deep Learning

Decision Tree

- ☒ Automatically Optimize
- Maximal Depth: 20

Random Forest

- ☒ Automatically Optimize
- Number of Trees: 20
- Maximal Depth: 20

Gradient Boosted Trees (XGBoost)

- ☒ Automatically Optimize
- Number of Trees: 20
- Maximal Depth: 20

Figure 35

Figure 35 shows rest of the models and the options that is available to optimize. When we are done at this stage, we can click on the Run button to run the auto model with the options chosen.

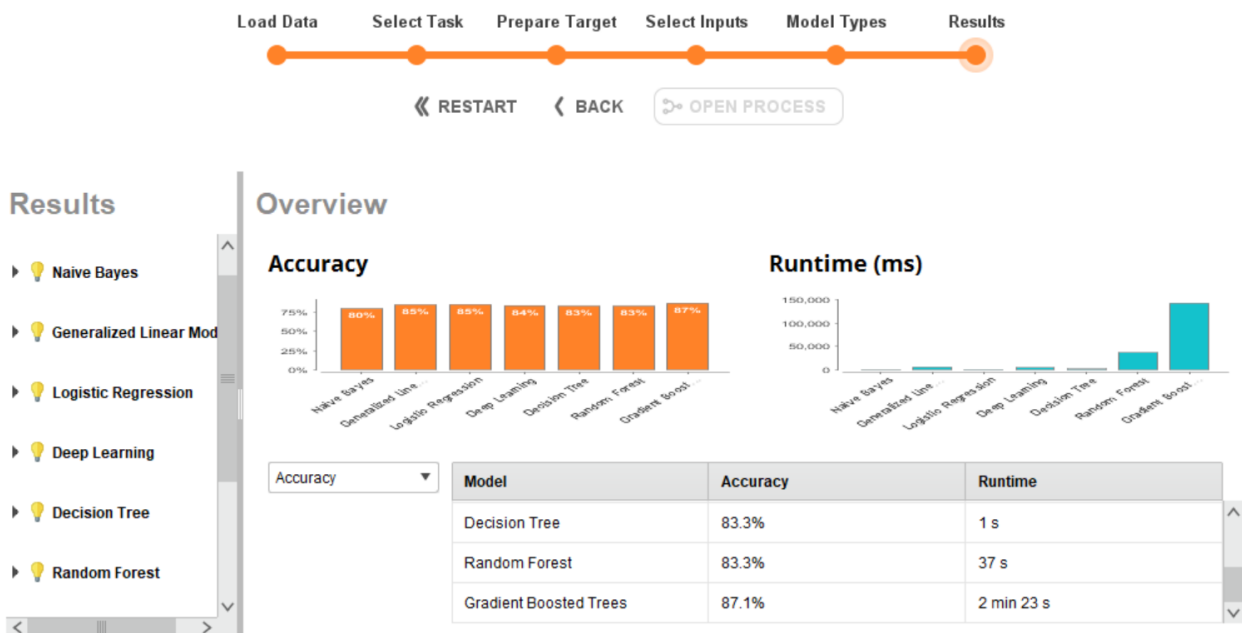


Figure 36

Figure 36 shows the final window, where the results are available. It gives an overview of terms such as accuracy, classification error, AUC, precision, recall, f-measure, sensitivity and specificity. We get to see the overview of all these measures as data and as well as on the plot. The Results tab has all the models listed. We can see the overview of each of them by selecting the algorithm.

The option comparison under Results gives an overall comparison of the models. There are two options under Comparison, Overview and ROC comparison. A ROC comparison is done on a plot like the Figure

37, which is below.

ROC Comparison

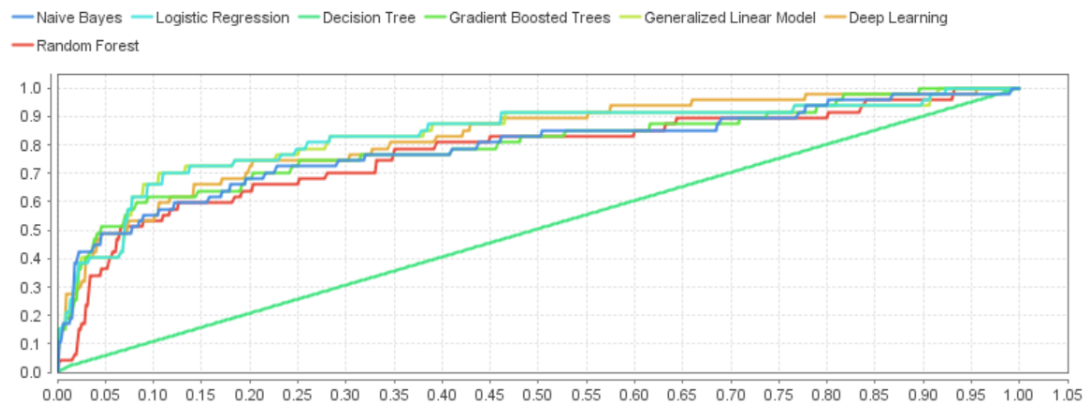


Figure 37

Chapter 4 Data Analysis and Findings

The dataset chosen for dissertation is based on “Employee Attrition” from SAMPLE DATA: HR Employee Attrition and Performance from IBM website (SAMPLE DATA: HR Employee Attrition and Performance, McKinley Stacker IV, 2015)(Appendix A). The dataset has about 1470 rows and 35 attributes. There are various attributes in the dataset such as the Age, Employee number, Education field and all the attributes that we would usually find in an employee database. The dataset has the column Attrition with values Yes and No, in which Yes is for people who have left the organization and No is for people who are still in the organization.

Here are some of the visualization of the dataset. (Appendix F)

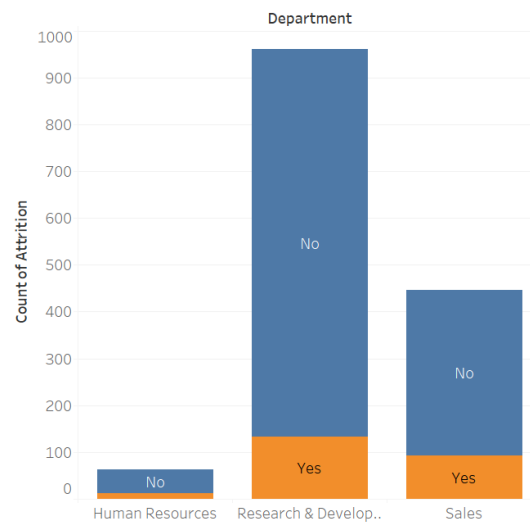


Figure 38

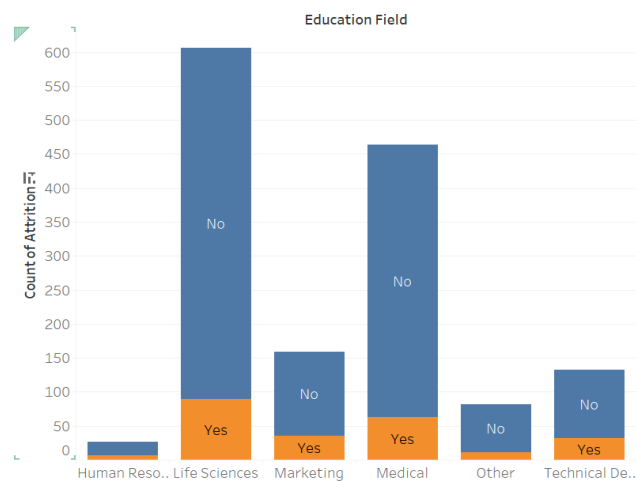


Figure 39

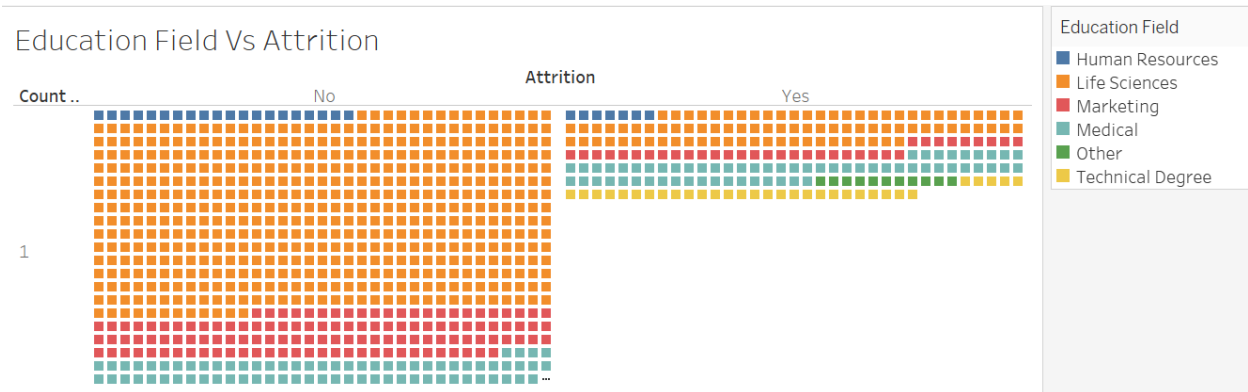


Figure 40

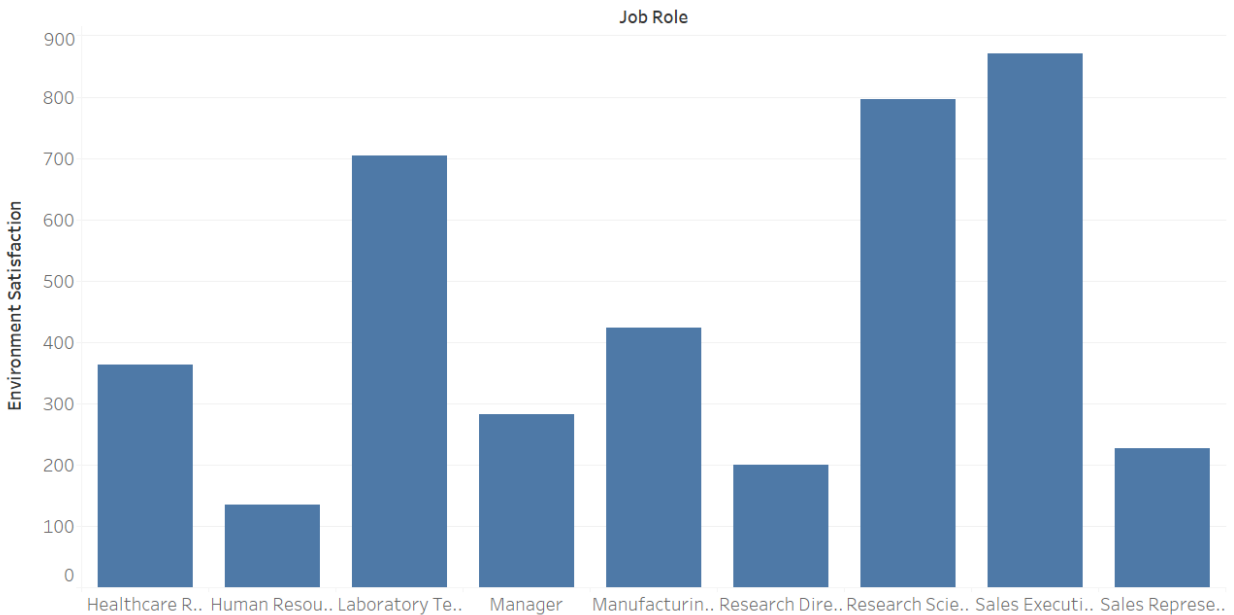


Figure 41

The dataset is a combination of discrete and continuous values. The attributes that are categorical are Attrition, Business Travel, Department, Education Field, Gender, Job Role, Marital Status and Overtime. The attributes that are numerical are Age, Daily Rate, Distance from home, Education, Employee count, Employee number, Environment Satisfaction, Hourly Rate, Job involvement, Job level, Job Satisfaction, Monthly Income, Monthly Rate, Number of companies worked, Percent Salary Hike, Performance rating, Relationship Satisfaction, Standard hours, Stock Option Level, Total working years, Training time last year, Work life balance, Years at company, Years in current role, Years since last promotion and Years

with current manager. Attributes like Over 18, Standard Hours, Employee count and Employee number. Except employee number all other attributes have the same value throughout. Employee number is just an identification number and does not have much to do with other attributes.

Business Travel has three categories, Travel_Rarely, Travel_Frequently and Non-Travel. The number of employees under the category Travel_Frequently is the most which is followed by Travel_Rarely and Non-Travel.

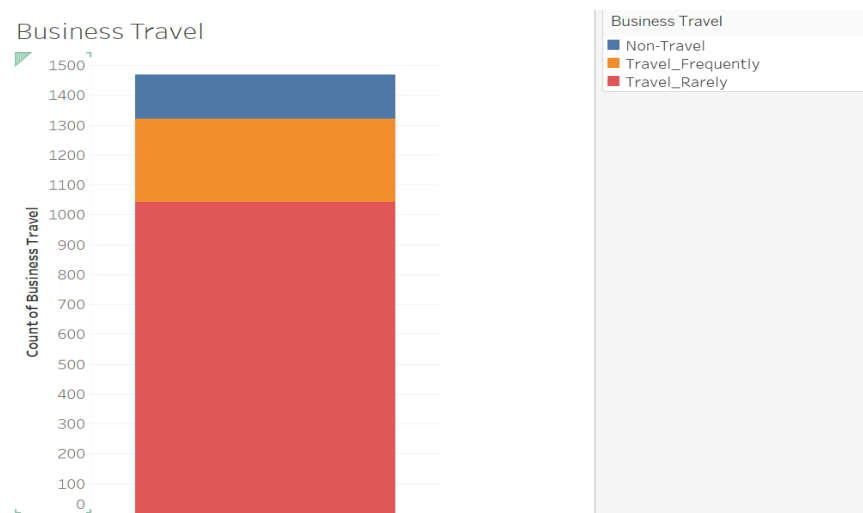


Figure 42

Department is an attribute which is again a categorical value and has three categories Sales, Research & Development and Human resources. The count for Research & Development is more which is followed by Sales and Huma Resource.

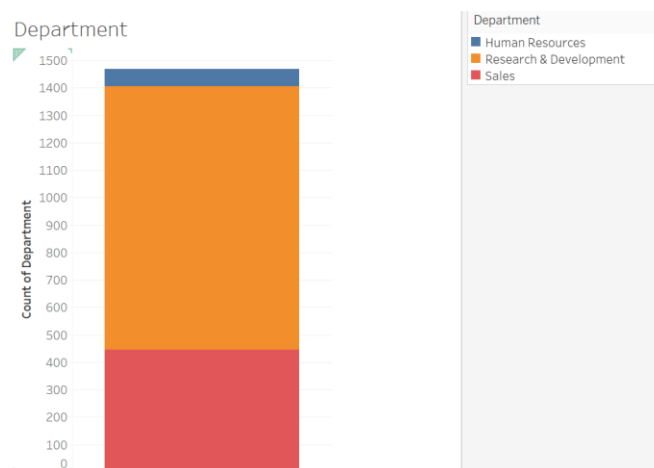


Figure 43

Let's look at the Education Field Attribute, which has categories such as Human Resources, Life Sciences, Marketing, Medical, Technical and others.

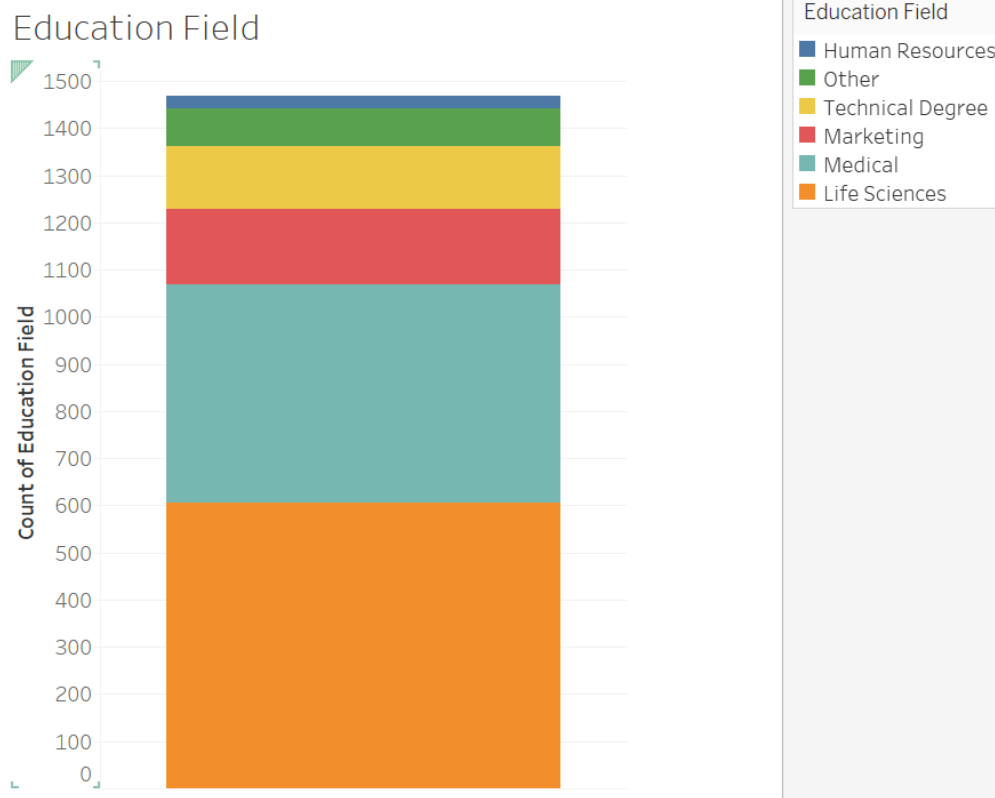


Figure 44

The next visualization is on the Male Female ratio.

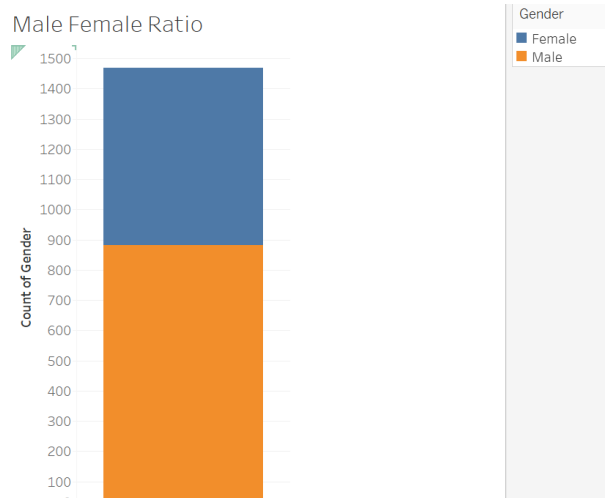


Figure 45

The next attribute is the Job Role which has categories such as Healthcare Representative, Human Resources, Laboratory Technician, Manager, Manufacturing Director, Research Director, Research Scientist, Sales Executive and Sales Representative. There are about 326 Sales Executive, 292 Research Scientist, 259 Laboratory technician, 145 Manufacturing Director, 131 Healthcare Representative, 102 Managers, 82 Sales Representatives, 100 Research Directors and 52 Human Resource.

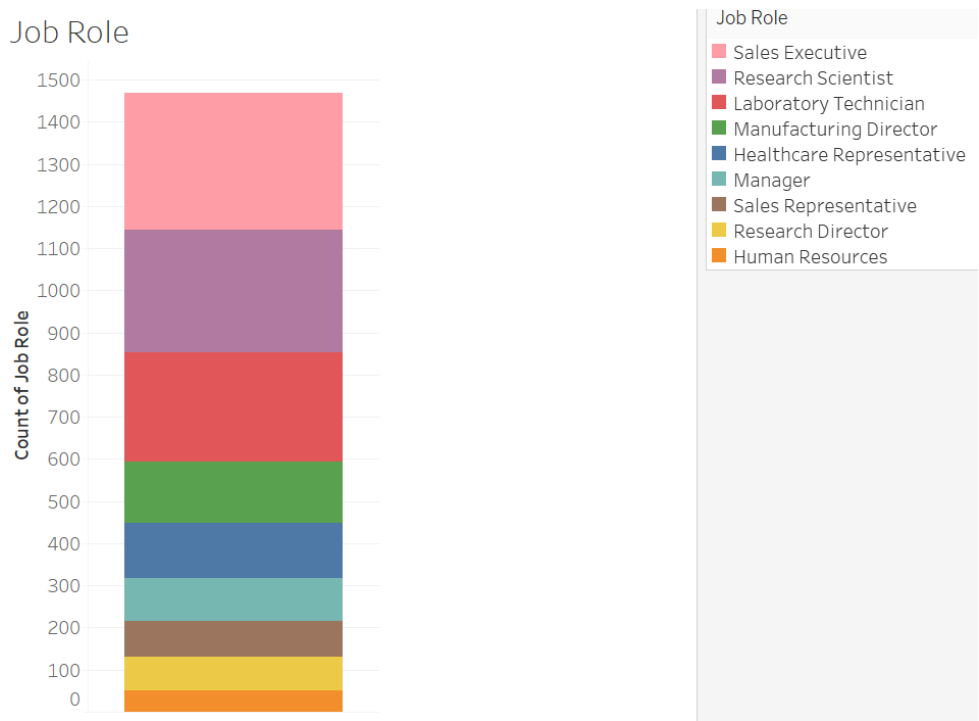


Figure 46

There are three categories in Marital Status which are Married, single and divorced. Most of the employees are married, followed by a lesser number of singles and divorced individuals.

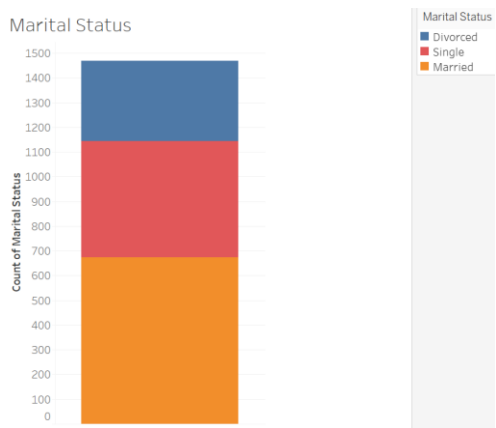


Figure 47

The visualization below tells us the amount of people working overtime and the ones that are not working overtime.

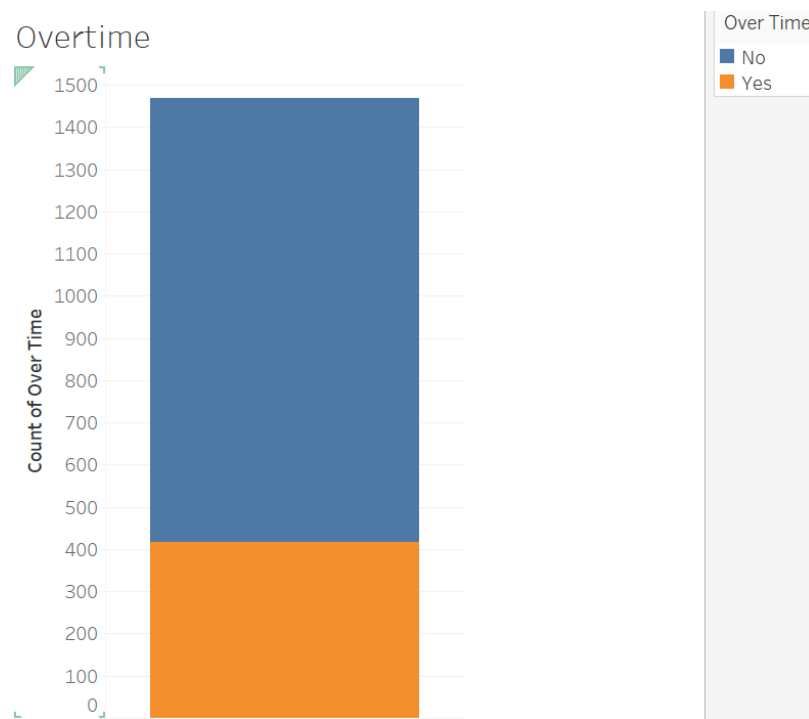


Figure 48

Let's look at the relationship between Attrition and Overtime. There is no linear relationship between Attrition and overtime that we can point out as Overtime a reason for Attrition.

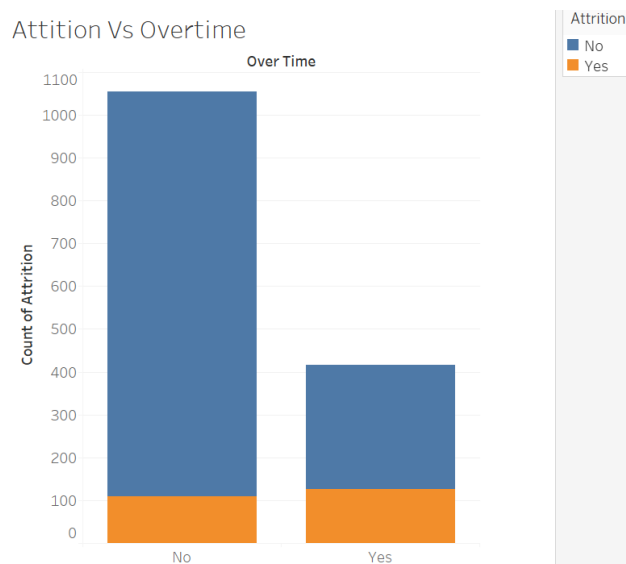


Figure 49

Chapter 5 Discussion

5.1 Comparison with Monte Carlo Simulation

Monte Carlo simulations were run on five sets of repetition such as 5, 10, 50, 100 and 200. The accuracy for models Naïve Bayes, Decision Tree, Random Forest and Support vector machines were calculated and plotted. The percentage of trainset used were 50, 60, 70, 80 and 90. We are going to look at the accuracy plotted for the said repetitions.

5.2 Monte Carlo Simulations With Different Iterations (Appendix G)

5.2.1 Monte Carlo with nrep = 5

The plots have the spread of accuracy on all five percentages of trainset ratio, run for 5 repetition. Here DT means Decision Tree, NB means Naïve Bayes, RF means Random Forest, SVM_lin means Support Vector Machine with kernel Linear and SVM_Pol means SVM algorithm with kernel and Polynomial.

Table 7

| Trainset % | 50% | 60% | 70% | 80% | 90% |
|---------------|--------|--------|--------|--------|--------|
| Decision Tree | 0.8302 | 0.8438 | 0.8295 | 0.8381 | 0.8231 |
| Random Forest | 0.8512 | 0.8602 | 0.8553 | 0.8523 | 0.8571 |
| SVM – Linear | 0.8795 | 0.8745 | 0.8825 | 0.8796 | 0.8966 |
| SVM – Polygon | 0.8319 | 0.8336 | 0.8349 | 0.8551 | 0.8286 |
| Naïve Bayes | 0.7951 | 0.7956 | 0.7583 | 0.7748 | 0.7769 |

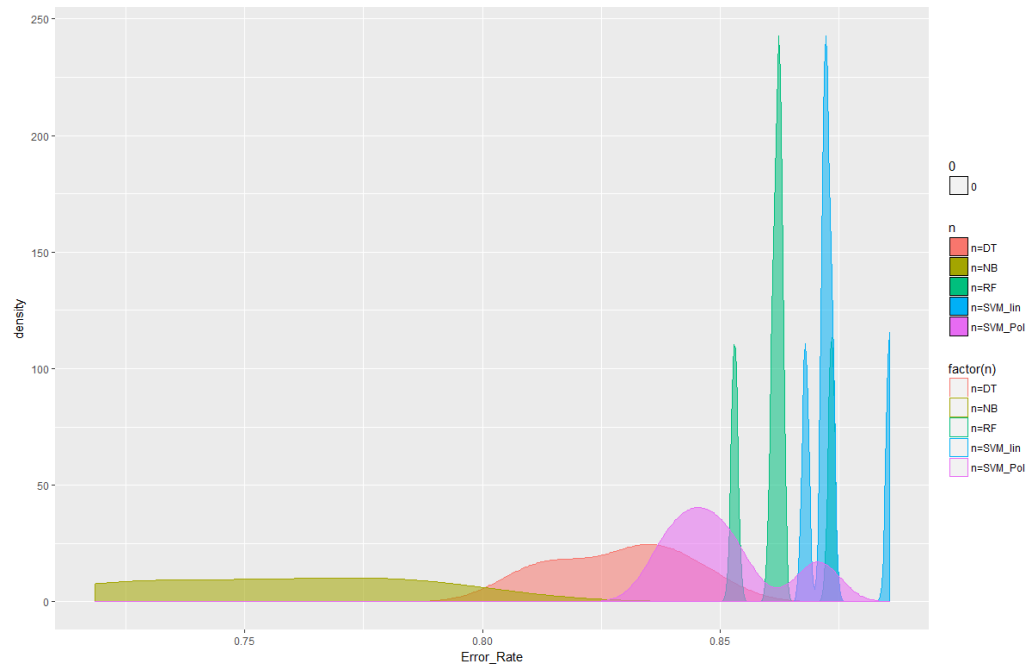


Figure 50

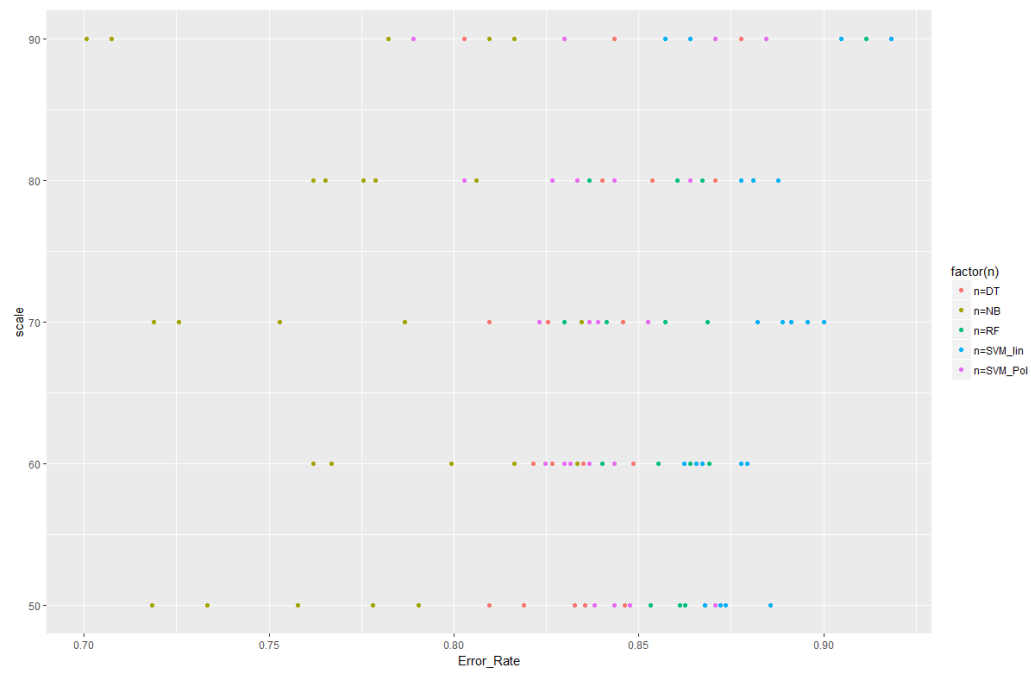


Figure 51

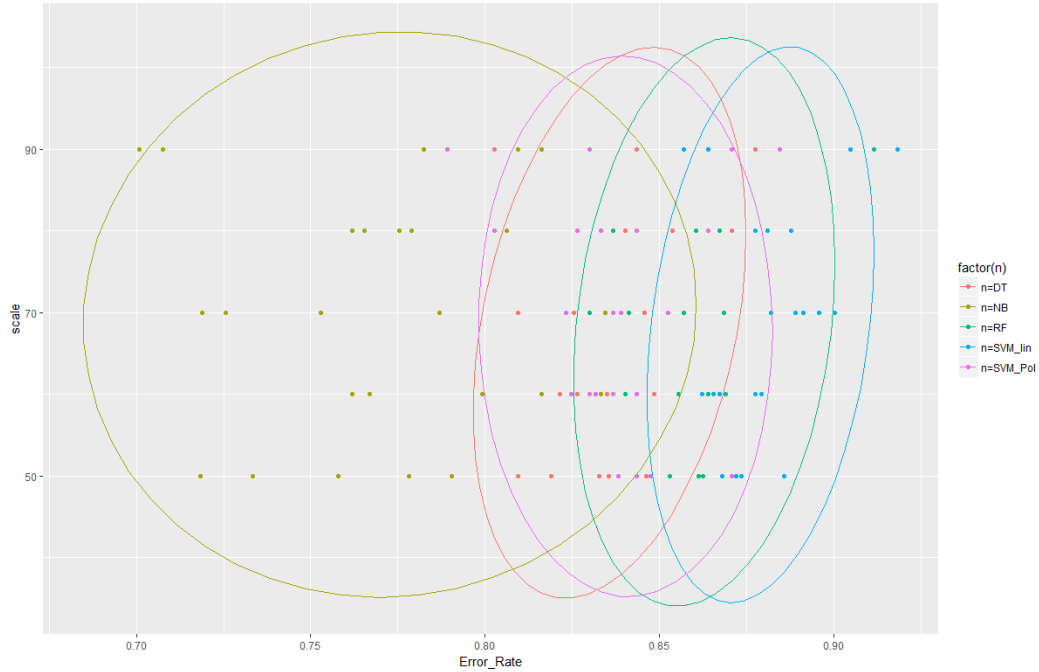


Figure 52

The accuracy for Naïve Bayes ranges from 0.65 to 0.85, while the accuracy range for Decision Tree is from 0.79 to 0.87. The accuracy range for SVM_Poly is from 0.79 to 0.87, the accuracy range for Random Forest is from 0.83 to 0.87 and finally the accuracy range for SVM_Lin is from 0.87 to more than 0.9. Looking at this plot, we can say that the Support Vector Machines with Linear Kernel is the optimum algorithm. The time taken for Monte Carlo runs with 5 repetition was 1 minute 23 seconds.

5.2.2 Monte Carlo Simulation with nrep = 10

Monte Carlo simulation was run with 10 repetitions and the plot is as below. The accuracy range is almost similar to the previous plot. The time take for Monte Carlo runs with 10 repetitions was 2 minutes.

Table 8

| Trainset % | 50% | 60% | 70% | 80% | 90% |
|---------------|--------|--------|--------|--------|--------|
| Decision Tree | 0.8299 | 0.8321 | 0.8395 | 0.8408 | 0.8510 |
| Random Forest | 0.8571 | 0.8568 | 0.8596 | 0.8674 | 0.8503 |
| SVM – Linear | 0.8739 | 0.8729 | 0.8846 | 0.8830 | 0.8762 |
| SVM – Polygon | 0.8372 | 0.8367 | 0.8386 | 0.8353 | 0.8449 |

| | | | | | |
|-------------|--------|--------|--------|--------|--------|
| Naïve Bayes | 0.7789 | 0.7983 | 0.7723 | 0.8061 | 0.7707 |
|-------------|--------|--------|--------|--------|--------|

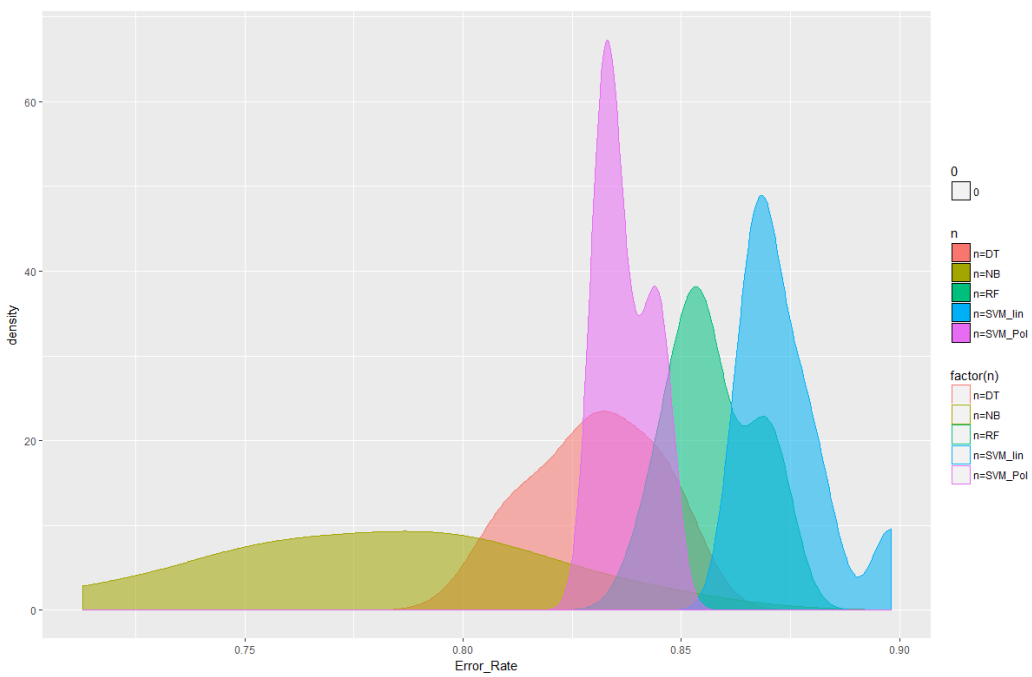


Figure 53

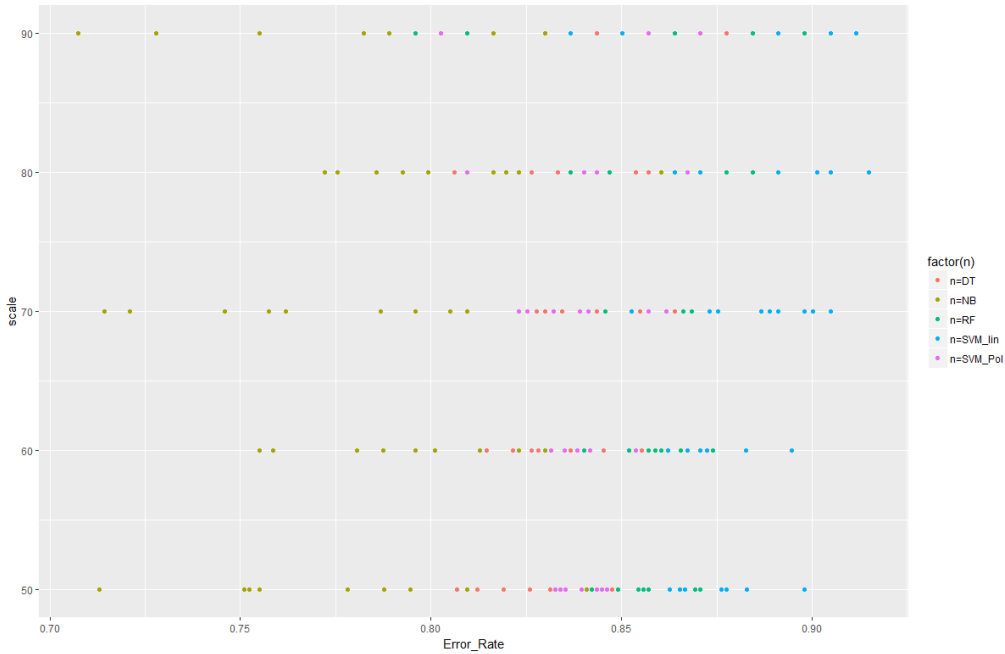


Figure 54

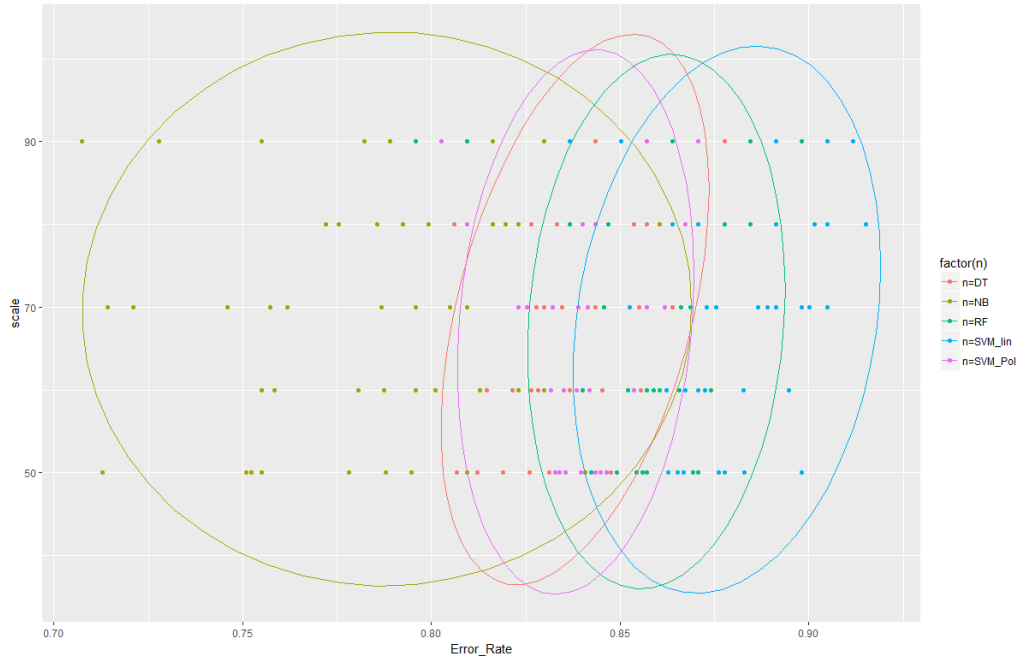


Figure 55

5.2.3 Monte Carlo Simulation with nrep = 50

Monte Carlo simulation with 50 repetition took about 10 minutes to complete and the accuracy range is again similar here, but we can see that the peaks are sharper now. The plot is as below

Table 9

| Trainset % | 50% | 60% | 70% | 80% | 90% |
|---------------|--------|--------|--------|--------|--------|
| Decision Tree | 0.8332 | 0.8320 | 0.8447 | 0.8400 | 0.8487 |
| Random Forest | 0.8546 | 0.8572 | 0.8589 | 0.8611 | 0.8573 |
| SVM – Linear | 0.8755 | 0.8771 | 0.8808 | 0.8840 | 0.8814 |
| SVM – Polygon | 0.8402 | 0.8397 | 0.8370 | 0.8349 | 0.8515 |
| Naïve Bayes | 0.7958 | 0.7874 | 0.7814 | 0.7790 | 0.7745 |

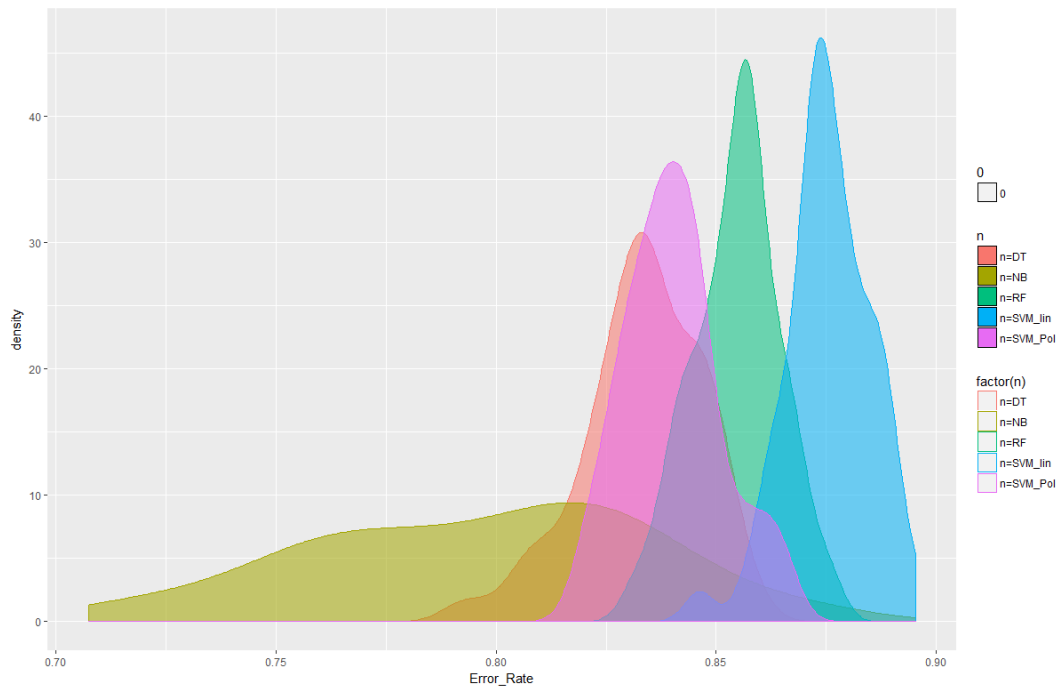


Figure 56



Figure 57

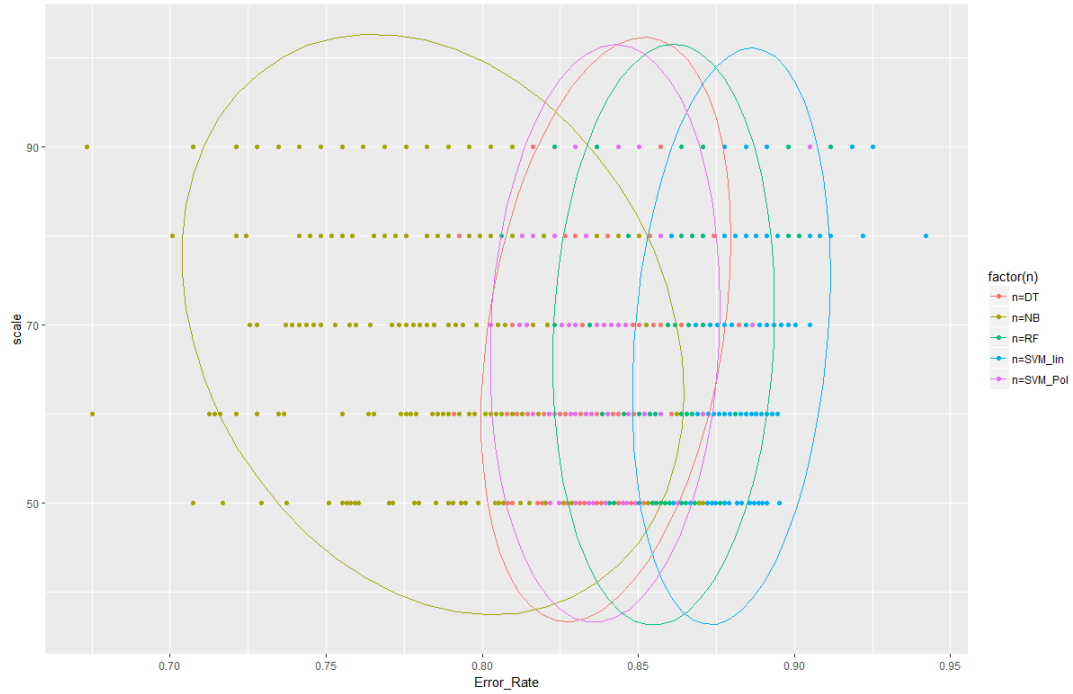


Figure 58

5.2.4 Monte Carlo Simulation with nrep = 100

Table 10

| Trainset % | 50% | 60% | 70% | 80% | 90% |
|---------------|--------|--------|--------|--------|--------|
| Decision Tree | 0.8317 | 0.8351 | 0.8359 | 0.8390 | 0.8417 |
| Random Forest | 0.8551 | 0.8572 | 0.8596 | 0.8580 | 0.8623 |
| SVM – Linear | 0.8750 | 0.8785 | 0.8817 | 0.8803 | 0.8830 |
| SVM – Polygon | 0.8395 | 0.8405 | 0.8391 | 0.8383 | 0.8407 |
| Naïve Bayes | 0.7686 | 0.7796 | 0.7781 | 0.7815 | 0.7813 |

Next Monte Carlo simulation was run for 100 repetitions and time taken was 20 minutes. In this plot we see some higher peaks with the range similar.

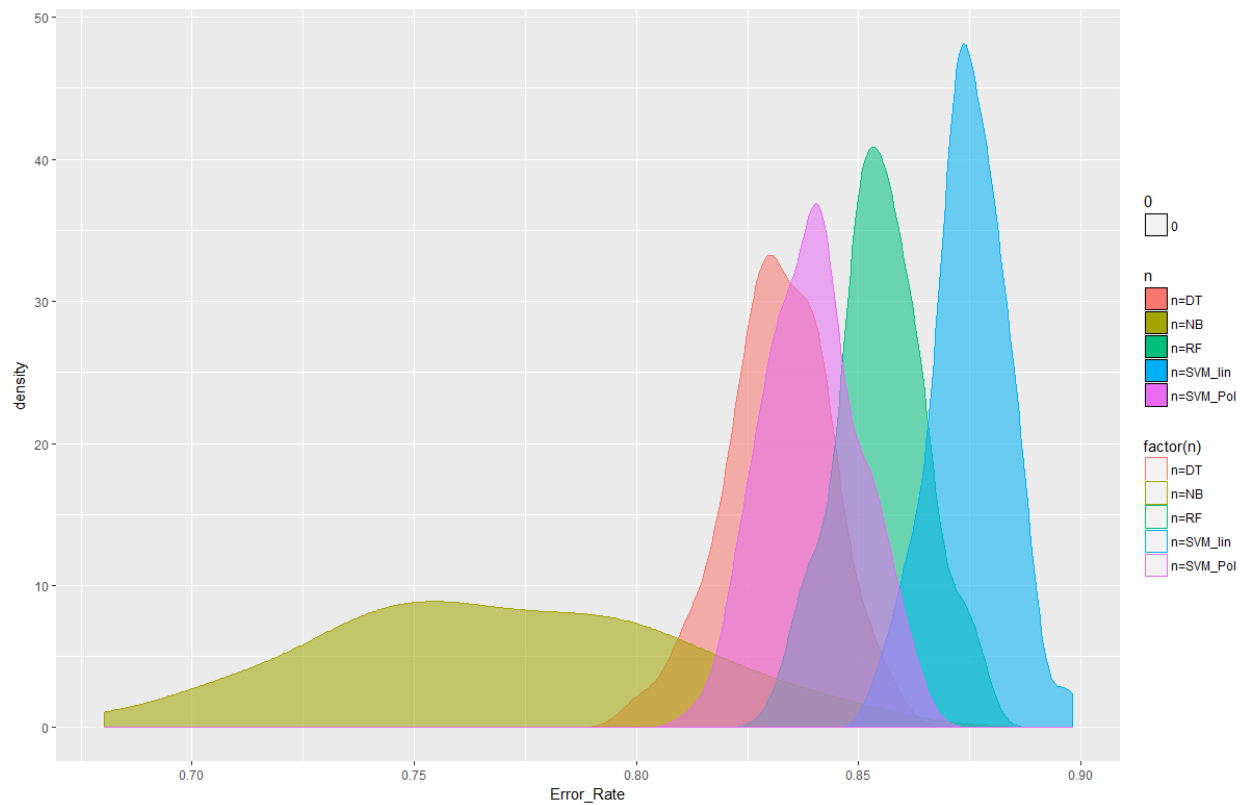


Figure 59

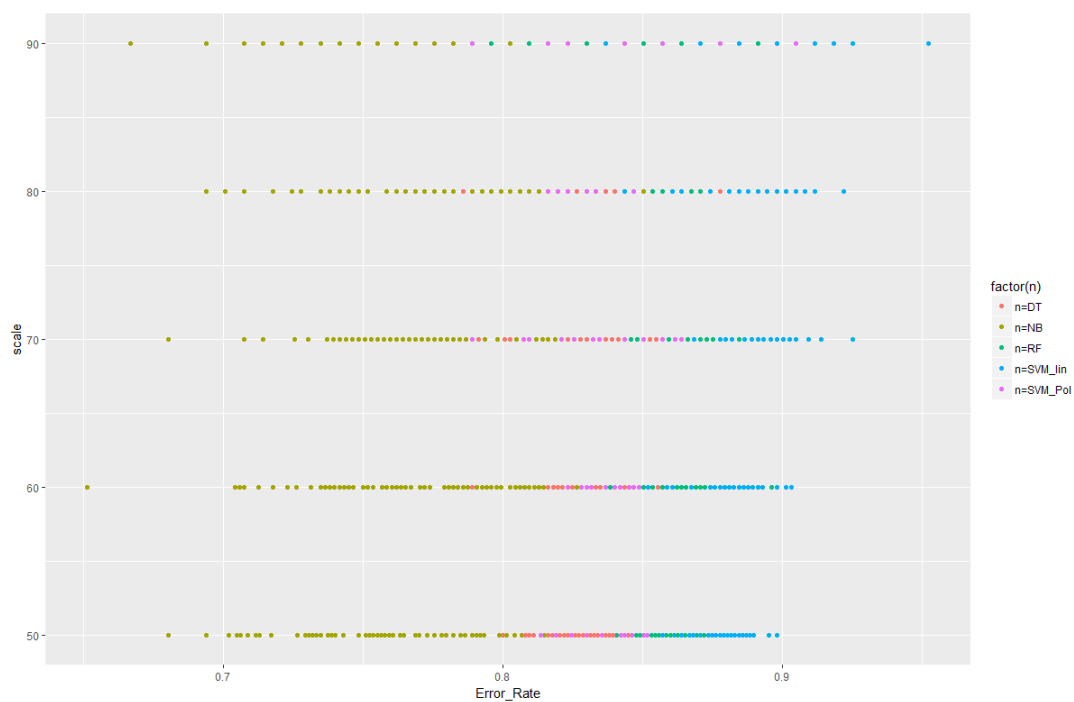


Figure 60

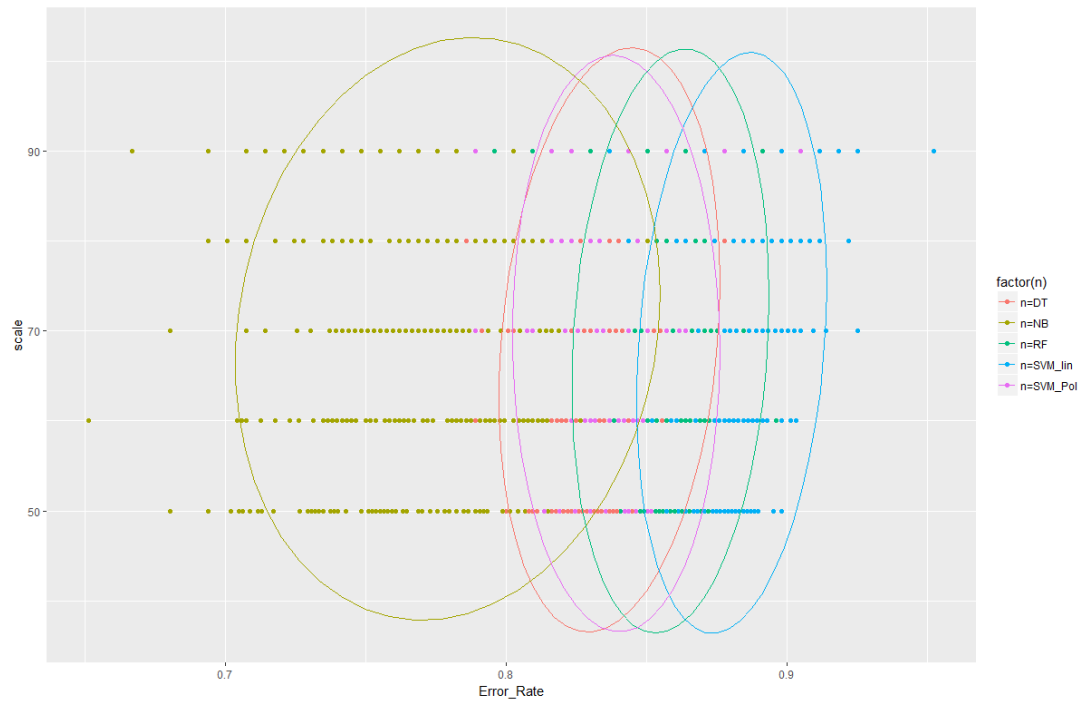


Figure 61

5.2.5 Monte Carlo Simulation with nrep = 200

Monte Carlo simulation with 200 repetition takes about 48 minutes and the plot is as below. The curves are getting sharper as the number of repetition increases.

Table 11

| Trainset % | 50% | 60% | 70% | 80% | 90% |
|---------------|--------|--------|--------|--------|--------|
| Decision Tree | 0.8323 | 0.8368 | 0.8402 | 0.8451 | 0.8436 |
| Random Forest | 0.8559 | 0.8565 | 0.8572 | 0.8584 | 0.8605 |
| SVM – Linear | 0.8746 | 0.8782 | 0.8795 | 0.8821 | 0.8828 |
| SVM – Polygon | 0.8393 | 0.8374 | 0.8416 | 0.8409 | 0.8411 |
| Naïve Bayes | 0.7762 | 0.7772 | 0.7795 | 0.7809 | 0.7795 |

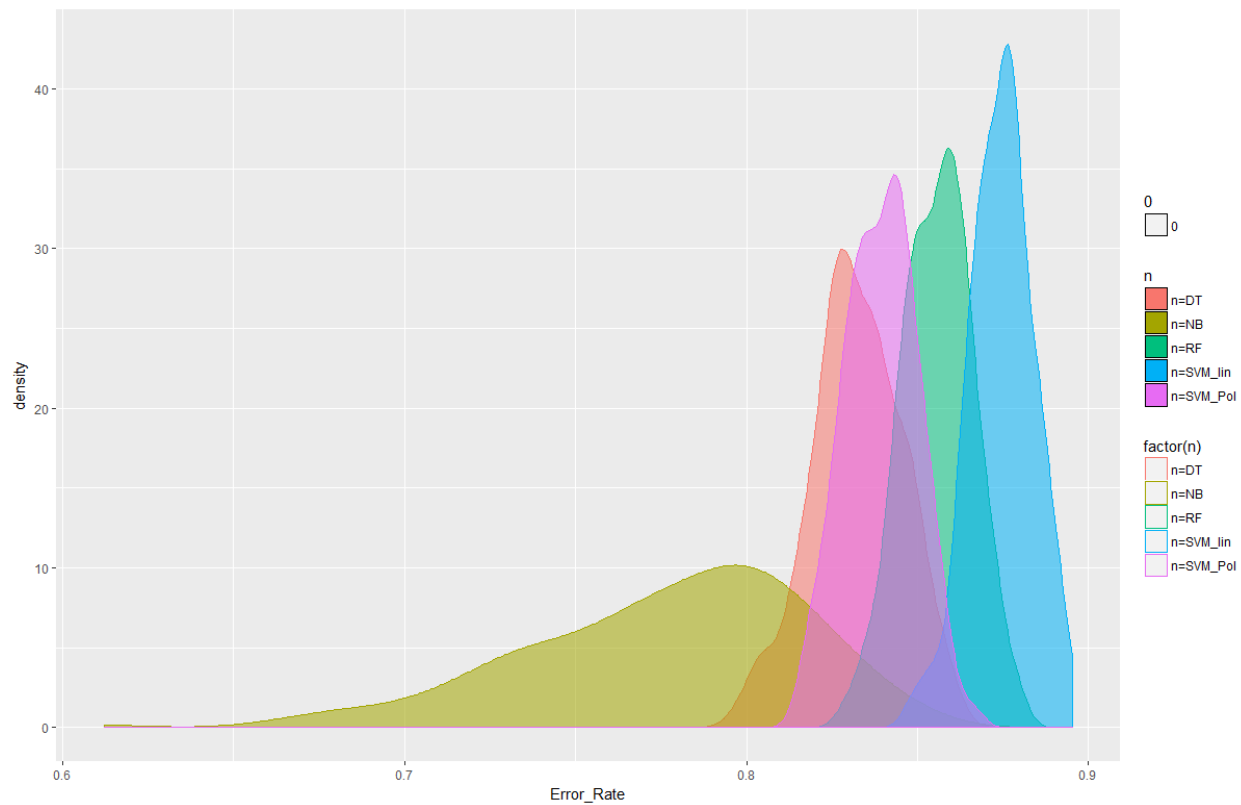


Figure 62

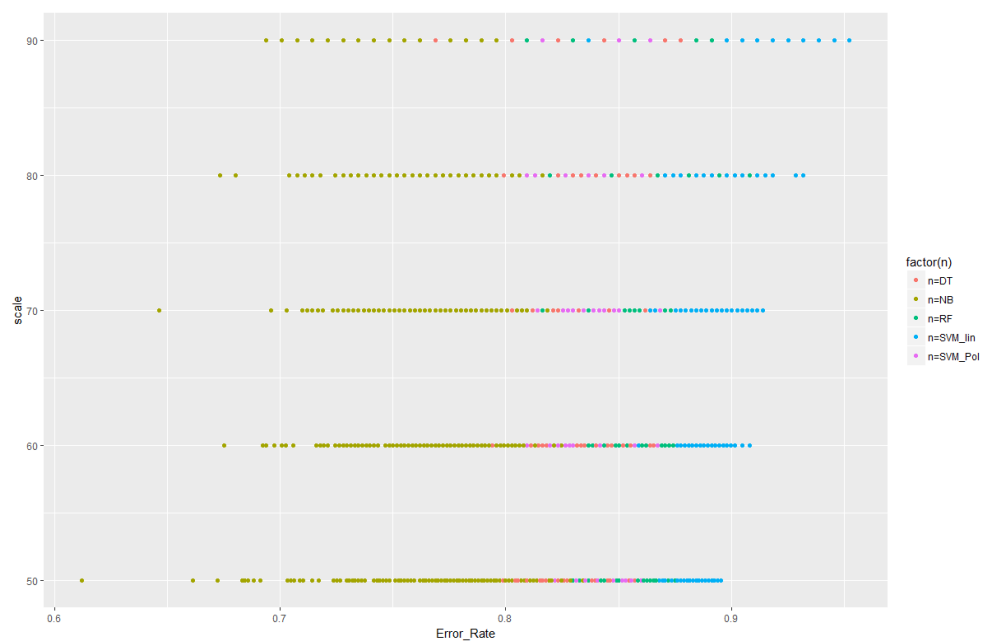


Figure 63

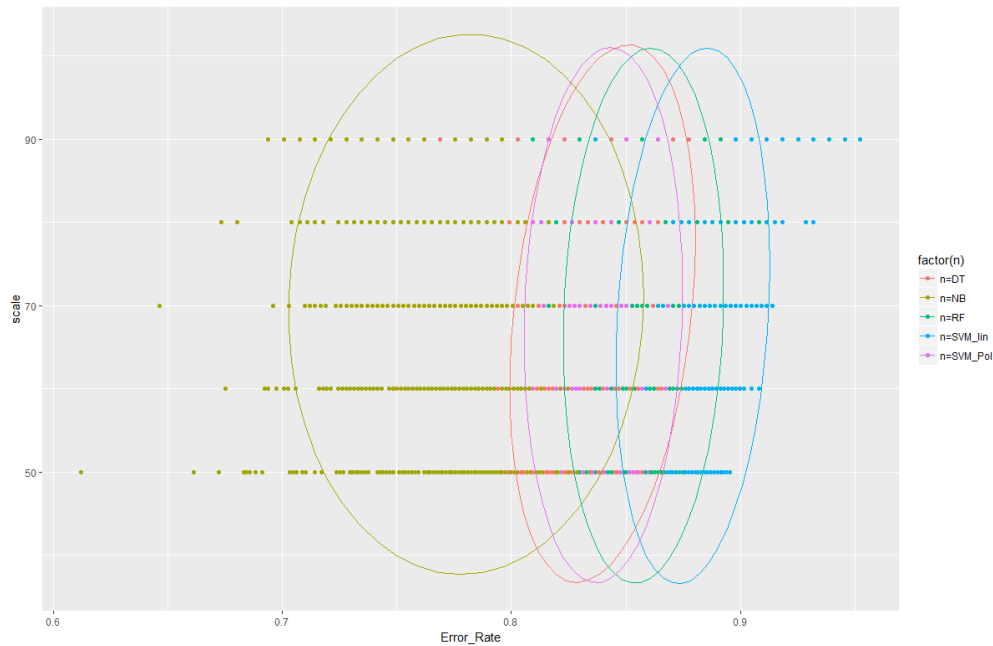


Figure 64

From all the simulations we can say that Support Vector Machine makes a better algorithm with a higher accuracy.

5.3 Comparison with RapidMiner

To know which a better model is, we just have to find the curve that is closest to the upper left corner. From this plot we can say that Deep Learning and Gradient Boosted Trees are better models.

ROC Comparison

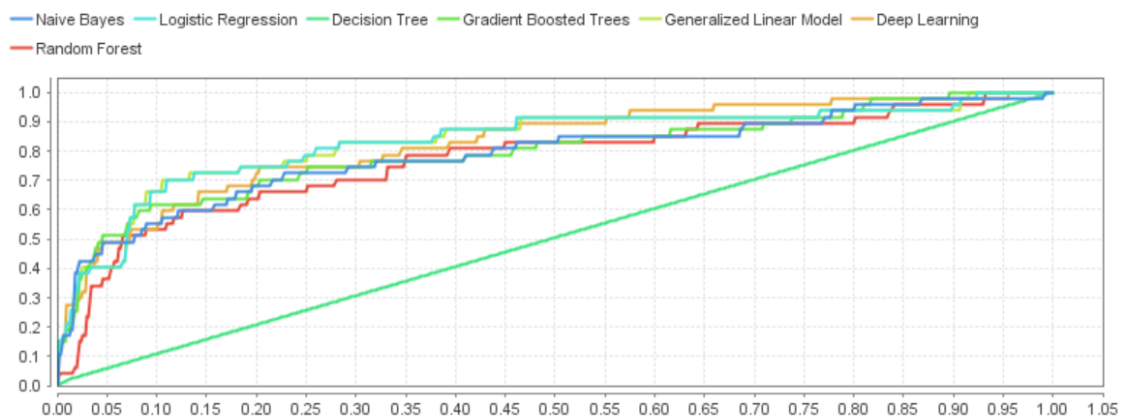


Figure 65

Under each of the models under Results, there are several options to choose. They are Model, Simulator, Performance and Lift Chart. When the option Model is chosen, we get to see plots on the relationship between each attribute and the target variable. Under Simulator there are options where the parameters can be adjusted, and the most important factors will be displayed.

The option Performance shows the parameters in detail. There is a confusion matrix, then all the other values displayed like in Figure 66.

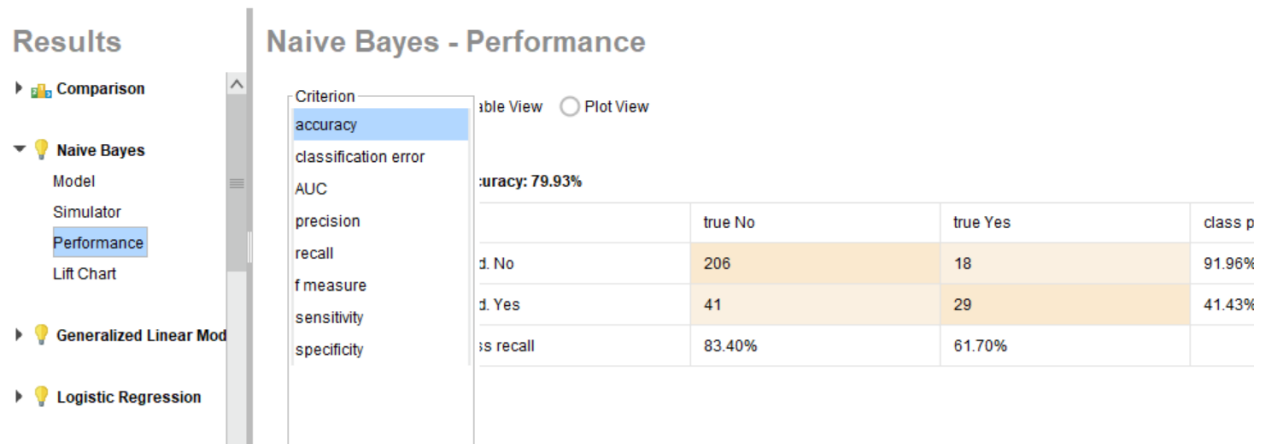


Figure 66

5.3.1 Naïve Bayes

Table 12

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 206 | 18 | 91.96% |
| Predicted Yes | 41 | 29 | 41.43% |
| Class Recall | 83.40% | 61.70% | |

- **Accuracy:** 79.93%
- **Error Rate:** 20.07%
- **AUC:** 0.788 (Positive Class: Yes)
- **Precision:** 41.43%
- **Recall:** 61.70%
- **F measure:** 49.57%
- **Sensitivity:** 61.70%
- **Specificity:** 83.40%

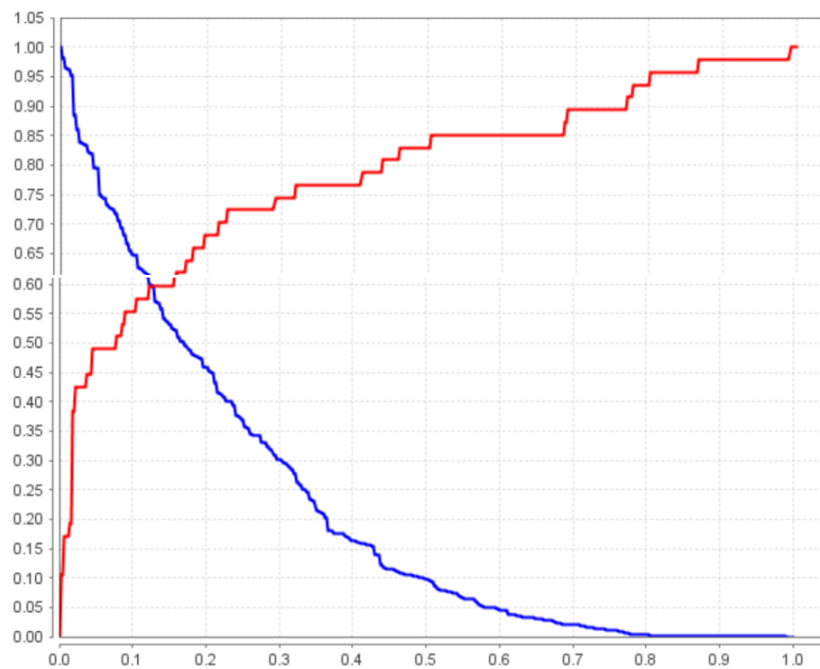


Figure 67

5.3.2 Logistic Regression

Table 13

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 230 | 27 | 89.49% |
| Predicted Yes | 17 | 20 | 54.05% |
| Class Recall | 93.12% | 42.55% | |

- **Accuracy:** 85.03%
- **Error Rate:**14.97%
- **AUC:** 0.834 (Positive Class: Yes)
- **Precision:** 55.26%
- **Recall:** 44.68%
- **F measure:** 49.41%
- **Sensitivity:** 44.68%
- **Specificity:** 93.12%

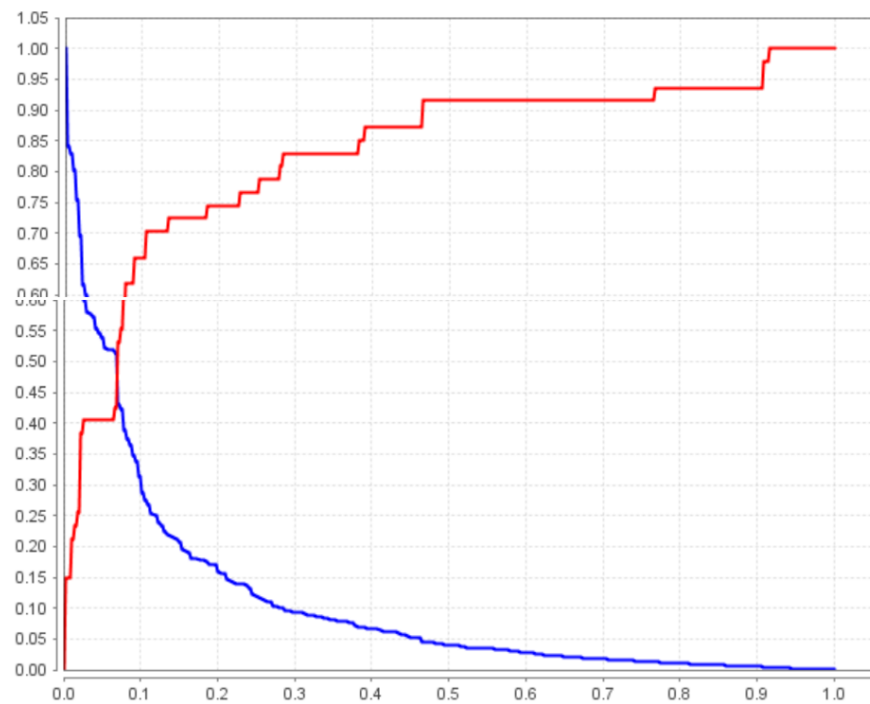


Figure 68

5.3.3 Deep Learning

Table 14

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 221 | 21 | 91.32% |
| Predicted Yes | 26 | 26 | 50.00% |
| Class Recall | 89.47% | 55.32% | |

- **Accuracy:** 84.01%
- **Error Rate:** 15.99%
- **AUC:** 0.826
- **Precision:** 50.00%
- **Recall:** 55.32%
- **F measure:** 52.53%
- **Sensitivity:** 55.32%
- **Specificity:** 89.47%

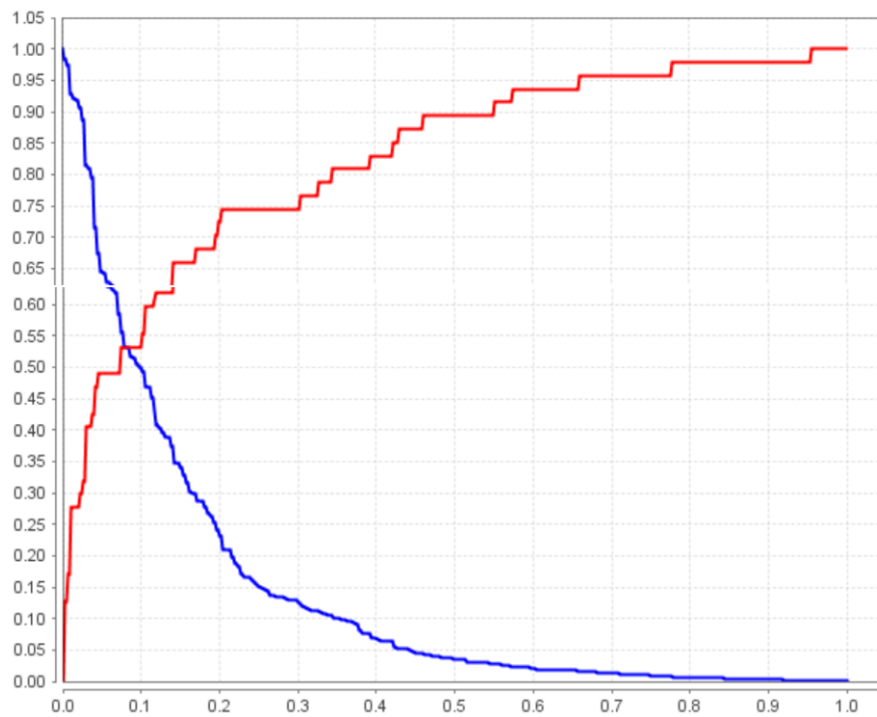


Figure 69

5.3.4 Decision Tree

Table 15

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 224 | 46 | 84.14% |
| Predicted Yes | 3 | 1 | 25.00% |
| Class Recall | 98.79% | 2.13% | |

- **Accuracy:** 83.33%
- **Error Rate:** 16.67%
- **AUC:** 0.505
- **Precision:** 25.00%
- **Recall:** 2.13%
- **F measure:** 3.92%
- **Sensitivity:** 2.13%
- **Specificity:** 98.79%

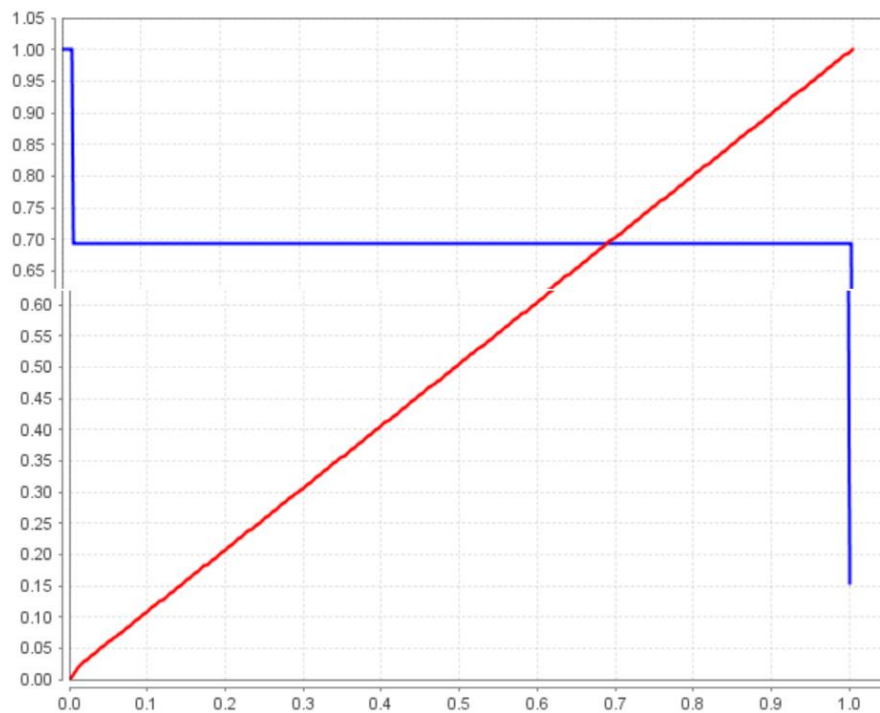


Figure 70

5.3.5 Random Forest

Table 16

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 242 | 44 | 84.62% |
| Predicted Yes | 5 | 3 | 37.50% |
| Class Recall | 97.98% | 6.38% | |

- **Accuracy:** 83.33%
- **Error Rate:** 16.67%
- **AUC:**0.771
- **Precision:** 37.50%
- **Recall:** 6.38%
- **F measure:** 10.91%
- **Sensitivity:** 6.38%
- **Specificity:** 97.98%

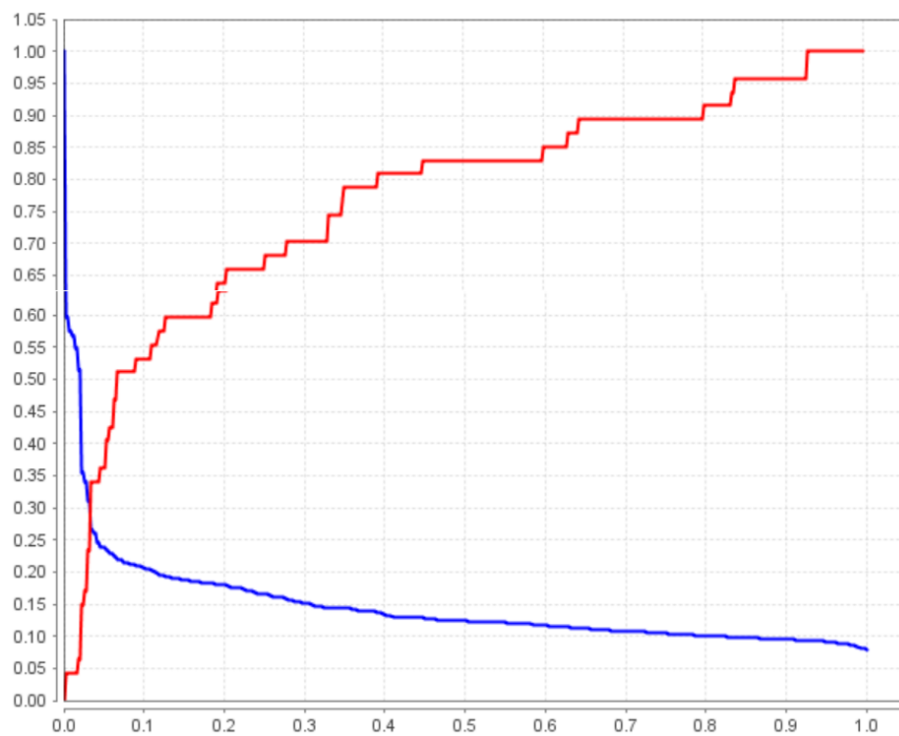


Figure 71

5.3.6 Gradient Boosted Trees

Table 17

| | True No | True Yes | Class Precision |
|---------------|---------|----------|-----------------|
| Predicted No | 242 | 33 | 88.00% |
| Predicted Yes | 5 | 14 | 73.68% |
| Class Recall | 97.98% | 29.79% | |

- **Accuracy:** 87.07%
- **Error Rate:** 12.93%
- **AUC:** 0.796
- **Precision:** 73.68%
- **Recall:** 29.79%
- **F measure:** 42.42%
- **Sensitivity:** 29.79%
- **Specificity:** 97.98%

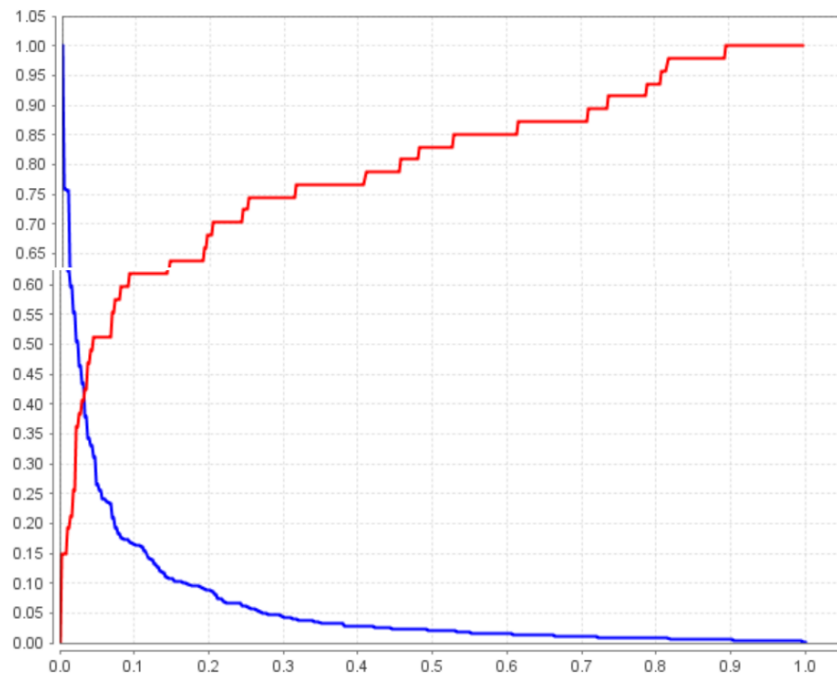


Figure 72

Chapter 6 Conclusion and Recommendation

Two methods were chosen to calculate and compare the accuracies of various machine learning algorithms on Employee attrition data and we have got different results on accuracy, error rate and many more. One was Monte Carlo Simulations and the other is Auto Model in RapidMiner.

The algorithms chosen for comparison were also run separately and the results are as below

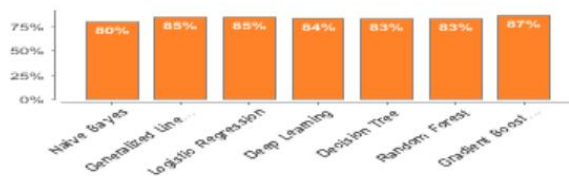
Table 18

| Model | Accuracy | Error Rate | Sensitivity | Precision |
|---------------------|----------|------------|-------------|-----------|
| Naïve Bayes | 79.59% | 0.20408 | 0.3972 | 0.64444 |
| Decision Tree | 82.99% | 0.17006 | 0.53571 | 0.22846 |
| Random Forest | 85.87% | 0.14130 | 0.8000 | 0.7500 |
| SVM(Linear) | 85.71% | 0.17007 | 0.5000 | 0.3200 |
| Logistic Regression | 88.18% | 0.11818 | 0.32432 | 0.92308 |

Based on the results that were calculated individually, we see that going by the accuracy Random Forest and SVM Linear has better performance than the others. Going by the error rate, we see it is less for Random Forest and Decision Tree and SVM(Linear) model. Lesser error rate should be one of the features while choosing a model.

Overview

Accuracy



Runtime (ms)

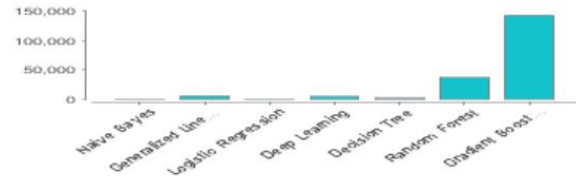


Figure 73

Naïve Bayes Generalized Linear Model, Logistic Regression, Deep Learning, Decision Tree, Random Forest, Gradient Boost

Table 19

| Model | Accuracy | Error Rate | AUC | Runtime |
|--------------------------|-----------------|-------------------|------------|----------------|
| Naïve Bayes | 79.9% | 20.1% | 0.788 | 315 ms |
| Generalized Linear Model | 85.0% | 15.0% | 0.833 | 6 s |
| Logistic Regression | 85.4% | 14.6% | 0.834 | 518 ms |
| Deep Learning | 84.9% | 16.0% | 0.826 | 4 s |
| Decision Tree | 83.3% | 16.7% | 0.505 | 1 s |
| Random Forest | 83.3% | 16.7% | 0.771 | 37 s |
| Gradient Boosted Trees | 87.1% | 12.9% | 0.796 | 2 m 23 s |

Table 19 has the accuracy, Classification error, AUC and Runtime for all the models. Accuracy of a model determines how often is the classifier correct or the rate at which the classifier is correct. We can see that Gradient Boosted Trees have the highest accuracy making it the model which can be chosen as best.

Error rate is otherwise called misclassification rate. This is the measure to see how often the classifier goes wrong. From Table 19, we can see that the error rate is the lowest for Gradient Boosted Trees, which is followed by Logistic Regression and Generalized Linear Model. Going by the error rate we can say that the Gradient Boosted Trees algorithm is the best.

The value of Area Under the Curve or AUC tells us how efficient is the model in classification. That is how well can it predict the values as it is. In our dataset we are predicting Attrition which is a binary.

Therefore, the ability to predict Yes as Yes and No as No is AUC. Hence looking at Table 12, we see that Logistic Regression is a better model at classification. Generalized Linear Model and Deep Learning are in the second and third place respectively according to Table 8. Hence, we can say that Logistic Regression is better at classification with respect to AUC value.

Comparing the runtime of all the models in the table, we see that Naïve Bayes is the quickest. Followed by Logistic Regression and Decision Tree. To conclude, even if Gradient Boosted Trees model is better with the accuracy and error rate, but when it comes to runtime the model is not very effective. From the results the second-best model would be Deep Learning, which is quicker than Gradient Boosted Trees algorithm.

Gradient Boosted Trees is one of the ensemble methods. This machine learning algorithm is suitable for both regression and classification. Gradient Boosted Trees works by bringing all the weak models together as one complex predictor. The weak models are basically decision trees; hence it is a complex collection of many decision trees. The algorithm works in such a way that it finds the pattern of residual to a large extent that would strengthen a model and make it more powerful. When this is done repetitively, there will be a point where there is no set pattern for these residuals. This is more like minimizing the loss function.

Some of the advantages of Gradient Boosted Trees are that it reduces the loss function, it can handle missing values, it is suitable for numerical and categorical data and the last but most important advantage is flexibility. Some of the downfalls of the algorithm is that it is time consuming as it has more than 100 decision trees. Apart from time it consumes memory as well. The algorithm might result in overfitting while trying to minimize the loss function. Hence it is important to cross validate.

Chapter 7 Reflection

I am happy to be sharing my experience about the Masters course and dissertation. The journey throughout was fabulous and very productive. I have learnt a lot of things that has improved me personally as well. As it said and like how many people believe there are two sides in a coin, so there were positive and not so positive experience. But I believe that when it is an experience there is no positive or negative to it. Experiences are there as a learning for an individual to evolve out of that little shell that they keep themselves in. So, the Masters in Data Analytics program in Dublin Business School has been a package for me where I have plenty of takeaways.

Being in the Customer Support industry for the past few years, I wanted a change in my career. Though I was doing well and was enjoying and working on the challenges, at a point I wanted to take a break from my career and pursue higher studies. While I had this in mind, I started looking at different articles and news on different technologies, advancements in different fields and many more. All these advancements lead me on to my further search into Artificial Intelligence, Machine Learning, Data Visualization and Data Analytics. Hence the interest grew that I wanted to study and pursue my Masters' in it. I am student with a bachelor's degree in Electrical and Electronics Engineering, so it was a tough call for me to decide on studying Data Analytics. But I was determined to do this course because of the various automations and betterment it can do for people.

Artificial Intelligence is applied in every day to day activities and it has made people's life easy. Not only has it made easy for people, it has made life better for some. Things like driver less cars, robots that assist people, tools used for handicapped people all these innovations are truly a wonder. I am a kind of person who stands by and would like to be part of technology that can help people in day to day activities. It is my dream to become an expert in Artificial Intelligence and bring good things for the mankind. I have many more things to learn to achieve that and I am sure one day I will become one.

The journey throughout the course and the few months of dissertation was challenging yet fruitful for me. Within the few months of studying here in Dublin Business School, I was able to understand the different topics in Data Analytics such as machine learning, data visualization, data analytics and other techniques at a basic level. Through the few months of dissertation I can say that I learnt a lot.

The research methodologies module that we had for the second semester was very helpful with the dissertation. We were allowed to think about the research topic we were interested in and we were let to discuss in the class. This is again similar to group projects. Different people had different unique ideas

that the other could learn from. By doing this we had more clarity on thoughts. The best part is I was able to think about certain topics and wonder if such things even existed. Once we have our ideas discussed we were able to get inputs from our Lecturer and other members in the class.

I must mention the programming module that we had for the first semester. The concepts were taught in such a systematic way, that if we had attended all the classes, it was easy to do the assignments without much help and do well in the examination. We were given challenging questions hence we worked among ourselves to find solutions.

Another important activity that helped me understand concepts better were through the group projects. We got to work as a team and pitch in different ideas. The activity was a learning for me. Apart from the classroom training and the videos and online articles that I read. When we work as a team, not everybody will have the same idea. Each one will have unique techniques and unique ideas that others can learn. This is possible only when we work as a team. I am sure all this experience will be helpful to work as a team in an organization. The lecturer set the standards of how a group project should be and that helped my team to work upon it which was very helpful. Last but not the least the experiences that I shared with my classmates was great. I was able to get different ideas from different people, which helped me throughout the dissertation apart from the advice and guidance from my supervisor.

Appendices

Appendix A

Dataset is being taken from 'SAMPLE DATA: HR Employee Attrition and Performance, McKinley Stacker IV, 2015'. Available at: <https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/>

Appendix B

R Code for Naïve Bayes Algorithm. The libraries that are necessary are mentioned in the beginning. There are separate codes for different ratios of trainset data. And the respective accuracy values are calculated. File name: Naive Bayes codes

Appendix C

R Code for Decision Tree Algorithm. The libraries are included at the beginning of the file. Codes are written separately for Decision Tree that can be split based on Gini Index and Information Gain. The respective accuracies are also calculated. File name: Dissertation - Decision Tree

Appendix D

R Code for Random Forest Tree. The libraries are included right at the beginning. Codes are written to model the dataset with Random Forest and find the importance of the attributes.

File name: Dissertation - Random Forest

Appendix E

R Code for Support Vector Machine Algorithm. The dataset is modeled with four different kernels linear, radial, sigmoid and polynomial. File name: Dissertation - SVM

Appendix F

Tableau workbook with all the visualization of the dataset. File name: Visualization.

Appendix G

R code for Monte Carlo simulations. Repetition selected are 5,10,50,100 and 200. Output for each of the reps are like in the files Output for 5 MCs, Output for 10 MCs, Output for 50 MCs, Output for 100 MCs and Output for 200 MCs. File name (R code): Code for Monte Carlo

Bibliography

1. 'Data Science Case Studies, Rapid Miner, no date'. Available:
<https://rapidminer.com/resource/case-studies/> (Accessed on 5/12/2018)
2. 'Data Science Plus, Logistic Regression, no date'. Available:
<https://datascienceplus.com/perform-logistic-regression-in-r/> (Accessed 23/11/2018)
3. 'Ian Goodfellow, Yoshua Bengio, Aaron Courville, page 592': Deep Learning
4. 'Matt Dancho, Using Machine Learning to Predict and Explain Employee Attrition, no date'.
Available at: <https://www.kdnuggets.com/2017/10/machine-learning-predict-employee-attrition.html> (Accessed on 2/12/2018)
5. 'Naïve Bayes Theorem, June 30, no year'. (Available at: <https://becominghuman.ai/naive-bayes-theorem-d8854a41ea08> (Accessed 02/12/2018)
6. 'Rapid Miner, Wikipedia, no date'. Available: <https://en.wikipedia.org/wiki/RapidMiner>
(Accessed on 5/12/2018)
7. 'RapidMiner Auto Model, RapidMiner, no date'. Available at:
<https://rapidminer.com/products/auto-model/> (Accessed on 5/12/2018)
8. 'Typical recruitment process', Available at:
<https://ge.usembassy.gov/embassy/jobs/recruitment-process/> (Accessed 2/12/2018)
9. 'Variable Importance for Random Forest Models, Dragonfly Statistics, 31/12/2017' Available at:
<https://www.youtube.com/watch?v=-2DIAMYioqY>. (Access on 20/11/2018)
10. Kernel Functions-Introduction to SVM Kernel & Examples, 2018, Available: <https://data-flair.training/blogs/svm-kernel-functions/> (Accessed 3/12/2018)
11. McKinley Stacker IV, 'SAMPLE DATA: HR Employee Attrition and Performance, 2015'. Available
at: <https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/>
Accessed on 07/11/2018
12. Niklas Donges, 'The Random Forest Algorithm, February 22, no year '. Available at:
<https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd> (Accessed on
9/12/2018)
13. Peter Dizikes (2010) 'Explained: Monte Carlo Simulations', MIT News Office. Available at:
<http://news.mit.edu/2010/exp-monte-carlo-0517>. Access on: 10/12/2018
14. Prakash Saxena, 'Things to keep in mind while working with Decision Trees, August 6, 2017'.
Available at <https://www.techleer.com/articles/247-things-to-keep-in-mind-while-working-with-decision-trees/> (Accessed 02/12/2018)

15. Vijay Kotu and Bala Deshpande Phd (2015), Predictive Analytics and Data Mining
16. Vijay Kotu and Bala Deshpande, Chapter 5, Pg 183, Figure 5.14, Predictive Analytics and Data Mining
17. Vijay Kotu, Bala Deshpande Phd, 2015, Pg 64, Predictive Analytics and Data Mining
18. Vijay Kotu, Bala Deshpande Phd, 2015, Pg 70, Predictive Analytics and Data Mining
19. Will Kenton, Attrition, Available: <https://www.investopedia.com/terms/a/attrition.asp>
(Accessed on 2/12/2018)

