Employee Attrition Data Set Analysis

Introduction:

The issue of keeping one's Employees cheerful, satisfied and fulfilled is a lasting and age-old Challenge. In the organisation, on the off chance that an employee you have contributed so much time and cash leaves for "greener fields", at that point this would imply that employer would need to invest considerably more energy and cash to employ another person. Let us along these lines swing to our prescient demonstrating abilities and check whether we can foresee worker whittling down on this Employee Attrition Dataset.

Data Set Description:

For the prediction of Employee Attrition, Employee Attrition dataset is used which contains 35 Features and 1470 observations. In this dataset, 'Attrition' is the target variable which provides the results in true or false statements stating on the basis of other 34 features that an employee will leave the organization or not. There are many features in the dataset which holds the value ranges between1-5 such as education, Job involvement job satisfaction etc. which are varying 1 as low to the 5 as highest level of that feature. All the features present in the dataset describes different aspects of employee information which may play a vital role in prediction of employee attrition.

Examination of Data:

For examination of data, the foremost work is to import all the required packages and libraries. After importing the dataset, I looked into the first few rows of the dataset with the 'head' function. Then the next and the most important step in data analysis is Data Exploration. In this step, the dataset has been very minutely observed to perform data exploration. First of all, I take a look at the structure of the dataset by the 'str' function. I found out that the dataset contains attributes of only two data types, i.e. integer and factor. For more examining the data, I checked the dataset with the missing values as they create too noises and problems while building a good model. So, there were no missing values present in the dataset, as the sum of the NA values comes out to be zero. After looking more closely in to the structure I observed that there are some features whose values are

same for each observation and hence they are the least important variables in predicting the attrition. Those features were Employee-count, Employee-number, over18 and standard hours. So I dropped them and excludes them from my further analysis. Then the next step is to observe the trends inside the dataset using visualizations. Bar plots had been used to make a comparative examine of the functions. First of all I looked into the factors type variables. I got the results that there are more male members than the Female members in the dataset and the employers travels rarely are in a majority. Then I tried to have a comparative study of employees who left the organization and who did not, based on the different attributes. So I decided to plot different attributes with the attrition and after plotting I got the following inferences from the plots:

- Employees with the job level of 4 & 5 shows the least attrition.
- Employees with the Education level 3 are the highest in number of leaving the organization while the Education level 5 are mostly tend to stay in the organizations.
- Environment satisfaction does not look to be a deciding factor in attrition as all levels were predicting almost similar numbers.
- High job involvement people were also more tend to leave the organization.
- The ratio of non-leaving personnel to leaving personnel is higher in the revenue hike range of 11% - 14%, this ratio is almost comparable & lower than the previous for the range of 15% - 22%. Personnel with revenue hike of 25 % display the least tendency of leaving the organization.
- Employees with performance level 4 are more tend to stay as compare to performance level 3.
- Employees who've spent 11 years or more within the organization have the least tendency of leaving.
- Employees who have spent 5 or more years in a current role tend to stay.

Now after getting the above results my next step was to find out the correlation between the different attributes. So for that I required all values to be in either integer or in numeric

form so I copied the dataset into the dummy dataset and then I converted all the factors variables into the number form. After converting the variables, I build the correlation table and I used the 'melt' function to extract all the values which then sorted and examined easily.

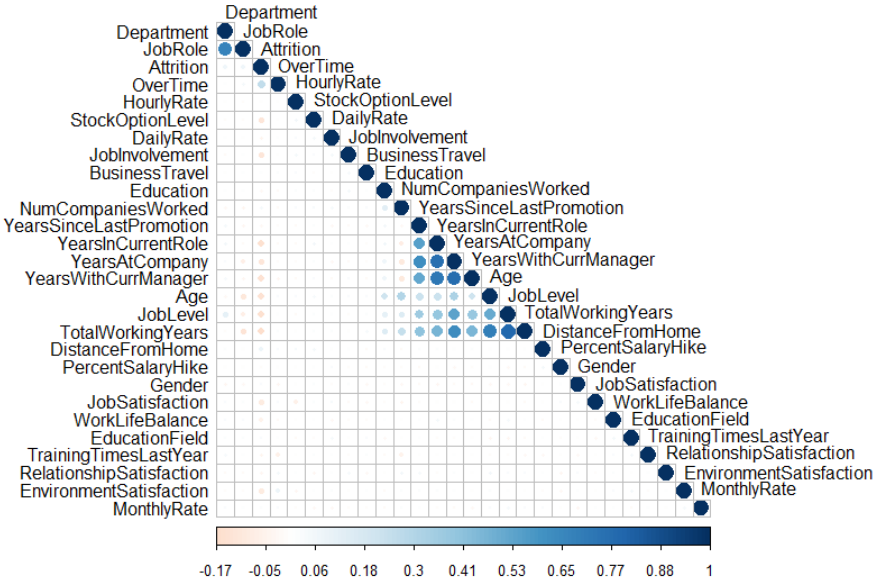

Figure 1. 1 correlation plot

After carefully analysing the correlation plot and desk, I discover that there are a lot of correlated features. I run the simple logistic regression model to have a look at which functions are to be dropped from the pair of every correlated features. This has been executed through apart from one characteristic and dropping the other and vice-versa at the same time as running the set of rules. After several iterations, it is concluded that features like Marital-Status, Monthly-Income and Performance-Rating degrade the model. So I consequently drop these columns.

There were different important variables on which attrition varies but the most important variables were Job Role, Over-Time and Training Time.

1. Job Role: There are different job roles present in the employee dataset such as sales representative, laboratory technicians, human resource, sales executives, research scientists, manufacturing directors, healthcare representatives, managers and research directors among which sales representative are having highest tendency to

leave the organization followed by the laboratory technicians while the Research Directors are the least tend to move.

2. Over Time: This Variable is one of the categorical variables which contains values of 'yes' for the people who does overtime and 'no' for the people who does not. People who do overtime are more likely to leave the organization as compare to the people who do not do overtime.
3. Training Time: This variable contains the level from one to six which indicates the training time period. It has been analysed that the time level of 2 are having most tendency to leave the organisation which is followed by level 3 while both the people who spent least and most time in training are in very less percentage of leaving the organisation and tends to stay within the organisation.

Model Building:

Now, for the building of the model the first step I used is the dividing the dataset into the train and test data sets. Before splitting the dataset I set seed to 233,seed is been set so that the random objects and simulations can be created which can be reproduced again. These random numbers remains the same no matter how far out in the sequence I went. For splitting the dataset I choose the 70/30 split .As the total no. of observations were 1470 in the employee attrition dataset which divided into the Test dataset containing 441 observations while the train dataset contains 1029 observations after applying the 70/30 splitting. I choose the Decision tree model for building up the simple tree. Decision trees are the tree model which displays the range of possible outcomes and subsequent decisions made after an initial decision. The most useful aspect of decision tree is that they force you to consider as many possible outcomes of a decision as you can think of. Decision Trees can be generated for both categorical as well as for continuous variables. Decision Tree has provided a framework to consider the probabilities and payoffs the decisions, which can help in analysing a decision to make the most informed possible decision. For building the decision tree I used the bucket size of 20 and 'fancyRplot' function is used to generate the decision tree as it is very interpretable and it beautifully explains which features are the prime deciding factors of attrition. It also explains the different factors and the percentage

attrition it may lead to. I then perform prediction on the test set and check the accuracy of the model based on the predicted data and also generate the confusion matrix for the test data. In the confusion matrix I got the 346 true positives and 26 true negatives with which I got the accuracy of 84.4% which is quite a good number but the area under curve value is very low in this model as it is only 0.71 . However this model is more helpful in interpreting the results and then making decisions based on the results. For improving the accuracy of the model, two ensemble techniques naming 'Bagging' and the 'extreme boosting' has been used by me.

Ensemble Techniques:

1. Extreme Gradient Boosting:

Extreme gradient boosting method is one of the tree ensemble techniques which are used for building more better and accurate model. It produces a prediction model in the form of an ensemble of weak prediction models, generally decision trees. It also builds the model in the stage wise approach as of other boosting techniques and it generalizes them by applying optimization of an random differentiable loss function.

Gradient tree boosting is typically used decision trees as base learners. There are two main reasons to use the Extreme Gradient boosting, the first one is the execution speed and another one is the model performance.

1. XGBoost is fast in the execution when compared to the other gradient boosting techniques implementations.
2. XGBoost dominates structured or tabular datasets on classification and regression predictive Modelling problems.

XGboost requires the numeric or integer matrix for its input. So, I need to convert all the factor variables into the numeric forms by creating matrix for factors variables using the one hot encoding technique. For both the train and test data I made the matrix using the sparse model matrix, with this function all the variables which were in numeric format remains the same only the factor variables changes into the numeric format.

After getting the sparse matrix the next step is to build the model using the 'XGB.train' method, this method is used to train the XGBoost model. This method

takes several inputs including the train and test sparse matrix containing all the values in numeric form, data as the train data, list of parameters containing the booster parameter and the max number of the iteration count which is used for the no. of times running the model. So, I kept the no. of iteration count as 6749. Then I Plot the training and Test errors for the 6749 iterations generated by the model.

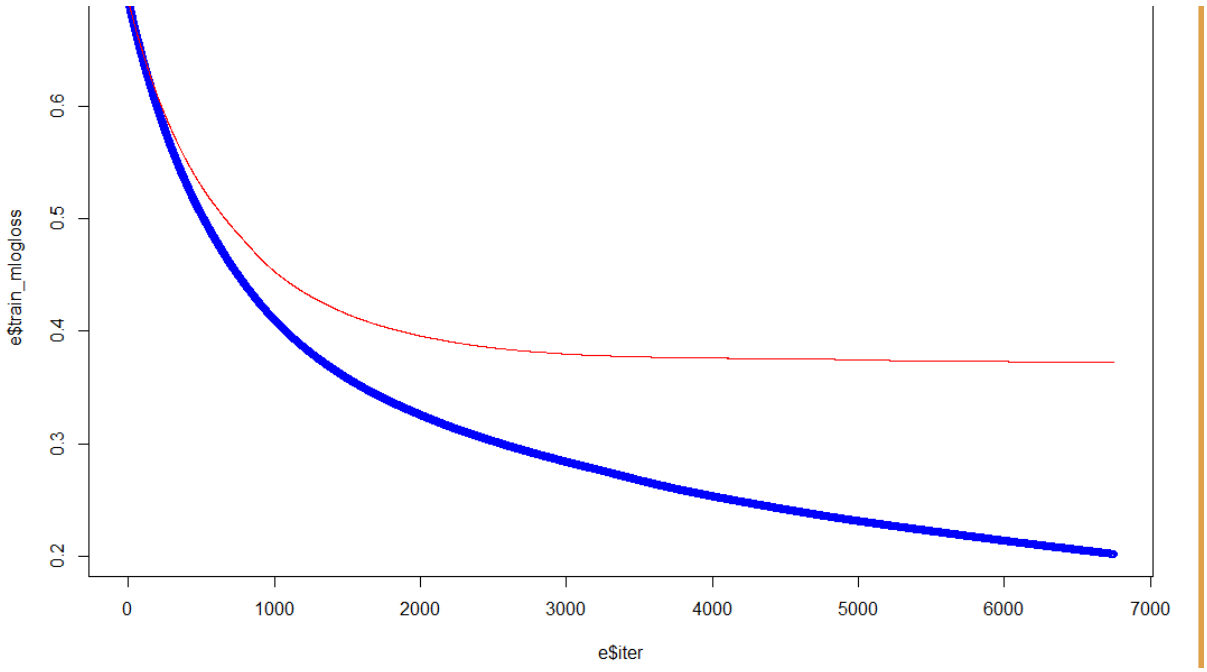

Figure 1. 2 Iteration VS error plot

From the plot, it can be seen clearly that after the 2000 iterations, the error becomes constant for the test data while the train error continuously reduces till the last iteration of the model. The minimum error recorder for the Test data is 0.372 while the minimum error for train data was recorded as 0.202.

Now, the step was to get the important features for the model, so I build a plot between the features and with their frequencies. From that plot, Monthly income tends to be the highest frequency of 0.08 followed by the Overtime with the 0.06.

After successful Building of the model now, was the time to predict the accuracy and builds the confusion matrix of the model to check the efficiency of the Extreme Gradient Boosting ensemble model. So from the confusion matrix, the true positives count comes out to be 363 and the true negatives count comes out to be 15 for the employee attrition Test data. And the accuracy of the extreme Gradient Boosting

ensemble model comes out to be 85.71% which is better than the previous single tree model of 84.4%. So, with these results I can say that Extreme Gradient Boosting ensemble has built the better and more accurate model than the Decision tree Model.

Also, another interesting result comes out in this was the confidence interval of the accuracy which ranges 82.1% to 88.8%. The other Results of this ensemble model includes the Sensitivity of the model which comes out to be 0.986 and the Specificity of the model which comes out to be 0.205. With the help of results of Sensitivity and Specificity I build a plot for calculation the area under curve and that comes out to be 0.806.

2. Bagging Ensemble Model:

Bagging is the ensemble technique in which multiple models are built from different subsamples of the training dataset. Bagging stand for the Bootstrap Aggregating which is a method to decrease the variance of the prediction made by model by generating additional data for training from the original dataset using combinations with repetitions to generate Multi-sets of same cardinality as of original dataset.

For the data attrition dataset, I already build the decision tree so with the help of bagging I tried to build the bagged tree and check the different results. First of all, I build the Bagged tree with the default parameters that is the nbagg count =25 i.e the 25 bootstrap replications, cp=0 and coob = false with these parameters I got the area under the curve value be very less as 0.78 which is still better than the decision tree but it was less than the previous ensemble technique. So, I changed the parameters while building the bagged tree as then I changed the nbagg count =100, cp=0.0001 and coob = true so that the misclassification error out of the bag will be computed, after changing the parameters I got more accurate results as the area under curve comes out to be 0.81 which is slightly more than the Extreme gradient boosting results and Out-of-bag estimate of misclassification error was 0.1584. For predicting the bagging model I used the probability as type of prediction.

After getting these results I experimented little bit and tries to build the bagged tree model using the test data set instead of training data and by keeping other parameters same as nbagg =100, cp=0.001 and the coob=true. I got the very interesting results as the area under the curve came out to be 0.97.

Comparison of the Models:

For the Employee Attrition Data-set, The Three Models which I build are the Decision Tree model, Extreme Gradient Boosting Model and the Bagged Tree Model. Though after cleaning up the data, I got the good results with Decision Tree Model as the accuracy of the model came out as 84.4% which can be considered as a good model with this much accuracy but the area under the curve value was low as 0.71. So, I applied the first Ensemble method, Extreme Gradient boosting which performs very well as It provides the accuracy of 85.71 % which again can be considered as a better model than the decision tree. Also, Extreme Gradient Boosting ensemble has provided the good sensitivity results which results into the 0.805 value of area under the curve. The Another Ensemble Technique, Bagging Tree model has also provided the interesting results as after performing different experiments on bagging the area under the curve comes out to be 0.81 which is better than other two models. Also with the Test data this model provides the 0.97 area under the curve value. I build the Roc curve (The plot between the sensitivity and the specificity) for all the three models which can be seen in the graph.

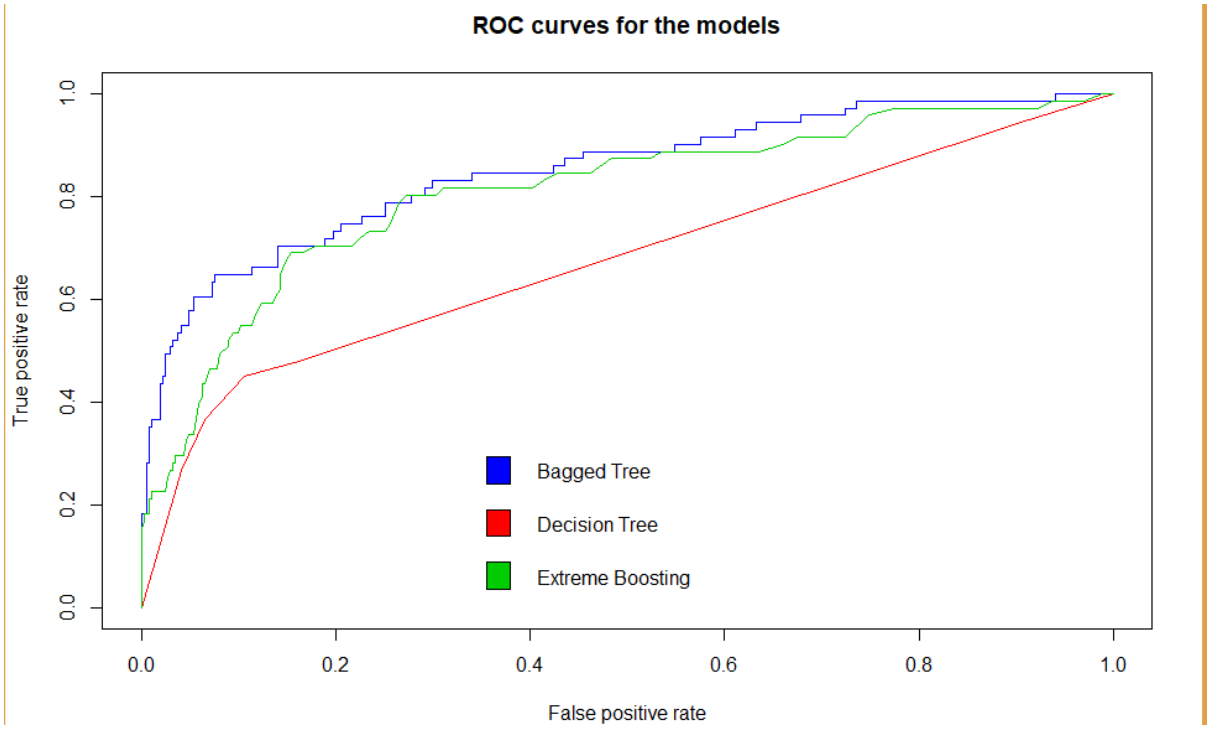

Figure 1. 3 ROC curve

As the Graph is depicting that the best ROC curve is of the Bagged Tree model followed by the Extreme Gradient Boosting and The Least Accurate is the Decision Tree model which was the simple general tree model built for the employee attrition.

Fish Dataset Analysis

Introduction

Fish data analysis report is based on latest data and economic reports on the EU fishing fleet. This analysis is done based on two main datasets (**economic**) which includes economic variables per fishing boat while the second dataset has data on the types of fish(**fish2**) with corresponding monetary value caught for each boat across the years. The purpose of this analysis was to build a model to predict net profit based on the monetary as well as non-monetary feature values.

Data Set Description:

For the prediction net profit, Economic dataset is used which contains 39 Features and 1432 observations. In this dataset, 'Net-Profit' is the target variable which is a continuous type of variable feature which predicts the results on the basis of other 38 features (monetary and non-monetary).Most of the features are monetary features and for calculating the net profit the important variables are more from the monetary type but sometimes non-monetary variables also plays a crucial role so I cannot discard or neglect the non-monetary variables as there is a possibility that some non-monetary variable may play a vital role in the prediction. There is another dataset present i.e. fish dataset that contains the monetary values corresponds to the different species of the fish based on the different years. There are in total 103 types of different species of fish are present in the dataset. So I need to analyse the dataset on the basis of fish species as well that which fish species demand or value increased in which year.

Examination of Data:

For examination of data, my first thought was to merge both the economic and fish dataset as both contains same year and vessel Id feature and I merged both the dataset, after merging both the datasets into one data frame there is much noise associates due to the merging of the datasets. So, I drop that thought and then I analyse both the dataset separately and tries to merge them after removing all the missing data and the noise from both the datasets. So, the foremost work is to import all the required packages and libraries

and the dataset into a data-frame. After importing the dataset, I looked into the first few rows of the dataset with the 'head' function. Then the next and the most important step in data analysis is Data Exploration. In this step, the dataset has been very minutely observed to perform data exploration. First of all, I take a look at the structure of the dataset by the 'str' function. I found out that the dataset contains attributes of three data types, i.e. factors, integer and numeric and most of them were the factors. For more examining the data, I checked the dataset with the missing values as they create too noises and problems while building a good model. There were too many missing values present in the economic dataset, so I choose two basic approaches to deal with the missing values. First approach was the standard scaling technique in which I replace the missing value in a column with the average value of the column while in the other technique I replaced the missing values with the zero (this is done in the columns where averaging is creating noise).There are many variables which were of factor type so I convert all of them into the numeric form. After conversion, I decided to build the plots for the 3 most important variables Net profit which is a continuous variable, Segment variable and size category variable which both are of categorical type variables. So, I build three plots.

- The first plot is between the Fishing Income and the Net Profit in which the net profit fluctuation can be seen on the basis of fishing income of different segments. As seen from the graph the High rated Profits are generated through TM segment while

there are more segments like DTS and FPO which generates the good profits but there profit level is quite low.

- The second plot is between Fishing Income and Size Category in which Income is depends upon Size category of different segments of fishes. In this, It can be seen very clearly that TM segment is of 40-70 size category and they are generating highest Fishing Income while the DTS segment is lying between 18-40 size category and they are too generating more fishing income. However, there are other size category segments are also present which are contributing in the Fishing Income.
- The Third plot is between the Count and the size category of the Fish segments in which count is dependable while size category is independent entity. The Highest no of segment fishes are present in the 10-20 size category while the segment which are most indulge in making profits and in generating incomes that are in lowest count i.e. the 40-70 size category segment.

After plotting these graphs, I further predict the Net Profit values according to the year and then I plot the amount of net profit with the year. With '23306286' this much profit calendar year 2012 leads in the profit table of fishing which is followed by 2011 accounted '19361728' profit . only the calendar year 2008 accounted the loss of '525597' in fishing. All other years' amounts can be seen in the graph present in Appendix B.

Model Building:

Now, for the building of the model the first step I used is the dividing the dataset into the train and test data sets. Before splitting the dataset I set seed to 101,seed is been set so that the random objects and simulations can be created which can be reproduced again. These random numbers remains the same no matter how far out in the sequence I went. For splitting the dataset I choose the 70/30 split .As the total no. of observations were 1432 in the economic dataset which divided into the Test dataset containing 430 observations while the train dataset contains 1002 observations after applying the 70/30 splitting. I choose the Single tree model for building up the simple tree. Tree model displays the range of possible outcomes and subsequent decisions made after an initial decision. The most useful aspect of decision tree is that they force you to consider as many possible outcomes of a decision as you can think of. Decision Tree has provided a framework to consider the probabilities

and payoffs the decisions, which can help in analysing a decision to make the most informed possible decision. For building the tree I used the 'fancyRplot' function is used to generate the tree as it is very interpretable and it beautifully explains which features are the prime deciding factors of Net profit/Loss. I then perform prediction on the test set and check the accuracy of the model based on the predicted data and also predict the RMSE and MAE matrix. I got the accuracy of 65.1 % which is quite a less number. I also generate the cp plot which can be seen in appendix. However this model is more helpful in interpreting the results and then making decisions based on the results. For improving the accuracy of the model, two ensemble techniques naming 'Random Forest' and the 'Stochastic Gradient boosting' has been used by me.

Ensemble Techniques:

1. Random Forest:

A Random forest is an ensemble approach that can likewise be thought of as a type of closest neighbour indicator. Ensembles are isolated and-vanquish approaches which are utilized to enhance the execution of the model. The fundamental rule behind ensemble techniques is that a gathering of weak learner model can meet up to frame a strong learner model. The Random Forest starts with a standard machine learning framework called a "Decision tree". In a Decision tree, data is entered at the top and as it crosses down the tree the data gets bucketed into close to nothing and little sets and in the wake of investigating every one of these sets a strong learner model has assembled i.e. the Random Forest model. Here are the means by which such a framework is prepared;

for some number of trees T:

1. Select some specimen cases aimlessly with substitution to make a subset of the information

2. At every node: 1. For some number x, x indicator factors are chosen aimlessly from all the indicator factors. 2. The indicator variable that gives the best split, as per some goal work, is utilized to do a twofold split on that hub. 3. At the following node, pick another x variable at arbitrary from all indicator factors and do likewise. Contingent on the estimation of x, Random forest: x << number of indicator factors. At the point when another info is gone

into the framework, it keeps running down the greater part of the trees. The outcome may either be a normal or weighted normal of the greater part of the terminal hubs that are come to, or, on account of absolute factors, a voting dominant part.

• With an expansive number of indicators, the qualified indicator set will be very unique in relation to the node to node.

• The more noteworthy the tree relationship, the more noteworthy the irregular random forest mistake rate, so one weight on the model is to have the trees as uncorrelated as would be prudent.

• As x goes down, both between tree relationship and the quality of individual trees go down. So some ideal estimation of x must be found.

Random forest performs fast modelling, I used different parameters and checks the results by building up the model again and again but I got the best results with the 250 no. of tree count and with that random forest ensemble method provides the accuracy of 91.19% which is a very good result and the accuracy of this ensemble method is way more than the previous single tree model. I also checked two more matrix RMSE and MAE for the model. The random forest model accuracy is present at Appendix B.

2. Gradient Boosting Model:

Gradient boosting Model method is one of the tree ensemble techniques which are used for building better and accurate model. It produces a prediction model in the form of an ensemble of weak prediction models, generally decision trees or a single tree. It also builds the model in the stage wise approach as of other boosting techniques and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Gradient tree boosting is typically used decision trees of a fixed size as base learners. There are two main reasons to use the Gradient boosting, the first one is the execution speed and another one is the model performance. Gradient Boost is fast in the execution when compared to the other gradient boosting techniques implementations. Gradient Boost dominates structured or tabular datasets on classification and regression predictive Modelling problems.

For building the ensemble I choose the type of model as gbm and train the model and after that I predict using the test data for the net profit as the target variable.

After successfully building the model I checked the accuracy and I got 84.5 % accuracy which is better than the single tree model. I also checked the RMSE and MAE matrix results with that I build the plot which is present in the Appendix B.

Comparison of the Models:

For the Economic Data-set and the fish Data set, The Three Models which I build are the Single Tree model, Gradient Boosting Model and the Random Forest Model. Though after cleaning up the data, I got the good results with Decision Tree Model as the accuracy of the model of just 65.1% which cannot be considered as a good model with this much accuracy and the area under the curve value was low. So, I applied the first Ensemble method, Random Forest which performs amazingly well as It provides the accuracy of 91.19 % which again can be considered as a way better model than the single tree model. Also, Random Forest ensemble has provided the good sensitivity and specificity results. The Another Ensemble Technique which I used, Gradient Boosting model has also provided the interesting results as It provides the accuracy of 84.5 %.though it is having lower accuracy than the Random Forest But it is also a way better model than the Single tree Model. Also, after performing different experiments on boosting the area under the curve comes out to be good as well.

Conclusions:

1. Random Forest ensemble builds the Best Model among Three with 91.1% accuracy.
2. Highest Net Profit earned in the calendar year 2012.
3. DFN segment generates Maximum Net Profit in all calendar Years but the count of this segment is very less as this segment is having largest size category i.e. this is the segment of big vessels.
4. Despite low fuel prices, consumption and fuel use intensity decreased by 16 % and 20 % respectively from 2008 to 2013, an indication that many EU fleets have become more efficient.

5. Improved performance is largely a result of increased income from landings and lower costs, in particular due to low fuel prices. Revenue increased almost 5 % while total costs only increased by 1 %.

Appendix A

Fish Dataset Analysis

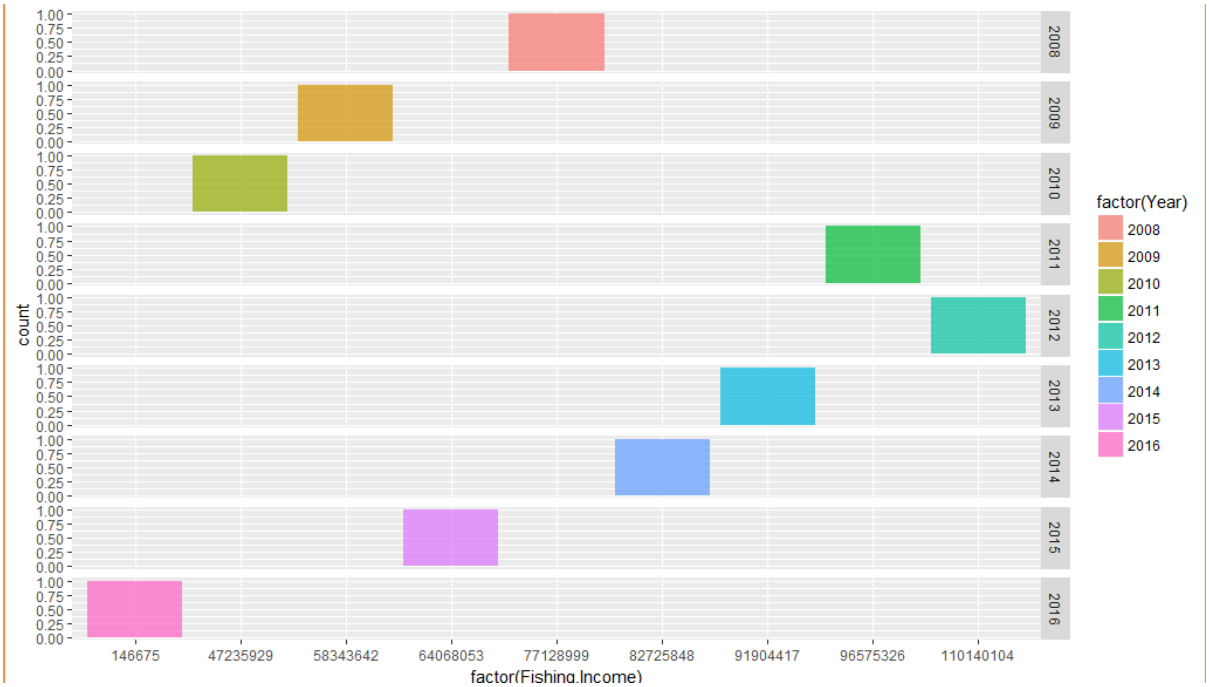

Figure 2. 1 plot between year and Net Profit

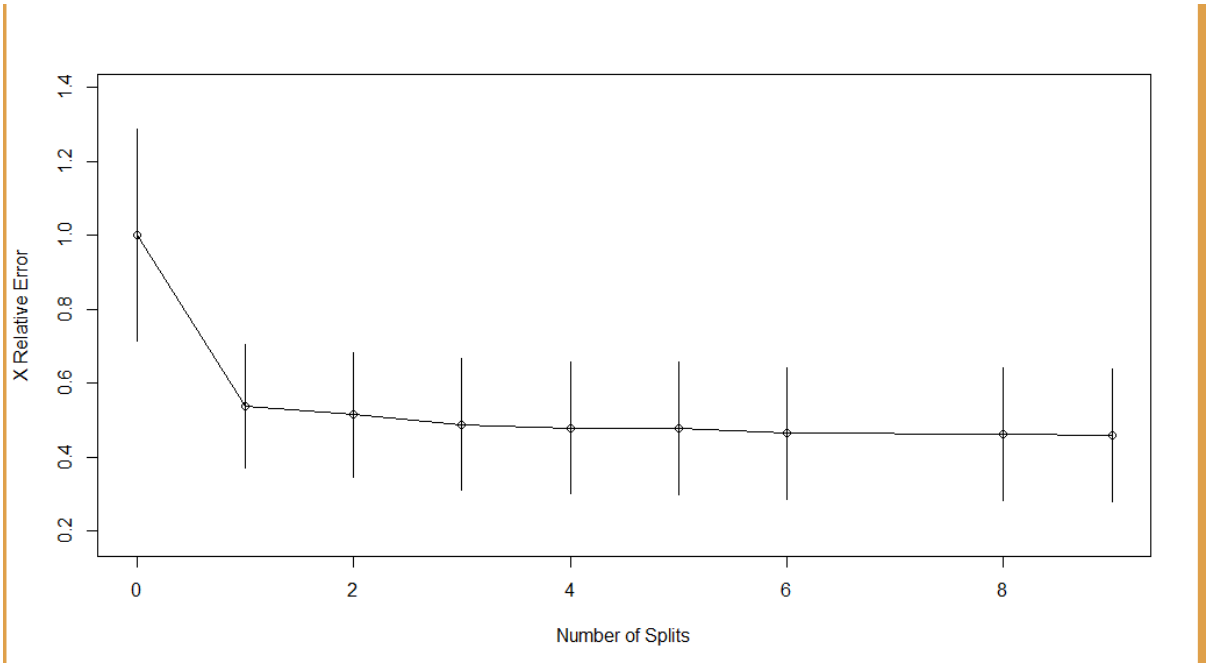

Figure 2. 2 cp for single tree

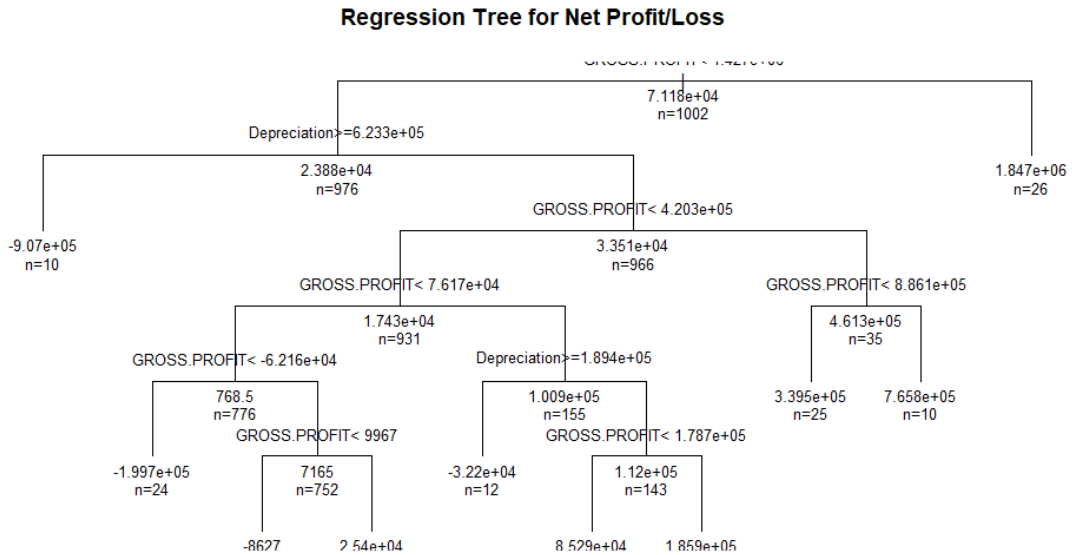

Figure 2. 3 Regression tree

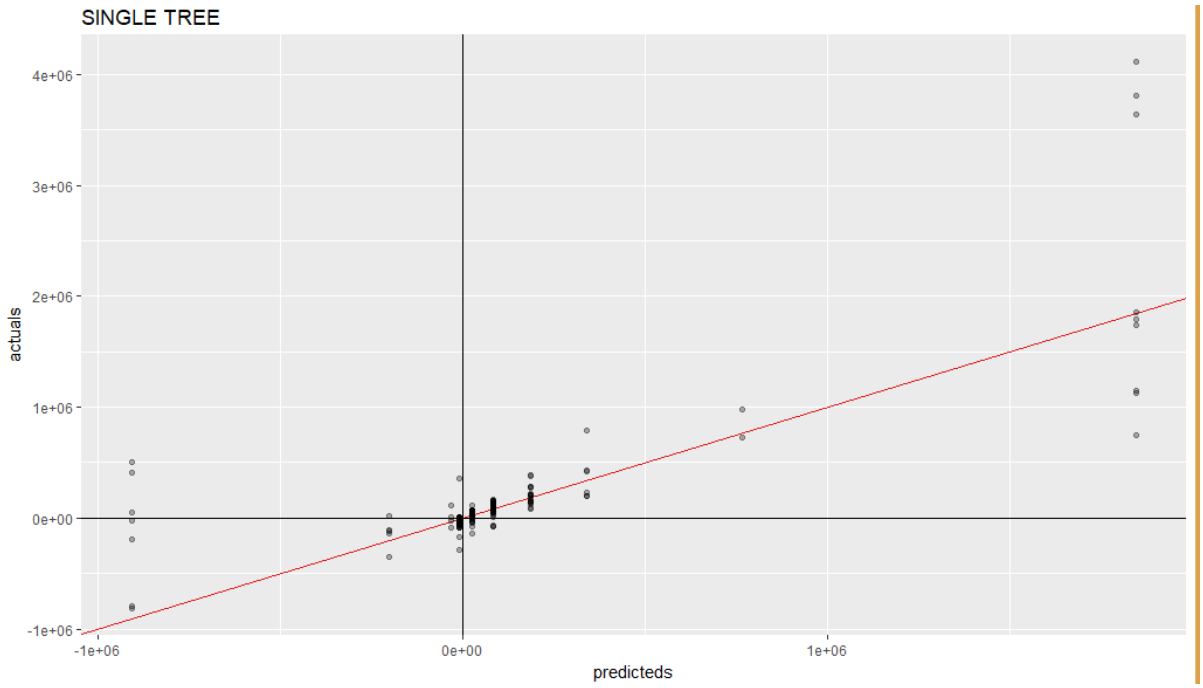

Figure 2. 4 Tree Accuracy curve

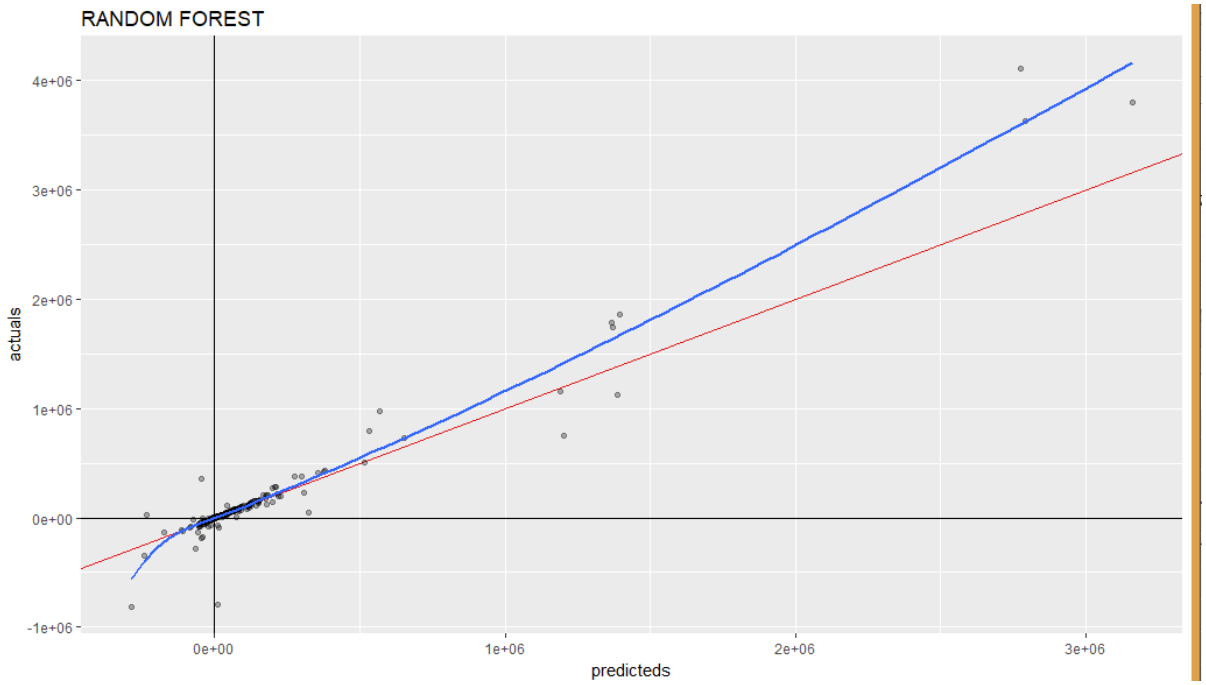

Figure 2. 5 Random Forest Accuracy Curve

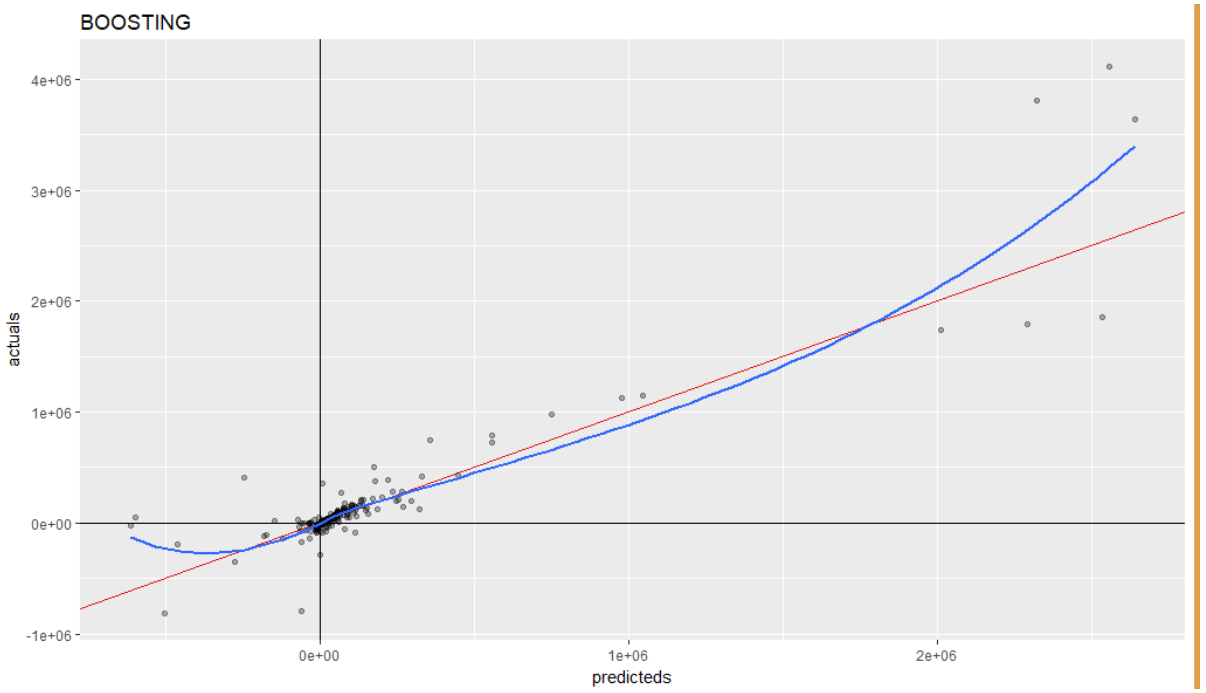

Figure 2. 6 Boosting curve