Human Resources Analytics

Name - Manasi Sheth Class - Stats 6620 Section - 03

# Project Proposal -

Data - Human Resource Analytics Data

Source - Kaggle - <https://www.kaggle.com/c/sm/data>

Implementation - The dataset is about staff attrition. The dataset asks the question “Why our best and most experienced employees prematurely?” Staff attrition can never be good for business. High Staff turnover rate can be very costly. Using data-driven approach, managers would be able to predict which employees are likely to leave the company soon. Acting upon the predictions, the managers would be then able to persuade employees who are likely to leave .

The dataset comprises of 14999 records with 10 variables. The algorithms that will be perfomed on the dataset will be logistic regression, decision trees and random forest.

# Implementation of Project

## Introduction

Employee attrition is one of the biggest challenges that the companies face.

There are several factors that lead to attrition. While it may not be easy to control all the factors, it may be worth the efforts to look into those factors that seem controllable. Factors such as average number of hours spend per month by the employees, salary, promotions, number of projects which an employee workied on are a few which are easier to manage.

If we are able to extract cut-off levels for some of the above mentioned factors through our analysis, then we should be able to have a better understanding about the factors that are responsible for the employees leaving the company prematurely.

## Project Implementation

### Step - 1 - Collecting the Data

The Human Resources Analytics Dataset is collected from Kaggle at <https://www.kaggle.com/c/sm/data>. This data was donated by Mr. Ludovic Benistant and contains following fields :

* Employee satisfaction level - Satisfaction Level of the Employees in the company which can be between 0 to 1.
* Last evaluation - The score which Employees received in their last evaluation
* Number of projects - The number of projects employees has received
* Average monthly hours - The average monthly hours which employees work
* Time spent at the company - Total years spend by a employee at a company
* Whether they have had a work accident - This field would have answer yes or no for question whether an employee had an accident at work or not.
* Whether they have had a promotion in the last 5 years
* Department - Department in which employee is working
* Salary - If the salary of the employee is “Low”, “Medium” and “High”
* Whether the employee has left - This field is answer to the question - if the employee is still working or not for the company.

The correspdong variable names used in the code are -

satisfaction\_level  
last\_evaluation  
number\_project  
average\_montly\_hours time\_spend\_company  
Work\_accident  
left promotion\_last\_5years . sales  
salary

The outcome variable is “Left” which has values 0 and 1. Hence the models used will be initially Logistic Regression, then the model is improved by Decision Trees and Random Forests.

### Step - 2 - Exploring and Preparing the Data

The data is explored in the report and we see whether we can shine some light in the relationships. The project begins by importing the CSV data file - “HR\_comma\_sep.csv”.

After the file is exported, First, we begin with exploring data on broader sense and obtaining basic information.

## [1] 14999 10

## [1] 14999

## [1] 10

We can see that our data set comprises of 14999 rows and 10 columns. The column names are as listed below -

## [1] "satisfaction\_level" "last\_evaluation"   
## [3] "number\_project" "average\_montly\_hours"   
## [5] "time\_spend\_company" "Work\_accident"   
## [7] "left" "promotion\_last\_5years"  
## [9] "sales" "salary"

Next, we take a look at high-level, non statistical summary of entire data frame i.e. we look at the structure of the data

## 'data.frame': 14999 obs. of 10 variables:  
## $ satisfaction\_level : num 0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...  
## $ last\_evaluation : num 0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...  
## $ number\_project : int 2 5 7 5 2 2 6 5 5 2 ...  
## $ average\_montly\_hours : int 157 262 272 223 159 153 247 259 224 142 ...  
## $ time\_spend\_company : int 3 6 4 5 3 3 4 5 5 3 ...  
## $ Work\_accident : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ left : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ promotion\_last\_5years: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ sales : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...  
## $ salary : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...

From the above results, we can see that we have 2 variables - satisfaction\_level and last\_evaluation of data type number, Then the variables - number\_project, average\_monthly\_hours, time\_spend\_company, work\_accident, left are of datatype integer and sales and salary are of type factor. Next we look at the statistical summary of the data set.

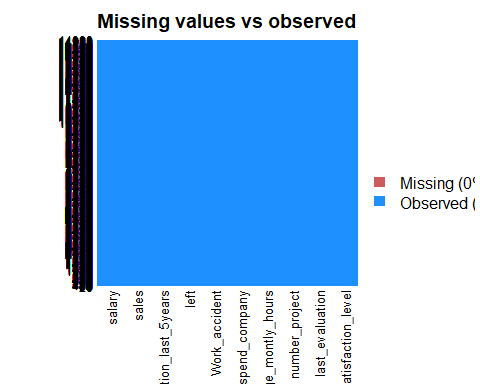
## satisfaction\_level last\_evaluation number\_project average\_montly\_hours  
## Min. :0.0900 Min. :0.3600 Min. :2.000 Min. : 96.0   
## 1st Qu.:0.4400 1st Qu.:0.5600 1st Qu.:3.000 1st Qu.:156.0   
## Median :0.6400 Median :0.7200 Median :4.000 Median :200.0   
## Mean :0.6128 Mean :0.7161 Mean :3.803 Mean :201.1   
## 3rd Qu.:0.8200 3rd Qu.:0.8700 3rd Qu.:5.000 3rd Qu.:245.0   
## Max. :1.0000 Max. :1.0000 Max. :7.000 Max. :310.0   
##   
## time\_spend\_company Work\_accident left   
## Min. : 2.000 Min. :0.0000 Min. :0.0000   
## 1st Qu.: 3.000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median : 3.000 Median :0.0000 Median :0.0000   
## Mean : 3.498 Mean :0.1446 Mean :0.2381   
## 3rd Qu.: 4.000 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :10.000 Max. :1.0000 Max. :1.0000   
##   
## promotion\_last\_5years sales salary   
## Min. :0.00000 sales :4140 high :1237   
## 1st Qu.:0.00000 technical :2720 low :7316   
## Median :0.00000 support :2229 medium:6446   
## Mean :0.02127 IT :1227   
## 3rd Qu.:0.00000 product\_mng: 902   
## Max. :1.00000 marketing : 858   
## (Other) :2923

We can see distribution of variables in the above output. We can see that there are no NA’s present in the data, hence we can say that there is no missing data in our dataset.

To confirm if there is no missing data in the dataset, Amelia package is used which has a special plotting function missmap() that will plot hr\_data dataset and highlight missing values. We also confirm from sapply function that there are no missing values.

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.5, built: 2018-05-07)  
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##



## satisfaction\_level last\_evaluation number\_project   
## 0 0 0   
## average\_montly\_hours time\_spend\_company Work\_accident   
## 0 0 0   
## left promotion\_last\_5years sales   
## 0 0 0   
## salary   
## 0

We can see that there are no missing values in the datasest from the missmap function and sapply function. Next, we find out how many values are unique in the dataset

## satisfaction\_level last\_evaluation number\_project   
## 92 65 6   
## average\_montly\_hours time\_spend\_company Work\_accident   
## 215 8 2   
## left promotion\_last\_5years sales   
## 2 2 10   
## salary   
## 3

We can see that there are very less distinct values in the dataset. To model the output variable “left”, the variable is converted into factor.

#### Exploratory Data Aanalysis

We start the Exploratory Data Analysis by seeing the exploring the categorical variable “Left”. In contrast to data, categorical data is typically examined using tables rather numeric than summary statistics. With the help of “table()” function, a one-way table is generated for “left” variable.

##   
## 0 1   
## 11428 3571

We can see that the number of employees working in the company are 11428 and number of employees who left the company are 3571. We can visualize the same data with histogram as follows -

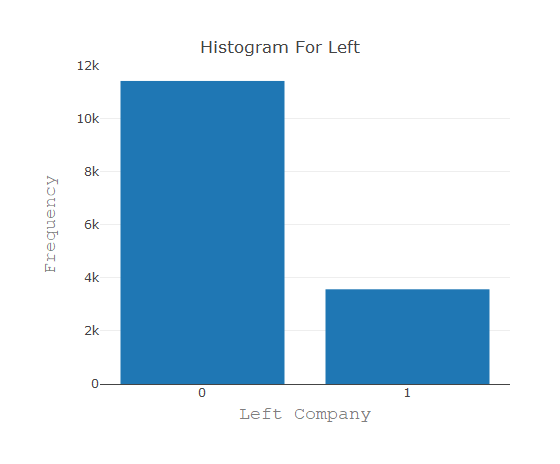
##   
## Attaching package: 'plotly'

## The following object is masked from 'package:Hmisc':  
##   
## subplot

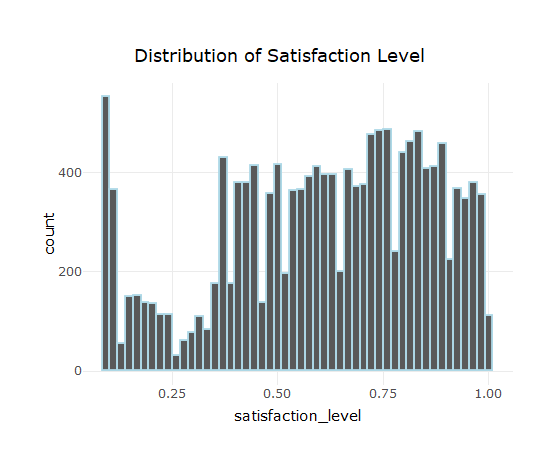
## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

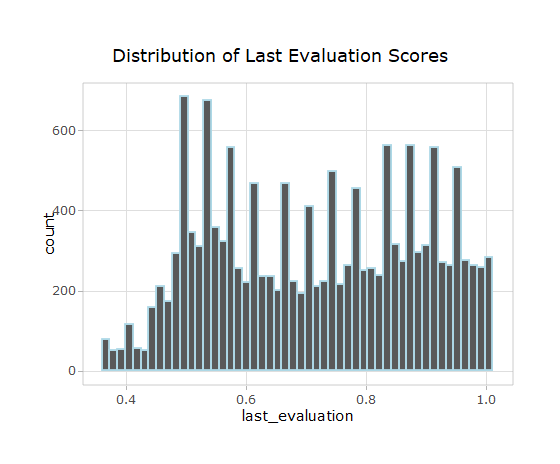


Next, we distribution of the variable “Satisfaction Level”.



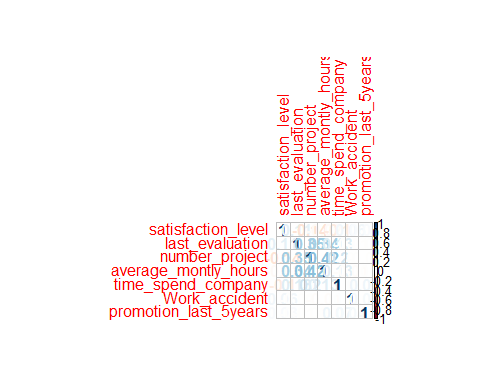
From the histogram, we can see that the maximum counts of satisfaction level are for values approximately equal to 0.09. The minimum counts of satisfaction level are for values approximately equal to 0.25. There are less than 200 records with value 1.

Next, we see distribution of variable last\_evaluation\_score.



We can see from the histogram that the minimum value of last evaluation score is 0.36 and 77 employeees had that score. The maximum number of records are for value 0.497. The maximum last evaluation score is 1.005.

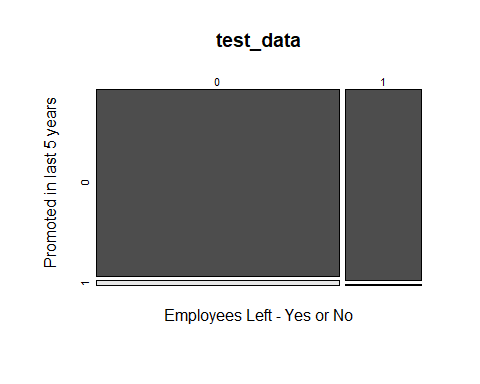
Then we answer the questions if the numeric variables are correlated.



We can see from the correlation matrix that all the variables are positively correlated to each other. The variables - last evaluation score and number of projects have moderate positive correlation with the variable average monthly hours.

Next, we see what is the relation between employees leaving and getting promoted in last 5 years

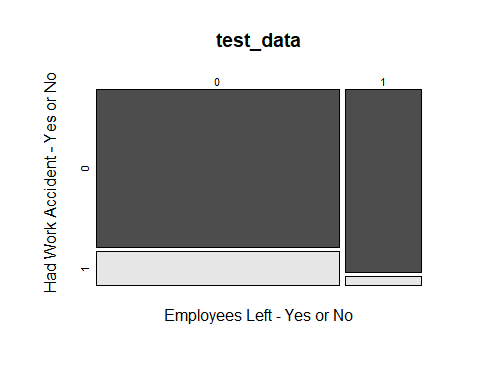
##   
## 0 1  
## 0 11128 300  
## 1 3552 19



From the tabular output and mosaic plot we can see that if the majority of the employees that are not promoted in last 5 years tend to not leave the job. However there are 3352 employees who left the job even if they were not promoted. Out of the total employees who did not leave the job after getting promoted were 300 and 19 employees who were promoted in last 5 years left the job. We can say from the graphs that promotion in last 5 years does not play mamjor role in determining if the employees would leave the company or not.

Then we explore the relationship between employees leaving and having work accident.

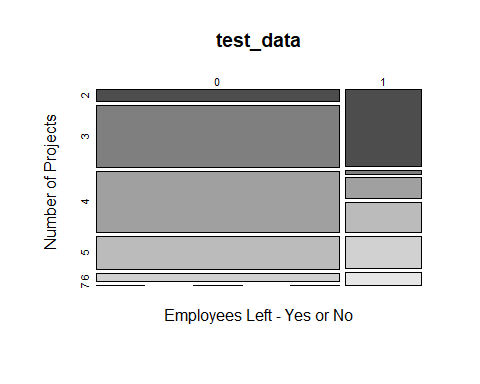
##   
## 0 1  
## 0 9428 2000  
## 1 3402 169



We can see from the mosaic plot and tabular output that the majority of the workers who have work accident tend to not leave the job. We can see there are 2000 such records, also there are 169 records of employees who had work accident and left the job.

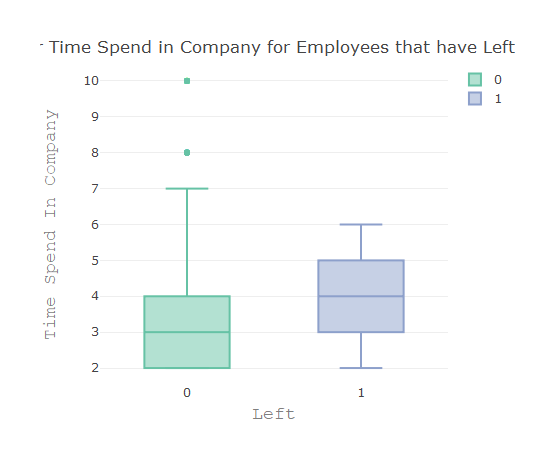
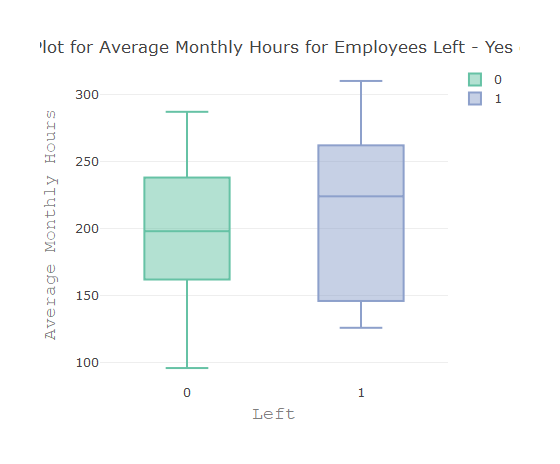
The question would be to see, generally after how many projects employees tend to leave the company?

##   
## 2 3 4 5 6 7  
## 0 821 3983 3956 2149 519 0  
## 1 1567 72 409 612 655 256

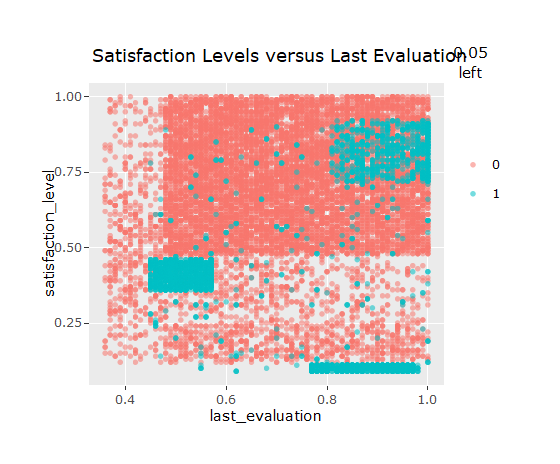


We can see from the mosaic plot and the tabular output, that the majority of the employees who left the company left it after doing 2 projects. Minimum number of employees left the company after doing three projects. We can also see that there are no employees who did not left but also worked on 7 projects. There are however 256 employees that worked on 7 projects and left the company. So we can safely say that if the employees work on 7 projects, then they tend to leave the company.

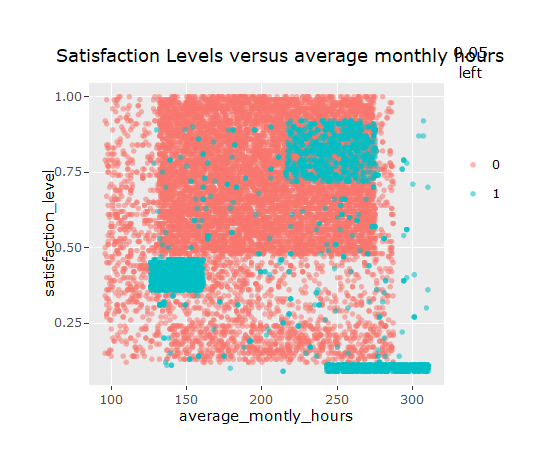
We have compared numereic variables with our outcome variable using box plots.



Following are the observations from the box plot. From the first plot we can see that if the satisfaction level is low then the employees have left the job. From the second box plot we can that the median value of the last evaluation score is high for the people who have left the job. From the third box plot we can see that people how have left the job tend to put in more hours. From the fourth plot we can see that there is no specific trend with respect to time spent in the company.



The above plot shows the satisfaction levels with respect to the last evaluation. We can see that teal dots, for employees who have left, are clustered in some regions.



The above plot shows the satisfaction levels with respect to the average monthly hours. We can see that teal dots, for employees who have left, are clustered in some regions.

Next, the data is split into test and training dataset to build the logistic regression model and to evaluate the performance of the model on new data. The data is randomized, and the first 90% is used for training and the rest of the data is used for testing.

## Step - 3 - Training a logistic regression model on the data

In this section we being by training the logistic regression model using glm function.

The logistic regression model looks as follows:

##   
## Call: glm(formula = left ~ ., family = binomial(link = "logit"), data = hr\_train)  
##   
## Coefficients:  
## (Intercept) satisfaction\_level last\_evaluation   
## -1.485290 -4.116651 0.666263   
## number\_project average\_montly\_hours time\_spend\_company   
## -0.304008 0.004732 0.263469   
## Work\_accident promotion\_last\_5years saleshr   
## -1.499329 -1.359880 0.263300   
## salesIT salesmanagement salesmarketing   
## -0.207982 -0.535651 -0.055261   
## salesproduct\_mng salesRandD salessales   
## -0.221588 -0.534263 -0.025812   
## salessupport salestechnical salarylow   
## 0.041641 0.100681 1.890103   
## salarymedium   
## 1.383615   
##   
## Degrees of Freedom: 13498 Total (i.e. Null); 13480 Residual  
## Null Deviance: 14820   
## Residual Deviance: 11590 AIC: 11620

We can see the intercept values and the values of slopes for different variables in the data set. The model has 13498 degress of freedom and the AIC value is 11620.

By using function summary(), we obtain the results of the model.

##   
## Call:  
## glm(formula = left ~ ., family = binomial(link = "logit"), data = hr\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2313 -0.6663 -0.4037 -0.1189 3.0298   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.4852896 0.2027110 -7.327 2.35e-13 \*\*\*  
## satisfaction\_level -4.1166506 0.1030029 -39.966 < 2e-16 \*\*\*  
## last\_evaluation 0.6662632 0.1568365 4.248 2.16e-05 \*\*\*  
## number\_project -0.3040083 0.0224704 -13.529 < 2e-16 \*\*\*  
## average\_montly\_hours 0.0047321 0.0005449 8.684 < 2e-16 \*\*\*  
## time\_spend\_company 0.2634692 0.0163591 16.105 < 2e-16 \*\*\*  
## Work\_accident -1.4993285 0.0933577 -16.060 < 2e-16 \*\*\*  
## promotion\_last\_5years -1.3598804 0.2658466 -5.115 3.13e-07 \*\*\*  
## saleshr 0.2633004 0.1374765 1.915 0.055462 .   
## salesIT -0.2079819 0.1290126 -1.612 0.106939   
## salesmanagement -0.5356511 0.1703439 -3.145 0.001664 \*\*   
## salesmarketing -0.0552614 0.1399913 -0.395 0.693028   
## salesproduct\_mng -0.2215877 0.1383656 -1.601 0.109274   
## salesRandD -0.5342629 0.1515505 -3.525 0.000423 \*\*\*  
## salessales -0.0258120 0.1081589 -0.239 0.811378   
## salessupport 0.0416408 0.1152865 0.361 0.717954   
## salestechnical 0.1006805 0.1123383 0.896 0.370132   
## salarylow 1.8901026 0.1327844 14.234 < 2e-16 \*\*\*  
## salarymedium 1.3836153 0.1335604 10.359 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14823 on 13498 degrees of freedom  
## Residual deviance: 11585 on 13480 degrees of freedom  
## AIC: 11623  
##   
## Number of Fisher Scoring iterations: 5

Now we have run the ANOVA function on the model to analyze the table of deviance.

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: left  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 13498 14823   
## satisfaction\_level 1 2051.38 13497 12772 < 2.2e-16 \*\*\*  
## last\_evaluation 1 18.11 13496 12754 2.084e-05 \*\*\*  
## number\_project 1 89.79 13495 12664 < 2.2e-16 \*\*\*  
## average\_montly\_hours 1 82.71 13494 12581 < 2.2e-16 \*\*\*  
## time\_spend\_company 1 164.76 13493 12416 < 2.2e-16 \*\*\*  
## Work\_accident 1 344.24 13492 12072 < 2.2e-16 \*\*\*  
## promotion\_last\_5years 1 60.79 13491 12011 6.363e-15 \*\*\*  
## sales 9 91.91 13482 11919 6.723e-16 \*\*\*  
## salary 2 334.14 13480 11585 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

From the anova output we can see that all the variables are significant. Since all the variables are significant, we will not remove any variables and this will be the final model. The difference between the null deviance and the residual deviance shows how our model is doing against the null model. The wider this gap, the better. Analyzing the ANOVA output we can see that as we add each variable one at a time the residual deviance has dropped.

## Step - 4 - Evaluating Model Performance

##### Logistic Regression

Now, we would like to see the model performance for predicting with new data set. By setting the parameter type=‘reponse’, R will output probabilities in form of P(y=1|X). Our decision boundary is 0.5. If P(y=1|X)>0.5 then y=1 otherwise y=0.

## [1] "Accuracy 0.796666666666667"

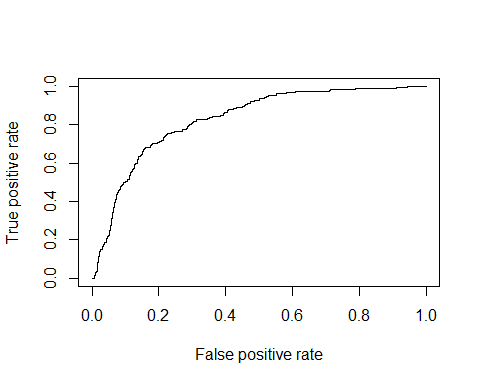
The accuracy of the model is 0.79 which is a good result. We are going to plot the ROC curve and calculate the AUC (area under the curve), which are typical performance measurements for the binary classifiers. As a rule of thumb, a model with good predictive ability should have an AUC closer to 1, than to 0.5.

## Warning: package 'ROCR' was built under R version 3.4.4

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess



A popular way for summarizing the discrimination ability of the model is to report the area under the ROC curve. In a model with good discrimination ability the ROC curve will go closer to the left corner. We have calculated the AUC to estimate the model’s predictive ability.

## [1] 0.8288136

Since the AUC is 0.82 we can say that model has good predictive abilities.

However this is result is somewhat dependent on the manual split of the data we did earlier. So we will be looking at improving the model performance.

## Step 5 - Improving Model Performance

To improve the model performance we have first constructed the decision trees, then boosted them and then created a random forest model.

#### Training and Testing Data for Decision Trees

The same training and testing dataset is used for decision tree.

#### create a model for decision trees

The decision tree is build using c5.0 algorithm. The model is created by excluding the ‘left’ class variable from the training data set. The ‘left’ variable is set as target factor vector for classification.

The basic data about the tree is as follows:

##   
## Call:  
## C5.0.default(x = hr\_c50\_train[-7], y = hr\_c50\_train$left)  
##   
## Classification Tree  
## Number of samples: 13499   
## Number of predictors: 9   
##   
## Tree size: 41   
##   
## Non-standard options: attempt to group attributes

From the output it can be seen that the tree size 41, which means that the tree is 41 decisions deep.

The confusion matrix for the tree is as follows:

The confusion matrix has displayed the incorrectly classified records. It can be seen that out of 13499 records, 246 records are incorrectly classified giving error rate of 1.8%. 28 values which were actually employees in the company were wrongly classified as left and 218 employees who left were misclassified as working.

#### Decision Trees prediction

Using the predict function the decision tree is applied to test data set.

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1500   
##   
##   
## | predicted default   
## actual default | 0 | 1 | Row Total |   
## ---------------|-----------|-----------|-----------|  
## 0 | 1142 | 3 | 1145 |   
## | 0.761 | 0.002 | |   
## ---------------|-----------|-----------|-----------|  
## 1 | 19 | 336 | 355 |   
## | 0.013 | 0.224 | |   
## ---------------|-----------|-----------|-----------|  
## Column Total | 1161 | 339 | 1500 |   
## ---------------|-----------|-----------|-----------|  
##   
##

From the confusion matrix it can be seen that out of 1500 records 22 records were misclassified. This resulted in an accuracy of 98.5 and an error rate of 1.5%. 3 values which were actually employees in the company were wrongly classified as left and 19 employees who left were misclassified as working.

#### boosted decision trees

Next we have used boosted decision trees to improve the model performance of decision trees. The number of trials are set to 10.

##   
## Call:  
## C5.0.default(x = hr\_c50\_train[-7], y = hr\_c50\_train$left, trials = 10)  
##   
## Classification Tree  
## Number of samples: 13499   
## Number of predictors: 9   
##   
## Number of boosting iterations: 10   
## Average tree size: 45.1   
##   
## Non-standard options: attempt to group attributes

We can see that the average tree size has increased from 41 to 45.1 by using boosted decision trees.

The confusion matrix has displayed the incorrectly classified records. It can be seen that out of 13499 records, 176 records are incorrectly classified giving error rate of 1.3%. 28 values which were actually employees in the company were wrongly classified as left and 148 employees who left were misclassified as working.

Using the predict function the boosted decision tree is applied to test data set.

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1500   
##   
##   
## | predicted default   
## actual default | 0 | 1 | Row Total |   
## ---------------|-----------|-----------|-----------|  
## 0 | 1141 | 4 | 1145 |   
## | 0.761 | 0.003 | |   
## ---------------|-----------|-----------|-----------|  
## 1 | 14 | 341 | 355 |   
## | 0.009 | 0.227 | |   
## ---------------|-----------|-----------|-----------|  
## Column Total | 1155 | 345 | 1500 |   
## ---------------|-----------|-----------|-----------|  
##   
##

From the confusion matrix it can be seen that out of 1500 records 18 records were misclassified. This resulted in an accuracy of 98.8 and an error rate of 1.2%. 4 values which were actually employees in the company were wrongly classified as left and 14 employees who left were misclassified as working.

#### create a model for random forests

Next we evaluate random forsests to improve the model. We keep the same training and test dataset.

#### Training and Testing Data For Random Forests

The random forest model is fitted using randomForest() function in the randomForest package.

## 11.02 sec elapsed

##   
## Call:  
## randomForest(formula = left ~ ., data = hr\_rf\_train)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 0.81%  
## Confusion matrix:  
## 0 1 class.error  
## 0 10265 18 0.001750462  
## 1 91 3125 0.028296020

The output shows that the random forest included 500 trees and tried 3 variables at each split. The out-of-bag error rate is 0.81%, which is an unbiased estimate of the test set error. The error rate in the confusion matrix is same 0.8%.

#### Prediction of data using Random Forests

The random forest performance is evaluated using the predict() function.

## [1] "Accuracy 0.994666666666667"

The accuracy of the random forest model is 99.4%, which is higher than logistic regression, decision tree and boosted decision tree.

# Conclusion

# Appendix