PREDICTING EMPLOYEE CHURN USING HR-EMPLOYEE DATA

BUAN 6356
Project Report

By
Abinitha Shivakumar
Nikitha Kodathala
Pooja Nitin Banthia
Sneha Liza George
Sumit Paul

Project supervisor: Prof. Sourav Chatterjee
Business Analytics with R
University of Texas at Dallas

November 2018

Table of Contents

Background:

An organization's biggest asset is the employees that work for it. The people within the organization is what can make it grow and thrive in a competitive market or dwindle and crash. Therefore, having the right people, the right goals, and the right satisfaction among an organization's people are very important for the success of the organization and everyone within it. But why do people leave an organization? Why does someone want to leave a company that he or she works for and go somewhere else to work? What factors influence an employee to leave an organization, and can an employee's leaving an organization be predicted based on the factors that can be measured on the employee? Many organizations often deal with employee or worker turnover. Employee turnover is the act of replacing an old employee with a new employee. There are many factors that cause an employee to leave an organization or even get fired. The factors that influence an employee to leave might be the salary level, job satisfaction, employee position within the company, and even the type of department or field of work. If an organization can use the measured factors to predict whether a person will leave the organization or not, the organization can save millions of dollars per year by either letting the employee go early or find ways to satisfy the employee to make him or her stay within the organization and provide more value overall.

Many companies would benefit from analysis about their employees' work attributes. Using those attributes to predict if an employee will leave the company or have varied productivity can be useful for companies when deciding to let go of an employee or give an employee a promotion. Practically, every company which utilizes employees for work, sales, and production can benefit from doing analysis on employee attribute information and save/increase profits by being able to correctly predict which employees to get rid of or which employees to promote to better positions. Just to name a few companies that can benefit from this kind of analysis include Walmart, Target, USAA, State Farm, etc. All industries ranging from technology to sales can utilize this kind of analysis.

The dataset that our group will utilize for the project is an employee churn dataset that was found on Kaggle. The dataset has 14,999 rows and 10 columns. Each row represents a different employee, and the columns represent the satisfaction level, last evaluation, number of projects, average monthly hours, time spent with company, work accidents, left (left the company), promotion within last 5 years, sales, and salary.

Our group was looking for a dataset in which there was an ample number of unique rows to where the dataset could be properly split into training and validation datasets. This is because the predictive models need enough training data in order to increase accuracy, and at the same time the validation data must be sufficient enough to see results. We also wanted a dataset that had enough employee attributes related to the employee's job, and we wanted to use all those attributes to predict one column within the dataset which represents whether the employee left the job or not. This is why we found that the human resources dataset is sufficient for analysis, building predictive models from what we have learned in class, and getting proper results.

Objective:

The objective of this project is to analyze the human resources dataset and find the cause of employee turnover within the dataset, as well as predict if an employee will leave the organization or not based on the other attributes about the employee. In terms of the dataset that we will be using, our group will be predicting if an employee has left the company (left = 1) using all the other columns within the dataset.

Data Exploration:

- We began exploring the data by looking at the structure of the Employee data frame by using the R function str().

```
> str(Emp.df)
'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 ...
 $ Departments              : 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 ...
```

- As our dataset has only 10 variables, we have not reduced the dimension of the dataset.

- We made sure that no columns in the dataset had any missing values in them

```
> apply(Emp.df, 2, function(x) any(is.na(x)))
  satisfaction_level  last_evaluation    number_project  average_montly_hours
       FALSE               FALSE              FALSE              FALSE
  time_spend_company     Work_accident    left          promotion_last_5years     Departments
       FALSE                  FALSE      FALSE             FALSE              FALSE
  salary
     FALSE
```

- Turnover rate depicts the percentage of employees that have left organization in a given period of time which is calculated as the ratio of employees who left to the total number of employees where 1 indicates inactive and 0 indicates active

- We then created some data visualizations using the dataset to understand the dataset better.

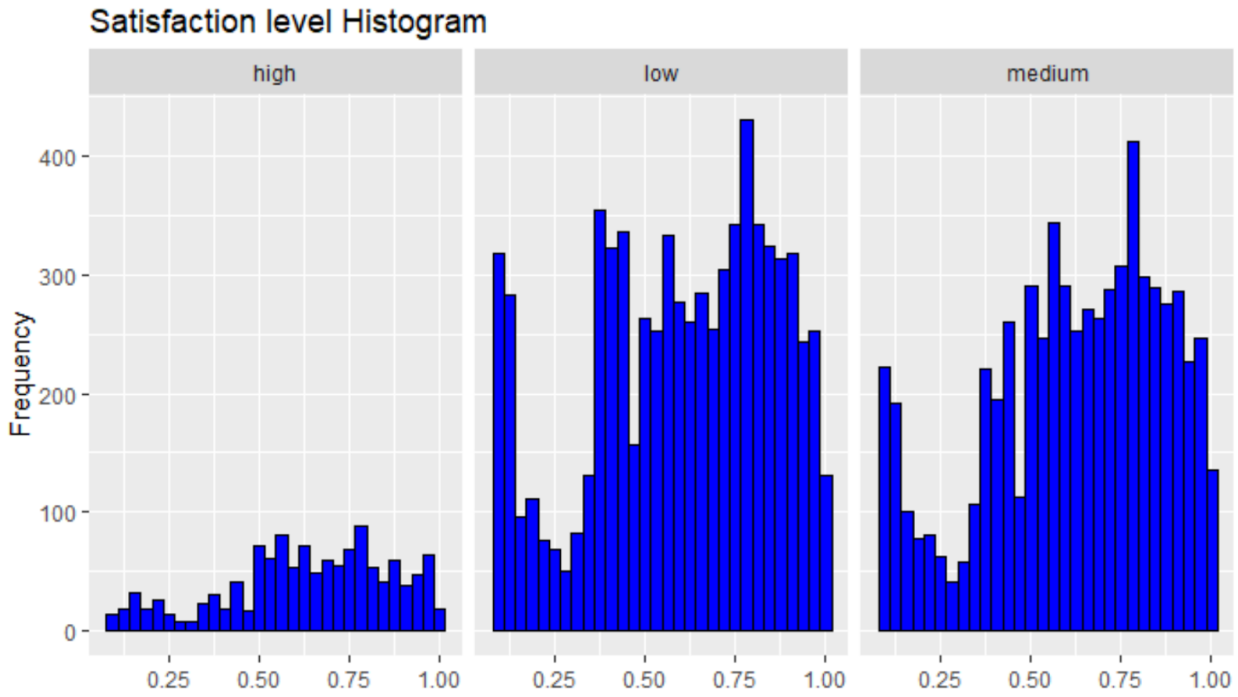

Figure 1

- The distribution of satisfaction level for each class of Salary Ranges is almost the same
- Employees having high salary have a little bit more satisfaction level compared to that of medium and low salary employees
- Employees with High salary are less in number

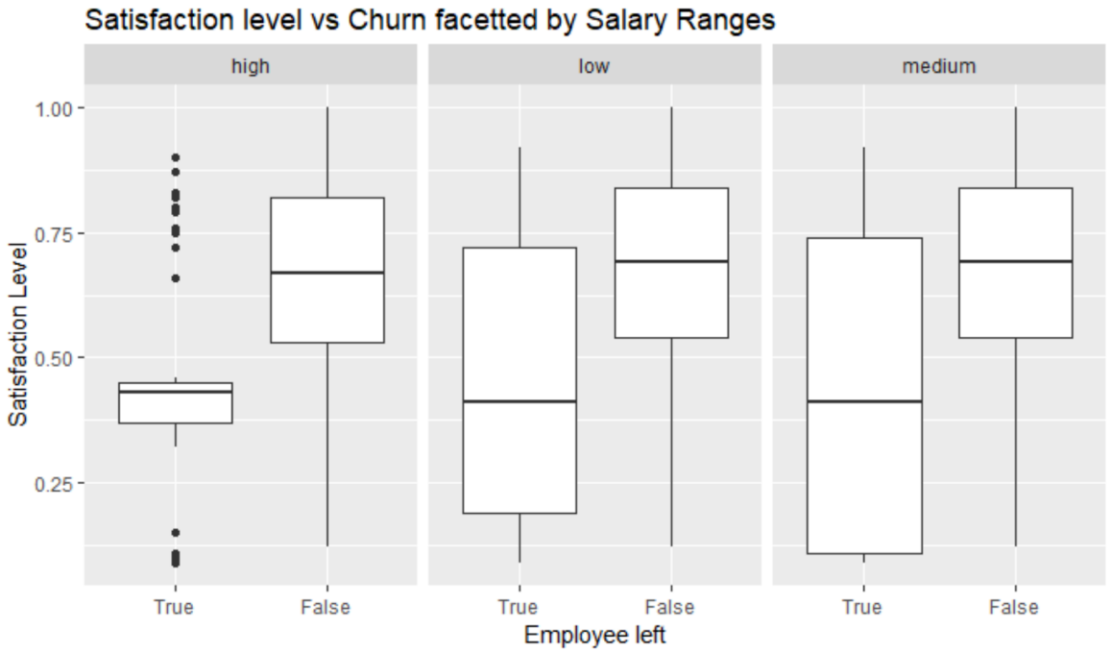

Figure 2

- As expected, the satisfaction level of employees who churned was less compared to those who stayed

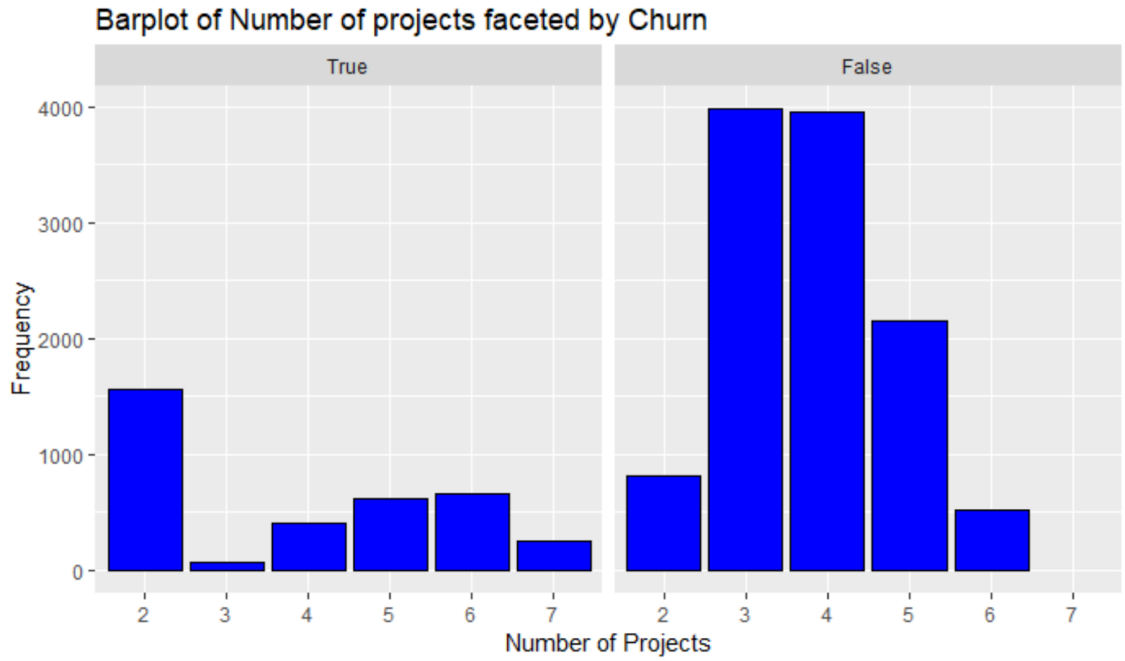

Figure 3

- Most employees who churned out had 2 projects whereas those who stayed back had 3 or 4 projects

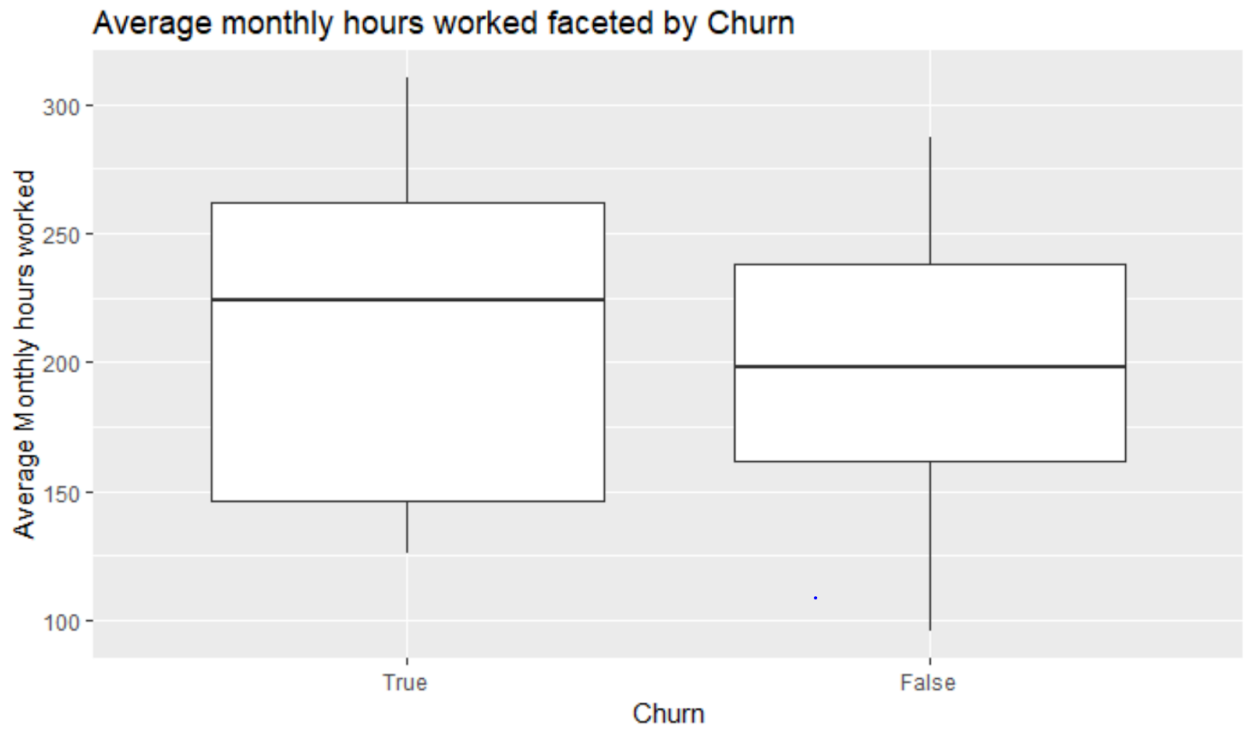

Figure 4

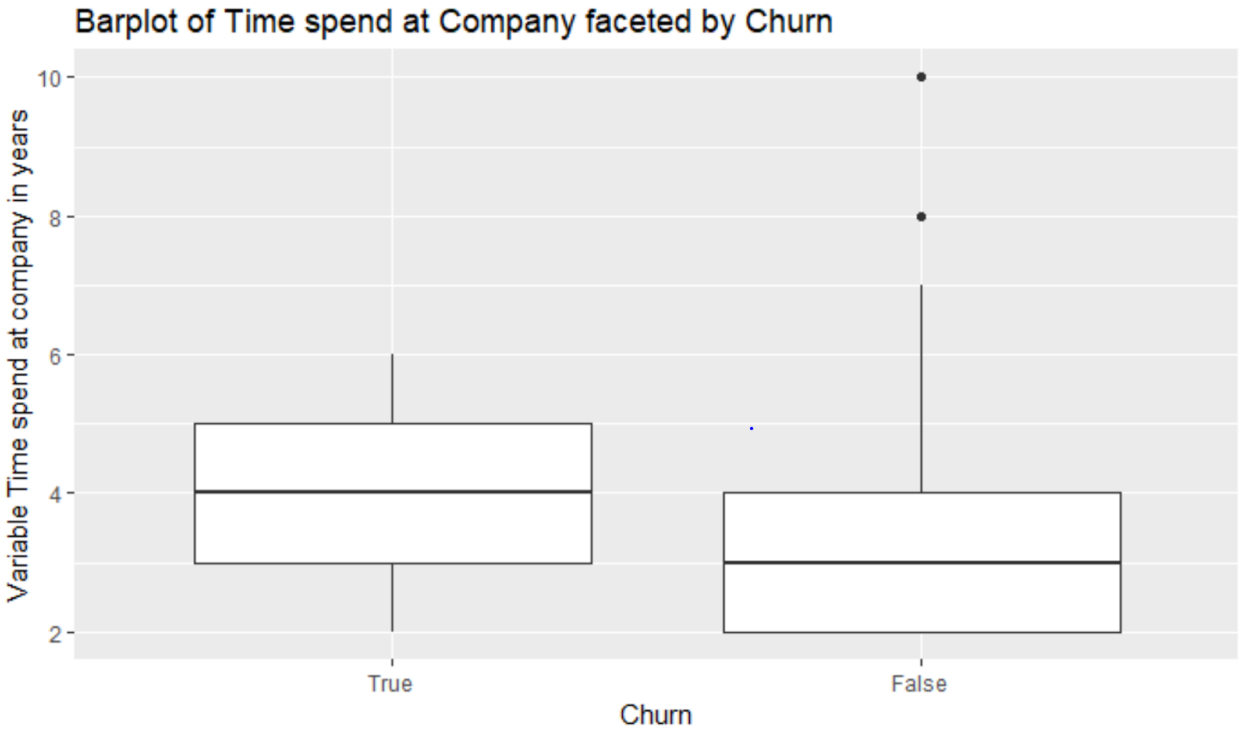

Figure 5

- Employees who left the company worked more time than those who did not leave, hence it might be possible that they left because they were over pressurized and stressed by their bosses with lots of work

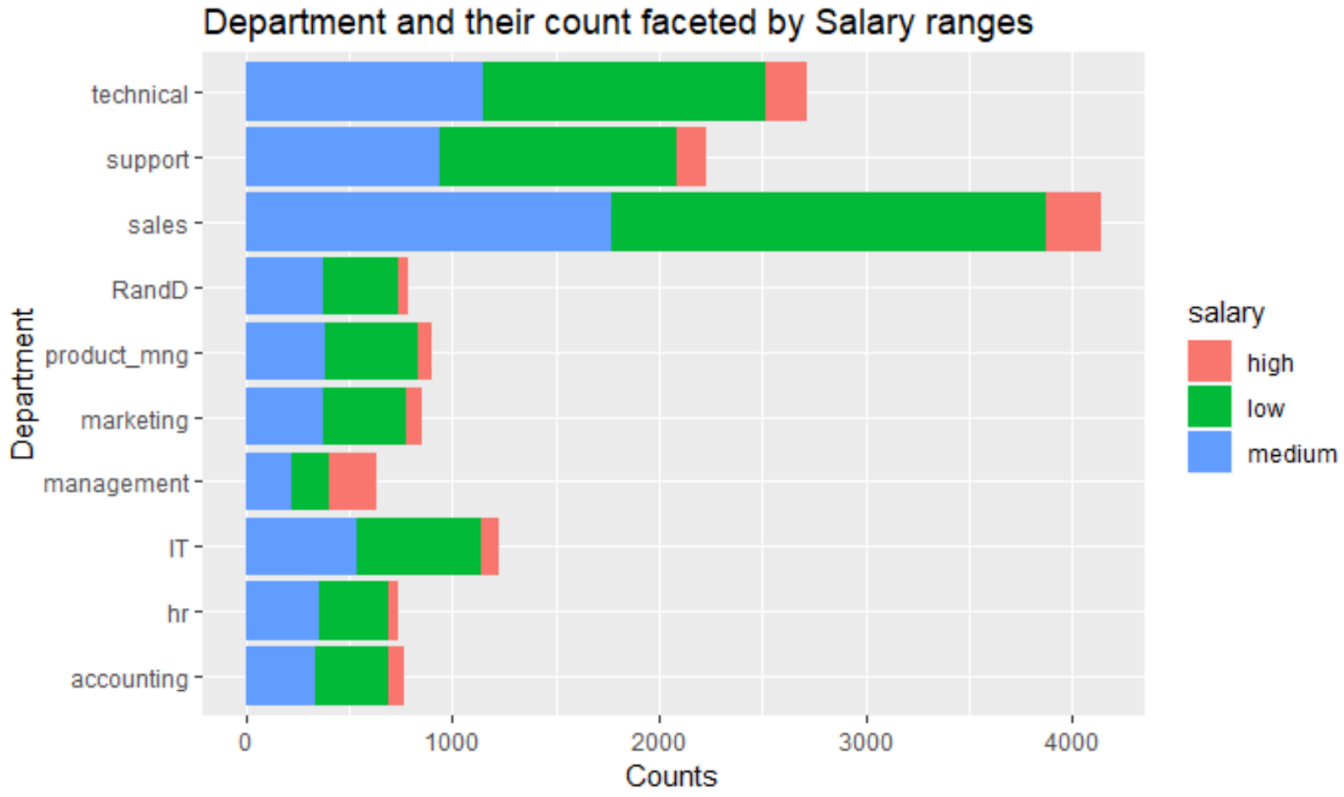

Figure 6

- Majority of the employees work in Sales department then Technical and least in Management

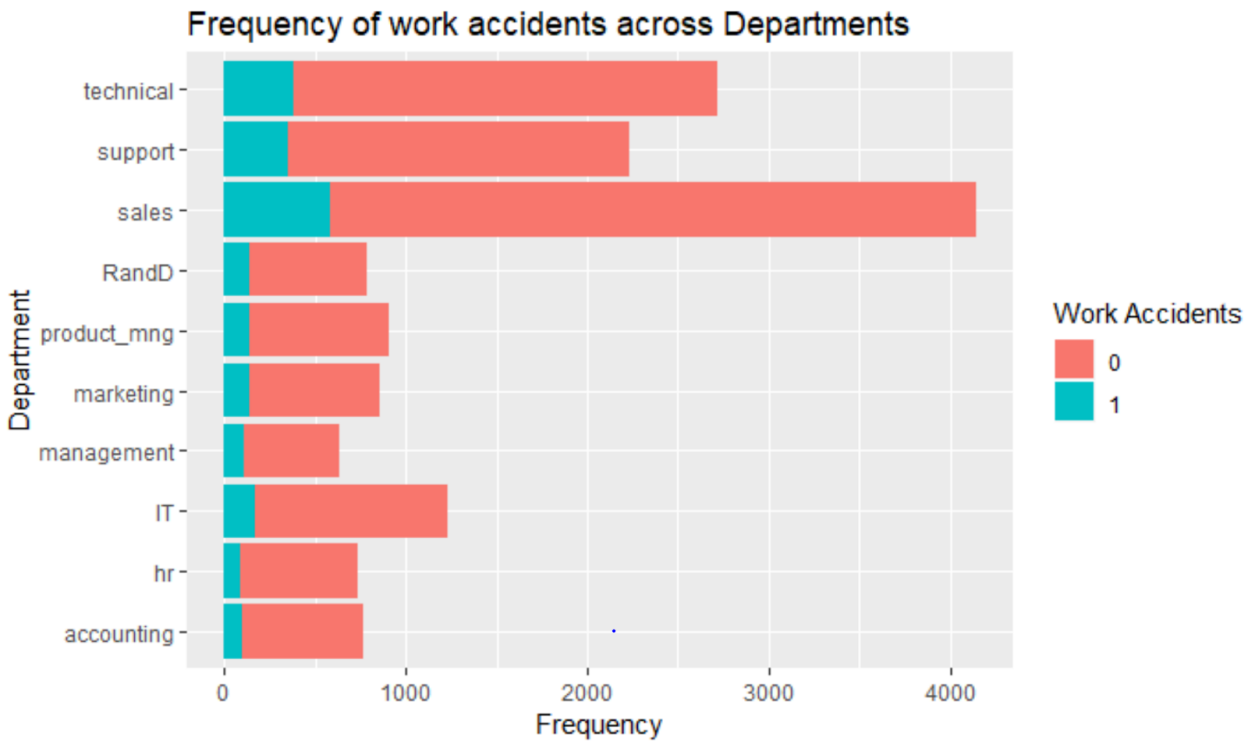

Figure 7

- Work accidents tend to happen the most in Sales followed by Technical

- 76% of the employees stayed and 24% employees left the company
- From the heatmap, there is a positive(+) correlation between number_project, averageMonthlyHours, and last_evaluation.
- This could mean that the employees who spent more hours and did more projects were evaluated highly.
- For the negative(-) relationships, left and satisfaction are highly correlated.
- We can say that people tend to leave a company more when they are less satisfied.

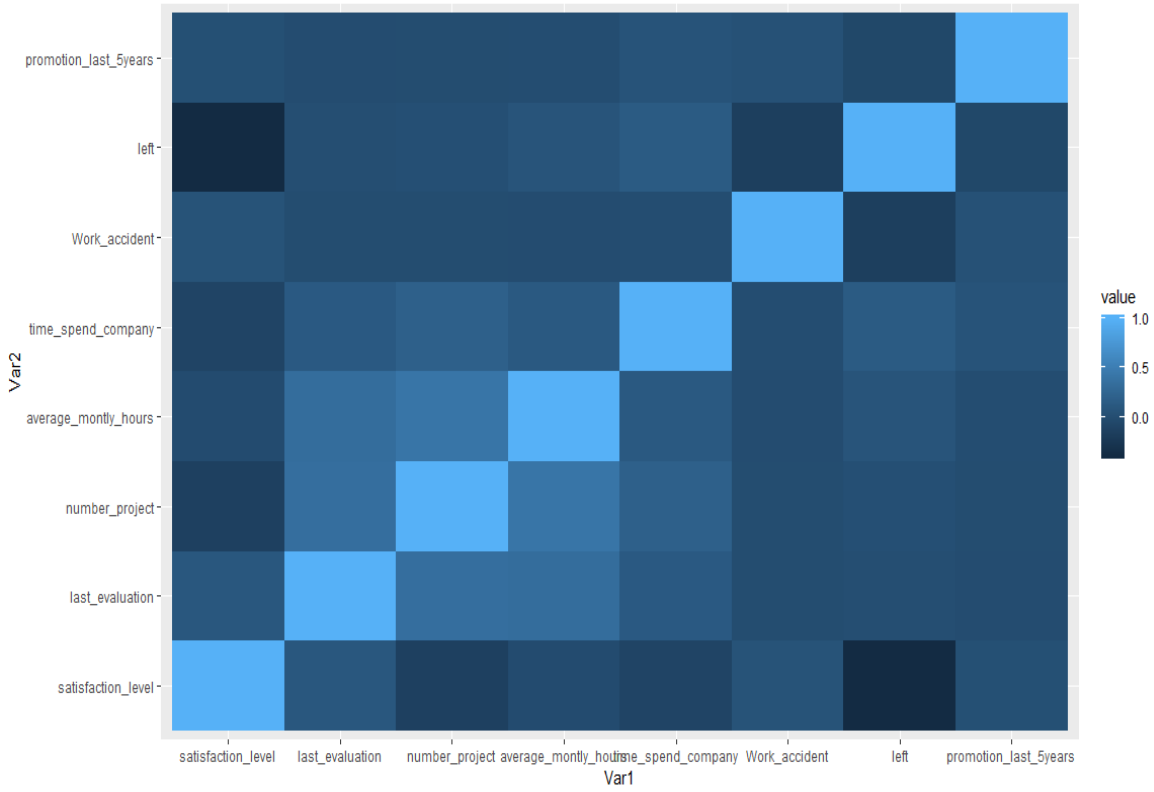

Figure 8

- Employees with 2 projects left as they weren't much satisfied with their work and had low salary
- Employees with 6 and 7 projects may have left because of extra workload
- The employees with 3-5 years of experience are leaving more because of no promotions in last 5 years
- Employees with more than 6 years experience are not leaving because they are promoted and have medium or high salary.
- Those who were promoted did not leave. Many of those who were not promoted left because of low salary and low satisfaction

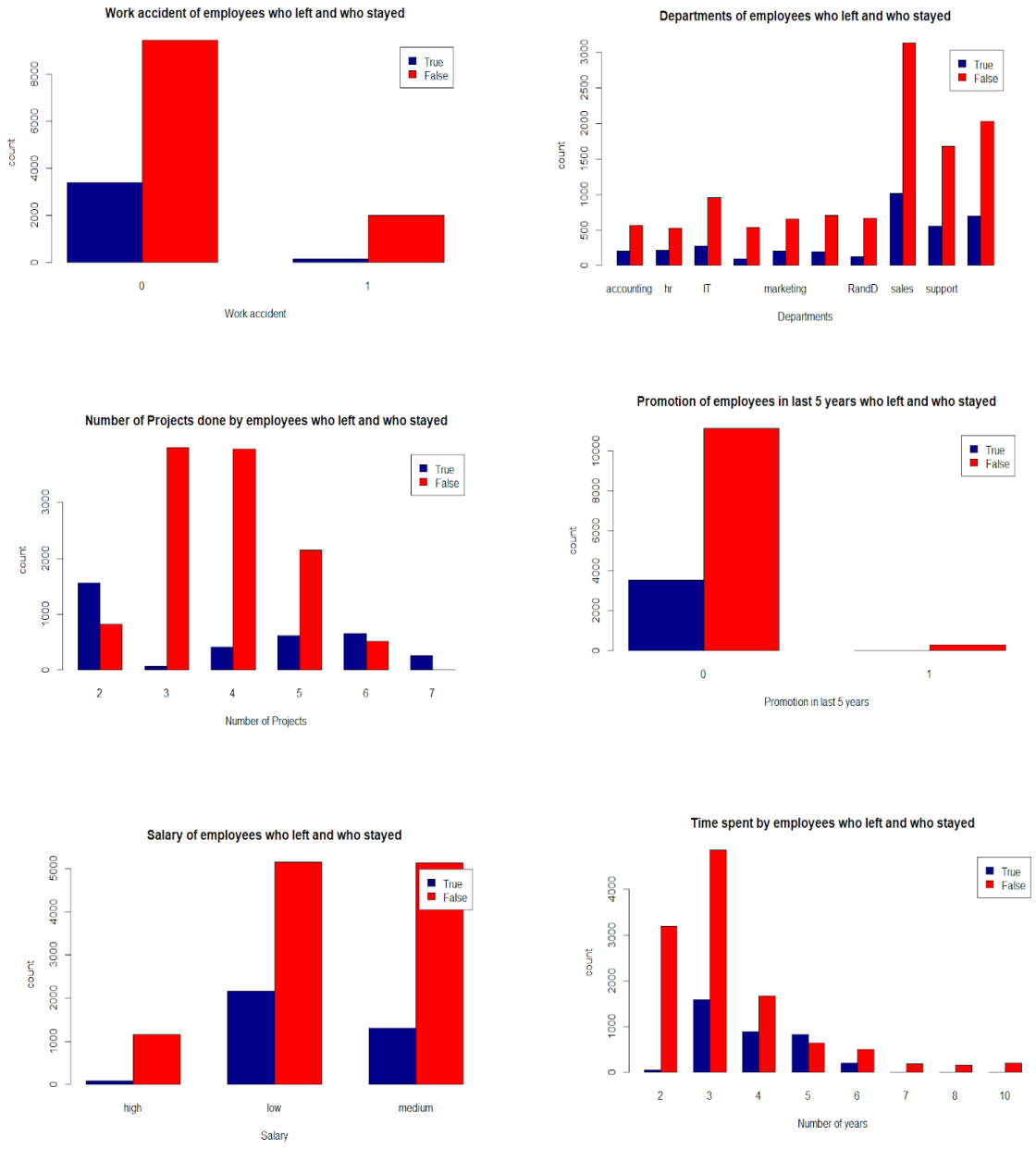

Figure 9

Few Employees who left the company in spite of getting highly paid, following could be the reasons:

- The plots show the count of employees who have left the company
- Even with high salary, many employees had only 2 projects and they left. These employees do not seem to enjoy the work and expect more number of projects
- On the other hand, people with more number of project have extra workload
- They were not promoted in spite of the efforts
- Average satisfaction level is low

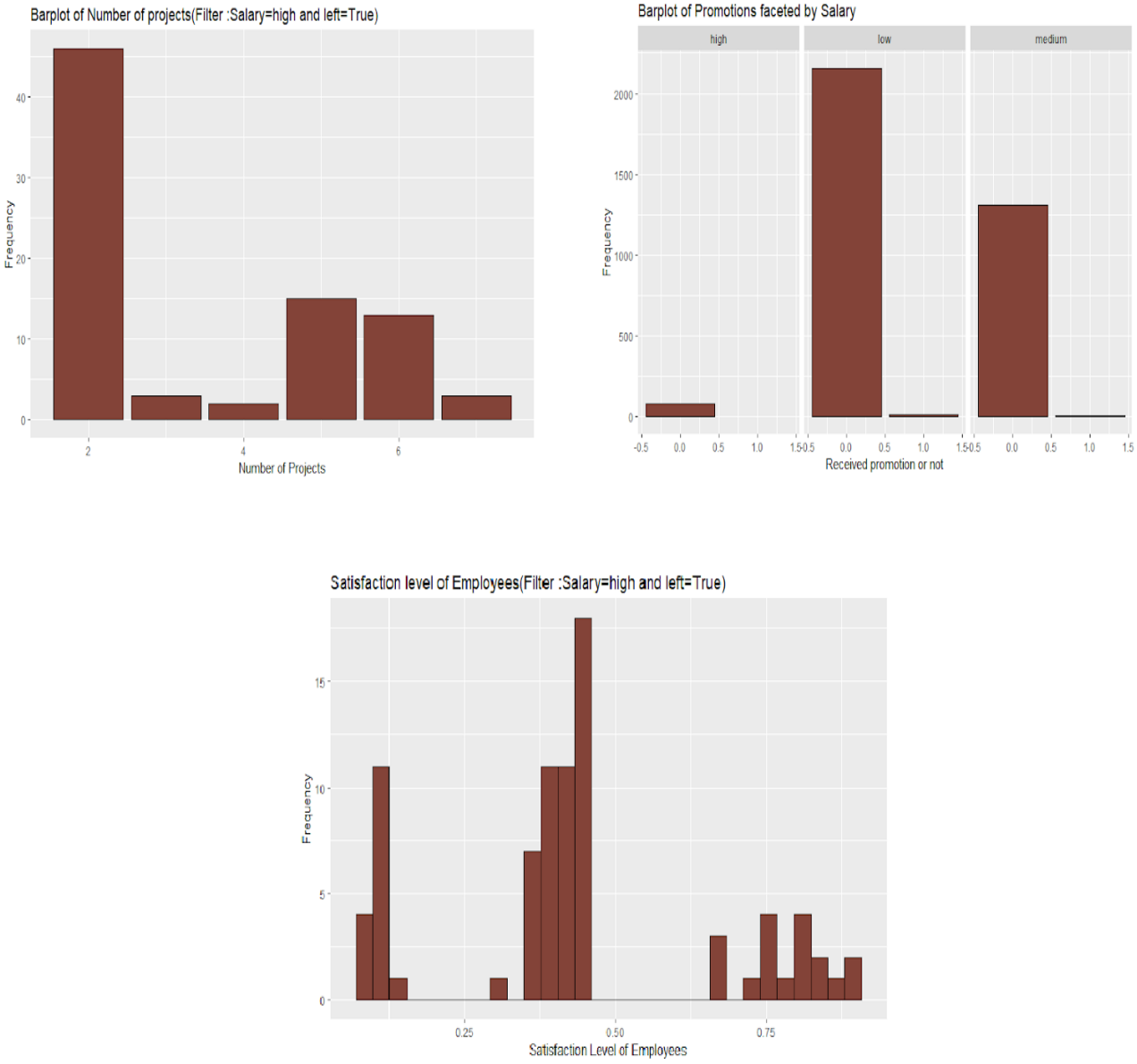

Figure 10

Number of project = 7, all left because of the following possible reasons:

- no promotion was given.
- majority had low or medium salary
- average monthly hours is high
- average satisfaction level is 0.12

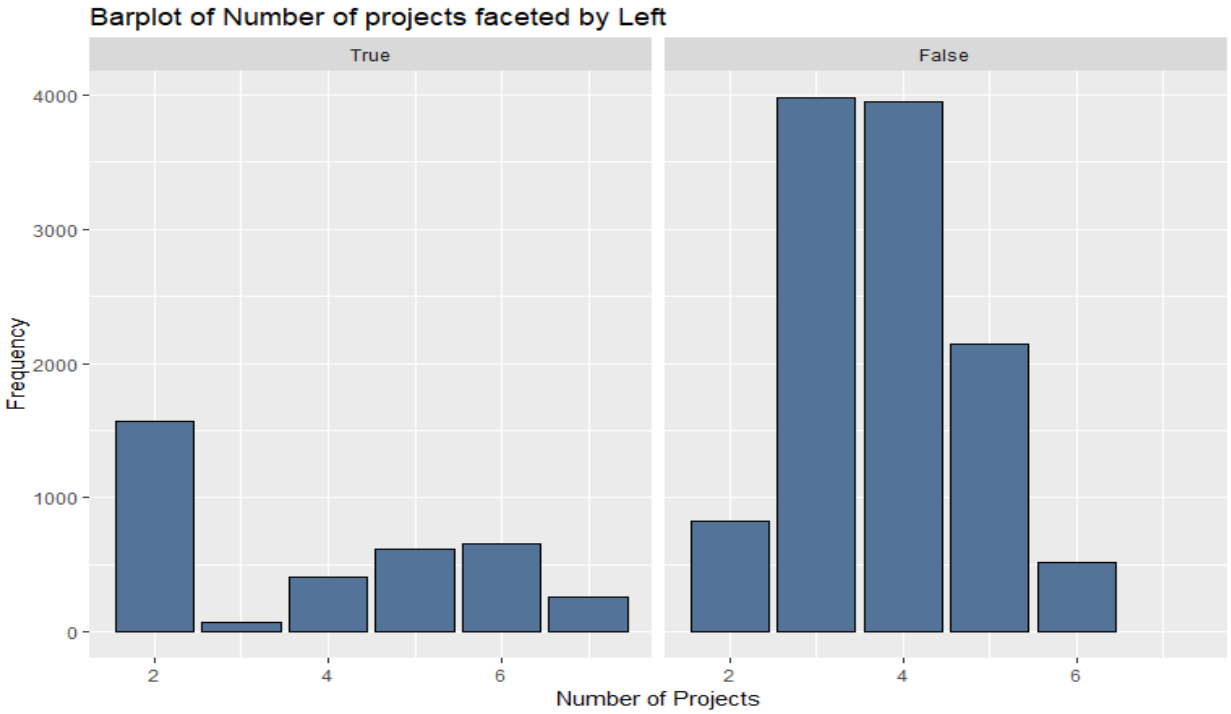

Figure 11

Majority of the employees who continued to work for the same company had 3-5 projects. But we can also see that people with 3 or lesser number of projects left the company. So there could be other deciding factors for the employee turnover.

table(hrm$number_project,hrm$left)

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

From the above table we can see that all the people who have been assigned seven projects have left the company

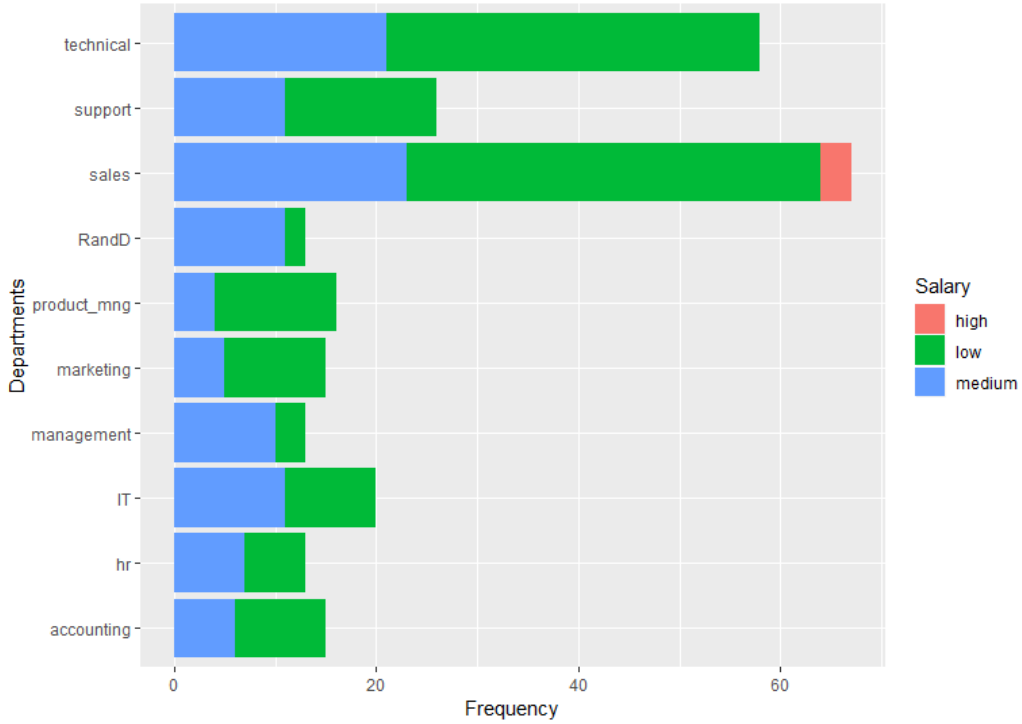

Figure 12

From the above bar chart we can see that a very low percentage of people who have worked on seven projects are given high salaries. This could be one of the most important factors for employee turnover.

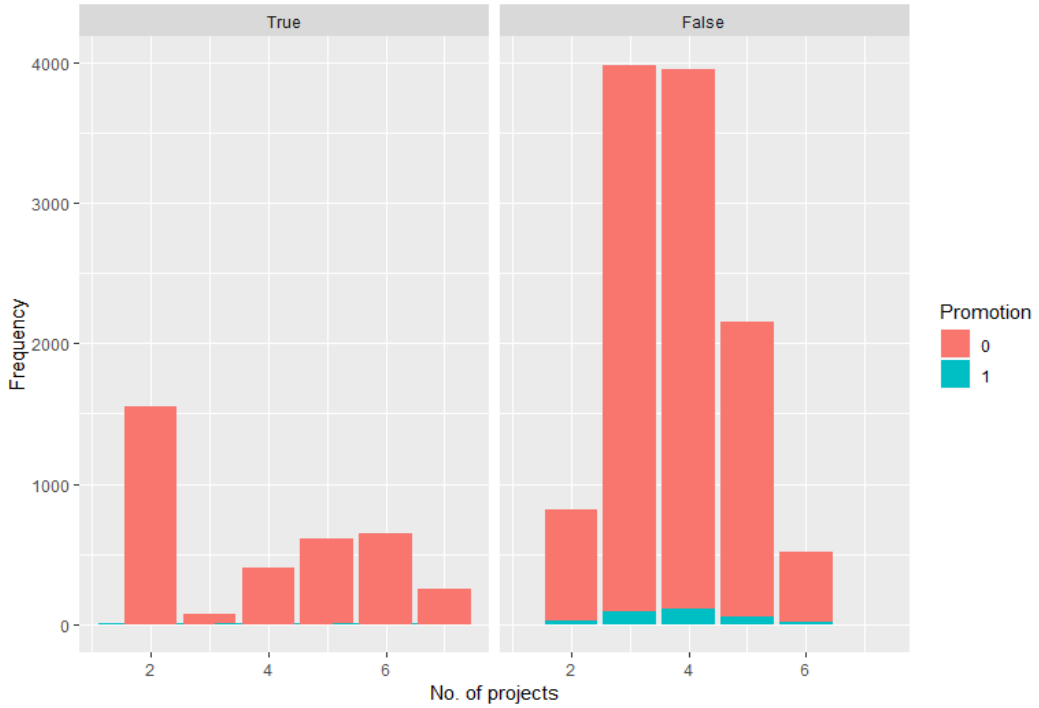

Figure 13

The employees who left the company were not given any promotion during their period of employment though they have worked on large number of projects. Even for the current employees, the percentage of employees who were given promotion is really low.

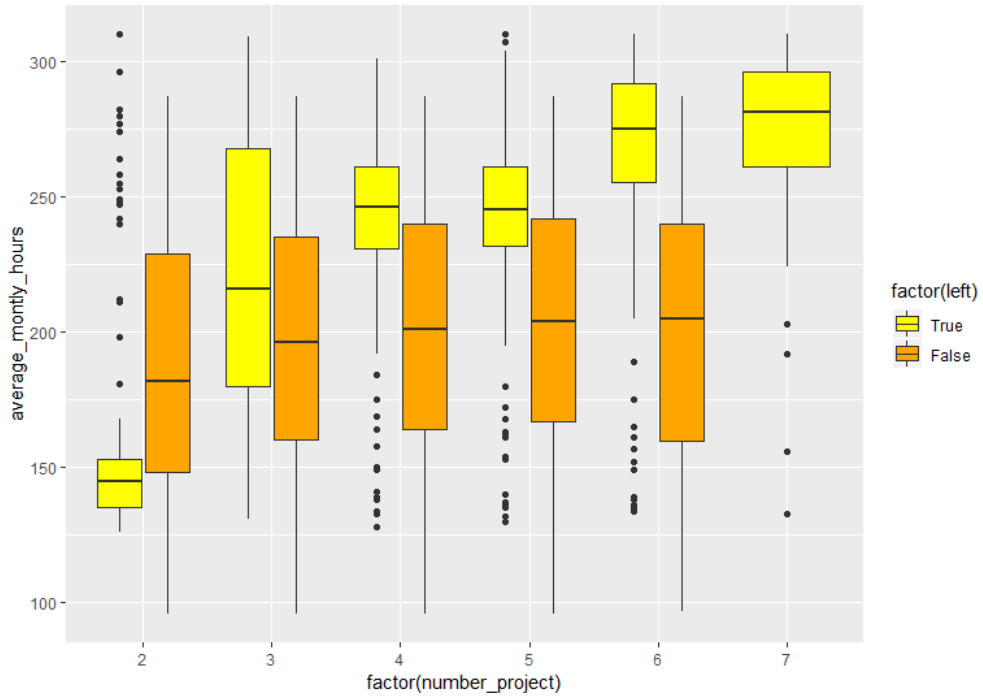

Figure 14

- As project count increased, so did average monthly hours
- Employees who did not have a turnover had consistent averageMonthlyHours, despite the increase in projects
- In contrast, employees who did have a turnover had an increase in averageMonthlyHours with the increase in projects
- The average working hours of the employee is around 200. Majority of the employees who left the company had working hours of 250

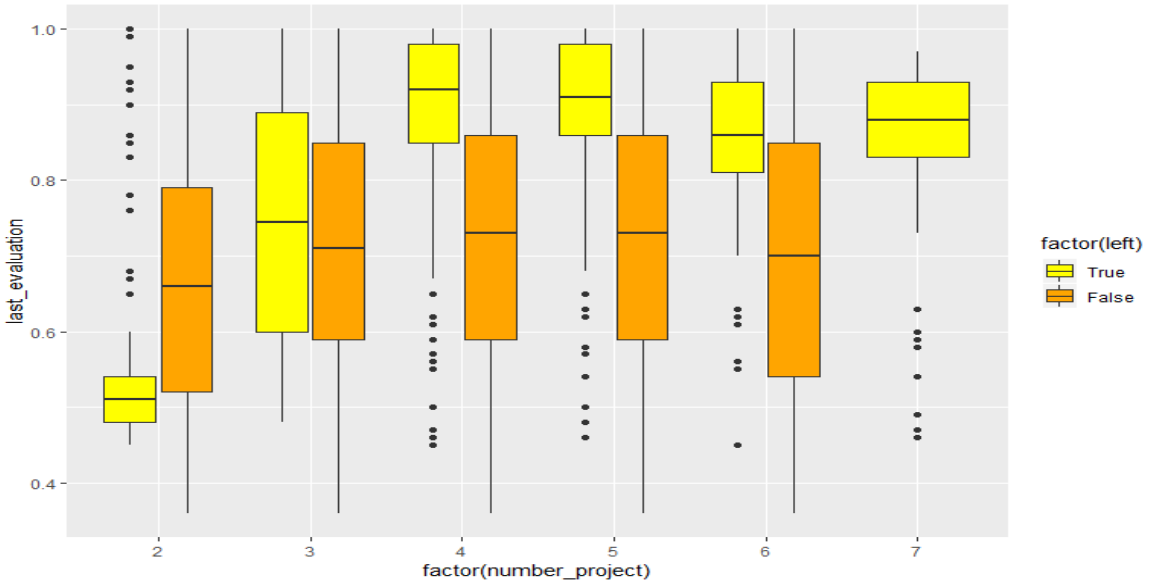

Figure 15

- The average evaluation of employees is around 70%
- Employees with less than two projects and bad evaluation left the company
- Employees with more than three projects and high evaluations also left the company

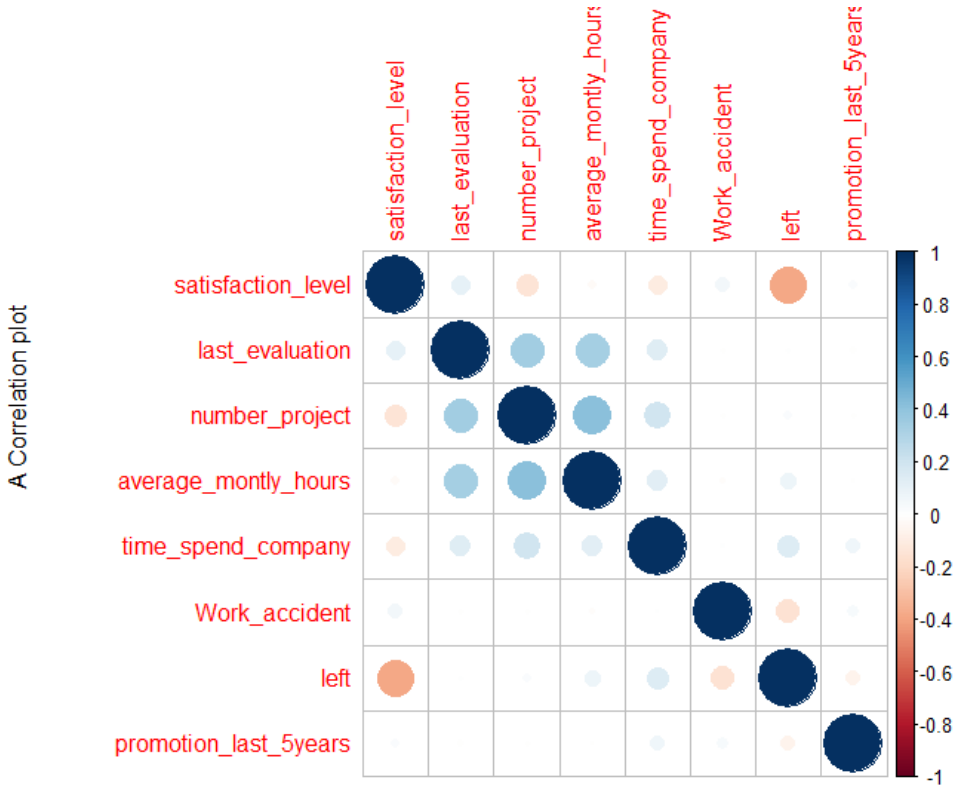

Figure 16

Inference from correlation plot:

We see from the correlation matrix that Satisfaction level, work accident, time spent in the company has most correlation with the left attribute. We can say that with lesser satisfaction levels the employee is more likely to leave. With more time spent in the company, an employee is likely to shift to another company. If work accidents have occured, the employee is very likely to leave the company. Last evaluation has a positive correlation with number of projects and average monthly hours.We can infer that an employees who work extra hours are more likely to receive a better evaluation

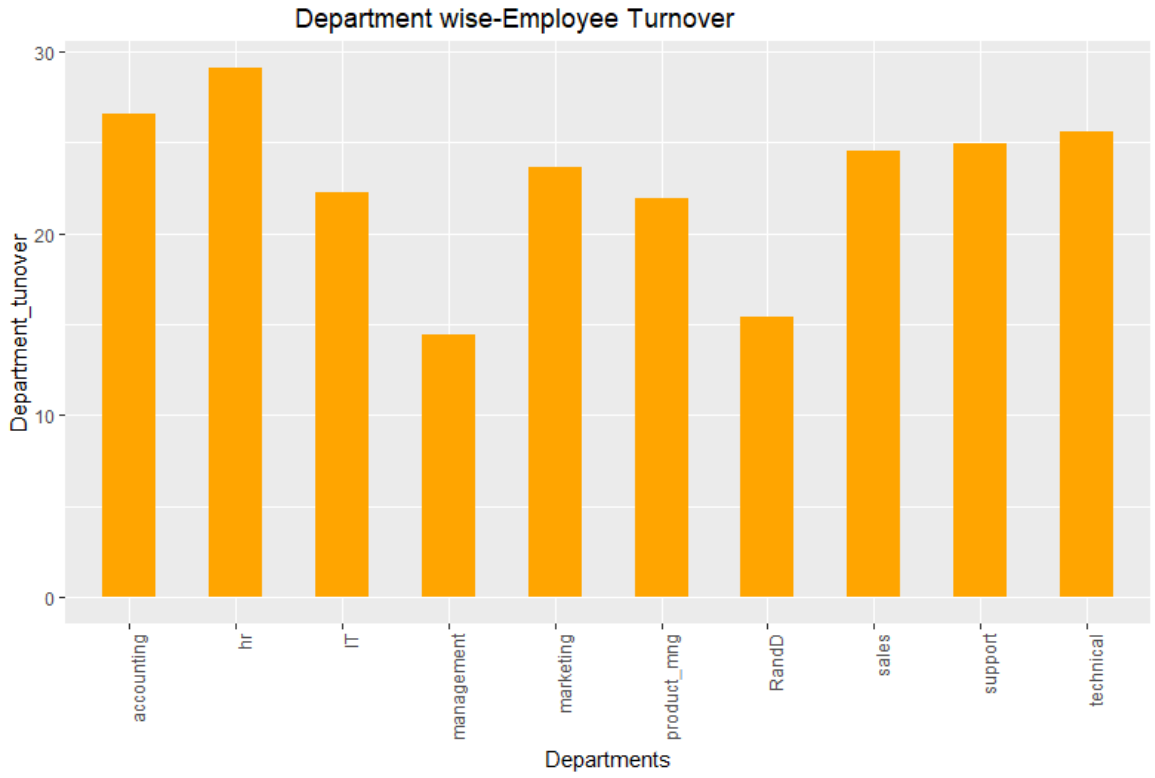

Figure 17

- HR and R & D have lowest turnovers

ALGORITHMS USED:

1) Linear Discriminant Analysis

Linear discriminant analysis, known as LDA, reduces the dimensionality of the dataset while also maintaining the variance within the data for class discrimination. LDA finds a line that finds the best separation between clusters of data. Our group chose to use this algorithm for our dataset because what we are trying to predict is if the left column will take a value of 0 or 1 according to all the other columns within the dataset. The results of using LDA is shown below. The algorithm was first trained using the training portion of the dataset and then used to predict values in the validation portion of the dataset. Linear discriminant analysis has its strengths and weaknesses. A strength that it has is that it allows for a ranking of predictors based on their importance. Ultimately, it may be useful for variable selection. A weakness that LDA has is that outliers within the dataset can influence the algorithm and cause it to perform poorly.

Confusion Matrix **Percent Accurate**

	0	1	
0	2077	479	
1	200	243	0.7735912

The confusion matrix shows that out of the 2,999 rows in the validation dataset, about 77.36% of the “left” column value was correctly identified.

Next, a lift chart and decile wise lift chart was made to evaluate the LDA algorithm and check it’s effectiveness at correctly predicting if the “left” column value will be 0 or 1.

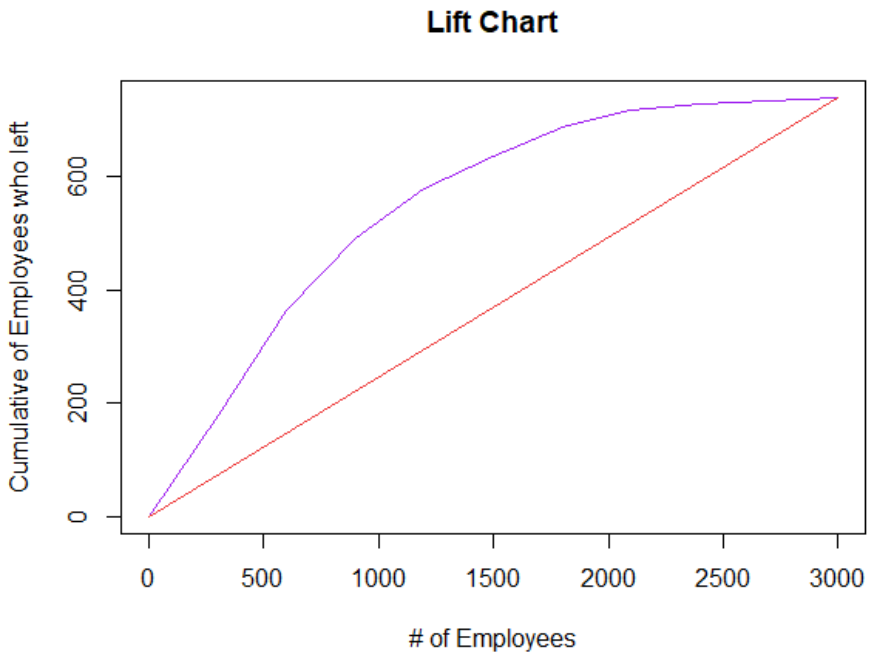

Figure 18

The lift chart measures how the model compares the effectiveness of the model as compared to randomly assigning the "left" column a value of 0 or 1. This lift chart shows that the LDA model did in fact do better than a random assignment as indicated by the purple line in comparison to the red line.

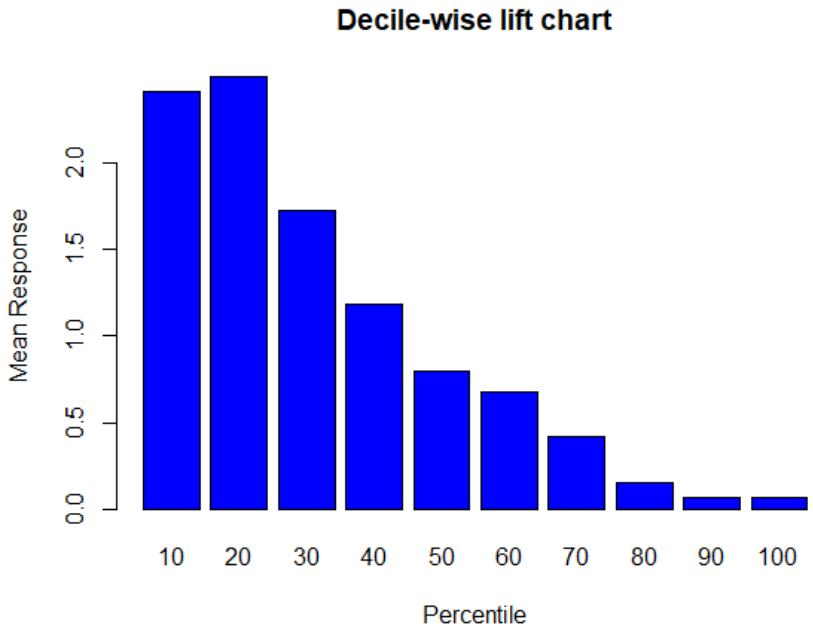

Figure 19

The decile wise lift chart shows that the records are sorted by their predicted scores in descending order for their bins that they are grouped by. The ideal effect that should be seen is the staircase effect, and in this case the chart is exhibited that effect. This shows that the model is binning the records correctly from highest response rate to lowest.

2) Logistic Regression

In the logistic regression model, the log odds of the outcome attribute, in our case the "left" column is modeled as a linear combination of the predictor variables.
We run the glm command with the response variable being "left" and including all other attributes as predictors
The summary of the model will display the call that we are running - logistic regression and the residuals which are a measure of the fit of the model.
We can interpret from the following from the coefficients, their standard errors, the z value and the p value
We observe that satisfaction_level, last_evaluation, number of projects, average monthly hours, time spent in the company, work accidents, promotion in the last 5 years and salary being low and medium are all statistically significant in predicting if the employee will leave the company or not
We can see that:

- with every unit increase in satisfaction level, the log of odds of an employee leaving decreases by 4.11
- with every unit increase in last evaluation, the log of odds of an employee leaving increases by 0.72
- similarly, we can observe that number of projects, work accident, promotion in the last 5 years all have a negative impact on the employee turnover
- -The factor variables Department and Salary are interpreted slightly differently
- -Having a low salary versus high salary changes the log odds of turnover by 1.98
- -Being in a the marketing department versus being in the accounting department decreases the log odds of employee leaving by 0.33
- We observe that the confusion matrix gives an accuracy of 77.9 ~ 78%

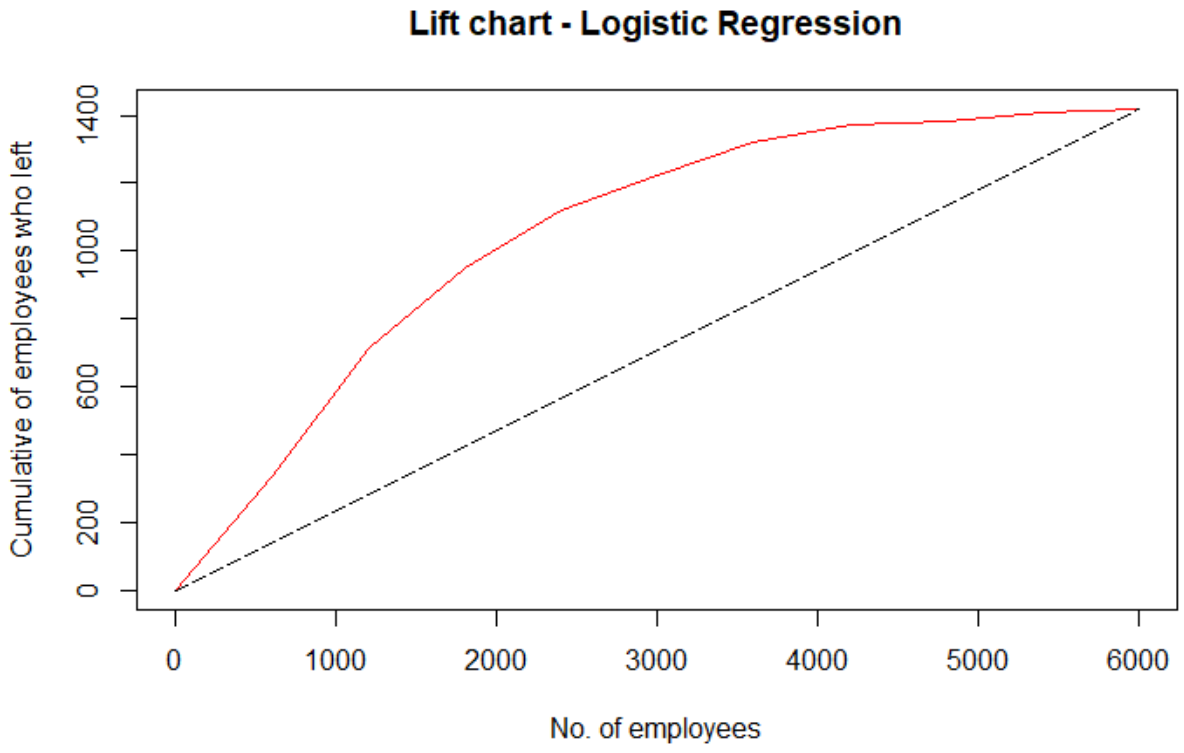

Figure 20

Lift Chart- Interpretation

The lift chart shows that in the naive model, among 500 employees, the model would predict only 160 case of employee who are going to leave, whereas, the logistic regression model predicts about 260 employees who will be leaving the company.

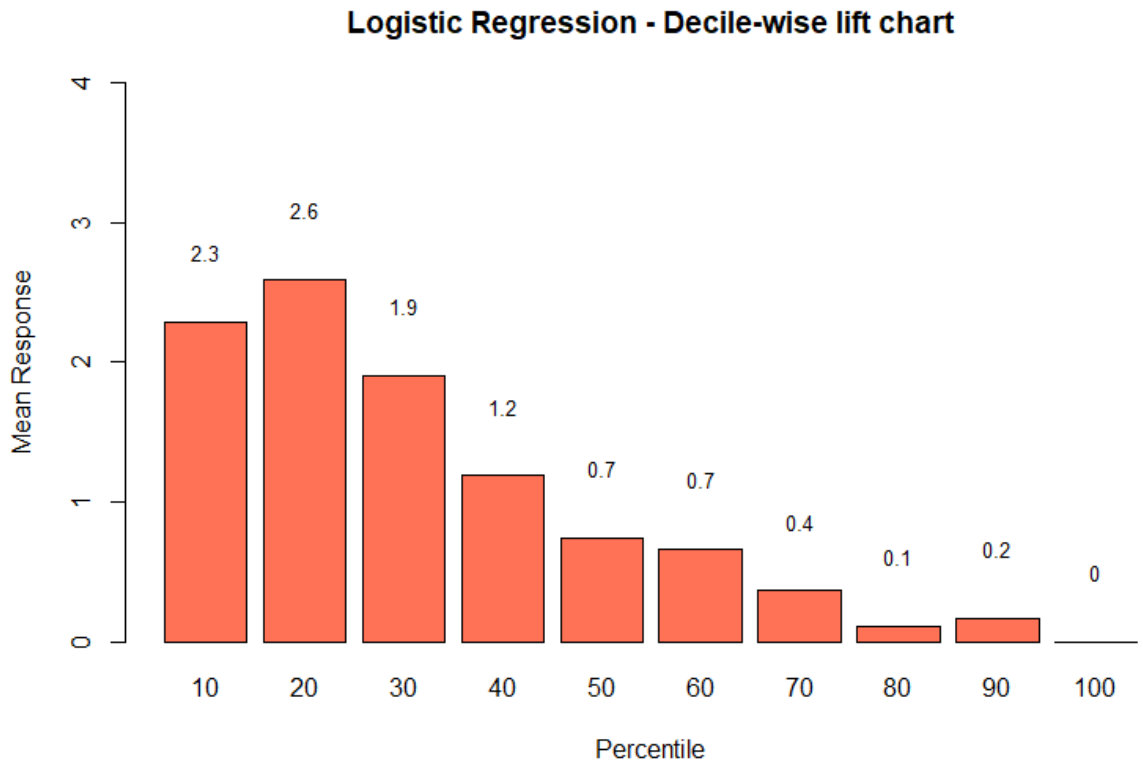

Figure 21

Decile-wise chart- Interpretation

We can interpret that this model is 2.3 times more effective than the naive model in predicting employee turnover when using only the top 10 percentile of data.

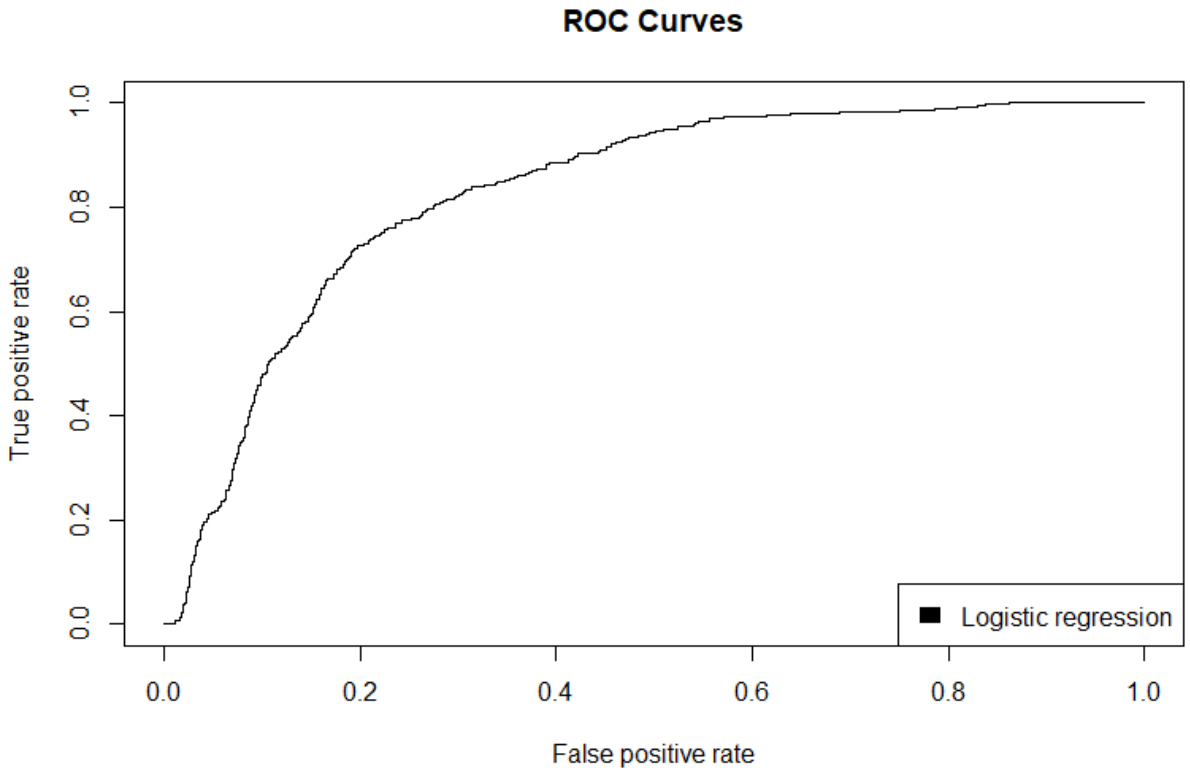

Figure 22

The Area under the curve for the ROC plot is 82.8 percent

Confusion Matrix:

	FALSE	TRUE
0	2077	181
1	480	261

Accuracy of the logistic regression model = 77.95 ~ 78 percent

3) Linear Regression

To predict the values of whether the employee would leave the company or not, we use statistical methods like linear and logistic regression to establish the relationship between variables and therefore make an inference about the turnover

The two regression methods will be evaluated on their accuracy and the better performing model will be picked.

Linear regression analyzes the relationship between the dependent variable - in our case, "left" and the rest of the independent variables. With the summary generated we can infer the effect of each predictor on the response variable.

From our regression model, we can observe the following:

Area under the curve = 0.8232

```
> confusion_matrix
   FALSE TRUE
 0  2115  143
 1   550  191
```

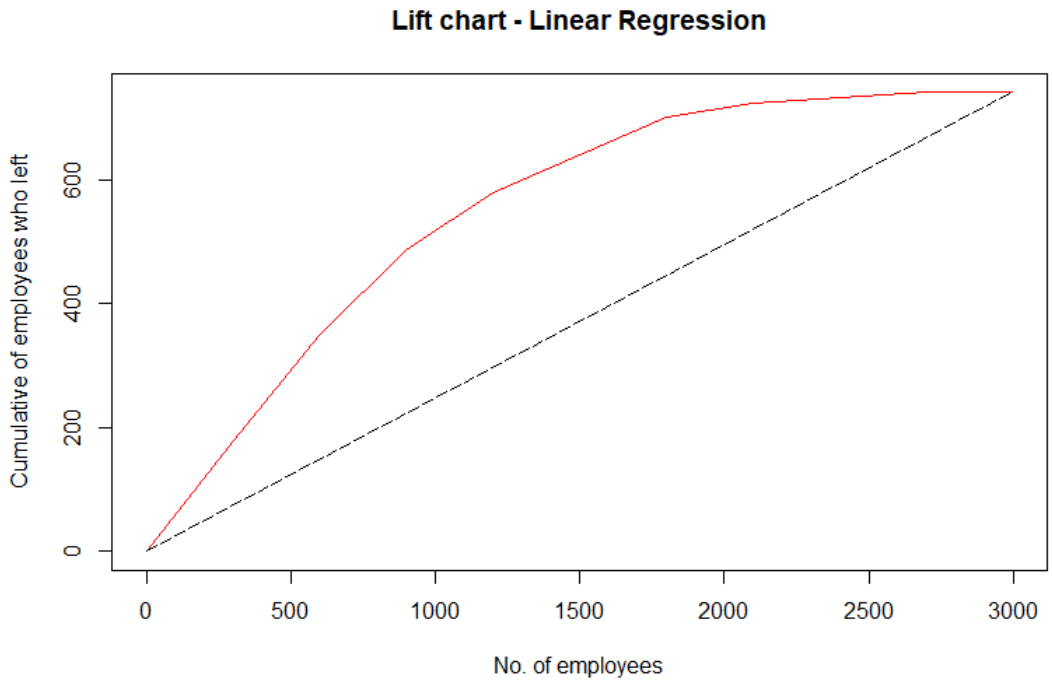

Figure 23

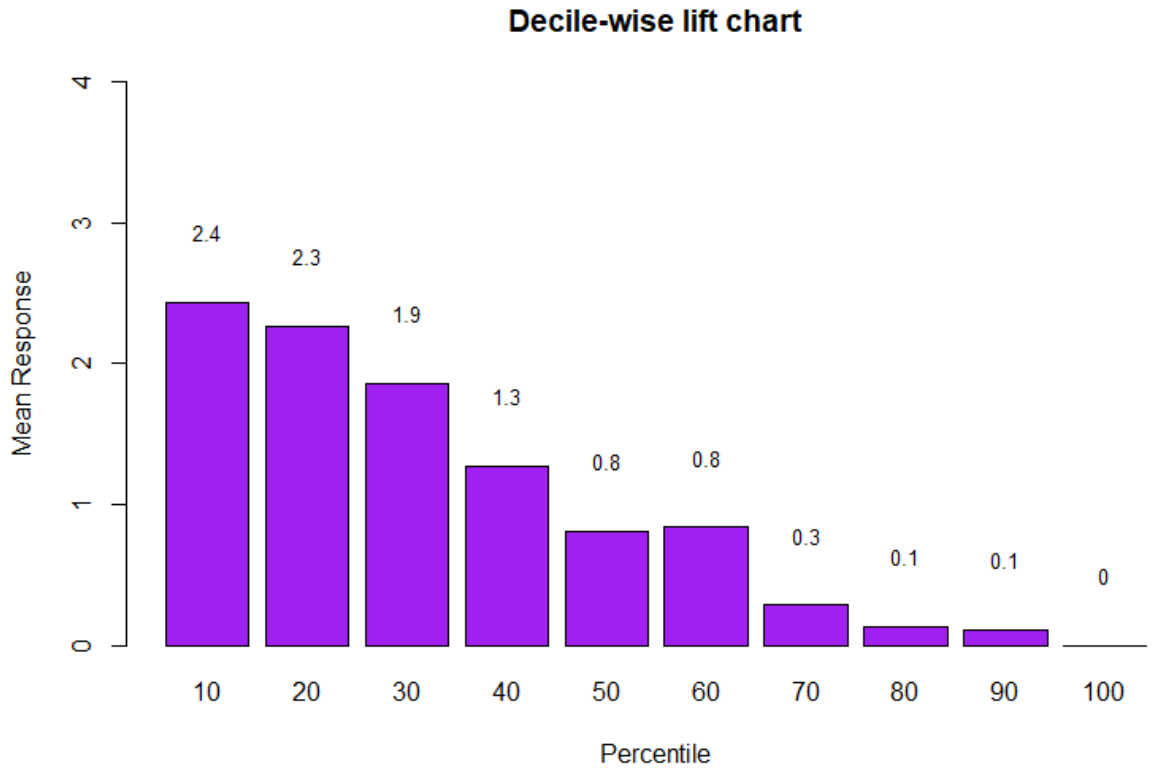
Decile-wise lift chart
Mean Response
0
1
2
3
4
2.4
2.3
1.9
1.3
0.8
0.8
0.3
0.1
0.1
0
10
20
30
40
50
60
70
80
90
100
Percentile

Figure 24

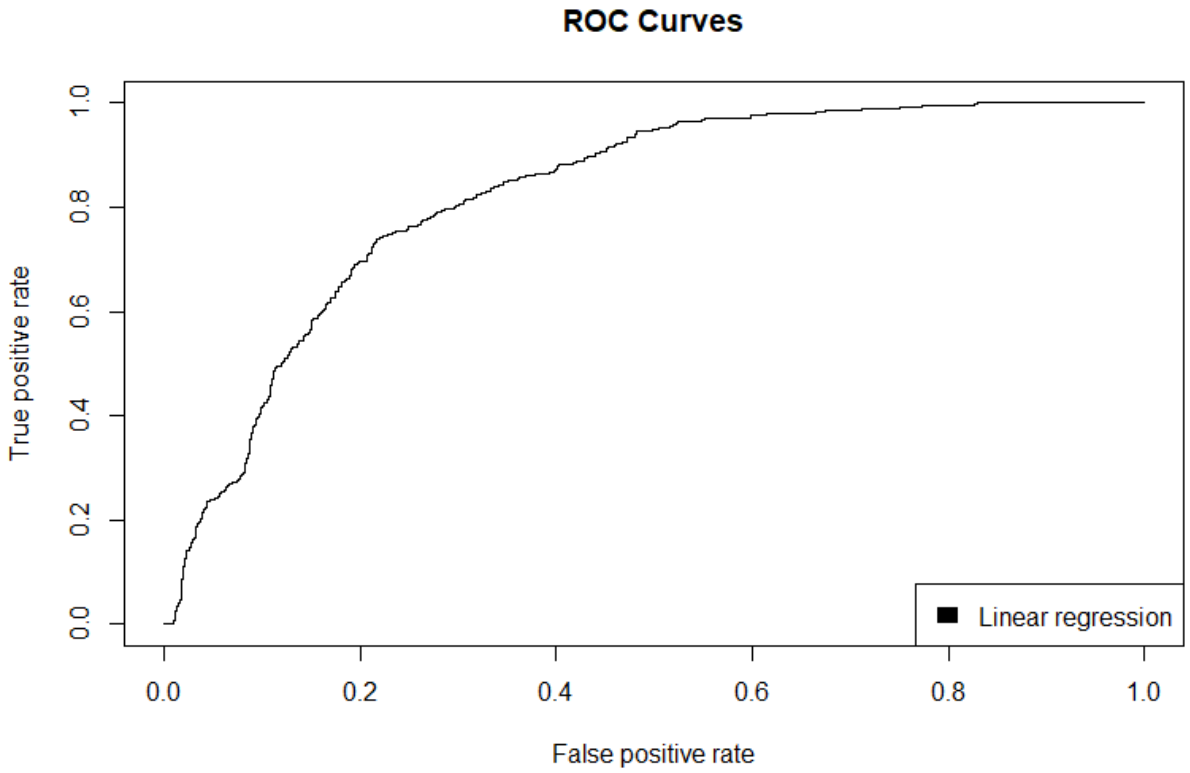

Figure 25

4) Random Forest

Random Forest is a supervised learning algorithm. Like it's name, it creates a forest and makes it somehow random. The "forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

The number of variables tried at each split i.e the mtry value is 3. The default value of mtry is determined dynamically based on number of input variables in training set. The number of features in the employee churn dataset is 10 so the default mtry value is square root of number of features i.e 3. Out of bag (OOB) error rate is computed across samples that were not selected into bootstrapped training sets. Since each tree in random forest is trained on bootstrapped sample of original data set, some of the samples will be duplicated in training set and some will be absent. The absent samples are the "out of bag" samples. The classification error across all out of bag samples is called out of bag (OOB) error. The confusion matrix is based on out of bag error rate.

Call:
randomForest(formula = left ~ ., data = train.df)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 3

OOB estimate of error rate: 0.82%
Confusion matrix:
    0   1 class.error
0 9146   12 0.00131033
1   87 2755 0.03061224

Confusion Matrix and Statistics
pred_rf   0   1
     0 2265   15
     1   5 714

Accuracy : 0.9933
95% CI : (0.9897, 0.9959)
No Information Rate : 0.7569
P-Value [Acc > NIR] : < 2e-16

Kappa : 0.9818
Mcnemar's Test P-Value : 0.04417
Sensitivity : 0.9978
Specificity : 0.9794
Pos Pred Value : 0.9934
Neg Pred Value : 0.9930
Prevalence : 0.7569
Detection Rate : 0.7553
Detection Prevalence : 0.7603
Balanced Accuracy : 0.9886

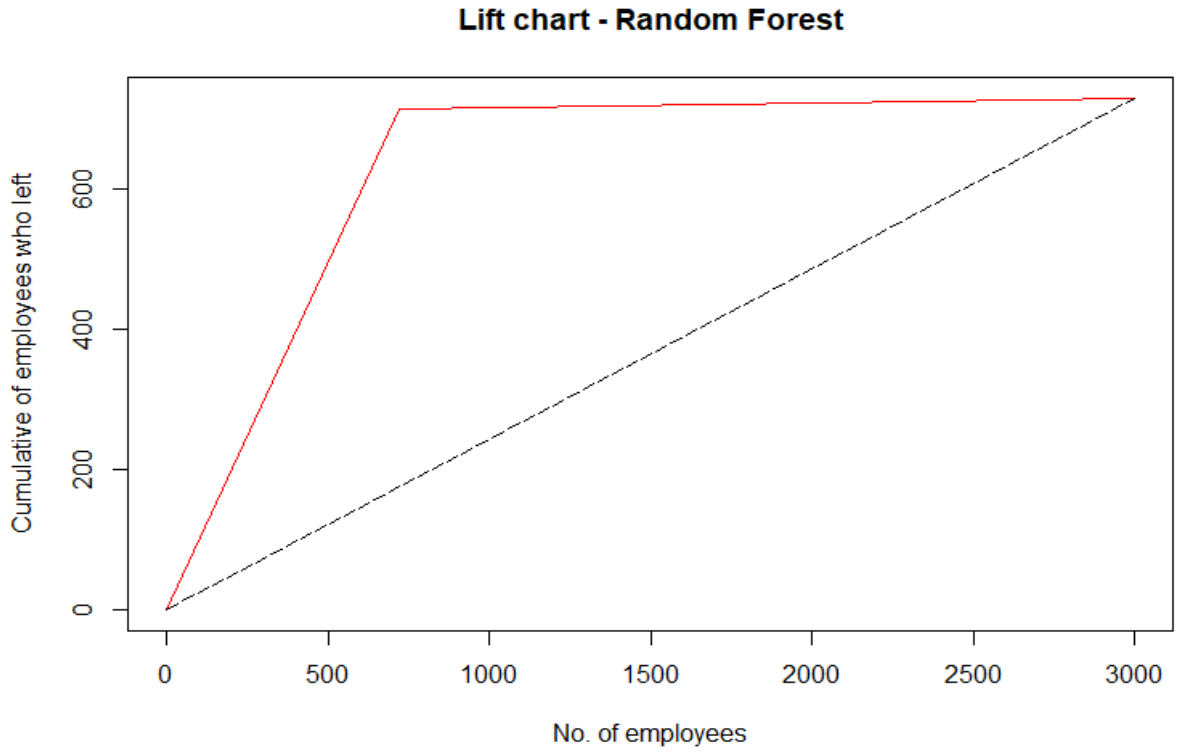

Figure 26

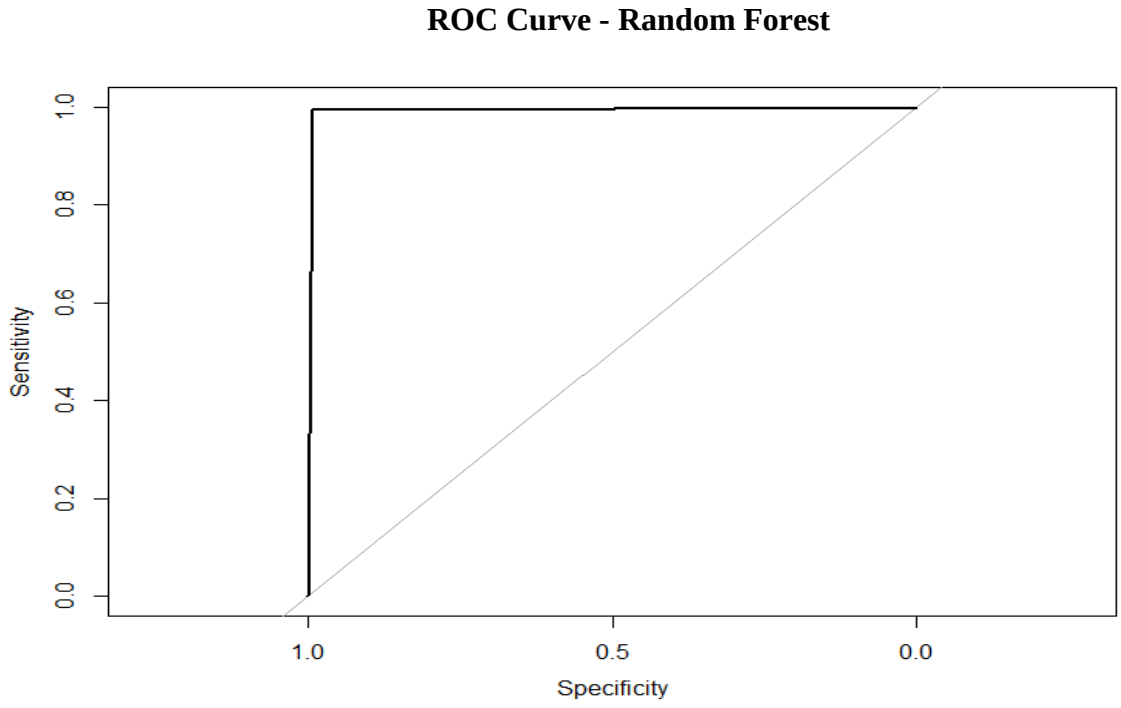

Figure 27

5) Decision Tree

Decision Tree is a Machine Learning algorithm used for both classification and regression. It is used because it is simple to understand, and we can make good interpretations from it. Here, we have used decision tree on training dataset for classifying whether the employee churned or not. This will help us predict results of the validation dataset.

'Left' is the dependent variable and all the remaining variables used in the algorithm are independent. The tree is initially allowed to grow fully and is then pruned with optimal CP value to avoid overfitting.

The node "0" represents that the number of employees who did not leave is greater than those who left. The node "1" represents that the number of employees who left is greater than those who stayed at that particular level. Initially, out of 12k employees in the training dataset, 9170 did not leave and 2830 employees left. That is why the first node (parent node) is labeled "0".

Each link(branch) represents a decision(rule) and each grey node(leaf) represents an outcome (0=did not leave, 1=left).

From the tree we can say that churn highly depends on the satisfaction level of the employee.

Consider the extreme right leaf node. If the rules are followed, then there are 62 employees who did not leave the company and 1191 who left. The rule to reach the leaf node is:

Satisfaction_level < 0.47 And number_project < 3 And last_evaluation < 0.58

When the model is tested on the validation dataset, we get an accuracy of **96.47%**

AUC: 0.9475

Lift Chart and ROC curve:

The lift chart shows that the decision tree model is more effective than the one without a predictive model.

The model seems to be quiet accurate since the ROC curve is closer to the upper left corner (high sensitivity)

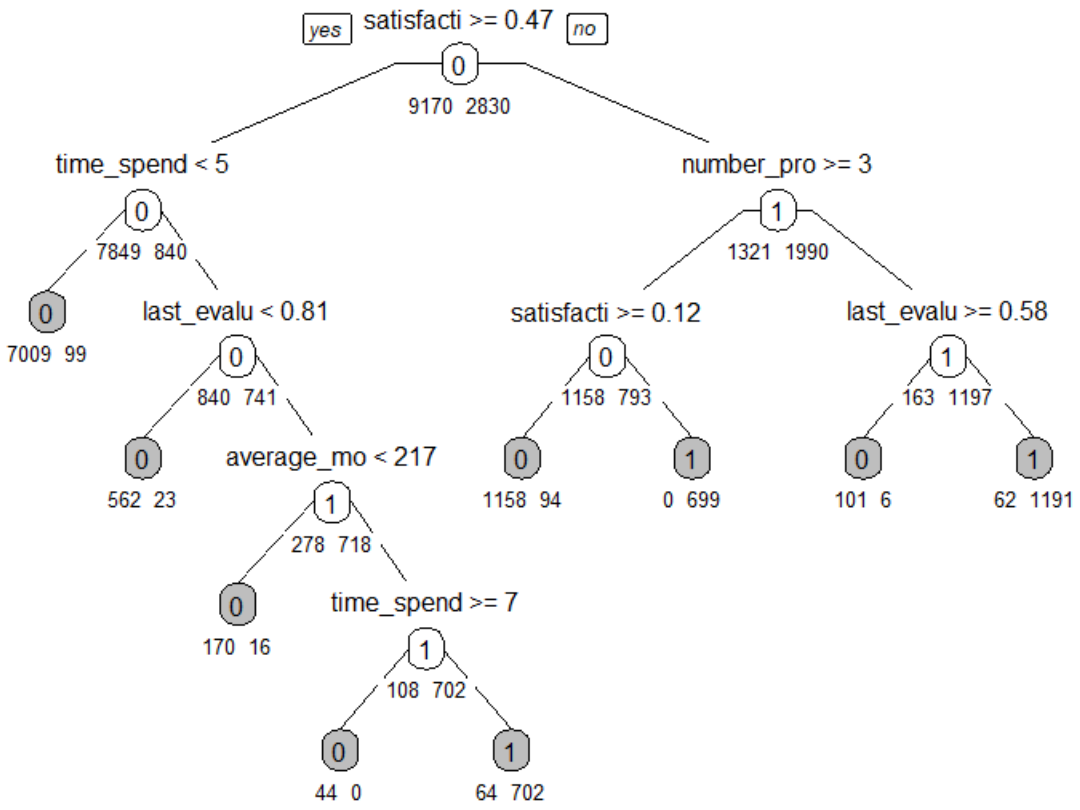

Figure 28

Confusion Matrix:

predict		
	0	1
0	2216	42
1	64	677

Accuracy: 0.9647

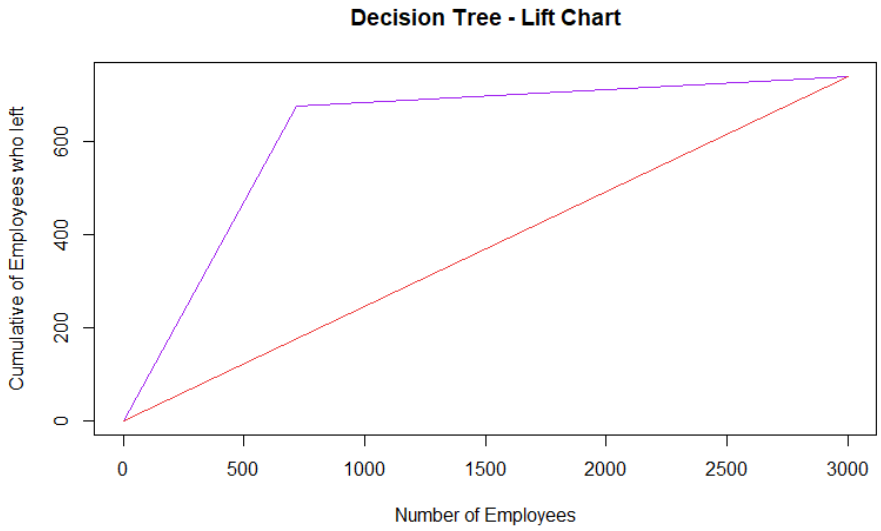

Figure 29

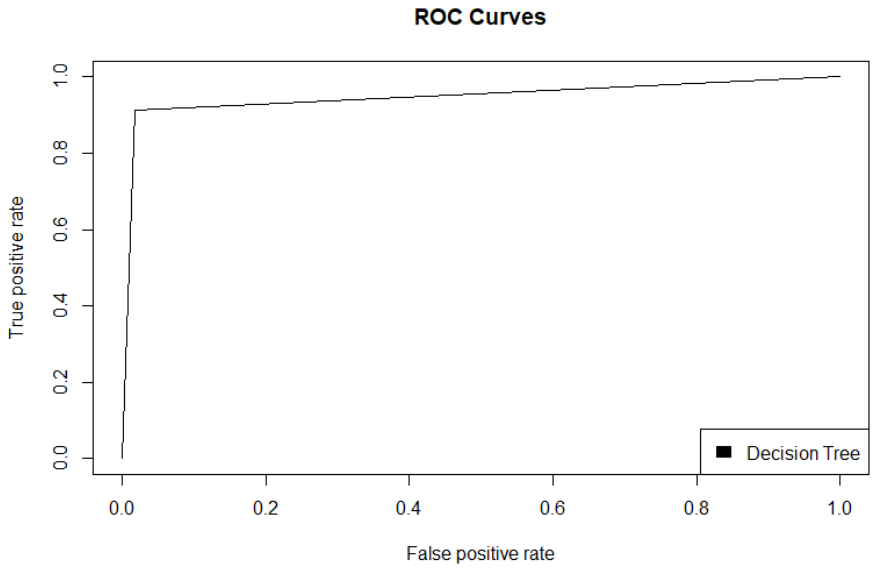

Figure 30

Model Selection:

On comparing the performances of each of the models by using an ROC curve, we can conclude the Random forest and Decision tree models are best suited for predicting the turnover of employees.

A better selection between the two would be to use Decision tree model as it allows for easier and clearer understanding of classifications and a comprehensive view of the attributing factors of employee churn.

The decision to select the model often falls on the data scientist factoring in computation power and size of dataset and level of comprehension that is required.

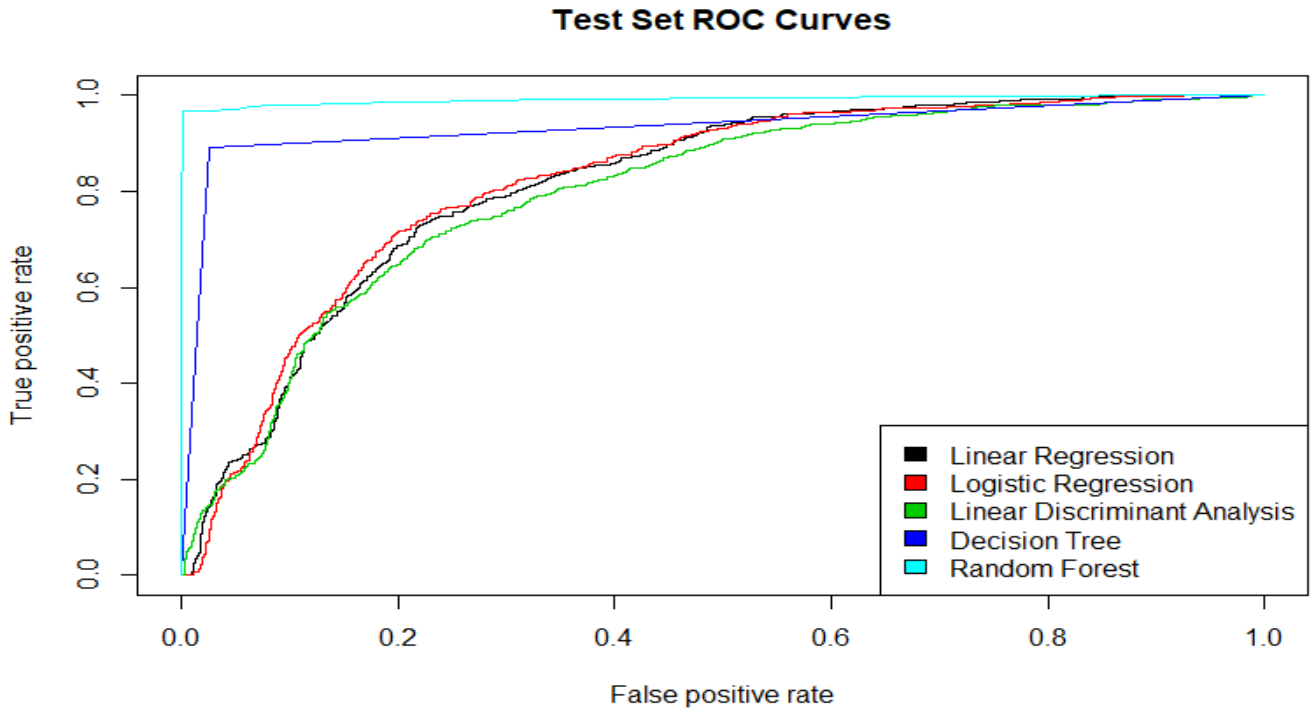

Figure 31

On comparing the performances of each of the the models by using an ROC curve, we can conclude the Random forest and Decision tree models are best suited for predicting the turnover of employees.

Model accuracy table

Index	Model name	Accuracy (percentage)	Area under the curve
1	Linear Regression	76.892 ~ 77%	0.823
2	Logistic Regression	77.959 ~ 78%	0.828
3	Linear Discriminant Analysis	77.359 ~ 77%	0.8101
4	Decision Tree	96.47 ~ 97%	0.9475
5	Random Forest	99.3 ~ 99%	0.9923

Recommendations:

- From our analysis we also came up with some recommendations for companies who want to retain their employees.
- We saw from the dataset that a very high number of projects was related to the employee leaving the company. Also, a very low number of projects was related to the employee leaving.
- Employee satisfaction also seemed to play a major role in the employee's decision to leave. Therefore, a company should try to measure various attributes related to the employee's safety, satisfaction, workload, etc. to try and understand if an employee will leave and utilize that information to increase employee satisfaction levels or decrease workload for an unsatisfied employee.
- Many companies overlook attributes related to job satisfaction of their employees, and ultimately lose talent within the organization and it can prove to be costly.
- Promoting an inclusive and comfortable work environment has proven to help retain employees
- In addition to our algorithms, we can conclude that Maternity leave, work places with childcare can help decrease churn rate.
- Promoting a healthy work-life balance directly affects the satisfaction levels of employees.
- Acknowledging and recognizing excellence with rewards makes a good motivating factor for employees to stay on
- Long hours at the workplace due to understaffing can result in unhappy employees. Therefore the companies must ensure that they are adequately staffed to prevent overworking their employees

Appendix - R Code

```
#Data Exploration and cleaning
#Including all the necessary library functions required in the project
library(tidyverse)
library(gains)
library(leaps)
library(caret)
library(dplyr)
library(forecast)
library(randomForest)

#Reading the csv datafile
Emp.df <- read.csv("HR_comma_sep.csv", strip.white = TRUE, blank.lines.skip = TRUE, header
= TRUE)

##Check to see if null values exist in any of the columns
apply(Emp.df, 2, function(x) any(is.na(x)))

#Exploring the data
str(Emp.df)
levels(Emp.df$Departments)
#Categorical data:
table(Emp.df$left)
count(Emp.df, Work_accident)
count(Emp.df, promotion_last_5years)

#Data Visualizations

#converting left variable to factor variable
hrm$left<-ifelse(hrm$left==1,'True','False')
hrm$left<-factor(hrm$left,levels=c("True","False"))

#Histogram
p1<-ggplot(aes(x=satisfaction_level),data=hrm) +
  geom_histogram(color="black",fill="blue",bins = 30) +
  labs(title="Satisfaction level Histogram",x='Satisfaction Level of Employees', y="Frequency")
p1

#Satisfaction level histogram facetted by salary classes
p2 =  p1  + facet_wrap(~salary)
p2

#Boxplot for Satisfaction level vs left facetted by Salary Ranges
ggplot(aes(x = left,y=satisfaction_level),data= hrm) +
  geom_boxplot() +
  ylab('Satisfaction Level') +
```

```
  xlab("Employee left") +
  facet_wrap(~salary) +
  labs(title = "Satisfaction level vs Churn facetted by Salary Ranges")

#faceted by whether an employee left or not
ggplot(aes(x=number_project),data = hrm) +
  geom_bar(color='black',fill='blue') +
  xlab("Number of Projects") +
  ylab("Frequency") +
  labs(title="Barplot of Number of projects faceted by Churn")+
  facet_wrap(~left)

#Average monthly hours worked faceted by Churn
ggplot(aes(y = average_montly_hours, x = hrm$left),data=hrm)+
  geom_boxplot() +
  xlab("Churn") +
  ylab("Average Monthly hours worked") +

  #Variable Time spend at Company faceted by Churn
  ggplot(aes(x = left, y = time_spend_company),data = hrm)+
  xlab("Churn") +
  ylab("Variable Time spend at company in years")+
  labs(title = "Barplot of Variable Time spend at Company faceted by Churn")+
  geom_boxplot()+
labs(title="Average monthly hours worked faceted by Churn")

# Department and their count faceted by Salary ranges
ggplot(aes(x =hrm$Departments),data = hrm ) +
  geom_bar(aes(fill=salary))  +
  xlab('Department') +
  ylab('Counts') +
  labs(title = "Department and their count faceted by Salary ranges")

#Frequency of work accidents across Departments
ggplot(aes(x = hrm$Departments),data = hrm) +
  geom_bar(aes(fill=factor(hrm$Work_accident))) +
  coord_flip() +
  labs(x = "Department",y ="Frequency", fill="Work Accidents" ,
       title = "Frequency of work accidents across Departments ")

##Number of projects
table(hrm$number_project,hrm$left)
##Employees who have worked on seven projects
proj <- filter(hrm,number_project==7)
summary(proj)
```

```
## Salary data for employees who have worked on 7 projects
ggplot(aes(x = Departments),data = proj) +
  geom_bar(aes(fill=factor(salary))) +
  coord_flip() +
  labs(x = "Departments",y ="Frequency", fill="Salary" )

#faceted by If a employee left or not
ggplot(aes(x=number_project),data = hrm) +
  geom_bar(color='black',fill='#547398') +
  xlab("Number of Projects") +
  ylab("Frequency") +
  labs(title="Barplot of Number of projects faceted by Left")+
  facet_wrap(~left)

## Number of projects vs promotion facetted by left
ggplot(aes(x = number_project),data = hrm) +
  geom_bar(aes(fill=factor(promotion_last_5years))) +
  labs(x = "No. of projects",y ="Frequency", fill="Promotion" )+
  facet_wrap(~left)

#ProjectCount VS Evaluation
#Looks like employees who did not leave the company had an average evaluation of around 70% even with different
#projectCounts.
#Employees that had two projects and a horrible evaluation left. Employees with more than 3 projects and super high evaluations
##left
p<-ggplot(hrm, aes(x = factor(number_project), y = last_evaluation, fill = factor(left))) +
  geom_boxplot() + scale_fill_manual(values = c("yellow", "orange"))
print(p)
##ProjectCount vs avaerage monthly hours
p<-ggplot(hrm, aes(x = factor(number_project), y = average_montly_hours, fill = factor(left))) +
  geom_boxplot() + scale_fill_manual(values = c("yellow", "orange"))
#print(p)[BOXPLOT]
#Looks like the average employees who stayed worked about 200hours/month. Those that had a turnover worked about 250hours/month
#and 150hours/month

#Correlation plot of variables:
#All int attributes have to be converted into numerics to feature in correlation matrix
Emp.df$number_project<-as.numeric(Emp.df$number_project)
Emp.df$average_montly_hours<-as.numeric(Emp.df$average_montly_hours)
Emp.df$time_spend_company<-as.numeric(Emp.df$time_spend_company)
Emp.df$Work_accident<-as.numeric(Emp.df$Work_accident)
Emp.df$left<-as.numeric(Emp.df$left)
Emp.df$promotion_last_5years<-as.numeric(Emp.df$promotion_last_5years)
```

```
str(Emp.df)

c<-data.frame(Emp.df)
correlation<-c[,-c(9:10)]
m<-cor(correlation)
library(corrplot)
corrplot(m)

#Calculation of employee turn_over rate using Emp.left column
#Turn_over rate = No. of employees who left / total no. of employees
#where 1 indicates inactive (left) and 0 indicates active

turnover_rate = mean(Emp.df$left)
turnover_rate
#Approximately 24 percent of employees are inactive (have left the firm) and 76 percent are active(currently employed at the firm)

#Turnover_rate with respect to each Department:
Emp.df %>%
  count(Departments, left)

df_Departments <- Emp.df %>%
  group_by(Departments) %>%
    summarize(turnover_Departments = mean(left))

#Plotting Department wise percentage turnover.
Department_tunover <- df_Departments$turnover_Departments*100
ggplot(df_Departments) + geom_bar(aes(x = Departments, y = Department_tunover), fill ="orange",  width = 0.5, stat = "identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1))

#We observe that management and RandD departments have the lowest turnover rates
#HR department has the highest turnover rate followed by the Accounting department

#Linear regression
set.seed(111)
#create data partition using predictor variable - left
train.index<- createDataPartition(Emp.df$left, p = 0.8, list =FALSE)
train.df <- Emp.df[train.index,]
valid.df <- Emp.df[-train.index,]

### Run regression
lm.reg <- lm(left ~ ., data = train.df)

options(scipen = 999)
sm <- summary(lm.reg)
```

```
### Generate predictions
lm.pred <- predict(lm.reg, valid.df)
some.residuals <- valid.df$left - lm.pred
plot(some.residuals, type = "p", pch = 16,
    col = "blue1",
    ylab = "Sample Residuals",
    ylim = c(0, 1), bty = "n",
    xlim = c(0, 100)
)

df <- data.frame("Predicted" = lm.pred, "Actual" = valid.df$left,
         "Residual" = some.residuals)
### Accuracy measures
accuracy(lm.pred, valid.df$left)
round(exp(coef(lm.reg)), 2)

lm.pred <- predict(lm.reg, valid.df[, -7], type = "response")
t(t(head(lm.pred, 10)))

str(lm.pred)
confusion_matrix<- table(valid.df$left, lm.pred > 0.5)
confusion_matrix

accuracy <- (sum(diag(confusion_matrix)) / sum(confusion_matrix))
accuracy

gain <- gains(valid.df$left, lm.pred, groups = 10)
#The accuracy of the Linear model is 0.7689, almost equal to 77 percent.

#Linear regression - Lift,Decile, ROC
#Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.df$left))~c(0,gain$cume.obs),
    xlab = "No. of employees", ylab = "Cumulative of employees who left", col = "red", main =
"Lift chart - Linear Regression", type = "l")
lines(c(0,sum(valid.df$left))~c(0, dim(valid.df)[1]), lty = 5)

# Decile-wise chart
heights <- gain$mean.resp/mean(valid.df$left)
midpoints <- barplot(heights, names.arg = gain$depth,  ylim = c(0,4), col = "purple",
                xlab = "Percentile", ylab = "Mean Response",
                main = "Decile-wise lift chart")
text(midpoints, heights+0.5, labels=round(heights, 1), cex = 0.8)

#ROC curve
library(pROC)
```

```
r <- roc(valid.df$left, lm.pred)
plot.roc(r, main = "Linear regression ROC")
### compute auc
auc(r)

#Area under the curve: 0.8232



#Logistic regression
#Splitting data into test and training data
set.seed(111)
#create data partition using predictor variable - left
train.index<- createDataPartition(Emp.df$left, p = 0.8, list =FALSE)
train.df <- Emp.df[train.index,]
valid.df <- Emp.df[-train.index,]
#when the dependent is categorical, we can use logistic regression
 ##to predict the probability of left - to see if the employee will stay or leave the company

log.reg <- glm(left ~ ., family ="binomial",data = train.df )
summary(log.reg)
round(exp(coef(log.reg)), 2)

log.reg.pred <- predict(log.reg, valid.df[, -7], type = "response")
t(t(head(log.reg.pred, 10)))

str(log.reg.pred)
confusion_matrix<- table(valid.df$left, log.reg.pred > 0.5)
confusion_matrix

accuracy <- (sum(diag(confusion_matrix)) / sum(confusion_matrix))
accuracy
gain <- gains(valid.df$left, log.reg.pred, groups = 10)
#We observe that the confusion matrix gives an accuracy of 77.9 ~ 78%

#Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.df$left))~c(0,gain$cume.obs),
    xlab = "No. of employees", ylab = "Cumulative of employees who left", col = "red", main =
"Lift chart - Logistic Regression", type = "l")
lines(c(0,sum(valid.df$left))~c(0, dim(valid.df)[1]), lty = 5)

# Decile-wise chart
heights <- gain$mean.resp/mean(valid.df$left)
midpoints <- barplot(heights, names.arg = gain$depth,  ylim = c(0,4), col = "blue",
                     xlab = "Percentile", ylab = "Mean Response",
```

```
                main = "Decile-wise lift chart")
text(midpoints, heights+0.5, labels=round(heights, 1), cex = 0.8)

#ROC curve for Logistic regression
##ROC curve
library(pROC)
r <- roc(valid.df$left, log.reg.pred)
plot.roc(r)
### compute auc
auc(r)

#Linear Discriminant Analysis
library(MASS)
library(caret)
library(ggplot2)
library(testthat)
library(gains)
library(lift)

hrm<-read.csv("HR_comma_sep.csv")
View(hrm)
as.numeric(hrm$Departments)
as.numeric(hrm$salary)

#hrm$time_spend_company <- (hrm$time_spend_company*8760)
set.seed(11)
training.index <- createDataPartition(hrm$left, p = 0.8, list = FALSE)
hrm.train <- hrm[training.index, ]
hrm.valid <- hrm[-training.index, ]

### Linear Discriminant Analysis - use lda() from MASS package
lda1 <- lda(left ~., data = train.df)
# output
lda1
# predict
pred1 <- predict(lda1, newdata=hrm.valid, type ="response")
lda.pred<- pred1$x
## class: predicted class
## posterior: posterior probabilities of belonging to different classes
## x: linear discriminant values

# check model accuracy
table(pred1$class, hrm.valid$left)  # pred v actual

mean(pred1$class == hrm.valid$left)  # percent accurate
```

```
gain <- gains(hrm.valid$left, pred1$x)

### cumulative lift chart
options(scipen=999)
### Compute gains relative to employees left
lefts <- hrm.valid$left[!is.na(hrm.valid$left)]
sum(lefts)
plot(c(0,gain$cume.pct.of.total*sum(lefts))~c(0,gain$cume.obs),
    xlab="# of Employees", ylab="Cumulative of Employees who left", main="Lift Chart",
    col = "purple", type="l")

### baseline
lines(c(0,sum(lefts))~c(0,nrow(hrm.valid)), col="brown2", lty=1)

###decile chart
barplot(gain$mean.resp/mean(lefts), names.arg = gain$depth,
       xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart",
       col = "blue")
##ROC curve
library(pROC)
r <- roc(hrm.valid$left, pred1$posterior[,2])
plot.roc(r)
### compute auc
auc(r)

#Decision Tree

library(caret)
library(gains)
library(lift)
library(rpart)
library(rpart.plot)

hrm <- read.csv("HR_comma_sep.csv")

## Splitting Dataset into training and testing
set.seed(111)
training.index <- createDataPartition(hrm$left, p = 0.8, list = FALSE)
hrm.train <- hrm[training.index, ]
hrm.test <- hrm[-training.index, ]
## Generate classification tree
tree <- rpart(left ~ ., data = train.df, method = "class")
prp(tree, type = 1, extra = 1, under = TRUE, split.font = 2, varlen = -10)
predict1 <- predict(tree, hrm.test, type="class")

## Fully-Grown Tree
```

```
deeper.ct <- rpart(left ~ ., data = hrm.train,
                   method = "class", cp = 0, minsplit = 1)

length(deeper.ct$frame$var[deeper.ct$frame$var == "<leaf>"])  # count number of leaves
prp(deeper.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
   box.col=ifelse(deeper.ct$frame$var == "<leaf>", 'gray', 'white'))



## Complexity Parameters
## xval:  # of folds to use cross-validation procedure
## CP: sets the smallest value for the complexity paraeter
cv.ct <- rpart(left ~ ., data = hrm.train, method = "class",
               cp = 0.00001, minsplit = 5, xval = 5)
printcp(cv.ct)

## Pruned Classificatin Tree using CP with lowest error
pruned.ct <- prune(cv.ct,
                   cp = cv.ct$cptable[which.min(cv.ct$cptable[,"xerror"]),"CP"])
length(pruned.ct$frame$var[pruned.ct$frame$var == "<leaf>"])
prp(pruned.ct, type = 1, extra = 1, split.font = 1, varlen = -10)

## Best-Pruned Tree
set.seed(1)
cv.ct <- rpart(left ~ ., data = hrm.train, method = "class", cp = 0.00001, minsplit = 1, xval = 5)

printcp(cv.ct)
# Print out the cp table of cross-validation errors.
pruned.ct <- prune(cv.ct, cp = 0.0154639)
prp(pruned.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
   box.col=ifelse(pruned.ct$frame$var == "<leaf>", 'gray', 'white'))

predict1 <- as.numeric(predict1)
gain <- gains(hrm.test$left, predict1)
head(predict1)

## cumulative lift chart
options(scipen=999)
## Compute gains relative to employees left
lefts <- hrm.test$left[!is.na(hrm.test$left)]
sum(lefts)
plot(c(0,gain$cume.pct.of.total*sum(lefts))~c(0,gain$cume.obs),
    xlab="# of Employees", ylab="Cumulative of Employees who left", main="Lift Chart",
    col = "purple", type="l")

## baseline
lines(c(0,sum(lefts))~c(0,nrow(hrm.test)), col="brown2", lty=1)
```

```
##ROC curve
library(pROC)
r <- roc(hrm.test$left, predict1)
plot.roc(r)
### compute auc
auc(r)

#Random Forest
context1 <- Emp.df
library(randomForest)
train.index <- createDataPartition(context1$left, p = 0.8, list = FALSE)
train.df <- context1[train.index,]
valid.df <- context1[-train.index,]
train.df$left <- as.character(train.df$left)
train.df$left <- as.factor(train.df$left)
rf.reg <- randomForest(left ~., data = train.df)
print(rf.reg)
pred_rf <- predict(rf.reg, valid.df, type = "prob")
head(pred_rf)
x <- pred_rf[,2]
library(pROC)
# Integrating all models into one ROC plot}
library(prediction)
library(ROCR)
# List of predictions
pred_list <- list(log.reg.pred, lm.pred, lda.pred, predict1, x)

# List of actual values (same for all)
m <- length(pred_list)
m
actual_list <- rep(list(valid.df$left), m)

# Plot the ROC curves
pred <- prediction(pred_list, actual_list)
rocs <- performance(pred, "tpr", "fpr")
plot(rocs, col = as.list(1:m), main = "Test Set ROC Curves")
legend(x = "bottomright",
    legend = c("Logistic Regression", "Linear regression", "Linear Discriminant Analysis",
"Decision Tree", "Random Forest"),
    fill = 1:m)
```

References:

- Link to dataset. https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Employee+Churn+in+Python/HR_comma_sep.csv

- Simon Jackson. 2016. "Visualising Residuals". Svbtle. AUGUST 23, 2016. https://github.com/drsimonj/blogR/blob/master/Rmd/visualising-regression-residuals.Rmd, https://drsimonj.svbtle.com/visualising-residuals

- University of California, Los Angeles. 2016. "LOGIT REGRESSION | R DATA ANALYSIS EXAMPLES". https://stats.idre.ucla.edu/r/dae/logit-regression/

- Yale University. 1997. "Linear Regression". http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

- https://campus.datacamp.com/courses/machine-learning-with-tree-based-models-in-r/random-forests

- https://www.datacamp.com/community/tutorials/decision-trees-R

- Susan Li.2017. "Predict Customer churn with R" https://towardsdatascience.com/predict-customer-churn-with-r-9e62357d47b4

- Niklas Donges. 2018. "The Random Forest Algorithm" https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

My Experience of Working on R Group Project

"Coming together is a beginning. Keeping together is progress. Working together is success."

This quote tells exactly what I have been experiencing as part of R Group Project.

One unique thing about R project was that the team was formed randomly without getting to choose the teammates. It is always challenging and exciting to work with people about whom you don't really know much. We started off by collecting the contact number of everyone in the team and creating a WhatsApp group to communicate effectively. We decided that we would bring one or two datasets per person before we meet based on everyone's availability. During the meet up, we went through various datasets brought by each one of us and discussed the pros and cons of using them. Some examples of the datasets we spoke about are studying Black Friday sales through consumer behavior, providing movie recommendations and analyzing global shark attacks. Since the meet up happened during the initial days of the course, we were biased to take a numerical dataset because linear regression is all we knew by then. That is how we all agreed on using a dataset for predicting the price of diamonds. But soon, we realized that we had agreed to disagree.

As the professor took us through different statistical concepts in R and while analyzing the dataset, I felt that the dataset we chose is too simple and not challenging enough for a graduate student. I was not really surprised when I came to know that even my teammates were thinking the same. We started our dataset hunt again with a wider approach and mindset.

I was always intrigued by the term "churn" and started looking out for datasets related to it. I came up with a dataset for predicting customer churn and my teammates were okay with it. Unfortunately, someone else in our class was using the same dataset and we had to change again. Since everyone of us was interested in churn prediction, we made sure that we got a dataset on similar lines. This is how we came up with our final dataset whose purpose was to predict employee churn in a company.

We calculated the number of days available before the due date and divided the tasks among the five of us based on everyone's availability. Since I had more time in the first half of the project comparatively, I took the initiative of doing the data exploration and telling others more about the dataset by studying them thoroughly. The journey of creating stories from data was fascinating for an amateur like me. I searched online on how to create different visualizations to provide meaningful insights. It was interesting to see how different variables are related to each other and coming up with hypothesis as to know the "why" behind the findings. Filtering out on which visualizations to use where was a major challenge that I faced. Through this project, I gained a wider perspective on the usage of different visualizations. Nikitha and Pooja came up with a few visualizations as addons as well.

We decided to use four algorithms so that we can compare the accuracy of each model. Linear Discriminant analysis was done by Sumit, Linear and Logistic Regression by Abhinitha, decision tree by Pooja. It is appreciable how Abhinitha and Nikitha came up with Random Forest Algorithm

which was outside of class lectures. I was able to provide suggestions for the analysis performed on how they can be improved during the meet ups. It became really challenging for all of us to catch up at the same time as the semester was ending. But the good thing is whoever was available met and discussed their respective findings and took suggestions from others. We made sure that whatever was discussed was updated in Google Docs so that everyone was aware of the progress.

I took the initiative of collecting and collaborating the individual contributions to a proper report format. While I was making the repot, I came across various areas where some major and minor changes had to be made and its commendable how my teammates made the necessary amendments at a fast pace. Along with the many technical things that I learnt in this journey, something that stands out for me is peer learning. Peer learning added another perspective to my individual thoughts. It was wonderful learning to accommodate others' thoughts and respecting them.