

PROJECT

DSA-6000

Data Science and Analytics

Title: Human Resource Analytics

Team members

Harjeet Singh Monga

Jayapriya Nagarajan

Ravikumar Sulepetkar

Shrankhala Jain

Contents

1	Abstract.....	3
2	Background.....	4
3	Methodology & Analysis.....	5
3.1	Set work directory and read the csv file	5
3.2	Dimensions	5
3.3	Summary	5
3.4	Change name of columns.....	6
3.5	Attrition Rate	6
3.6	Correlation Matrix	6
3.7	One-Sample T-Test (Measuring Satisfaction Level)	7
3.8	Distribution plots	9
3.8.1	Employee Satisfaction Distribution	9
3.8.2	Employee Last Evaluation Distribution	10
3.8.3	Employee Average Monthly Hours Distribution	10
3.8.4	Salary Vs Turnover	11
3.8.5	Number of projects Vs Turnover	12
3.8.6	Distribution Plots – Department / Salary	12
3.8.7	Turnover Vs Evaluation	13
3.8.8	Turnover Vs Average monthly hours.....	14
3.8.9	Turnover Vs Satisfaction level.....	15
3.8.10	Salary Vs Satisfaction level	15
3.8.11	Satisfaction level Vs Experience.....	16
3.8.12	Average monthly hours Vs Experience.....	17
3.8.13	Evaluation Vs Experience	17
4	Predictive Modelling:	19
4.1	Logistic Regression.....	19
4.2	Logistic Regression – Revised.....	20
4.3	Cross Validation Method	21
4.4	Confusion Matrix and Statistics.....	21
4.5	Variable Selection by using Boruta Algorithm.....	23
4.6	Decision tree	25
5	Results.....	27
6	Conclusion:	27
7	References:	28

1 Abstract

The article presents factors attributing to the employee attrition, which is one amongst the biggest challenges faced by organizations. There could be several reasons for employee turnover. In this project, we have considered several different parameters like satisfaction level, their experience with the company, average monthly working hours, last evaluation rating and promotion in the last five years.

The analysis was conducted based on the Human Resource Analytics dataset from kaggle. Two focus areas are addressed in this report. The first objective is why employees are leaving the organization. The second objective is to predict which set / section of employees may probably leave the organization. To answer the objective and to understand whether these variables have any impact on the attrition, data was analyzed using several different functions and libraries available in R studio. Predictive modelling techniques like logistics regression, ROC and Decision tree were used.

2 Background

The primary focus of the project is to understand why the companies best and valuable employees are leaving the organization and predict the employees who will be leaving the next. The Human Resource Analytics dataset consists of 14999 rows and 10 columns which means it includes the data of 14999 employees with respect to 10 variables that includes Satisfaction level, Last evaluation, Number of projects, Average monthly work hours, Time spent in the company, work accident, promotion in the last 5 years, Department and Salary. The company name remains anonymous, but a similar study / research can be performed by any HR organization by accumulating Employee data through surveys.

When good employee leaves the company, there is a multifold impact. There is quantifiable economic loss since it costs more to lose employees. According to the data drawn from various research papers, it costs additional 20% of their wages when the employee leaves. These costs reflect the loss of productivity after their departure, replacement cost, and the reduced productivity while the new employee gets up to speed. With multiple employees leaving every year, there is not only big dent in the budget but it also is detrimental to the moral of the current employees working in the company. Hence, it is important for organizations to find why their first-class employees are leaving prematurely and to predict who could be leaving the organization next. This will help them to create policies to improve Employee Retention.

3 Methodology & Analysis

The methodology followed in this project includes both Prediction and Inference. The details of the techniques used, is described in detail below:

3.1 Set work directory and read the csv file

```
setwd("/Users/Thiya/Desktop/Wayne/Analytics course/Project")
HR = read.csv(file = 'HR.csv', header = T, sep = ',')
```

3.2 Dimensions

```
dim(HR)
```

```
## [1] 14999    10
```

The dataset has 14999 rows and 10 column which indicates that the dataset consists of data of 14999 employees with respect to 10 different variables.

3.3 Summary

```
## satisfaction_level last_evaluation number_project average_monthly_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
```

3.4 Change name of columns

```
library("data.table")
```

```
setnames(HR, c('sales','time_spend_company'), c('Department','Experience'))
```

Used library data. Table for set names function, to change names of multiple columns. In this case, we changed the name for “sales” and “time_spend_company” variable as “Department” and “Experience” respectively.

3.5 Attrition Rate

```
attrition<-as.factor(left)  
summary(attrition)
```

```
##      0      1  
## 11428  3571
```

Parameter ‘Left’ denotes the employees who left the Organization.

- No. of Employees who left the organization → 3571.
- No. of Employees who stayed in the organization → 11428.

```
perc_attrition_rate<-sum(left/length(left))*100  
print(perc_attrition_rate)
```

```
## [1] 23.80825
```

Approximately 24 % of the Employees left the organization.

3.6 Correlation Matrix

```
corrplot::corrplot.mixed(correlate_HR,lower = "circle", upper = "number", tl.pos = "lt", diag = "l")
```



#Positively Correlated Features:

- Number_project vs last_evaluation: **0.35**
- Number_project vs average_monthly_hours: **0.42**
- Average_monthly_Hours vs last_evaluation: **0.34**

#Negatively Correlated Feature:

- Satisfaction_level vs turnover: -0.39

3.7 One-Sample T-Test (Measuring Satisfaction Level)

A one-sample t-test checks whether a sample mean differs from the population mean. Let's test to see whether the average satisfaction level of employees that had a turnover differs from the entire employee population.

Hypothesis Testing: Is there significant difference in the means of satisfaction level between employees who had a turnover and the entire employee population?

Null Hypothesis: ($H_0: p_{TS} = p_{ES}$) If the null hypothesis is true, there is no difference in satisfaction level between employees who did turnover and the entire employee population.

Alternate Hypothesis: ($H_A: p_{TS} \neq p_{ES}$) The alternative hypothesis would be that there is a difference in satisfaction level between employees who did turnover and the entire employee population.

```
emp_population_satisfaction <-mean(satisfaction_level)
left_pop<-subset(HR,left==1)
print( c('The mean for the employee population is: ', emp_population_satisfaction) )
```

```
## [1] "The mean for the employee population is: "
## [2] "0.612833522234816"
```

The above output indicates the mean satisfaction level of employees is 0.613

```
emp_turnover_satisfaction <-mean(left_pop$satisfaction_level)
print( c('The mean for the employees that had a turnover is: ',emp_turnover_satisfaction) )
```

```
## [1] "The mean for the employees that had a turnover is: "
## [2] "0.440098011761411"
```

The above output indicates the mean satisfaction level of employees who had left the organization is 0.44

```
#One Sample t-test
t.test(left_pop$satisfaction_level,mu=emp_population_satisfaction)
```

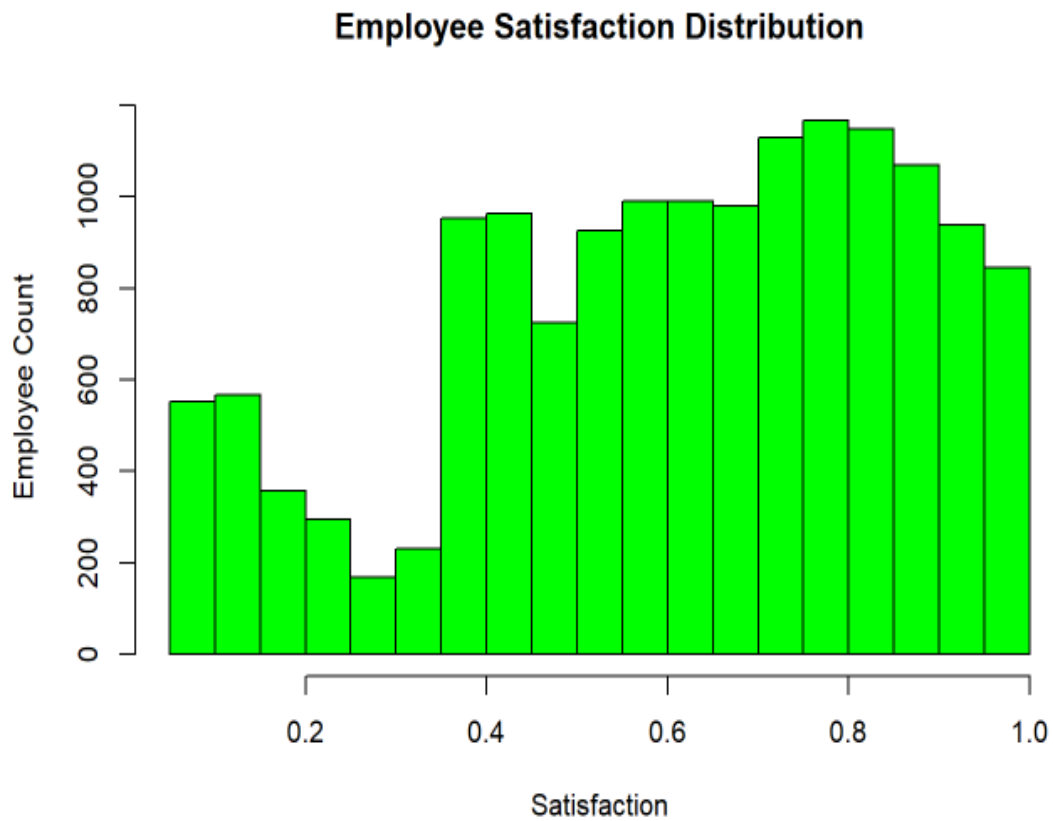
```
## data: left_pop$satisfaction_level
## t = -39.109, df = 3570, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0.6128335
## 95 percent confidence interval:
##  0.4314385 0.4487576
## sample estimates:
## mean of x
##  0.440098
```

The test result shows the test statistic value is equal to -39.109. This test statistic tells us how much the sample mean deviates from the null hypothesis. If the t-statistic lies outside the quantiles of the t-distribution corresponding to our confidence level and degrees of freedom, we reject the null hypothesis.

3.8 Distribution plots

3.8.1 Employee Satisfaction Distribution

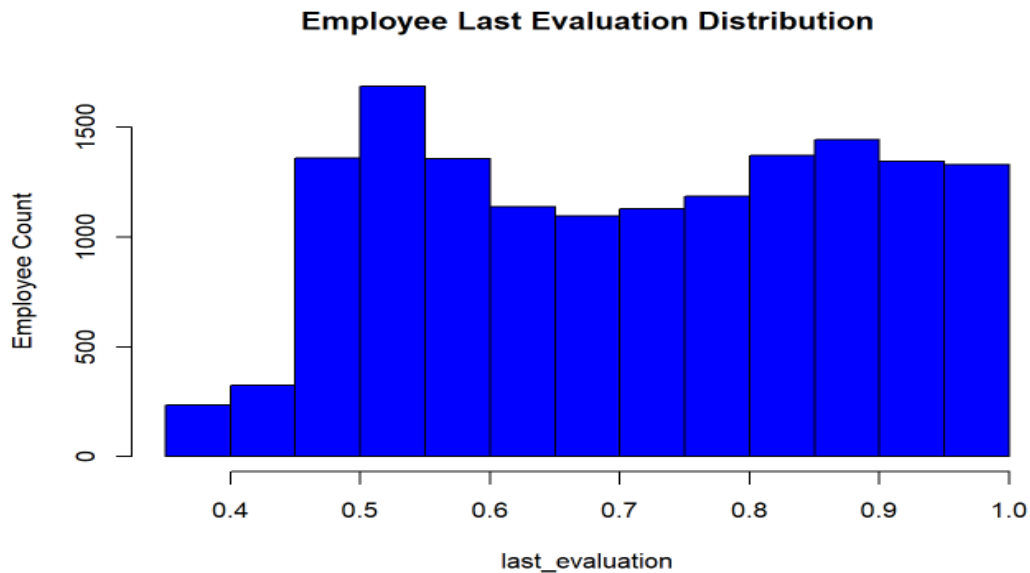
```
#Distribution Plots (Satisfaction - Evaluation - AverageMonthlyHours)
par(mfrow = c(1,1))
hist(satisfaction_level, col = 'green', xlab = "Satisfaction", ylab = "Employee Count", main = "Employee Satisfaction Distribution")
```



Employees with Low and High Satisfaction tend to leave the Organization.

```
hist(last_evaluation, col = 'blue', xlab = "last_evaluation", ylab = "Employee Count", main = "Employee Last Evaluation Distribution")
```

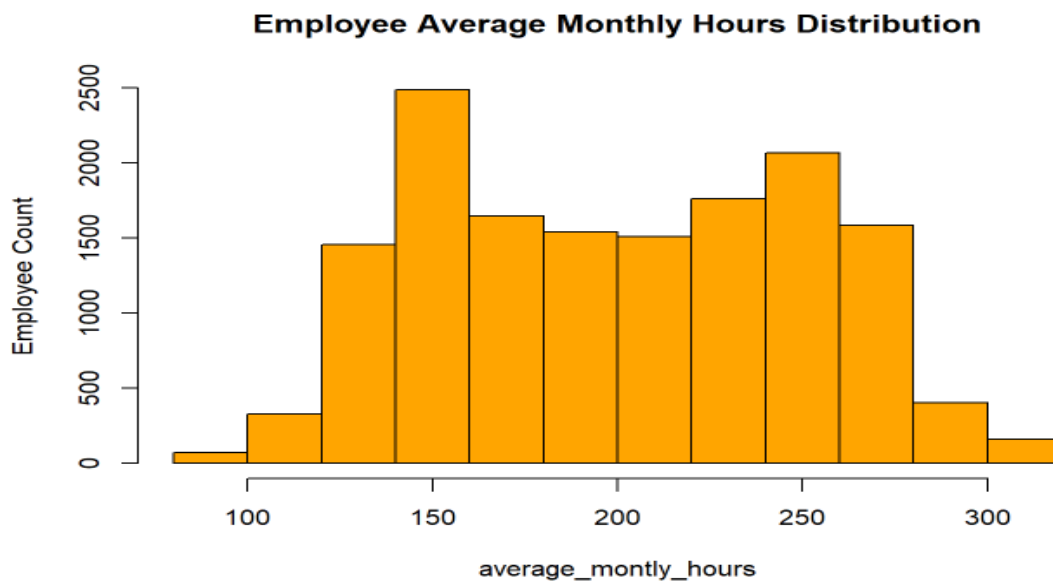
3.8.2 Employee Last Evaluation Distribution



```
hist(average_monthly_hours, col = 'orange', xlab = "average_monthly_hours", ylab = "Employee Count", main = "Employee Average Monthly Hours Distribution")
```

Employees with Low evaluation (< 0.6) and with High Evaluation (> 0.8) tend to leave the organization.

3.8.3 Employee Average Monthly Hours Distribution



Employees who work less than 150 hours or more than 250 hours tend to leave

3.8.4 Salary Vs Turnover

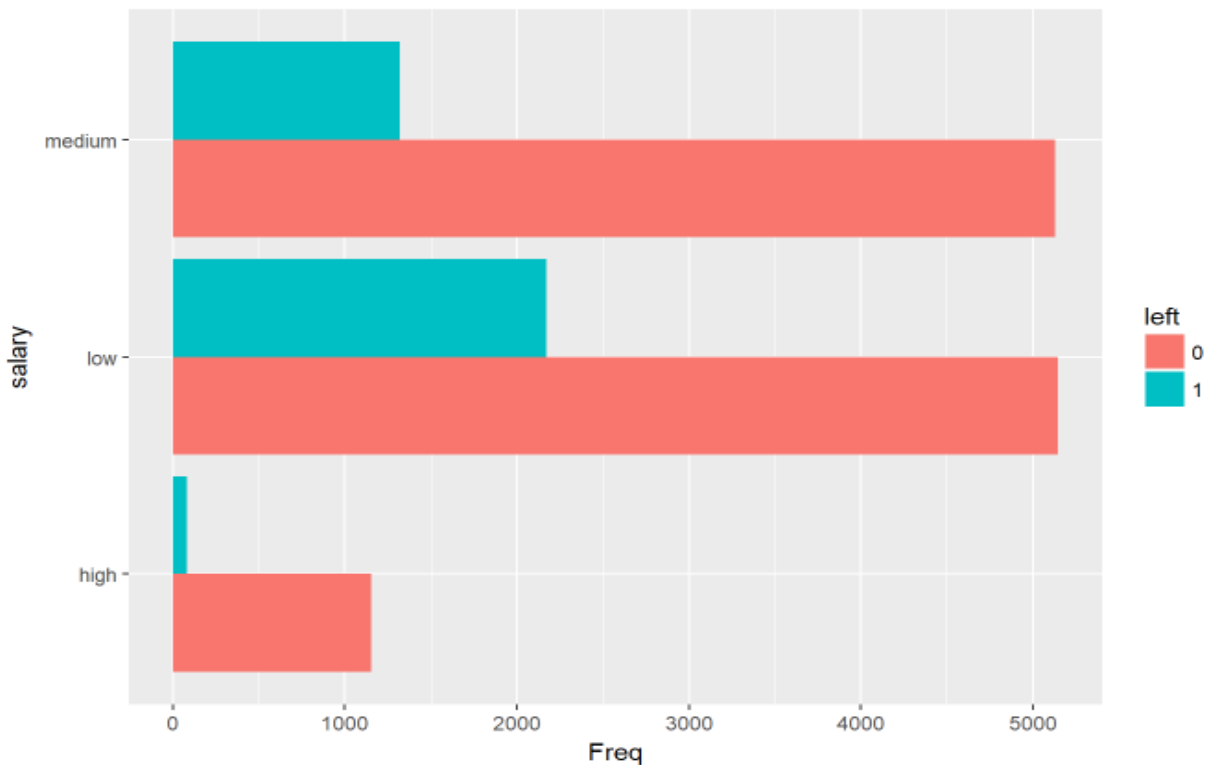
```
vara <- table(salary,left)
varb <- as.data.frame(vara)
print(varb)
```

```
##  salary left Freq
## 1   high    0 1155
## 2    low    0 5144
## 3 medium    0 5129
## 4   high    1   82
## 5    low    1 2172
## 6 medium    1 1317
```

```
library(ggplot2)
```

ggplot2 library used for data visualization & interpretation.

```
p<-ggplot(varb, aes(x=salary,y=Freq,fill=left)) +
  geom_bar(position="dodge",stat='identity') + coord_flip()
print(p)
```



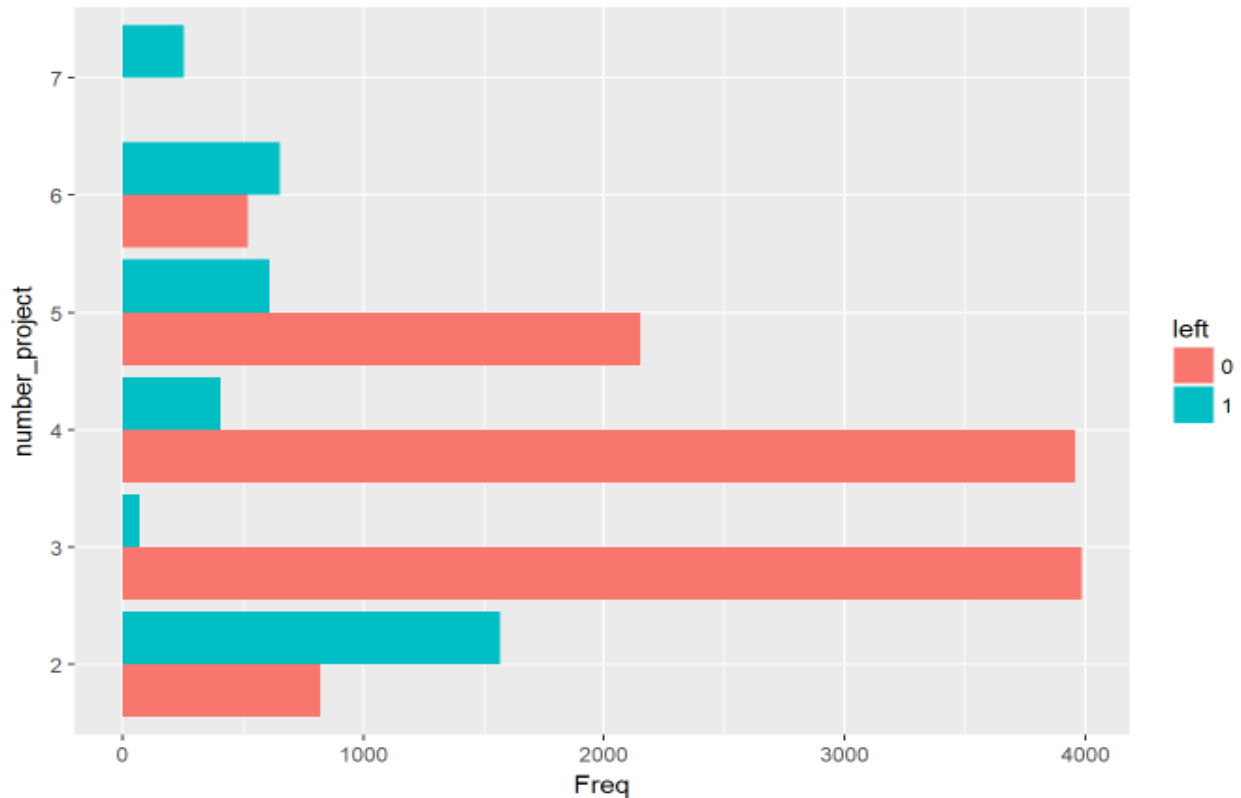
Majority of employees who left either had low or medium salary. Barely any employees left with high salary. Employees with low to average salaries tend to leave the company.

3.8.5 Number of projects Vs Turnover

```
varc<-table(number_project,left)
vard<-as.data.frame(varc)
```

```
p<-ggplot(vard, aes(x=number_project,y=Freq,fill=left)) +
  geom_bar(position="dodge",stat='identity') + coord_flip()

print(p)
```



More than half of the employees with 2,6, or 7 projects left the company. Majority of the employees who did not leave the company had 3,4, or 5 projects All the employees with 7 projects left the company. There is an increase in employee turnover rate as project count increases

3.8.6 Distribution Plots – Department / Salary

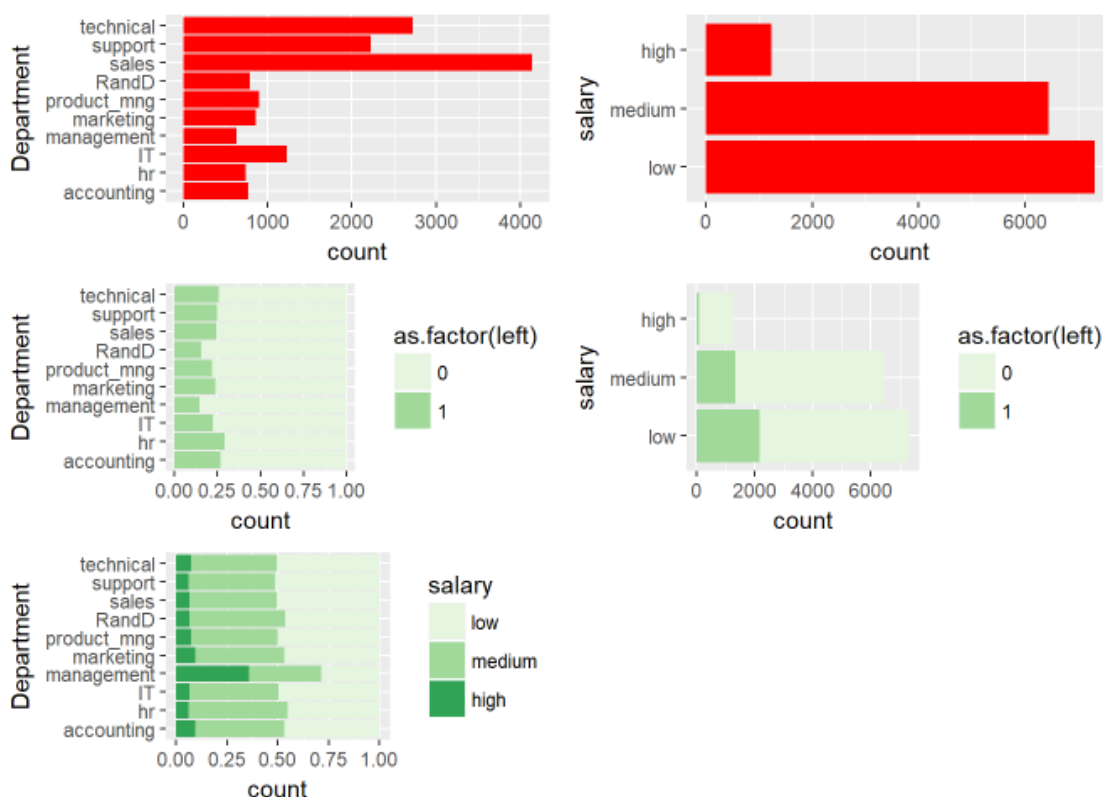
```
HR$Department<-as.factor(HR$Department)
HR$salary<-as.factor(HR$salary)
HR$salary<-ordered(HR$salary,levels=c("low","medium","high"))
```

```
plot_sales<-ggplot(HR,aes(x= Department))+geom_bar(fill="red")+coord_flip()
plot_salary<-ggplot(HR,aes(x=salary))+geom_bar(fill="red")+coord_flip()
plot_sales_left<-ggplot(HR,aes(x=Department,fill=as.factor(left)))+geom_bar(position="fill")+coord_flip()+scale_fill_brewer(
palette="Greens")
plot_salary_left<-ggplot(HR,aes(x=salary,fill=as.factor(left)))+geom_bar()+coord_flip()+scale_fill_brewer(palette="green")
```

```
library(gridExtra)
```

We used library grid. Extra for arranging multiple grid-based plots on a page by using grid. Arrange function

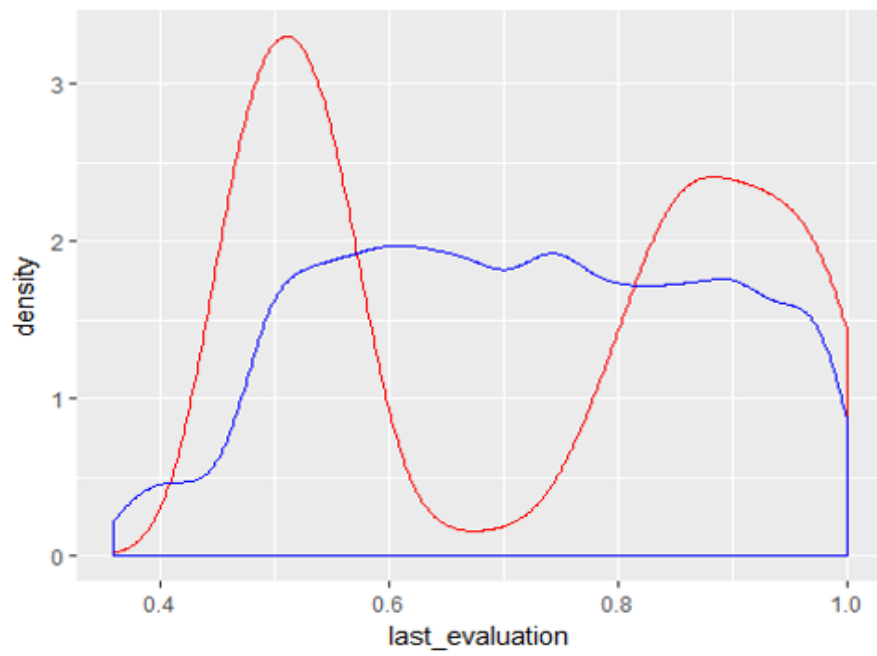
```
grid.arrange(plot_sales,plot_salary,plot_sales_left,plot_salary_left,plot_salary_sales,ncol=2)
```



3.8.7 Turnover Vs Evaluation

```
left_data<-subset(HR,left==1)
stay_data<-subset(HR,left==0)
```

```
ggplot() + geom_density(aes(x=last_evaluation), colour="red", data=left_data) +
geom_density(aes(x=last_evaluation), colour="blue", data=stay_data)
```



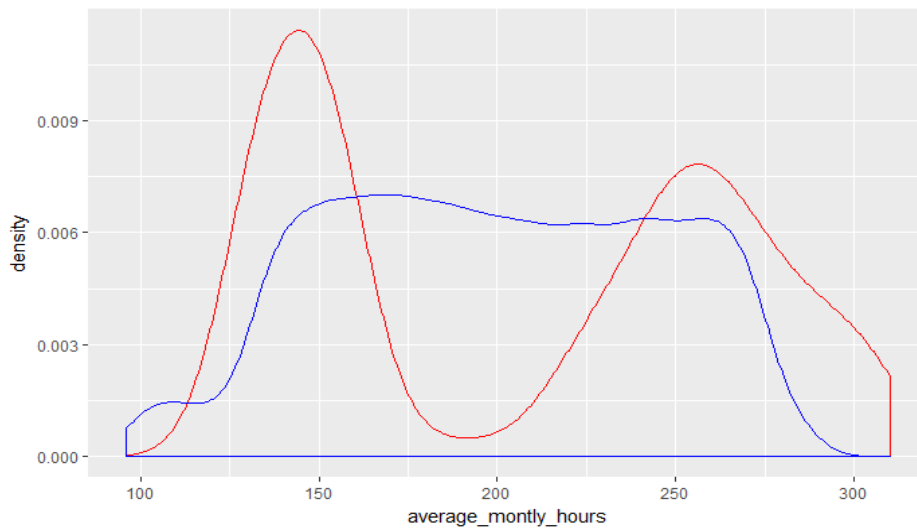
Red indicates Employees who left the organization

Blue indicates Employees who stayed with the organization.

There is a bi-modal distribution for those that had a turnover. Employees with low performance tend to leave the company more. Employees with high performance tend to leave the company more. The sweet spot for employees that stayed is within 0.6-0.8 evaluation.

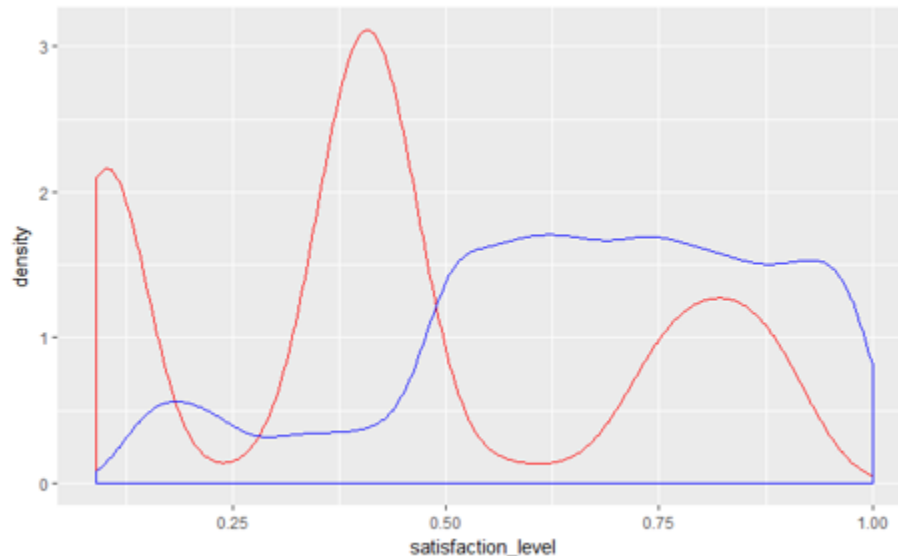
3.8.8 Turnover Vs Average monthly hours

```
ggplot() + geom_density(aes(x=average_monthly_hours), colour="red", data=left_data) +  
  geom_density(aes(x=average_monthly_hours), colour="blue", data=stay_data)
```



Bi-modal distribution for employees that turnover. Employees who had less hours of work (~150 hours or less) left the company more. Employees who had too many hours of work (~250 or more) left the company. Employees who left generally were underworked or overworked.

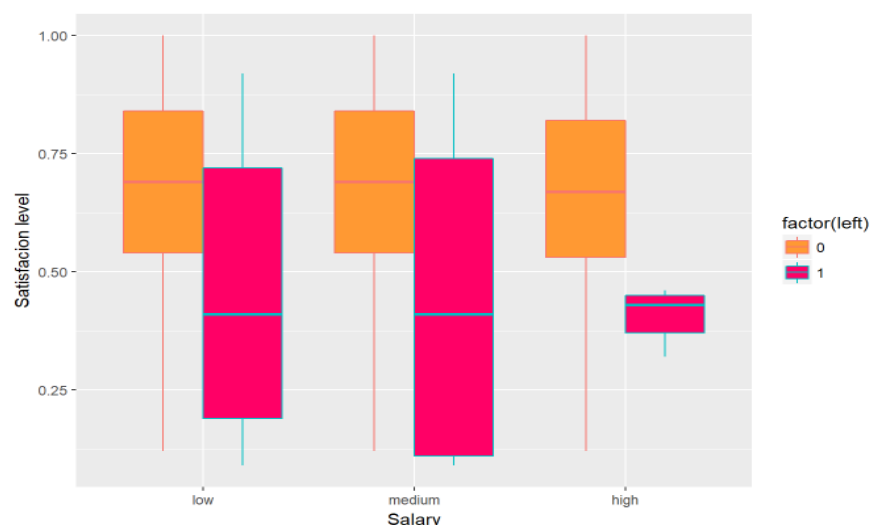
3.8.9 Turnover Vs Satisfaction level



There is a tri-modal distribution for employees that turnover. Employees who had low satisfaction levels (0.2 or less) left the company, employees who had low satisfaction levels (0.3~0.5) left the company more and employees who had high satisfaction levels (0.7 or more) left the company.

3.8.10 Salary Vs Satisfaction level

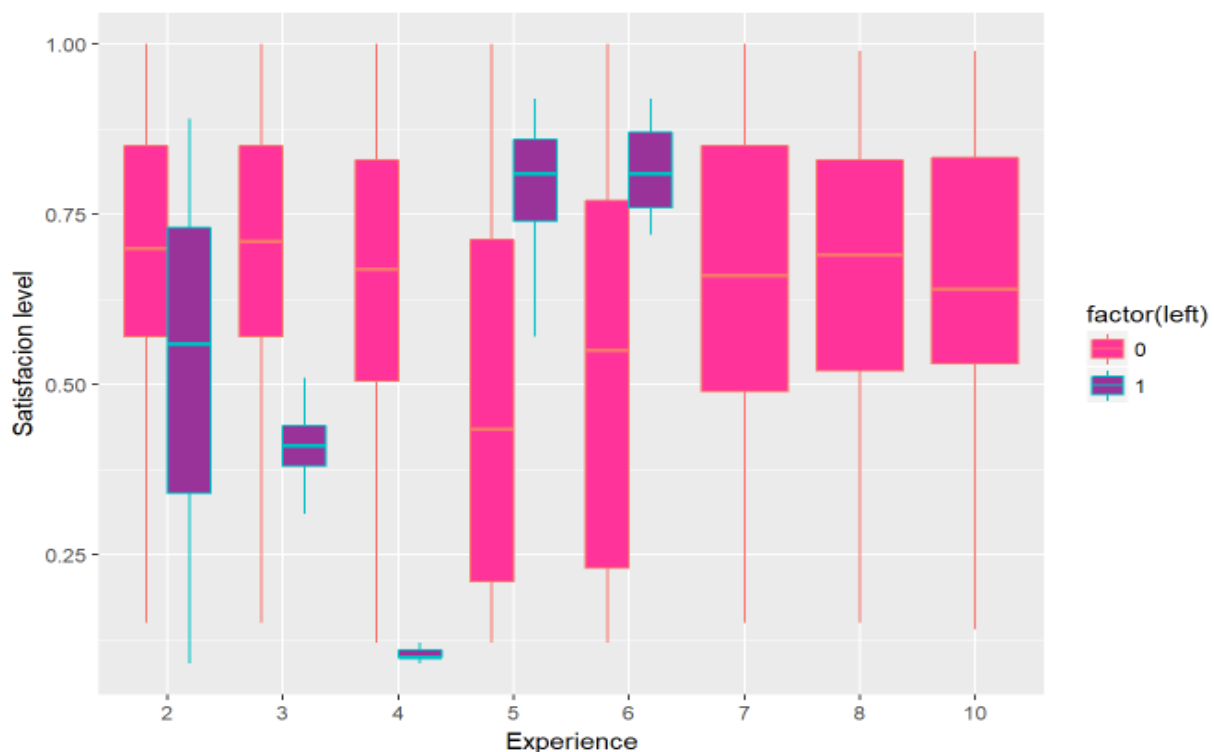
```
ggplot(HR, aes(x = salary, y = satisfaction_level, fill = factor(left), colour = factor(left))) + geom_boxplot(outlier.colour = NA) + xlab("Salary") + ylab("Satisfaction level") + scale_fill_manual(values = c("#FF9933", "#FF0066"))
```



The average satisfaction level of employees who left is lower than those who stayed with the organization. Employees earning high salary but having low satisfaction level, left the organization. Employees earning low or medium salary but having low average satisfaction level, left the organization.

3.8.11 Satisfaction level Vs Experience

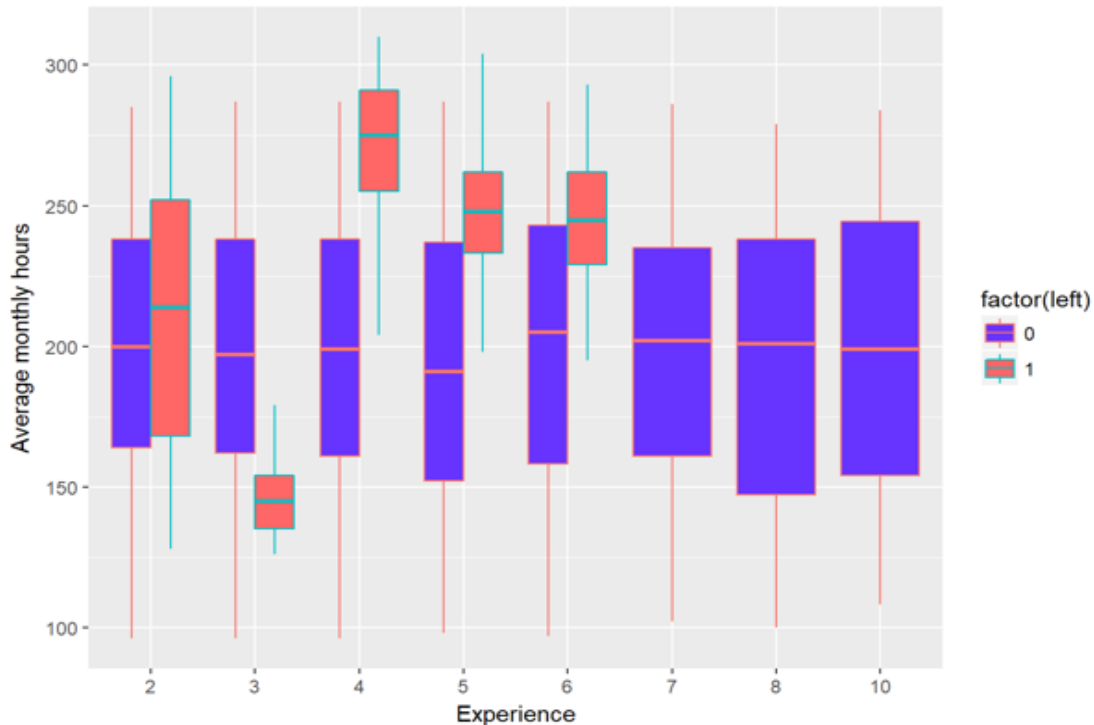
```
ggplot(HR, aes(x = factor(Experience), y = satisfaction_level, fill = factor(left), colour = factor(left))) + geom_boxplot  
(outlier.colour = NA) + xlab("Experience") + ylab("Satisfaction level") + scale_fill_manual(values = c("#ff3399", "#993399"))
```



Average satisfaction level of the employees having two years' experience is lower for those who left the organization than who stayed. Employees having three years' experience with average low satisfaction level of ~0.40, left the organization. Employees having four years' experience with very low satisfaction level of ~<0.12, left the organization. Employees having five to six years' experience having high satisfaction level ~above 0.75, left the organization. Employees more than 6 years' experience tend to stay with the organization.

3.8.12 Average monthly hours Vs Experience

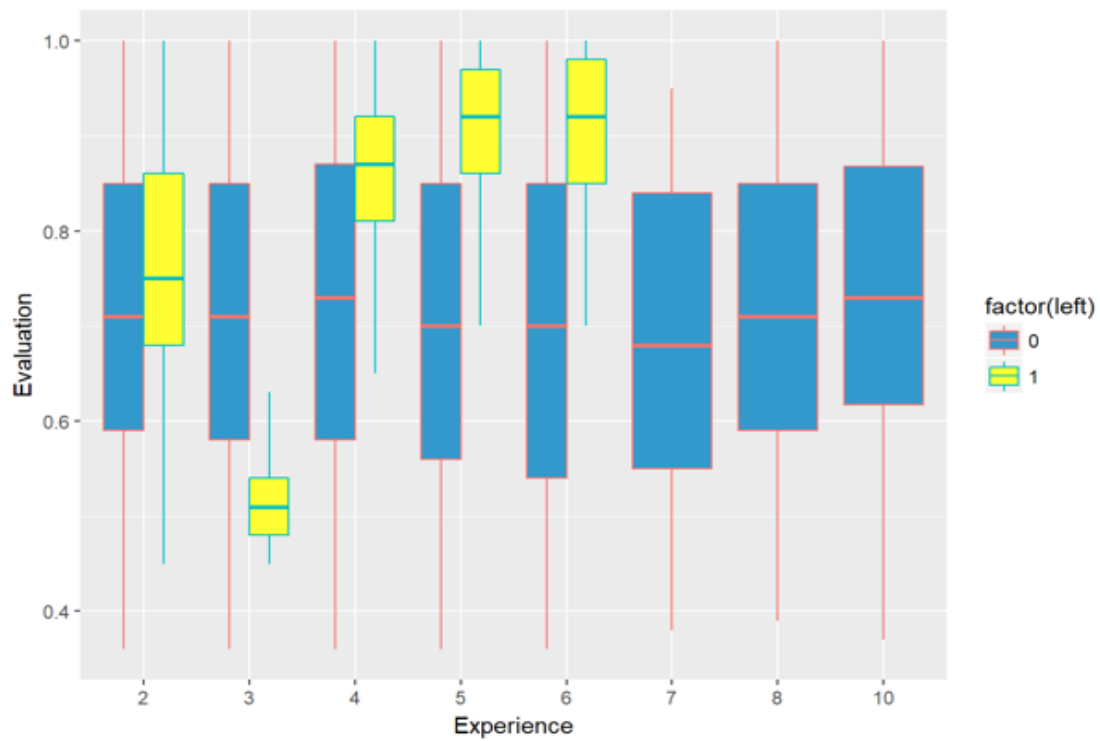
```
ggplot(HR, aes(x = factor(Experience), y = average_monthly_hours, fill = factor(left), colour = factor(left))) + geom_boxplot(
  outlier.colour = NA) + xlab("Experience") + ylab("Average monthly hours") + scale_fill_manual(values = c("#6633ff", "#ff6666"))
```



Employees with 2 years' experience who worked (average monthly hours) for ~215 left the organization. Employees with 3 years' experience who worked an average of ~140-150 hours monthly left the organization. Employees with 4-6 years' experience who were overworked left the organization. Employees with more than 6 years' experience tend to stay with the organization.

3.8.13 Evaluation Vs Experience

```
ggplot(HR, aes(x = factor(Experience), y = last_evaluation, fill = factor(left), colour = factor(left))) + geom_boxplot(
  outlier.colour = NA) + xlab("Experience") + ylab("Evaluation") + scale_fill_manual(values = c("#3399cc", "#ffff33"))
```



Employees having 3 years' experience and having low evaluation rating left the organization. Employees having 4-6 years' experience even with high evaluation rating left the organization.

4 Predictive Modelling:

Logistic regression and Decision were used for our predictive analysis. Logistic regression is used to find the probability of event=Success and event=Failure. Logistic regression is used for categorical / qualitative variables (0/ 1, True/ False, Yes/ No).

4.1 Logistic Regression

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

```
fit.model1 <- glm(left ~ satisfaction_level + last_evaluation + number_project + average_monthly_hours + Experience + Work_accident + promotion_last_5years + Department + salary, data = HR, family = 'binomial')
summary(fit.model1)
```

```
Call:
glm(formula = left ~ satisfaction_level + last_evaluation + number_project +
    average_monthly_hours + time_spend_company + Work_accident +
    promotion_last_5years + sales + salary, family = "binomial",
    data = hr.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2248  -0.6645  -0.4026  -0.1177   3.0688

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.4762862   0.1938373  -7.616 2.61e-14 ***
satisfaction_level -4.1356889   0.0980538 -42.178 < 2e-16 ***
last_evaluation    0.7309032   0.1491787   4.900 9.61e-07 ***
number_project    -0.3150787   0.0213248 -14.775 < 2e-16 ***
average_monthly_hours 0.0044603   0.0005161   8.643 < 2e-16 ***
time_spend_company  0.2677537   0.0155736  17.193 < 2e-16 ***
Work_accident    -1.5298283   0.0895473 -17.084 < 2e-16 ***
promotion_last_5years -1.4301364   0.2574958  -5.554 2.79e-08 ***
saleshr           0.2323779   0.1313084   1.770 0.07678 .
salesIT           -0.1807179   0.1221276  -1.480 0.13894
salesmanagement  -0.4484236   0.1598254  -2.806 0.00502 **
salesmarketing    -0.0120882   0.1319304  -0.092 0.92700
salesproduct_mng  -0.1532529   0.1301538  -1.177 0.23901
salesRandD        -0.5823659   0.1448848  -4.020 5.83e-05 ***
salessales        -0.0387859   0.1024006  -0.379 0.70486
salessupport      0.0500251   0.1092834   0.458 0.64713
salestechnical     0.0701464   0.1065379   0.658 0.51027
salarylow         1.9440627   0.1286272  15.114 < 2e-16 ***
salarymedium      1.4132244   0.1293534  10.925 < 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: 16465  on 14998  degrees of freedom
Residual deviance: 12850  on 14980  degrees of freedom
AIC: 12888

Number of Fisher Scoring iterations: 5
```

```

probs <- predict(fit.model1, type = "response")
pred.glm <- rep("Down", length(probs))
pred.glm[probs > 0.5] <- "Up"
table(pred.glm, left)

```

```

      left
pred.glm  0      1
Down 10614 2301
Up    814 1270

```

True Prediction rate is $(1270 + 10614) / (10614 + 2301 + 814 + 1270) = 0.7923$ or 79%. In other words, 21% is the training error.

4.2 Logistic Regression – Revised

```

sample_size <- floor(0.50 * nrow(HR))
training_data <- sample(seq_len(nrow(HR)), size = sample_size)
train <- HR[training_data, ]
test <- HR[-training_data, ]
left.20171129 <- left[-training_data]

```

```

Call:
glm(formula = left ~ satisfaction_level + last_evaluation + number_project +
    average_monthly_hours + time_spend_company + Work_accident +
    promotion_last_5years + salary, family = binomial, data = hr.data,
    subset = training_data)

```

```

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1132  -0.6692  -0.4067  -0.1071   3.0547

```

```

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.4919537   0.2380755   -6.267 3.69e-10 ***
satisfaction_level -4.1306611   0.1384408  -29.837 < 2e-16 ***
last_evaluation   0.7988393   0.2114856    3.777 0.000159 ***
number_project   -0.3349400   0.0300480  -11.147 < 2e-16 ***
average_monthly_hours 0.0047052   0.0007323    6.425 1.32e-10 ***
time_spend_company  0.2261912   0.0217313   10.409 < 2e-16 ***
Work_accident    -1.5028324   0.1216913  -12.350 < 2e-16 ***
promotion_last_5years -1.8088531   0.4058938   -4.456 8.33e-06 ***
salarylow        2.0860147   0.1755432   11.883 < 2e-16 ***
salarymedium     1.5101186   0.1768259    8.540 < 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: 8274.9  on 7498  degrees of freedom
Residual deviance: 6477.4  on 7489  degrees of freedom
AIC: 6497.4

```

```

Number of Fisher Scoring iterations: 6

```

```
fit.model2 <- glm(left ~ satisfaction_level + last_evaluation + number_project + average_monthly_hours + Experience + Work_accident + promotion_last_5years + Department + salary, data = HR, family = binomial, subset = training_data)
summary(fit.model2)
```

```
probs2 <- predict(fit.model2, test, type = "response")
pred.glm2 <- rep("Down", length(probs2))
pred.glm2[probs2 > 0.5] <- "Up"
table(pred.glm2, left.20171129)
```

	left.20171129	
pred.glm2	0	1
Down	5323	1084
Up	410	683

True Prediction rate is 80%.

4.3 Cross Validation Method

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

4.4 Confusion Matrix and Statistics

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. As our model is a binary classifier we can see the factors like accuracy, sensitivity(TPR), specificity (FPR), Kappa Value and more.

```
library("caret")
```

```
train_control<- trainControl(method="cv", number=5, repeats=3)
```

```
rpartmodel<- train(left~., data=HR, trControl=train_control, method="rpart")

# to make predictions
predictions<- predict(rpartmodel,HR)
HR_analytics_tree<- cbind(HR,predictions)
# For summarizing results and to check the performance we are creating confusion matrix with the predicting data.
confusionMatrix<- confusionMatrix(HR_analytics_tree$predictions,HR_analytics_tree$left)
confusionMatrix
```

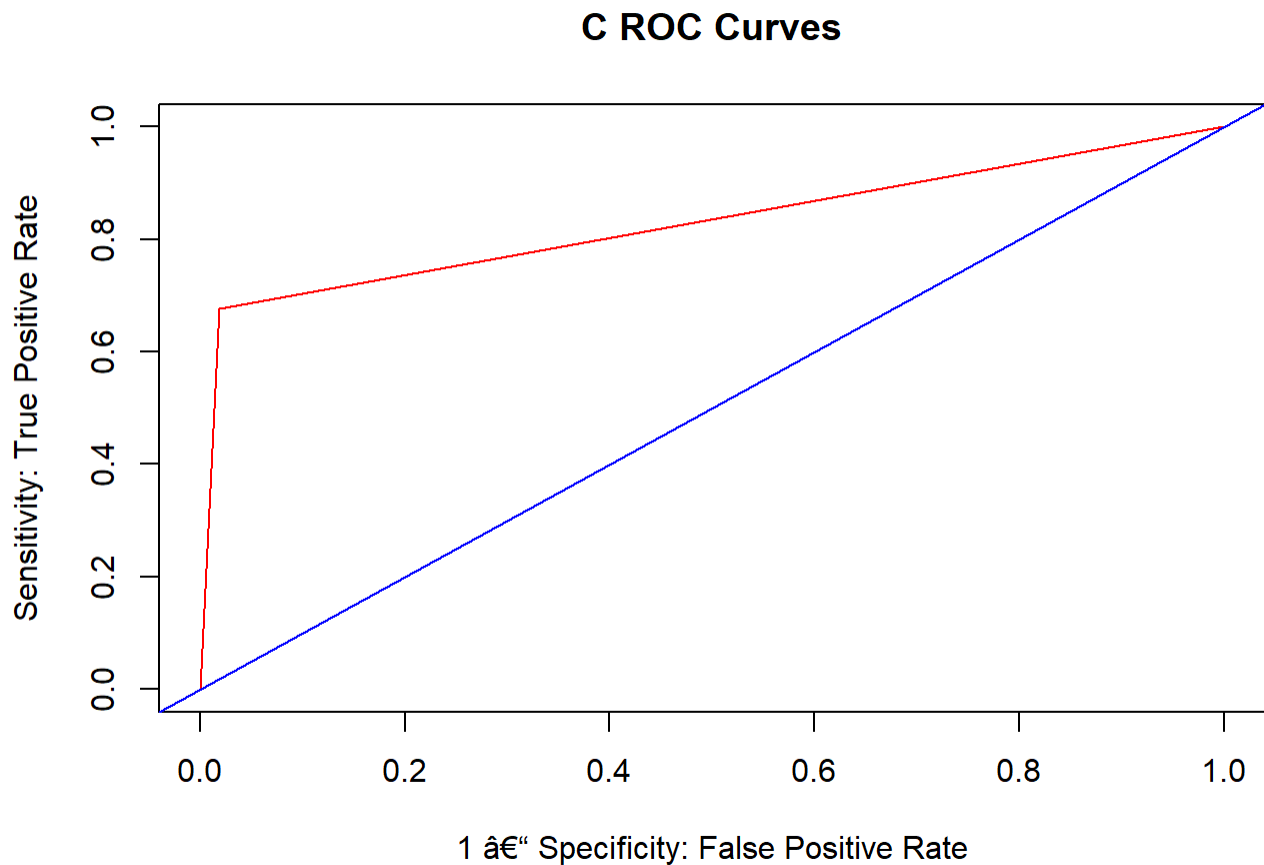
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##           0 11217  1156
##           1   211  2415
##
##           Accuracy : 0.9089
##           95% CI : (0.9041, 0.9134)
##           No Information Rate : 0.7619
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.7236
##           Mcnemar's Test P-Value : < 2.2e-16
##
##           Sensitivity : 0.9815
##           Specificity : 0.6763
##           Pos Pred Value : 0.9066
##           Neg Pred Value : 0.9196
##           Prevalence : 0.7619
##           Detection Rate : 0.7478
##           Detection Prevalence : 0.8249
##           Balanced Accuracy : 0.8289
##
##           'Positive' Class : 0
##
```

As we can see the accuracy for this method is 90% and the kappa value is 72%.

Plotting ROC with the above confusion matrix. ROC stands for Receiver Operating Characteristic curve, it is a graphical plot that illustrates the diagnostic ability of a binary classifier system. It is plotted between TPR and FPR, TPR stands for True positive rate also called as Sensitivity on y-axis and FPR that is False positive rate also called Specificity on x-axis for different threshold points. The closer the ROC curve is to the upper left corner, the higher the overall accuracy of model.

```
# As we can see using the tree analysis accuracy is 90% which is good.
# Now plotting the ROC curve
library("ROCR")
HR_analytics_tree$predictions <- as.numeric(paste(HR_analytics_tree$predictions))
perf.obj <- prediction(predictions=HR_analytics_tree$predictions, labels=HR_analytics_tree$left)
## Get data for ROC curve
roc.obj <- performance(perf.obj, measure="tpr", x.measure="fpr")
plot(roc.obj,
```

```
main="C ROC Curves",  
xlab="1 - Specificity: False Positive Rate",  
ylab="Sensitivity: True Positive Rate",  
col="red")  
abline(0,1,col="blue")
```



4.5 Variable Selection by using Boruta Algorithm

Boruta is a feature selection algorithm. It works as a wrapper algorithm around Random Forest. Variable selection is an important factor of model building. Here we used Boruta library to determine the most important variables for building a tree.

For predicting the key important variable in this dataset, Boruta package is used. We have total nine attributes in the dataset and according to Boruta variable selection method all are important, it means all attributes are responsible for employee to leaving his job.

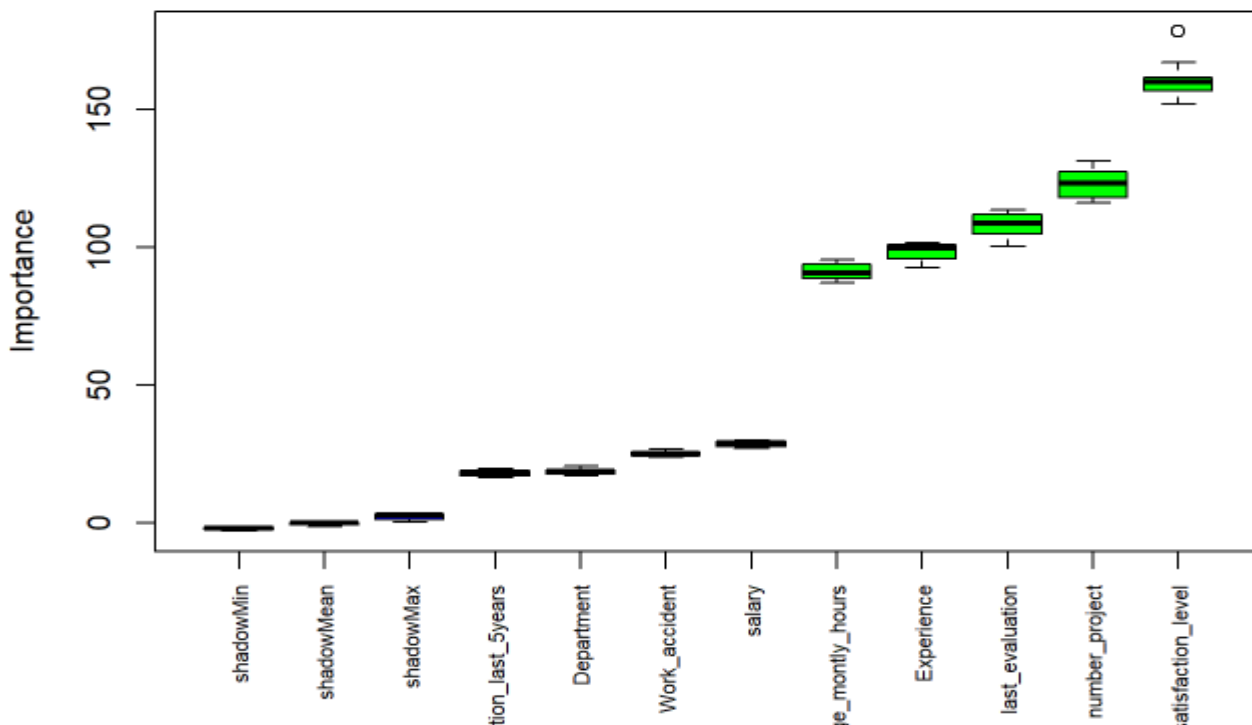
```
library("Boruta")
```

```
boruta.train <- Boruta(left~., data = HR, doTrace = 2)
```

```
print(boruta.train)
```

```
## Boruta performed 10 iterations in 2.257556 mins.  
## 9 attributes confirmed important: average_monthly_hours,  
## Department, Experience, last_evaluation, number_project and 4  
## more;  
## No attributes deemed unimportant.
```

```
plot(boruta.train, xlab = "", xaxt = "n")  
lz<-lapply(1:ncol(boruta.train$ImpHistory),function(i)  
boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])  
names(lz) <- colnames(boruta.train$ImpHistory)  
Labels <- sort(sapply(lz,median))  
axis(side = 1,las=2,labels = names(Labels),  
at = 1:ncol(boruta.train$ImpHistory), cex.axis = 0.7)
```



The following plot shows the relative importance of each factor explanatory attribute. The x-axis represents each of the factor, the green color indicates the attributes that are relevant to prediction. According to the plot as we can see the most important attribute for employees in leaving company is Satisfaction level, the second highest factor is number of projects undertaken.

4.6 Decision tree

Decision tree is a type of supervised learning algorithm (with a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

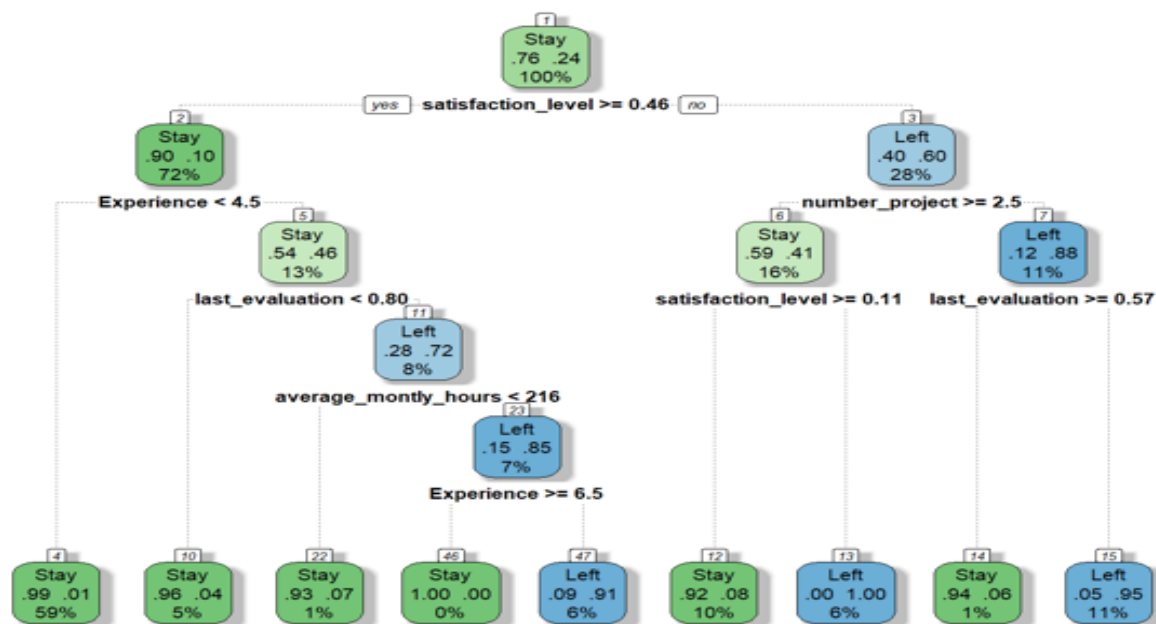
For Tree Learning we are using Rpart Library. This is a recursive partitioning fundamental tool in data mining. For predicting what is the most important factor responsible for leaving the company by employee.

```
library("rpart")
library("rpart.plot")
library("e1071")
```

```
library(rattle)
```

Rattle is unique feature of R for data mining with great GUI which provide easier way to analyze data.

```
model_dt <- rpart(left ~ ., data=train, method="class", minbucket=30)
fancyRpartPlot(model_dt)
```



```
predicted_dt <- predict(model_dt, test, type="class")
table(test$left, predicted_dt)
```

```
      predicted_dt
      Stay Left
Stay 3370    51
Left   87   992
```

```
mean(predicted_dt==test$left)
```

```
0.9693333
```

Decision Tree provides insight into categories of employees who are on the verge of leaving the company. HR teams can focus on Employees who are on the verge of leaving the organization. There were two sets of employees which could be targeted to reduce the attrition rate.

Employees who have a satisfaction level < 0.46 , with number of projects ≤ 2.5 and their last evaluation is < 0.57 may leave. If they were assigned more projects and if their satisfaction level is < 0.11 , the decision tree predict that they may leave.

Second set of employees with satisfaction level ≥ 0.46 , last evaluation is < 0.80 , average monthly hours is > 216 and experience is between 4.5 – 6.5 years. These are overworked employees and they may also leave the organization.

To conclude, successful and overworked employees as well as unhappy and underworked employees may leave the company.

The accuracy of this model is 96.9%

5 Results

- Satisfaction level is a key parameter. Highly satisfied Employees tend to leave the company.
- Employees having satisfaction level < 0.46 are more likely to leave and if the satisfaction level < 0.11 then attrition rate is 100%.
- Employees with low satisfaction level if assigned less than 2 projects are more likely to leave.
- Employees with 5-6 years' experience having higher satisfaction level (> 0.8) and working for > 240 hours per month with last evaluation > 0.75 , are more likely to leave.
- Employees with 4 years' experience and no promotion, leave the organization.
- On a scale of 0 to 1, Employees with lower evaluation (< 0.6) and employees with higher evaluation (> 0.8) tend to leave.
- Employees assigned 6 or 7 projects tend to leave the organization.
- Employees that work for less than 150 hours or more than 250 hours may leave the company.
- Employees between 4 to 6 years of service contribute to higher turnover.
- Employees with 6+ years' experience tend to stay with the organization.

6 Conclusion:

It was unclear why the Employees with High Satisfaction left the organization. Maybe they were not challenged enough, maybe they didn't see any scope to further grow in the organization. The attributes of the data set were limited. Hence there is scope for further learning and deep analysis, by collecting several different data points to conclude why the highly satisfied and successful employees leave the organization.

7 References:

[http://www.cookbook-r.com/Graphs/Colors_\(ggplot2\)/](http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/)

<http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization>

<https://www.rdocumentation.org/packages/corrplot/versions/0.84/topics/corrplot.mixed>

<https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html>

<https://stackoverflow.com/questions/29380447/using-data-tablesetnames-when-some-column-names-might-not-be-present>

<https://stat.ethz.ch/pipermail/r-help/2010-March/230314.html>

<https://www.statmethods.net/advstats/cart.html>

<https://stackoverflow.com/questions/26145525/using-packages-dplyr-and-data-table-in-same-session-causes-copy-error-in-mutate>

<https://onepager.togaware.com/DTreesG.pdf>

<https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/>

<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/#one>