EDUREKA

Certification Project

Kaustav Sadhukhan

email:-kaustav.sadhukhan@tcs.com

First we load the data-set

 # Importing the dataset

emp_data = read.csv("D:/Edureka Assignments/338_cert_proj_datasets_v3.0.csv")

head(emp_data)

Output:-

```
satisfaction_level last_evaluation number_project average_montly_hours
        0.38            0.53              2                 157
        0.80            0.86              5                 262
        0.11            0.88              7                 272
        0.72            0.87              5                 223
        0.37            0.52              2                 159
        0.41            0.50              2                 153
  time_spend_company Work_accident left promotion_last_5years department salary
        3             0     1                     0      sales    low
        6             0     1                     0      sales medium
        4             0     1                     0      sales medium
        5             0     1                     0      sales    low
        3             0     1                     0      sales    low
        3             0     1                     0      sales    low
```

Exploratory Data Analysis

 # Splitting the data-set into Independent and Dependent Variables

x<-emp_data[, -which(names(emp_data) == "left")]

y<-emp_data$left

 # Creating the correlation matrix of attributes

result = cor(x[,sapply(x,is.numeric)],use="complete.obs",method="pearson")

round(result,2)

Output:-

```
                      satisfaction_level last_evaluation number_project
satisfaction_level                  1.00            0.11          -0.14
last_evaluation                     0.11            1.00           0.35
number_project                     -0.14            0.35           1.00
average_montly_hours               -0.02            0.34           0.42
```

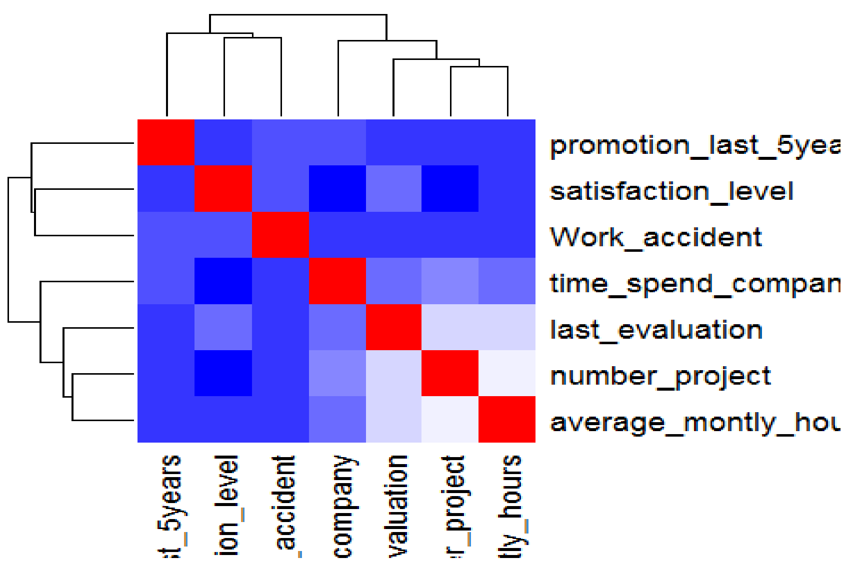
```
time_spend_company                      -0.10           0.13           0.20
Work_accident                            0.06          -0.01           0.00
promotion_last_5years                    0.03          -0.01          -0.01

                      average_montly_hours time_spend_company Work_accident
satisfaction_level                   -0.02              -0.10          0.06
last_evaluation                       0.34               0.13         -0.01
number_project                        0.42               0.20          0.00
average_montly_hours                  1.00               0.13         -0.01
time_spend_company                    0.13               1.00          0.00
Work_accident                        -0.01               0.00          1.00
promotion_last_5years                 0.00               0.07          0.04
                      promotion_last_5years
satisfaction_level                     0.03
last_evaluation                       -0.01
number_project                        -0.01
average_montly_hours                   0.00
time_spend_company                     0.07
Work_accident                          0.04
promotion_last_5years                  1.00
```

 # Plotting the Correlation matrix with a heatmap

col<- colorRampPalette(c("blue", "white", "red"))(20)

heatmap(x = result, col = col, symm = TRUE)



#We can also create a correlogram

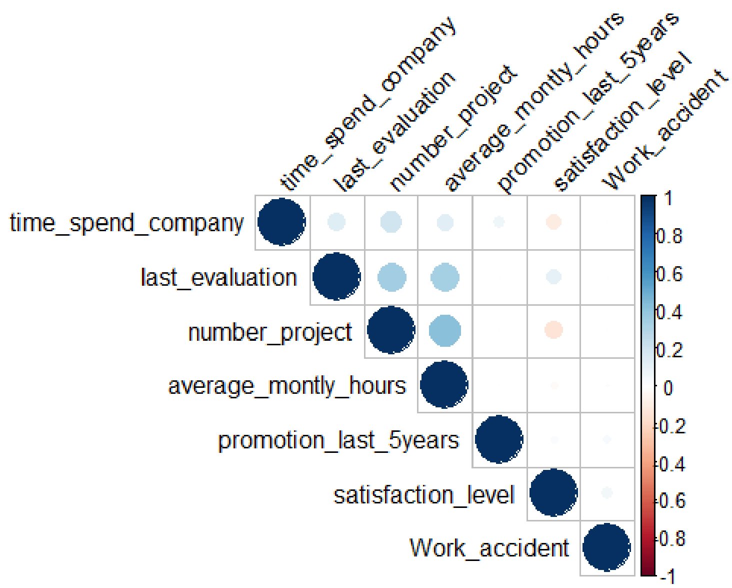
Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients. In the right side of the correlogram, the legend color shows the correlation coefficients and the corresponding colors.

install.packages("corrplot")

library(corrplot)

corrplot(result, type = "upper", order = "hclust",

tl.col = "black", tl.srt = 45)



#Visualizing the characteristics of whole data and only people who left, using plots

install.packages("ggpplot2")

library(ggplot2)

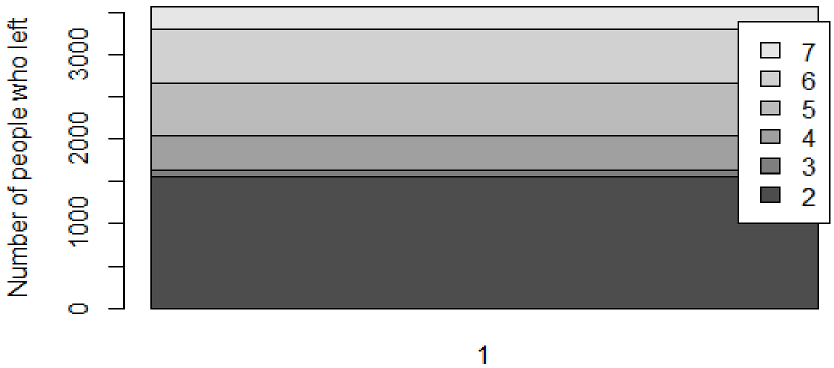
#Subsetting the data with only those who left the company

emp_data_left<-emp_data[emp_data$left=='1',]

#Plotting the barplot for number of projects vs only people who left.

tbl <- with(emp_data_left, table(number_project,left))

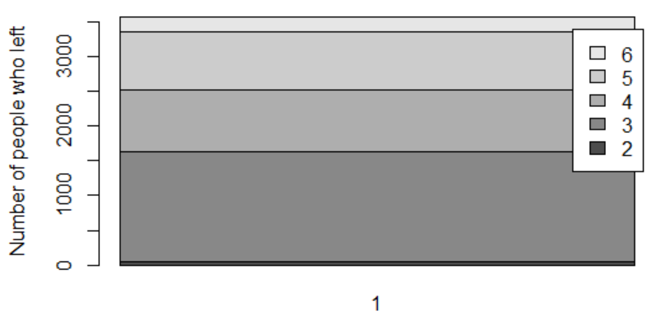
```
barplot(tbl,ylab="Number of people who left",
      legend=TRUE)
```



- **We find the maximum number of people who left the company have worked in only 2 projects.**

#Plotting the barplot for time spend in company vs only people who left.

```
tbl <- with(emp_data_left, table(time_spend_company,left))
barplot(tbl,ylab="Number of people who left",
      legend=TRUE)
```



- **We find the maximum number of people who left the company have 3 years of experience.**

#Plotting the barplot for work accident vs only people who left.

```
tbl <- with(emp_data_left, table(Work_accident,left))

barplot(tbl,ylab="Number of people who left",

    legend=TRUE)
```

- **We find the maximum number of people who left the company do not have work related accidents.**

#Plotting the barplot for promotion given in last 5 years vs only people who left.

```
tbl <- with(emp_data_left, table(promotion_last_5years,left))

barplot(tbl,ylab="Number of people who left",

    legend=TRUE)
```

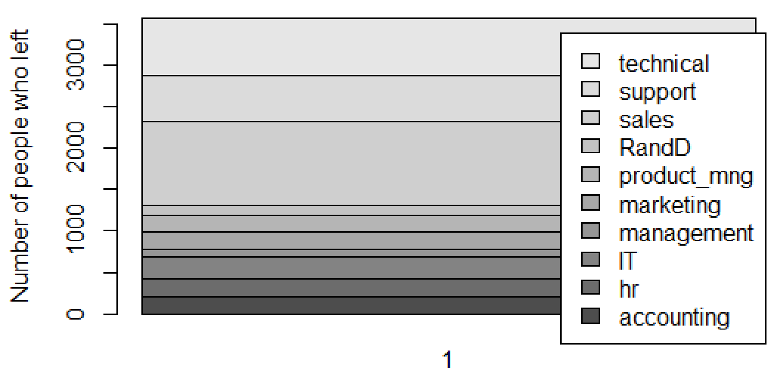
- **We find all of the people who left the company were denied promotions for the last 5 years.**

#Plotting the barplot for department vs only people who left.

tbl <- with(emp_data_left, table(department,left))

barplot(tbl,ylab="Number of people who left",

legend=TRUE)



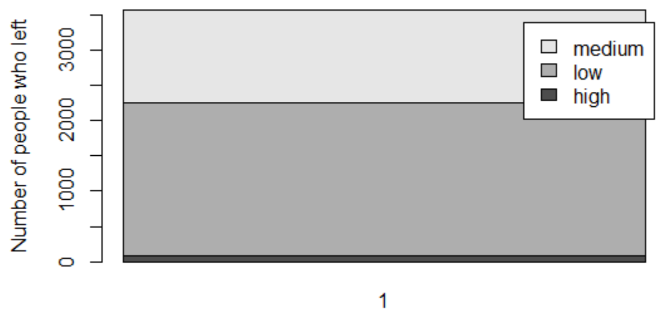
- **Maximum people are leaving the company from Sales department.**

#Plotting the barplot for salary vs only people who left.

tbl <- with(emp_data_left, table(salary,left))

barplot(tbl,ylab="Number of people who left",

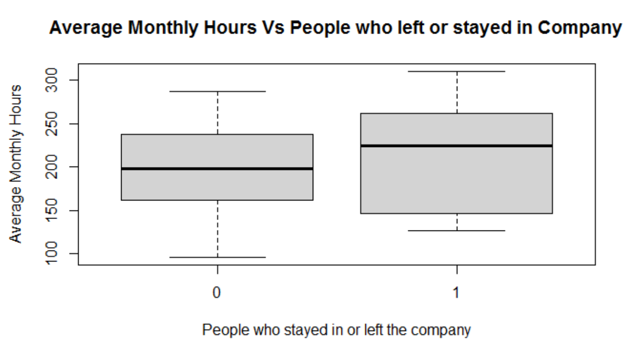
legend=TRUE)



- **We find maximum people who left the company had salary in the low bracket.**

#Plotting the boxplot for average monthly hours vs people who left or stayed in company

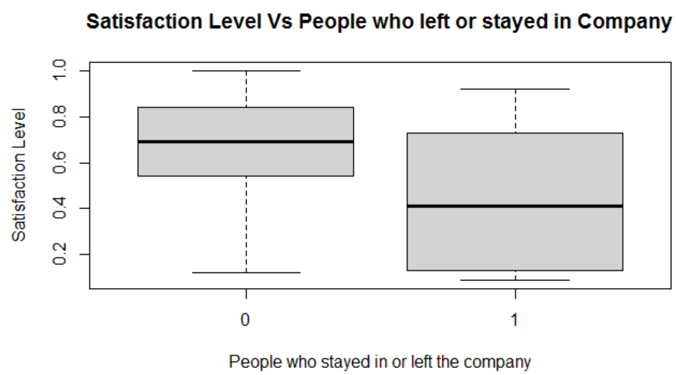
boxplot(emp_data$average_montly_hours~emp_data$left, data = emp_data, xlab = "People who stayed in or left the company", ylab = "Average Monthly Hours",main = "Average Monthly Hours Vs People who left or stayed in Company")



- **We find the median value of average monthly hours is greater for people who left the company.**

#Plotting the boxplot for satisfaction level vs people who left or stayed in company

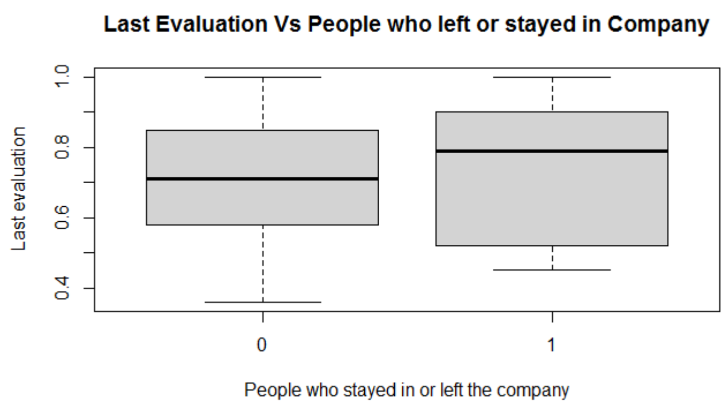
boxplot(emp_data$satisfaction_level~emp_data$left, data = emp_data, xlab = "People who stayed in or left the company", ylab = "Average Monthly Hours",main = "Satisfaction Level Vs People who left or stayed in Company")



- **We find the median value of satisfaction is lower for people who left the company.**

#Plotting the boxplot for last evaluation vs people who left or stayed in company

boxplot(emp_data$last_evaluation~emp_data$left, data = emp_data, xlab = "People who stayed in or left the company",ylab = "Average Monthly Hours",main = "Last Evaluation Vs People who left or stayed in Company")



- **We find the median value of last evaluation score is higher for people who left the company.**

Final Conclusion (Characteristics of the whole data and only the people who left)

1. ***We find the maximum number of people who left the company have worked in only 2 projects.***
2. ***We find the maximum number of people who left the company have 3 years of experience.***
3. ***We find the maximum number of people who left the company do not have work related accidents.***
4. ***We find all of the people who left the company were denied promotions for the last 5 years.***
5. ***Maximum people are leaving the company from Sales department.***
6. ***We find maximum people who left the company had salary in the low bracket.***
7. ***We find the median value of average monthly hours is greater for people who left the company***
8. ***We find the median value of satisfaction is lower for people who left the company.***
9. ***We find the median value of last evaluation score is higher for people who left the company.***

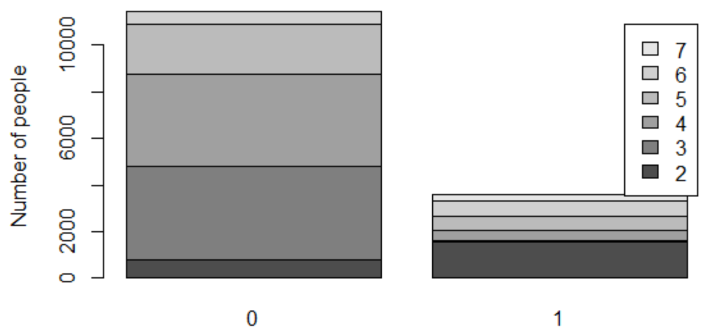
Now we evaluate the attributes for both left and non-left employees

 # Plotting the barplot for number of projects Vs only people who left and non-left.

tbl <- with(emp_data, table(number_project,left))

barplot(tbl,ylab="Number of people",

legend=TRUE)

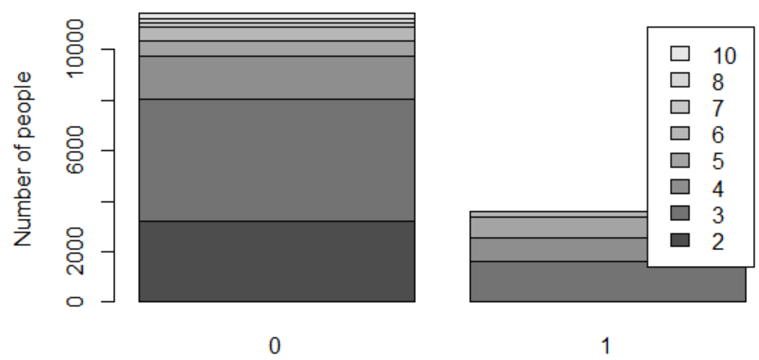


- ***We find that there is not much of an effect of the number of projects on people leaving or staying in company.***

#Plotting the barplot for time spend in company vs only people who left.

tbl <- with(emp_data, table(time_spend_company,left))

barplot(tbl,ylab="Number of people", legend=TRUE)



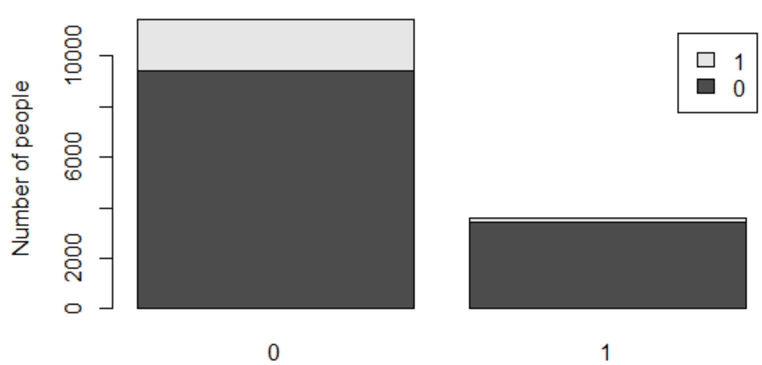
- ***We find that there is not much of an effect of time spent in company on people leaving or staying in company.***

#Plotting the barplot for work accident vs only people who left.

tbl <- with(emp_data, table(Work_accident,left))

barplot(tbl,ylab="Number of people",

legend=TRUE)



- ***We find that work related accidents have a major role in deciding whether people leave or stay in company.***

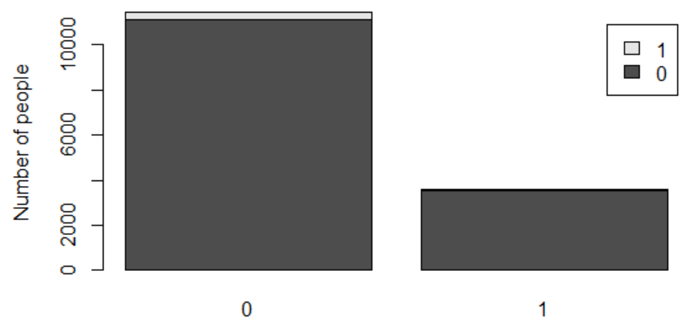
People who are leaving the company have no work related accidents.

#Plotting the barplot for promotion given in last 5 years vs only people who left.

```
tbl <- with(emp_data, table(promotion_last_5years,left))

barplot(tbl,ylab="Number of people",

      legend=TRUE)
```



- ***We find that promotion given in last 5 years have a major role in deciding whether people leave or stay in company.***

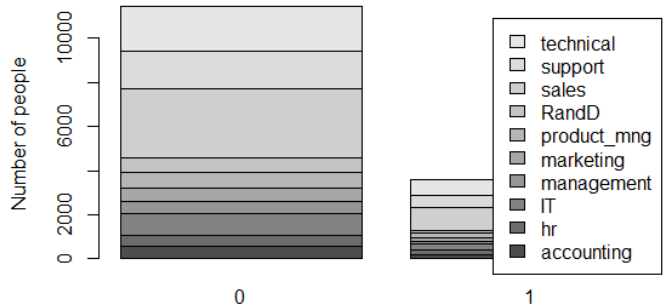
People who are leaving the company did not have promotion for the last 5 years.

#Plotting the barplot for department Vs only people who left.

```
tbl <- with(emp_data, table(department,left))

barplot(tbl,ylab="Number of people",

      legend=TRUE)
```



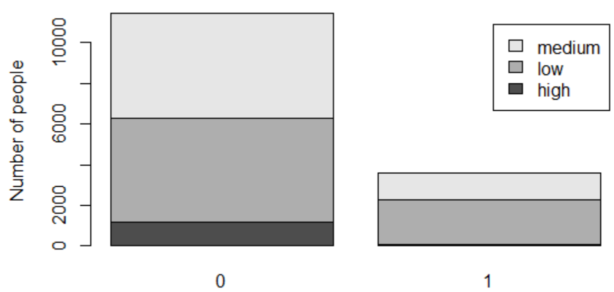
- ***No major effect.***

#Plotting the barplot for salary vs only people who left.

tbl <- with(emp_data, table(salary,left))

barplot(tbl,ylab="Number of people",

legend=TRUE)



- ***We find salary does have a major efffect in deciding whether people will leave the company or stay.***

Maximum of the people who are leave the company are in low salary bracket.

Conclusion (values of each attributes for both left and non-left employees)

1. *We find that there is not much of an effect of the number of projects on people leaving or staying in company*
2. *We find that there is not much of an effect of time spent in company on people leaving or staying in company.*
3. *We find that work related accidents have a major role in deciding whether people leave or stay in company. People who are leaving the company have no work related accidents.*
4. *We find that promotion given in last 5 years have a major role in deciding whether people leave or stay in company. People who are leaving the company did not have promotion for the last 5 years.*
5. *No major effect for department.*
6. *We find salary does have a major efffect in deciding whether people will leave the company or stay. Maximum of the people who are leave the company are in low salary bracket.*

Analyze the department wise turnouts and find out the percentage of employees leaving from each department.

- ***First we find the count of employees leaving the company from different departments and store this in freq***

freq<-table(emp_data_left$department)

freq

```
accounting          hr          IT  management   marketing product_mng       RandD 
       204         215         273          91         203         198         121 
     sales     support   technical 
      1014         555         697 
```

- ***Then we find the percentage of employees leaving from each department***

percentage_freq<-freq*100/sum(freq)

percentage_freq

```
accounting          hr          IT  management   marketing product_mng       RandD
  5.712686    6.020722    7.644917    2.548306    5.684682    5.544665    3.388407
     sales     support   technical
 28.395407   15.541865   19.518342
```

- ***We find the maximum percentage is from the Sales department, which shows, out of the people who are leaving the company, maximum is from Sales department.***

Build a classification model to forecast what the attributes of people who leave the company.

Build models using Decision Tree, Random Forest, Naïve Bayes and SVM techniques and find out the most accurate one.

Data pre-processing

 # Splitting the dataset into the Training set and Test set

```
install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(emp_data$left, SplitRatio = 0.8)
training_set = subset(emp_data, split == TRUE)
test_set = subset(emp_data, split == FALSE)
```

#First we relocate the dependent/target variable to the last column for ease of calculation.

```
library(dplyr)
left=training_set$left
newdata <- training_set %>%
  select(left=left)
training_set<-training_set %>% relocate(left, .after = last_col())
```

 # We do the same thing for test data

```
library(dplyr)
left=test_set$left
newdata <- test_set %>%
  select(left=left)
test_set<-test_set %>% relocate(left, .after = last_col())
```

 # Encoding the target feature as factor

```
training_set$left = factor(training_set$left, levels = c(0, 1))
```

#Encoding the remaining independent categorical variables

```
training_set$department <- factor(training_set$department, levels =
c('accounting','hr','IT','management','marketing','product_mng','RandD','sales','support','t
echnical'), labels = c(0,1,2,3,4,5,6,7,8,9))
training_set$department <- as.numeric(training_set$department)


training_set$salary <- factor(training_set$salary, levels = c('high','low','medium'), labels
= c(0,1,2))
training_set$salary <- as.numeric(training_set$salary)


training_set$Work_accident <- factor(training_set$Work_accident,levels = c(0,1))
training_set$Work_accident <- as.numeric(training_set$Work_accident)


training_set$promotion_last_5years <- factor(training_set$promotion_last_5years,levels
= c(0,1))
training_set$promotion_last_5years <- as.numeric(training_set$promotion_last_5years)


training_set$number_project <- factor(training_set$number_project,levels =
c('2','3','4','5','6','7'), labels = c(0,1,2,3,4,5))
training_set$number_project <- as.numeric(training_set$number_project)
```

training_set$time_spend_company <- factor(training_set$time_spend_company,levels = c('2','3','4','5','6','7','8','10'), labels = c(0,1,2,3,4,5,6,7))
training_set$time_spend_company <- as.numeric(training_set$time_spend_company)

summary(training_set)

Output:-

```
satisfaction_level last_evaluation number_project  average_montly_hours
 Min.   :0.0900     Min.   :0.360   Min.   :1.000   Min.   : 96.0
 1st Qu.:0.4400     1st Qu.:0.560   1st Qu.:2.000   1st Qu.:156.0
 Median :0.6400     Median :0.720   Median :3.000   Median :200.0
 Mean   :0.6122     Mean   :0.717   Mean   :2.804   Mean   :201.1
 3rd Qu.:0.8200     3rd Qu.:0.870   3rd Qu.:4.000   3rd Qu.:245.0
 Max.   :1.0000     Max.   :1.000   Max.   :6.000   Max.   :310.0
 time_spend_company Work_accident   promotion_last_5years   department
 Min.   :1.00       Min.   :1.000   Min.   :1.000         Min.   : 1.000
 1st Qu.:2.00       1st Qu.:1.000   1st Qu.:1.000         1st Qu.: 5.000
 Median :2.00       Median :1.000   Median :1.000         Median : 8.000
 Mean   :2.48       Mean   :1.144   Mean   :1.022         Mean   : 6.938
 3rd Qu.:3.00       3rd Qu.:1.000   3rd Qu.:1.000         3rd Qu.: 9.000
 Max.   :8.00       Max.   :2.000   Max.   :2.000         Max.   :10.000
     salary      left
 Min.   :1.000   0:9142
 1st Qu.:2.000   1:2857
 Median :2.000
 Mean   :2.346
 3rd Qu.:3.000
 Max.   :3.000
```

#Feature Scaling
#We leave out the target variable
training_set[-10]=scale(training_set[-10])

#Same preprocessing is done for test set

 # Encoding the target feature as factor
test_set$left = factor(test_set$left, levels = c(0, 1))

#Encoding the remaining independent categorical variables
test_set$department <- factor(test_set$department, levels = c('accounting','hr','IT','management','marketing','product_mng','RandD','sales','support','technical'), labels = c(0,1,2,3,4,5,6,7,8,9))
test_set$department <- as.numeric(test_set$department)

test_set$salary <- factor(test_set$salary, levels = c('high','low','medium'), labels = c(0,1,2))
test_set$salary <- as.numeric(test_set$salary)

test_set$Work_accident <- factor(test_set$Work_accident,levels = c(0,1))
test_set$Work_accident <- as.numeric(test_set$Work_accident)

test_set$promotion_last_5years <- factor(test_set$promotion_last_5years,levels = c(0,1))
test_set$promotion_last_5years <- as.numeric(test_set$promotion_last_5years)

test_set$number_project <- factor(test_set$number_project,levels = c('2','3','4','5','6','7'), labels = c(0,1,2,3,4,5))
test_set$number_project <- as.numeric(test_set$number_project)

test_set$time_spend_company <- factor(test_set$time_spend_company,levels = c('2','3','4','5','6','7','8','10'), labels = c(0,1,2,3,4,5,6,7))
test_set$time_spend_company <- as.numeric(test_set$time_spend_company)

summary(test_set)

```
satisfaction_level last_evaluation  number_project  average_montly_hours
 Min.   :0.0900     Min.   :0.3600   Min.   :1.000   Min.   : 96
 1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:2.000   1st Qu.:156
 Median :0.6500     Median :0.7100   Median :3.000   Median :201
 Mean   :0.6153     Mean   :0.7124   Mean   :2.799   Mean   :201
 3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:4.000   3rd Qu.:245
 Max.   :1.0000     Max.   :1.0000   Max.   :6.000   Max.   :310
 time_spend_company Work_accident   promotion_last_5years   department
 Min.   :1.000      Min.   :1.000   Min.   :1.00          Min.   : 1.000
 1st Qu.:2.000      1st Qu.:1.000   1st Qu.:1.00          1st Qu.: 5.000
 Median :2.000      Median :1.000   Median :1.00          Median : 8.000
 Mean   :2.499      Mean   :1.149   Mean   :1.02          Mean   : 6.928
 3rd Qu.:3.000      3rd Qu.:1.000   3rd Qu.:1.00          3rd Qu.: 9.000
 Max.   :8.000      Max.   :2.000   Max.   :2.00          Max.   :10.000
     salary      left
 Min.   :1.000   0:2286
 1st Qu.:2.000   1: 714
 Median :2.000
 Mean   :2.352
 3rd Qu.:3.000
 Max.   :3.000
```

#Feature Scaling
#We leave out the target variable
test_set[-10]=scale(test_set[-10])

Modelling

#First we apply SVM with linear kernel

install.packages("e1071")

library(e1071)

classifier = svm(formula = left ~ .,

    data = training_set,

    type = 'C-classification',

    kernel = 'linear')

 # Predicting the Test set results

y_pred = predict(classifier, newdata = test_set[-10])

 # Making the Confusion Matrix

cm = table(test_set[, 10],y_pred)

cm

```
y_pred
         0     1
  0  2140   146
  1   542   172
```

#Calculating the Accuracy

n = sum(cm)

n **#total Records**

nc = nrow(cm)

nc **#Total classes**

diag = diag(cm**) #Correctly classified points**

rowsums = apply(cm,1,sum)

rowsums

colsums = apply(cm,2,sum)

colsums

p = rowsums/n

q = colsums/n

accuracy = sum(diag)/n

accuracy

```
0.7706667
```

precision = diag/colsums

precision

recall = diag/rowsums

recall

f1 = 2*precision*recall/(precision+recall)

f1

data.frame(precision,recall,f1)

```
  precision    recall        f1
0 0.7979120 0.9361330 0.8615137
1 0.5408805 0.2408964 0.3333333
```

#We apply SVM with radial kernel

classifier = svm(formula = left ~ .,

data = training_set,

type = 'C-classification',

kernel = 'radial')

 # Predicting the Test set results

y_pred = predict(classifier, newdata = test_set[-10])

 # Making the Confusion Matrix

cm = table(test_set[, 10], y_pred)

cm

```
y_pred
         0     1
  0   2236    50
  1     69   645
```

#Calculating the Accuracy

n = sum(cm)

n **#total Records**

nc = nrow(cm)

nc **#Total classes**

diag = diag(cm) **#Correctly classified points**

rowsums = apply(cm,1,sum)

rowsums

colsums = apply(cm,2,sum)

colsums

p = rowsums/n

q = colsums/n

accuracy = sum(diag)/n

accuracy

0.9603333

precision = diag/colsums

precision

recall = diag/rowsums

recall

f1 = 2*precision*recall/(precision+recall)

f1

data.frame(precision,recall,f1)

```
  precision    recall        f1
0 0.9700651 0.9781277 0.9740797
1 0.9280576 0.9033613 0.9155429
```

#Now we use naive Bayes

classifier = naiveBayes(x = training_set[-10],

y = training_set$left)

 # Predicting the Test set results

y_pred = predict(classifier, newdata = test_set[-10])

 # Making the Confusion Matrix

cm = table(test_set[, 10], y_pred)

cm

```
y_pred
       0    1
  0 1881  405
  1  207  507
```

#Calculating the Accuracy

n = sum(cm)

n **#total Records**

nc = nrow(cm)

nc **#Total classes**

diag = diag(cm) **#Correctly classified points**

rowsums = apply(cm,1,sum)

rowsums

colsums = apply(cm,2,sum)

colsums

p = rowsums/n

q = colsums/n

accuracy = sum(diag)/n

accuracy

0.796

precision = diag/colsums

precision

recall = diag/rowsums

recall

f1 = 2*precision*recall/(precision+recall)

f1

data.frame(precision,recall,f1)

```
      precision    recall       f1
0     0.9008621  0.8228346  0.8600823
```

```
1       0.5559211   0.7100840   0.6236162
```

#Random Forest Classifier

```
install.packages("randomForest")

library(randomForest)

set.seed(123)

classifier = randomForest(x = training_set[-10],
                          y = training_set$left,
                          ntree = 500)
```

 # Predicting the Test set results

```
y_pred = predict(classifier, newdata = test_set[-10])
```

 # Making the Confusion Matrix

```
cm = table(test_set[, 10], y_pred)

cm
```

```
y_pred
       0    1
  0 2284    2
  1   50  664
```

#Calculating the Accuracy

```
n = sum(cm)
```

n **#total Records**

```
nc = nrow(cm)
```

nc **#Total classes**

diag = diag(cm) **#Correctly classified points**

rowsums = apply(cm,1,sum)

rowsums

colsums = apply(cm,2,sum)

colsums

p = rowsums/n

q = colsums/n

accuracy = sum(diag)/n

accuracy

0.9826667

precision = diag/colsums

precision

recall = diag/rowsums

recall

f1 = 2*precision*recall/(precision+recall)

f1

data.frame(precision,recall,f1)

```
  precision    recall        f1
0 0.9785775 0.9991251 0.9887446
1 0.9969970 0.9299720 0.9623188
```

#Decision Tree Classification

library(rpart)

classifier = rpart(formula = left ~ .,

                data = training_set)

 # Predicting the Test set results

```
y_pred = predict(classifier, newdata = test_set[-10], type = 'class')
```

 # Making the Confusion Matrix

```
cm = table(test_set[, 10], y_pred)

cm
```

```
  y_pred
       0    1
  0 2260   26
  1   89  625
```

 # Plotting the tree

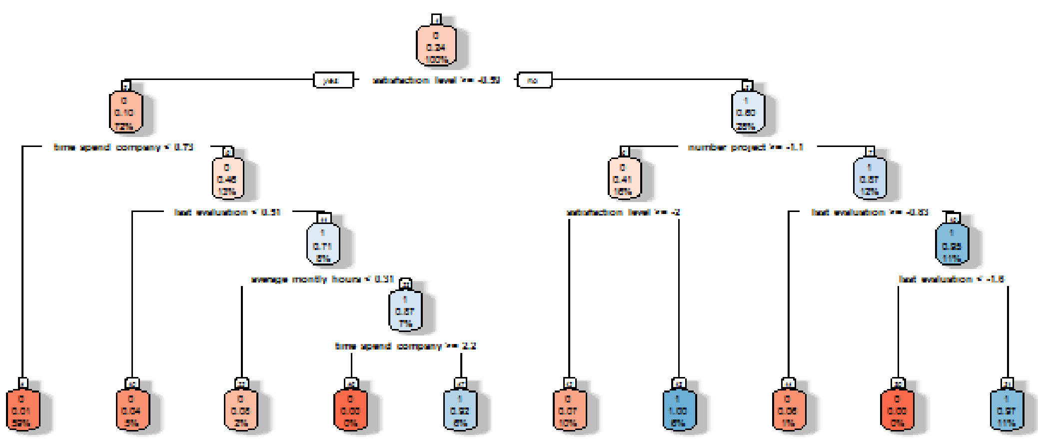
```
install.packages("rpart.plot")

library(rpart.plot)

par(mar = c(2, 2, 2, 2))

rpart.plot(classifier, box.palette="RdBu", shadow.col="gray", nn=TRUE)
```

Decision Tree Diagram



#Calculating the Accuracy

n = sum(cm)

n **#total Records**

nc = nrow(cm)

nc **#Total classes**

diag = diag(cm) **#Correctly classified points**

rowsums = apply(cm,1,sum)

rowsums

colsums = apply(cm,2,sum)

colsums

p = rowsums/n

q = colsums/n

accuracy = sum(diag)/n

accuracy

0.9616667

precision = diag/colsums

precision

recall = diag/rowsums

recall

f1 = 2*precision*recall/(precision+recall)

f1

data.frame(precision,recall,f1)

```
        precision    recall          f1
0       0.9621115  0.9886264  0.9751888
```

1 0.9600614 0.8753501 0.9157509

Comparative Analysis between the different models to find the most accurate one.

Model	*Accuracy*	*Confusion Matrix*	*Precision/Recall/f1 score*
SVM (linear kernel)	0.7706667	y_pred 0 1 0 2140 146 1 542 172	precision recall f1 0 0.7979120 0.9361330 0.8615137 1 0.5408805 0.2408964 0.3333333
SVM(radial Kernel)	0.9603333	y_pred 0 1 0 2236 50 1 69 645	precision recall f1 0 0.9700651 0.9781277 0.9740797 1 0.9280576 0.9033613 0.9155429
Random Forest Classifier(most accurate one)	0.9826667	y_pred 0 1 0 2284 2 1 50 664	precision recall f1 0 0.9785775 0.9991251 0.9887446 1 0.9969970 0.9299720 0.9623188
Naïve Bayes	0.796	y_pred 0 1 0 1881 405 1 207 507	precision recall f1 0 0.9008621 0.8228346 0.8600823 1 0.5559211 0.7100840 0.6236162
Decision Tree	0.9616667	y_pred 0 1 0 2260 26 1 89 625	precision recall f1 0 0.9621115 0.9886264 0.9751888 1 0.9600614 0.8753501 0.9157509

*From the above comparative analysis it is evident that the **Random Forest Classifier** is the most accurate one. We have a good accuracy and the **precision, recall and f1-score** are all quite **good** for the 1s (for the people who left the company)*