UNIVERSITY OF ECONOMICS AND LAW
FACULTY OF INFORMATION SYSTEMS

FINAL PROJECT REPORT

TOPIC: EMPLOYEE CHURN

Lecturer: Ho Nguyen Van, MSc.

Group 4

Ho Chi Minh City, May 27th, 2022

Members of Group 4

NO.	NAME	STUDENT ID
1	Trần Triệu Vy	K204162011
2	Hoàng Nguyễn Quỳnh Giang	K204161987
3	Lê Thị Thùy Linh	K204161990
4	Trương Công Vinh	K204162008
5	Nguyễn Thị Ái Linh	K204161991

Acknowledgments

My team would like to thank Mr. Nguyen Van Ho - Lecturer of Data Analysis with R who equipped us with basic knowledge and skills to complete the midterm project.

However, in the process of researching the topic, due to limited specialized knowledge, we still have many shortcomings when researching, evaluating and presenting the topic. We hope to receive the attention and suggestions of teachers to improve our topic.

Sincerely thanks.

Group 4

TABLE OF CONTENT

List of Figure

List of Table

ABSTRACT

In recent years the market for specialized talent in Portugal has seen a dramatic rise. This has been a catalyst for employee churn and so retaining an employee is a key strategy that can potentially reduce company costs by a large margin.

To face this issue, organizations are adopting proactive strategies in order to retain their employees. These strategies involve, amongst other things, the creation of a predictive model to identify employees in risk of churn.

Employee churn prediction is however a complex problem, and the reason for an employee to leave can stem from different sources. These reasons can be generally placed under three main groups: The employee, or his course in the company, has some observable intrinsic characteristic that is more associated with churn; The employee leaves because some observable event (or multiple events) happened in a given time window; The employee leaves for reasons that are not observable in the available data. This means that there is no consensus on why employees leave an organization. Furthermore, different organizations have different available data making it hard to develop a general solution to employee churn prediction.

This thesis comprises a framework to thoroughly and correctly study the problem of employee churn, for this specific organization. This frameworks embodies understanding the population and different sub-groups in the data and how they relate to churn; creating a classification model for employee churn prediction; assessing the main reasons that drive the decision to leave the company and studying cause-effect relationships between treatments the organization can give (e.g. promotions, raises, etc.) and their effect on retaining employees. This thesis also envisions an extended generic approach, through autoencoders and time-series embeddings, to thoroughly understand the different

types of churning employees and place them in one of the three groups mentioned in the previous paragraph.

Keywords: Employee churn, Predictive modeling, Machine learning, Deep-Learning, Time-series, Embeddings, Causal inference . . .

CHAPTER 01: INTRODUCTION

The issue of employee churn has become an important part of a company's strategy due to the, generally, negative impact it has on productivity, and also the high cost associated with it. Considering that, in the event of replacing an employee: the human resources team needs to dispend around two to eight weeks to find a good candidate for the given role; a number of interviews must be conducted in order to find the optimal candidate; the candidate must have a gradual learning process to get familiarized with the role; we rapidly come to the conclusion that replacing an employee is a costly process.

In a 2012 study from the American center for progress the median cost of replacing an employee has been identified as being around 21,4% of his annual wage. This cost is aggravated when it is necessary to give the new employee some sort of formation, or when the employee that left was a high performer. In larger organizations this cost represents a higher issue. In the fiscal year of 1997 an Israeli high-tech firm lost 2.8 million US dollars (16,5% of its before-tax income) due to employee churn. A more recent case study by Eric Siegel, founder of Predictive Analytics World, can be found in Hewlett-Packard (HP).In his blog post adapted from his book, he states that starting in 2011 the organization adopted to create a predictive model for employee churn with an estimated potential cost saving of 300 million US dollars. Having the ability to proactively act towards retaining identified potential churners makes it possible to transform the high cost associated with these events into a high return on investment (ROI). In companies of high dimension the adoption of employee retention strategies greatly reduces company costs as described in Hewlett-Packard's case study.

However it is important to note that churn is not in all cases a negative event, which makes it difficult to determine a true cost for churn. In some cases, especially those in which the employee is under-performing or unmotivated, it may be beneficial for both the worker and the organization if that employee is replaced. This might boost productivity and creativity in the organization.

This event can be further divided into three categories: voluntary, involuntary (induced by the organization) and retirement. The main interest in this problem is centered around voluntary churn. For involuntary churn the organization has a direct influence on the event, being ultimately the company's final decision to terminate a contract or not. Retirement is, generally, legally enforced and so these two types of churn are not random variables that make sense to predict. The decision to voluntarily leave the organization can be centered around multiple factors. These factors can be grouped in three generic groups: Intrinsic to the employee or its course in the company (A set of observable characteristics more associated with churn, e.g.: job dissatisfaction), Associated with an observable event happening in a specific time-window (e.g.: moving to a new city, having a child), or some external reason that is unobserved in the data.

Furthermore the contagious effect of churn can also be observed for employees, where the churn of a coworker can influence other people's decisions. There is no consensus on what are the main motivations for churn, with the factors ranging from age, tenure, pay, job satisfaction to education, recognition, burnout and many other different reasons. This makes it so that no organization is alike, since they are formed of different people and only historical data can be used to best determine the decision plan for HR management. This fact hinders the possibility of creating a generic solution to employee churn prediction that might work on multiple organizations.

CHAPTER 02: RELATED WORK

Churn prediction, particularly customer churn prediction, attracted huge attention from researchers. For instance, Coussement and Van den Poel studied the problem of optimizing the performance of a decision support system for churn prediction [1]. They studied the effect of textual information in the churn prediction method. They found that adding unstructured, textual information into a conventional churn prediction model resulted in a significant increase in predictive performance. In a similar study, Wei and Chiu propose churn prediction of telecommunication customers by analyzing call details of the customers [2]. Coussement and Van den Poel implement SVM method to predict customer churns [3]. Their study shows that supporting vector machines results in good generalization performance when applied to noisy marketing data. Burez and Van den Poel study class imbalances in customer churn prediction [4]. Results of the study show that under-sampling can lead to improved prediction accuracy.

In another study, Tsai and Chen use association rules to select important features and then apply neural networks and Decision Tree to predict customer churns in a telecommunication company [5]. Similar to us, they use four performance measurements to analyze their results, accuracy, precision, recall, and F-measure.

There are also other studies which implement well-known techniques of data mining to predict customer churns. Huang et al. proposes some new features to customer churn prediction and implement seven prediction techniques including Logistic Regression, Linear Classification, Naive Bayes, Decision Tree, Multilayer Perceptron Neural Networks, Support Vector Machines and the evolutionary data mining algorithms [6].

CHAPTER 03: THEORETICAL BASIC

3.1. Data mining techniques

Data mining refers to digging into or mining the data in different ways to identify patterns and get more insights into them. It involves analyzing the discovered patterns to see how they can be used effectively.

In data mining, you sort large data sets, find the required patterns and establish relationships to perform data analysis. It’s one of the pivotal steps in data analytics, and without it, you can’t complete a data analysis process.

In data mining, the predictive analysis task is undertaken through classifcation and regression techniques. Regression is a statistical method that is used to estimate relationships between dependent variables to one or more independent variables. It can also be used to assess the strength of the relationship between variables as well as to model future relationships between them, whereas classifcation is a predictive modeling problem where a class label is predicted for input data. They instigated classifcation as a procedure to find a model that demonstrates and identifies data concepts or classes. Afterward, the model has been used for predicting class labels of objects with unidentified labels.

3.2. Classification techniques in Data mining

Classification in data mining is a common technique that separates data points into different classes. It allows you to organize data sets of all sorts, including complex and large datasets as well as small and simple ones.

It primarily involves using algorithms that you can easily modify to improve the data quality. This is a big reason why supervised learning is particularly common with classification in techniques in data mining. The primary goal of classification is to connect a variable of interest with the required variables. The variable of interest should be of qualitative type.

The algorithm establishes the link between the variables for prediction. The algorithm you use for classification in data mining is called the classifier, and observations you make through the same are called the instances. You use classification techniques in data mining when you have to work with qualitative variables.

There are multiple types of classification algorithms, each with its unique functionality and application. In this, we used the datamining algorithms such as, Logistic Regression Classifier, Random Forest Classifier, SVM (Support Vector Machines)

3.3. Logistic Regression Classifier

Logistic Regression is a traditional classification algorithm involving linear discriminants, as originally proposed in 1958 by Cox. The primary output is a probability that the given input point belongs to a certain class. Based on the value of the probability, the model creates a linear boundary separating the input space into two regions.Logistic regression is easy to implement and work well on linearly separable classes, which makes it one of the most widely used classifiers.

3.4. Random Forest Classifier

Random forests take an ensemble approach that provides an improvement over the basic decision tree structure by combining a group of weak learners to form a stronger learner (see the paper by Breiman). Ensemble methods utilize a divideandconquer approach to improve algorithm performance. In random forests, a number of decision trees, i.e., weak learners, are built on bootstrapped training sets, and a random sample of m predictors are chosen as split candidates from the full set P predictors for each decision tree. As m P, the majority of the predictors are not considered. In this case, all of the individual trees are unlikely to be dominated by a few influential predictors. By taking the average of these uncorrelated trees, a reduction in variance can be attained, making the final result less variable and more reliable.

3.5. SVM (Support Vector Machines)

Support vector machine was initially proposed in 1995 by Vapnik and Cortes . SVM is commonly used as a discriminative classifier to assign new data samples to one Employee Turnover Prediction with Machine Learning 741
of two possible categories. The basic idea of SVM is to define a hyperplane which separates the n-dimensional data into two classes, wherein the hyperplane maximizes the geometric distance to the nearest data points, so-called support vectors. It is noteworthy that practical linear SVM often yields similar results as logistic regression

In addition to performing linear classification, SVM also introduces the idea of a kernel method to efficiently perform non-linear classification. It is a feature mapping methodology which transfers the attributes into a new feature space (usually higher in dimension) where the data is separable. For further details, refer to the paper by Muller and co-researchers.

RESEARCH	DATA MINING TECHNIQUE	ALGORITHMS
(1)	Classification	Logistic Regression Classifier
(2)	Classification	Random Forest Classifier
(3)	Classification	SVM (Support Vector Machines)

Table 3- 1 A summary of the datamining algorithms used in the previously related work

CHAPTER 04: PROPOSED METHODOLOGY

Employee attrition is a trivial issue for organization's loss. It leads to some crucial issues such as financial loss, cost and time to get the replacement and hiring, retraining of new employees and also customer dissatisfaction. Somehow organizations can bear the loss of attrition of employees that are not as experienced as those who have spent a significant amount of time that their attrition always results in some serious losses. Therefore, the key is to retain its experienced and trained workforce. Employee attrition can have a negative impression on existing employees. Employee churn can be classified into following categories:

- Best and experienced employees leaving prematurely.
- Fresher candidates churn.
- Department-wise churn.

4.1. Overview

First here we have a problem statement; we will be creating a dataset to our project by collecting the data. The data is stored as a csv (comma separated values) file. After creating the data we will be sending the data to the data preprocessing.

Here we will be carrying out our project in two phases. In the first phase we will be applying the basic methods such as Data Visualization, Cluster Analysis, and Correlation Analysis. By applying these methods we can draw the conclusions like what are the factors that are responsible for an employee leaving the company.

In the second phase, after predicting the factors that are responsible for an employee leaving a company we are going to check how accurate they are . First, we should split some of the data into the training phase and testing phase. Here we are sending 70% of the data to the training phase and remaining 30% to the testing phase. Here splitting the data into training phase and testing phase we are going to find the accuracy. We will find accuracy using three methods namely Logistic Regression Classifier, Random Forest Classifier, SVM (Support Vector Machine) Classifier. Now we will compare the accuracy obtained from the three methods and by comparing we get the best accuracy for the Random Forest Classifier and we will declare it as the best model.

Finally, we will construct the confusion matrix by using three methods namely Logistic Regression Classifier, Random Forest Classifier and SVM (Support Vector Machine) Classifier and we will also be calculating the precision and recall values.

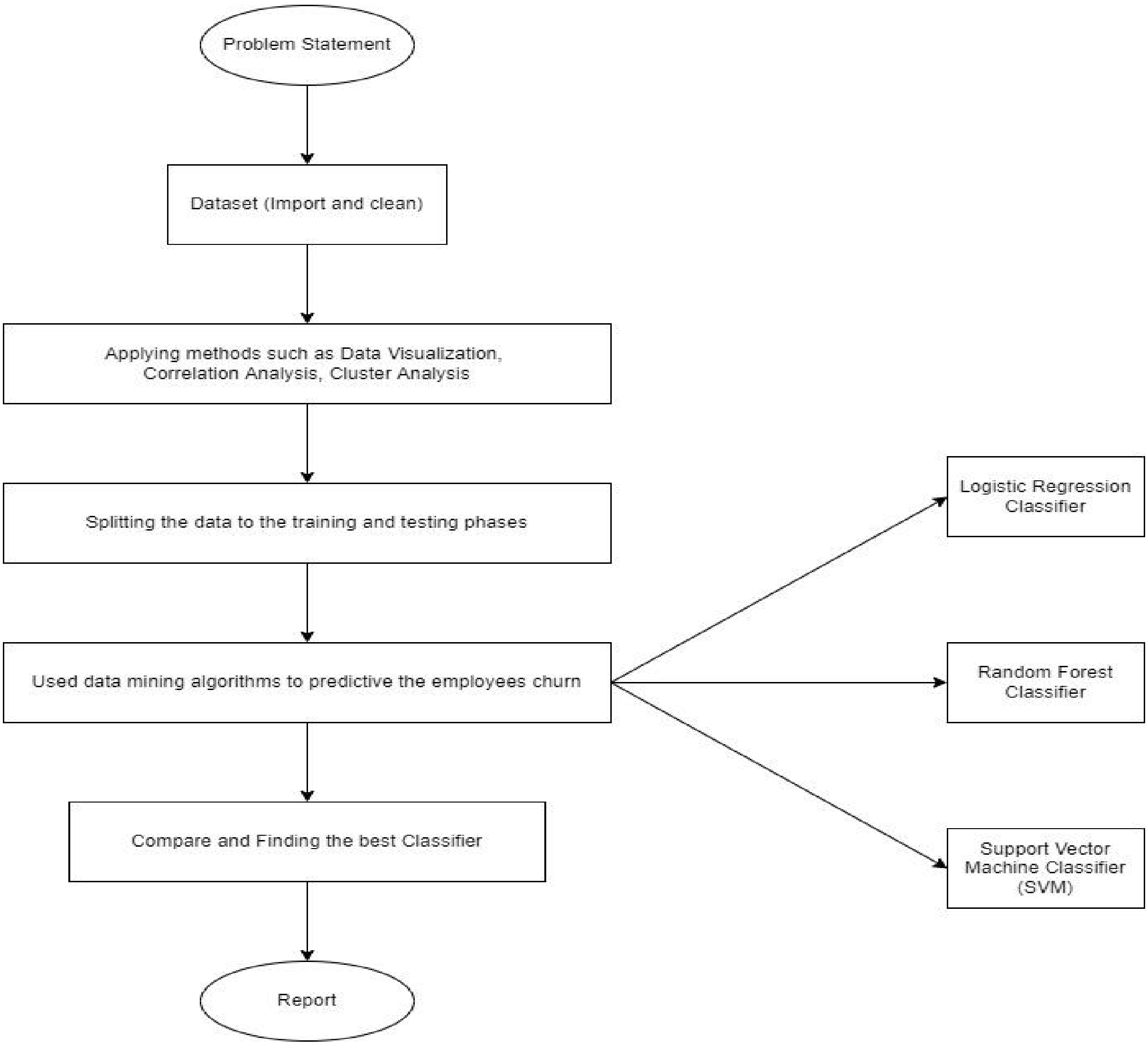


Figure 4- 1 The proposed Methodology of our project is represented in Flow Chaw Chart

4.2. Import Dataset

The data include 10 features for each record of the employee:

Num	Attributes / Features	Data Type	Description
1	Satisfaction Level	Numeric	"Satisfaction level of employee:", 0.0, 1.0, 0.07
2	Last Evaluation	Numeric	"Last evaluation result:", 0.0, 1.0, 0.07
3	Number of Projects	Int	"Number of project involved:", 0, 7, 7
4	Average Monthly Hours	Int	"Average monthly hour:", 96, 310, 280, step=1
5	Time spent in Company	Int	"How long work employee?", ("2", "3", "4", "5", "6", "7+")
6	Work accident	Int	"Has the employee ever had a work accident?", ("Yes", "No")
7	Left	Int	“Number of employees leaving the company:”, 1,0
8	Promotion last 5 years	Int	"Has the employee been promoted in the last 5 years?", ("Yes", "No")

9	Departments	chr	"Which department work employee for ?", ("sales", "accounting", "hr", "technical", "support", "management", "IT", "product_mng", "marketing", "RandD")
10	Salary	chr	"Salary level of employee", ("low", "medium", "high")

Table 4- 1 HUMAN RESOURCE DATASETS ATTRIBUTES

4.3. Data Preprocessing

Datasets in any data mining application can have missing data values. These missing values can get propagated due to lack of communication among the parameters in a data collection system. These missing values can affect the performance of a data mining system, and it should be noticed. We've checked it by function so we can see that it didn't have any missing data values here.

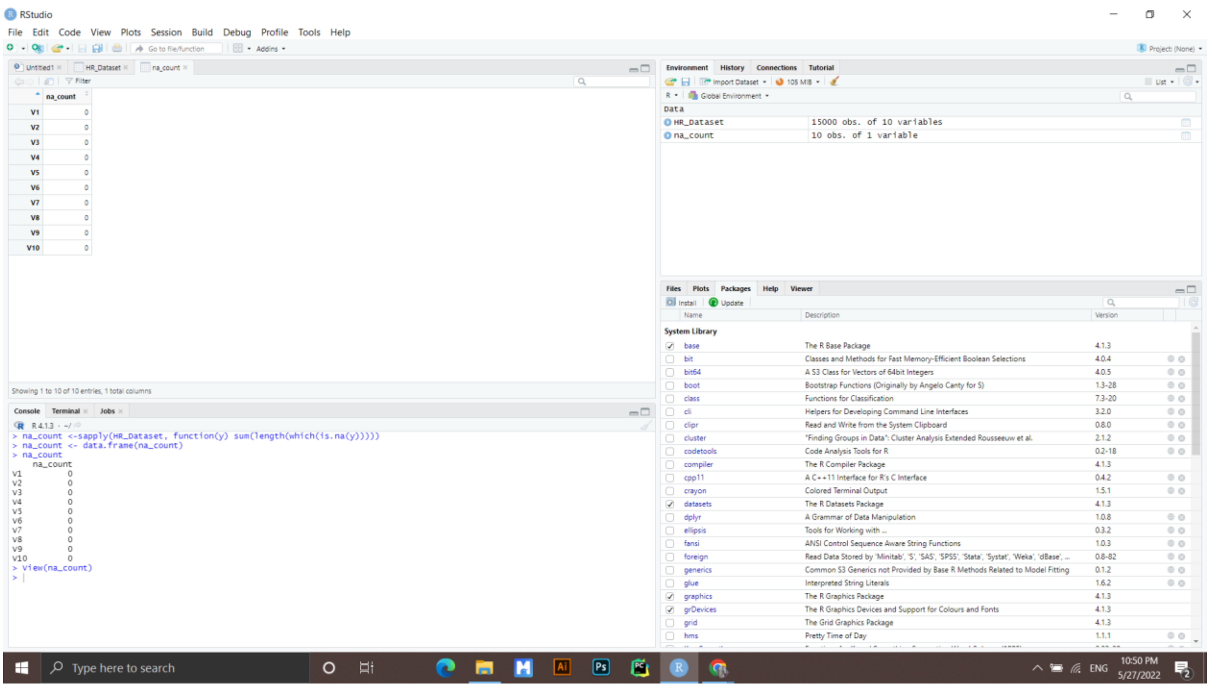


Figure 4- 2 The dataset has been checked for missing values

4.4. PHASE 1

The methods that we applied in phase 1 are Data Visualization, Cluster Analysis, and Correlation Analysis

4.4.1. Data Visualization:

Data visualization is the technique used to deliver insights in data using visual cues such as graphs, charts, maps, and many others. This is useful as it helps in intuitive and easy understanding of the large quantities of data and thereby make better decisions regarding it.Data Visualization is another form of visual art that grabs our interest and keeps eyes on the message. R is a language that is designed for statistical computing, graphical data analysis, and scientific research. It is usually preferred for data visualization as it offers flexibility and minimum required coding through its packages. Our project will use package **ggplot2** to Data visualization the data set to have a clearer view and make useful conclusions

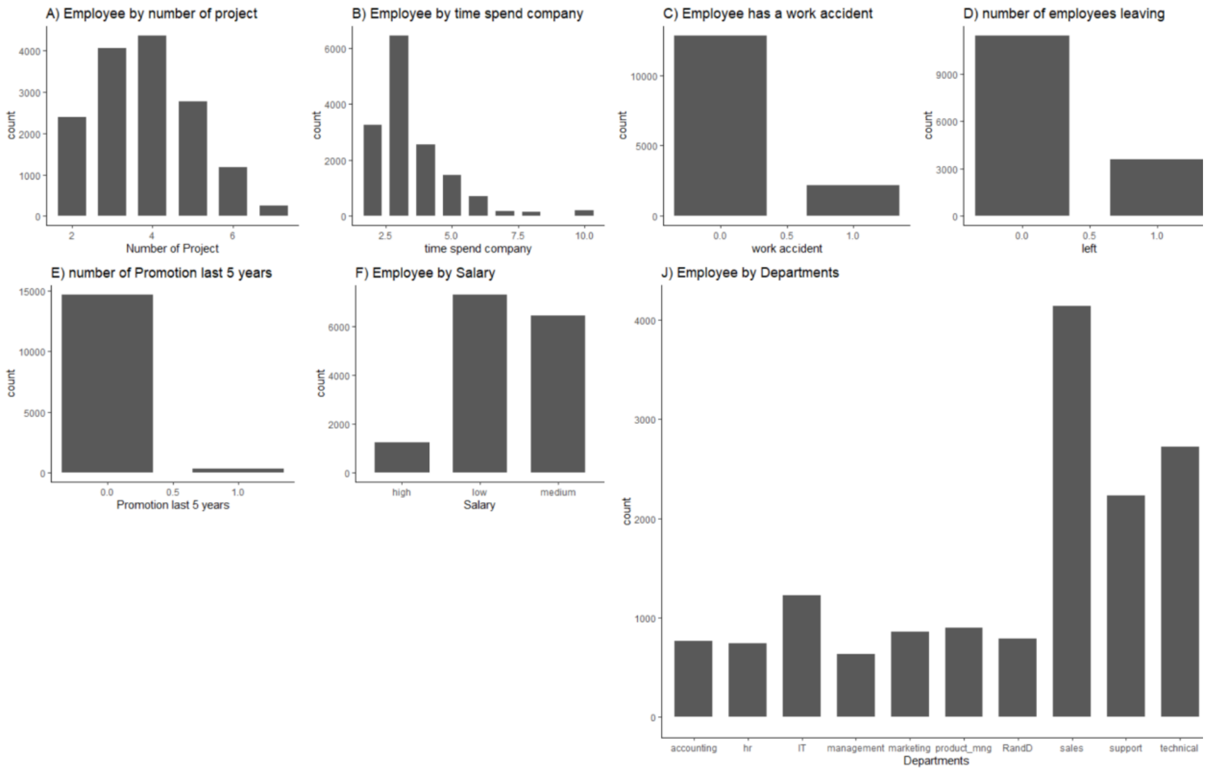


Figure 4- 3 Visualization of database created

1) Here in our project we apply data visualization techniques on attributes like satisfaction level, last evaluation, number of projects, monthly average hours, amount of time spend in company, employees left the company, promotions in last 5years, departments, salary..

The some of the conclusions are: As we can in the screenshots

- The number of projects is generally 3 to 4
- The number of promotions in last 5 years is very less.
- Most of the employees are in the sales category of the department
- Most of the salary is in the range between from low to medium.
- Fewer and fewer veteran employees (most of the time with the company is 2-3 years)

2) Here we are performing Data Visualization on the people who left the company and who do not leave the company. By drawing and analyzing the charts the some of the conclusion are:

- The employees who have number of projects from 6 to 7 are left more.
- The person who spends 5 years in a company is having more chances of leaving.
- People who did not get promotions left the company more.
- The people who are having low salary are left more

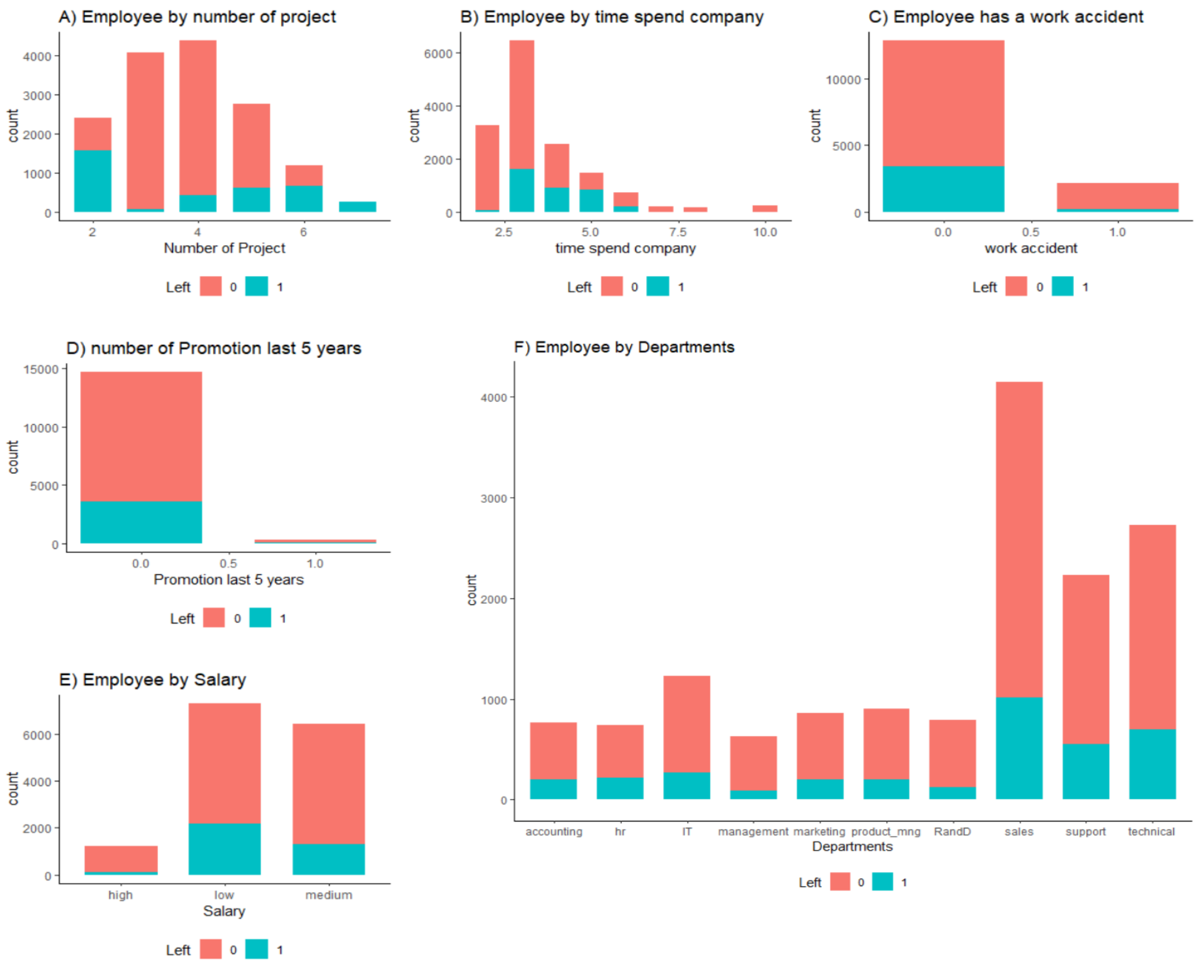


Figure 4- 4 Statistical description of different attributes who left and stay in different Departments

4.4.2. Correlation Analysis:

Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables (e.g. height and weight). This particular type of analysis is useful when a researcher wants to establish if there are possible connections between variables

A Correlation is number between -1 and +1 that measures the degree of association between two attributes(call them as X and Y).A positive value for the correlation implies positive association .In this case large values of X tend to be associated with large values of Y and small values of X tend to be associated with small values of Y. A negative value for the correlation implies that a negative or inverse association .In this case large values of X tend to be associated with small values of Y and small values of X tend to be associated with large values of Y.

In our project the use of correlation analysis is: Looking at the correlation matrix we can see that the people who left the company the highest negative correlation is with satisfaction_level. Which implies that **satisfaction_level** increases as the number of people who **left** the company decreases.

- **Satisfaction_level and Left**

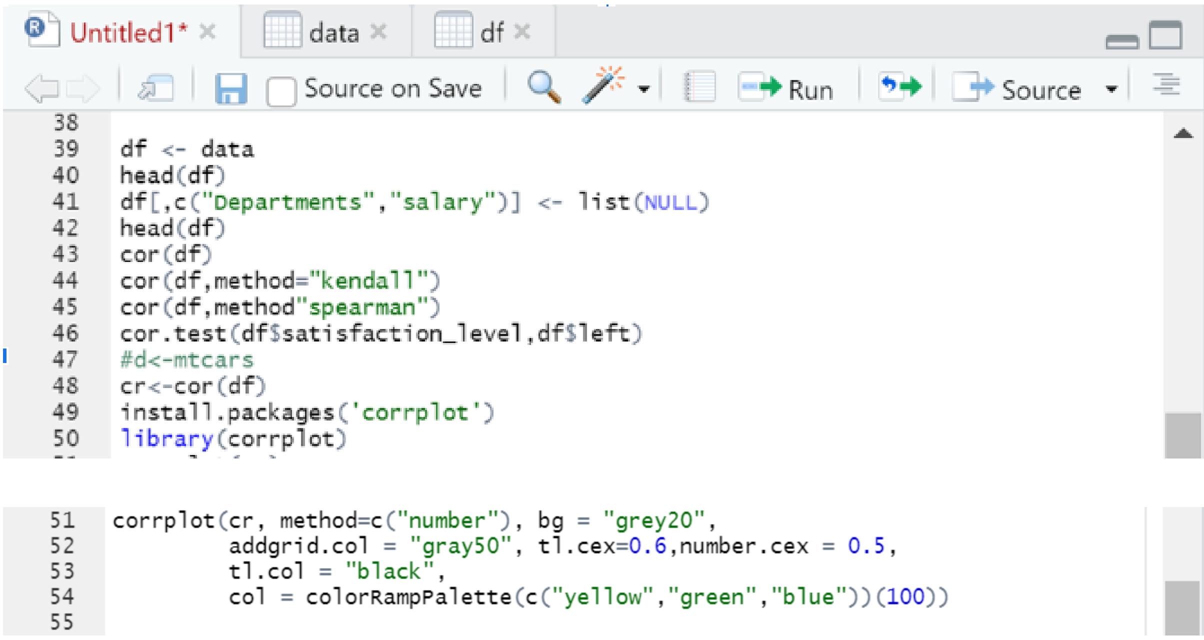
```
Console   Terminal ×   Jobs ×
R 4.1.2 · D:/2ndYEAR/HKII/R/employee_churn-master/
> setwd("D:/2ndYEAR/HKII/R/employee_churn-master")
> library("stats")
> library("psych")
> data <- read.csv("HR_Dataset.csv")
> View(data)
> #a, Test for normalty
```

```
Console   Terminal ×   Jobs ×
R 4.1.2 · D:/2ndYEAR/HKII/R/employee_churn-master/

> #a3
> describe(data$satisfaction_level)
   vars     n mean   sd median trimmed  mad  min max range  skew kurtosis se
X1    1 14999 0.61 0.25   0.64    0.63 0.28 0.09   1  0.91 -0.48    -0.67  0
> describe(data$left)
   vars     n mean   sd median trimmed mad min max range skew kurtosis se
X1    1 14999 0.24 0.43      0    0.17   0   0   1     1 1.23    -0.49  0
> cor.test(data$satisfaction_level,data$left)

	Pearson's product-moment correlation

data:  data$satisfaction_level and data$left
t = -51.613, df = 14997, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4018809 -0.3747001
sample estimates:
      cor
-0.388375

>
```

Figure 4- 5 Cor satisfaction_level and left

Cor: -0.388375 < 0 Negative correlation coefficient. That is, the value of variable Satisfaction_level increases, the value of variable Left decreases, and vice versa, the value of variable Left increases, the value of variable Satisfaction_level decreases.

Cor < -0.29, this is a weak correlation



Correlation plot:

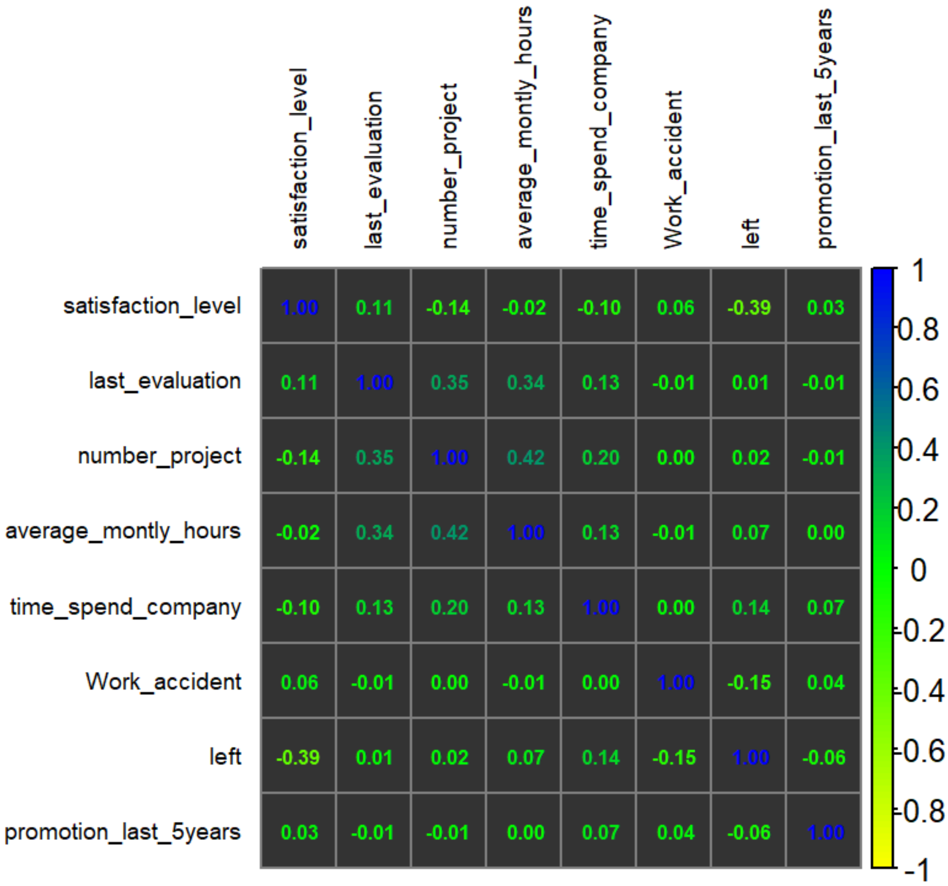


Figure 4- 6 Correlation plot

4.4.3. Cluster Analysis:

Clustering is an example for unsupervised learning

```
> df_clust <- kmeans(data,
+                    centers = 3,
+                    nstart = 35)
> summary(df_clust)
             Length Class  Mode
cluster      14999  -none- numeric
centers          6  -none- numeric
totss            1  -none- numeric
withinss         3  -none- numeric
tot.withinss     1  -none- numeric
betweenss        1  -none- numeric
size             3  -none- numeric
iter             1  -none- numeric
ifault           1  -none- numeric
> 
```

```
> fviz_cluster(df_clust,
+              data,
+              palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
+              geom = "point",
+              ellipse.type = "convex",
+              ggtheme = theme_bw())
> 
```

We try to conduct cluster analysis through K-Means Alg with center = 3, that is, choose k=3 clusters.

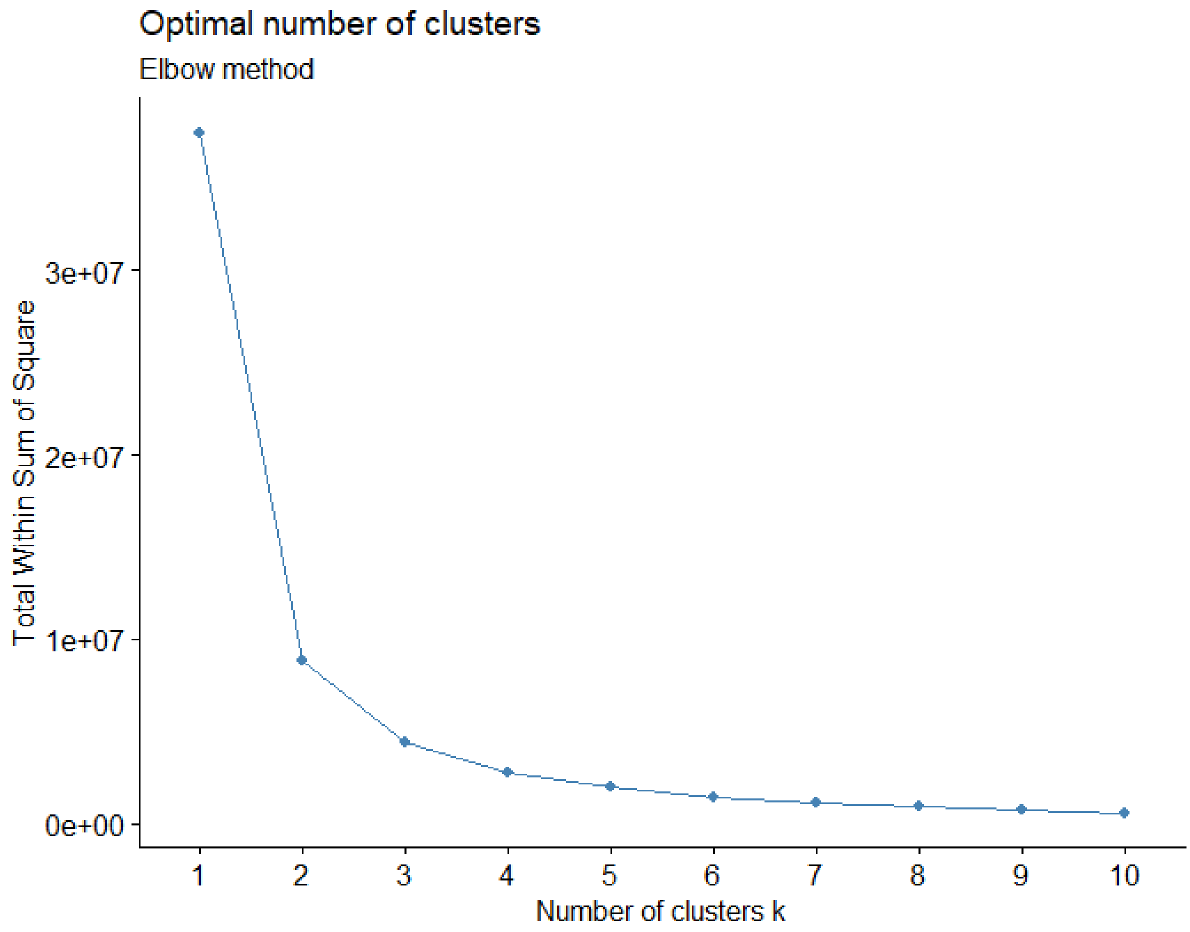


Figure 4- 7 Optimal number of clusters

```
> fviz_nbclust(df,
+              kmeans, # Sử dụng thuật toán K-Means
+              method='wss') +
+     labs(subtitle = "Elbow method")
>
> df %>% ggplot(aes(satisfaction_level, last_evaluation)) +
    geom_point(color="#333333") +
    ggtitle("House price of unit area via distance to the nearest MRT station ") +
    theme_minimal(ss
```

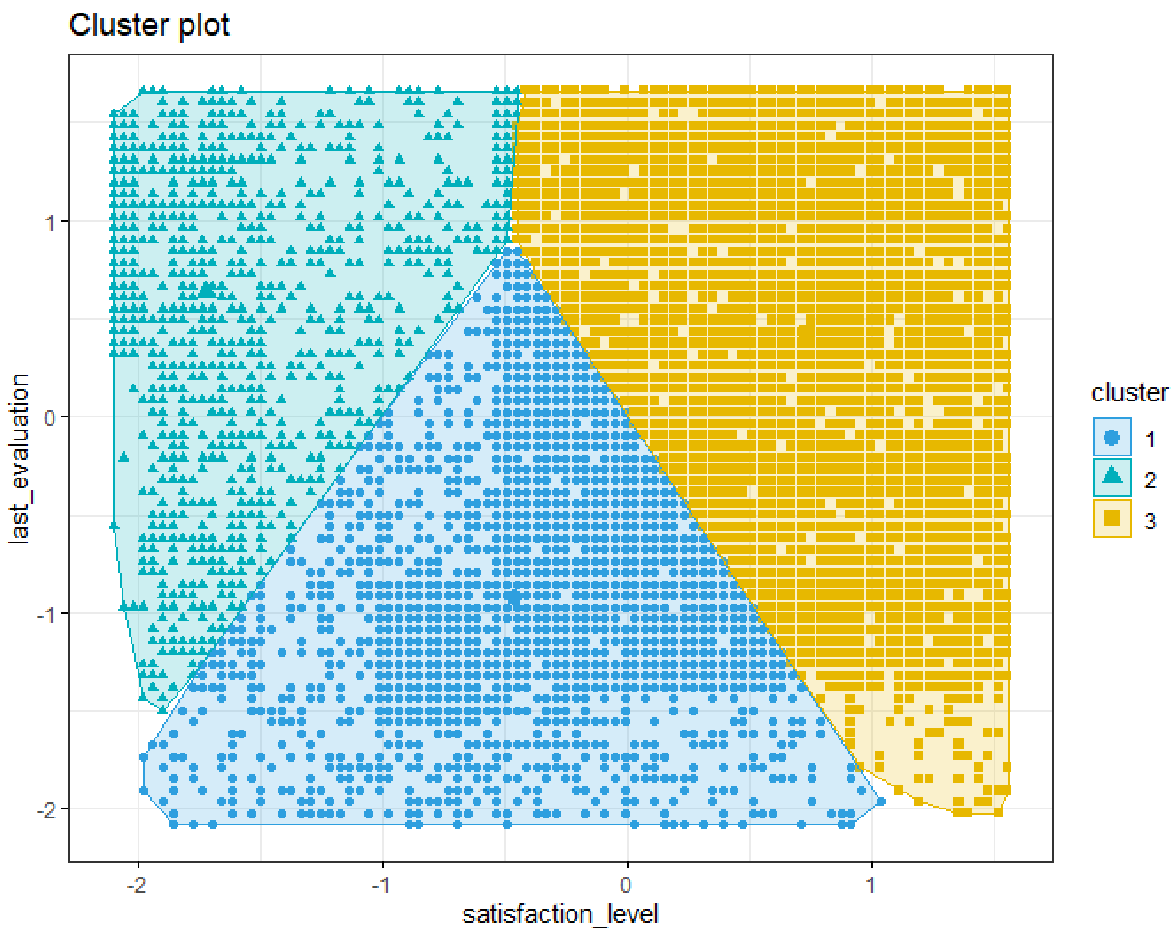


Figure 4- 8 Cluster plot

```
> tidy(df_clust)
# A tibble: 3 x 5
  satisfaction_level last_evaluation  size withinss cluster
               <dbl>           <dbl> <int>    <dbl> <fct>
1              0.498           0.559  5007    124.  1
2              0.184           0.825  2041     52.9 2
3              0.795           0.787  7951    295.  3
>
```

Extract content of parameters in df_clust into dataframe

```
> augment(df_clust, df) %>%
+     ggplot(aes(satisfaction_level,
+                last_evaluation,
+                color =.cluster)) +
+     geom_point(alpha = 2) +
+     ggtitle("Clustering: Employees who left") +
+     theme_minimal()
>
```

Figure 4- 9 Clustering : Employees who left

Here by using cluster analysis method for our employee churn prediction project we draw some conclusions like: There are three categories of employees:

Cluster 1: Employees with low satisfaction and low performance

Cluster 2: Employees with low satisfaction and high performance

Cluster 3: Employees with high satisfaction and high performance.

Here in our project we use K-means Clustering and here k value = 3 Here we apply cluster analysis on the satisfaction level of employees who left the company. Hence, these are methods we applied in the phase 1. By applying these methods such as Data Visualization, Correlation Analysis and Cluster Analysis, we are going to predict the factors that are responsible for an employee leaving the company.

Validation:

Estimate the Sillhoutte index S_i from -1 to 1

```
> sil = silhouette(df_clust$cluster, dist(df))
>
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1 5007          0.08
2       2 2041         -0.03
3       3 7951         -0.08
>
```

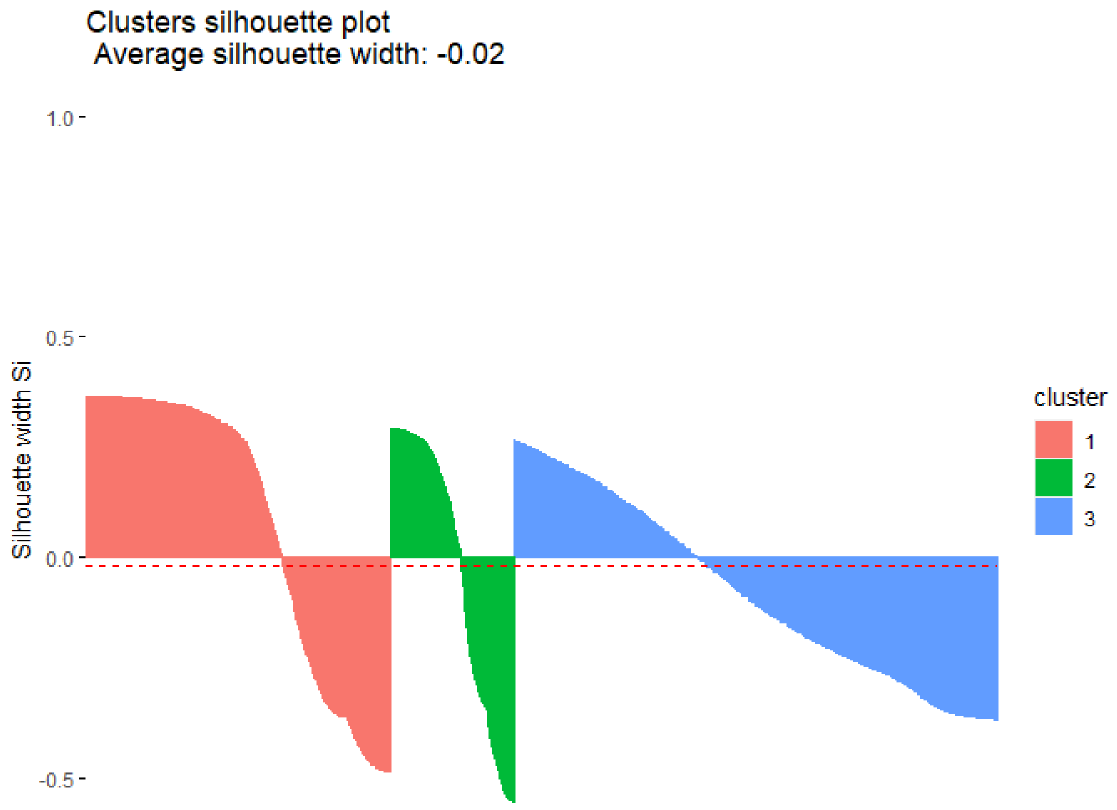


Figure 4- 10 Clusters silhouette plot

Calculating variables

```
> summary(df)
 satisfaction_level last_evaluation  number_project  average_montly_hours time_spend_company Work_accident         left
 Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0        Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0        1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.6400     Median :0.7200   Median :4.000   Median :200.0        Median : 3.000     Median :0.0000   Median :0.0000
 Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1        Mean   : 3.498     Mean   :0.1446   Mean   :0.2381
 3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0        3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0        Max.   :10.000     Max.   :1.0000   Max.   :1.0000
 promotion_last_5years
 Min.   :0.00000
 1st Qu.:0.00000
 Median :0.00000
 Mean   :0.02127
 3rd Qu.:0.00000
 Max.   :1.00000
```

Figure 4- 11 Calculating Variables

4.5. PHASE 2

Here in phase 1, we predicted the model and in phase 2 we are going to check how accurate the model is and which is the best model. In this, first we need to split the data into training test and testing set

4.5.1. Training and Testing a model:

Training and testing means that we need to send some of our code to training phase and testing phase. Here we are going to test our model. Here we split some our data to training phase and testing phase. Here we give more amount of our data to training phase and less amount of data to testing phase. After a model has been processed by using the training set, you test the model by making predictions against the test set. Because the data in the testing set already contains known values for the attribute that you want

We'll randomly split the data into training set (70% for building a predictive model) and test set (30% for evaluating the model).

```
#Use 70% of dataset as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(HR_Dataset), replace=TRUE, prob=c(0.7,0.3))
train <- HR_Dataset[sample, ]
test <- HR_Dataset[!sample, ]
```

This is about training and testing in our project. After sending, some data to training and testing phases we are going to calculate the accuracy of the model by using three Classifier namely Logistic Regression Classifier, Random Forest Classifier, SVM (Support Vector Machine) Classifier.

4.5.2. Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes.

Train the model. Here, as we have a small number of predictors (n = 10), we can select manually the most significant:

```
#Train the model using the training data
mymodel <- glm(left ~ satisfaction_level + last_evaluation + number_project + average_montly_hours + time_spend_company,
               data = train, family = binomial)
summary(mymodel)
```

In the output above, the first thing we see is the call, this is R reminding us what the model we ran was, what options we specified, etc.

```
Call:
glm(formula = left ~ satisfaction_level + last_evaluation + number_project +
    average_montly_hours + time_spend_company, family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1617  -0.6900  -0.4711  -0.2543   2.5539

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.1259976  0.1384046   0.910    0.363
satisfaction_level  -4.0461811  0.1123526 -36.013  < 2e-16 ***
last_evaluation      0.6871515  0.1684962   4.078 4.54e-05 ***
number_project      -0.2967943  0.0242743 -12.227  < 2e-16 ***
average_montly_hours 0.0044924  0.0005857   7.670 1.72e-14 ***
time_spend_company   0.1880572  0.0168987  11.129  < 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: 11608.2  on 10581  degrees of freedom
Residual deviance:  9788.1  on 10576  degrees of freedom
AIC: 9800.1

Number of Fisher Scoring iterations: 5
```

We'll make predictions using the test data in order to evaluate the performance of our logistic regression model.

The procedure is as follow:

1. Predict the class membership probabilities of observations based on predictor variables
2. Assign the observations to the class with highest probability score (i.e above 0.5)

The R function predict() can be used to predict the probability of leaving

```
summary(mymodel)
# Run the test data through the model
res <- predict(mymodel,test,type = "response")
res <- predict(mymodel,train,type = "response")
```

Accuracy - It determines the overall predicted accuracy of the model.

```
> res_accuracy <- (sum(diag(res_confmatrix)))/sum(res_confmatrix)
> cat("Logistic Regression Accuracy: ", res_accuracy)
Logistic Regression Accuracy:  0.7647059
```

Based on the data of the confusion matrix, The accuracy that we got using the Logistic Regression is **0.765.**

4.5.3. Random Forest

Using random forest algorithms to build models.

```
#Generate Random forest learning tree
train$left <- as.factor(train$left)
rfmodel <- randomForest(left~., data=train, proximity=T)
```

Output:

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

        OOB estimate of  error rate: 1%
Confusion matrix:
     0    1 class.error
0 7937   18 0.002262728
1   87 2413 0.034800000
```

The output notes that the random forest included 500 trees and tried 3 variables at each split. According to the confusion matrix—the error rate of 1%. However, this confusion matrix does not show a resubstitution error. Instead, it reflects the out-of-bag error rate (listed in the output as OOB estimate of error rate), which unlike resubstitution error, is an unbiased estimate of the test set error.

You can use the plot() function to plot the mean square error of the forest object:

```
plot(rfmodel)
```

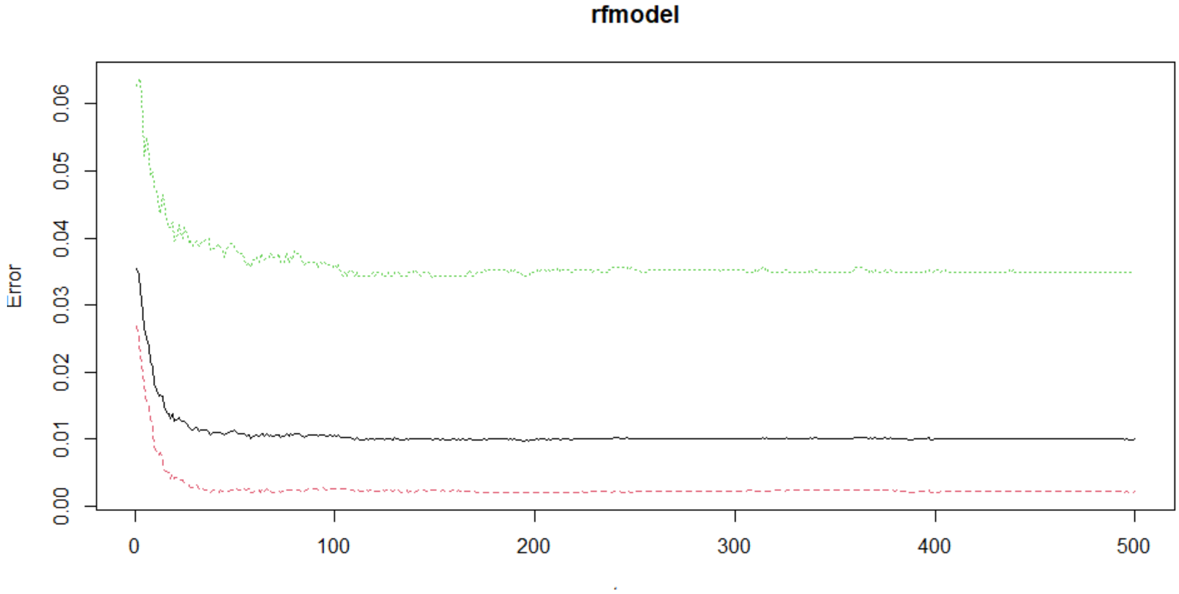


Figure 4- 12 Random forest Model

The plot seems to indicate that after 100 decision trees, there is not a significant reduction in error rate.

You can then examine the importance of each attribute within the fitted classifier:

```
importance(rfmodel)
```

```
                      MeanDecreaseGini
satisfaction_level          1264.29577
last_evaluation              469.24020
number_project               673.92177
average_montly_hours         571.97847
time_spend_company           695.71307
Work_accident                 21.35775
promotion_last_5years          3.79377
Departments                   48.35595
salary                        32.49869
```

Try to build random forest for testing data. Similar to other classification methods, you can obtain the classification table:

```
#Validate the model - Confusion Matrix
rfPred = predict(rfmodel, newdata=test)
```

Let's determine the misclassification rate. First, build a confusion matrix. Each column of the matrix represents the number of predictions of each class, while each row represents the instances in the actual class.

```
rf_confmatrix <- table(rfPred, test$left)
```

Second, build a *diagonal mark quality prediction*. Applying the diag function to this table then selects the diagonal elements, i.e., the number of points where *random forest* agrees with the true classification, and the sum command simply adds these values up.

```
> rf_accuracy <- (sum(diag(rf_confmatrix)))/sum(rf_confmatrix)
> cat("Random Forest Accuracy: ", rf_accuracy)
Random Forest Accuracy:  0.9940581
```

The model has a high overall accuracy that is **0.994**

4.5.4. SVM

SVM (Support Vector Machine) is a supervised machine learning algorithm which is mainly used to classify data into different classes. Unlike most algorithms, SVM makes use of a hyperplane which acts like a decision boundary between the various classes.

SVM can be used to generate multiple separating hyperplanes such that the data is divided into segments and each segment contains only one kind of data.

Before moving further, let's discuss the features of SVM:

1. SVM is a supervised learning algorithm. This means that SVM trains on a set of labeled data. SVM studies the labeled training data and then classifies any new input data depending on what it learned in the training phase.
2. A main advantage of SVM is that it can be used for both classification and regression problems. Though SVM is mainly known for classification, the SVR (Support Vector Regressor) is used for regression problems.
3. SVM can be used for classifying non-linear data by using the kernel trick. The kernel trick means transforming data into another dimension that has a clear dividing margin between classes of data. After which you can easily draw a hyperplane between the various classes of data.

In this demo, we’ll be using the Caret package and e1071 package:

The caret package is also known as the Classification And REgression Training, has tons of functions that helps to build predictive models. It contains tools for data splitting, pre-processing, feature selection, tuning, unsupervised learning algorithms, etc.

e1071 packeage: This package was the first implementation of SVM in R. With the svm() function, we achieve a rigid interface in the libsvm by using visualization and parameter tuning methods.

Fitting SVM to the training set

```
classifier = svm(formula = left ~ .,
                 data = train,
                 type = 'C-classification',
                 kernel = 'linear')
classifier
```

The output is:

- Classifier detailed

```
classifier                       List of  30
  $ call          : language svm(formula = left ~ ., data = train, type = "C-classification", kernel = "linear")
  $ type          : num 0
  $ kernel        : num 0
  $ cost          : num 1
  $ degree        : num 3
  $ gamma         : num 0.0526
  $ coef0         : num 0
  $ nu            : num 0.5
  $ epsilon       : num 0.1
  $ sparse        : logi FALSE
  $ scaled        : logi [1:19] TRUE TRUE TRUE TRUE TRUE TRUE ...
  $ x.scale       :List of 2
   ..$ scaled:center: Named num [1:7] 0.612 0.718 3.798 201.245 3.498 ...
   .. ..- attr(*, "names")= chr [1:7] "satisfaction_level" "last_evaluation" "number_project" "average_montly_hours" ...
   ..$ scaled:scale : Named num [1:7] 0.249 0.171 1.237 50.056 1.469 ...
   .. ..- attr(*, "names")= chr [1:7] "satisfaction_level" "last_evaluation" "number_project" "average_montly_hours" ...
  $ y.scale       : NULL
  $ nclasses      : int 2
  $ levels        : chr [1:2] "0" "1"
  $ tot.nSV       : int 4844
  $ nSV           : int [1:2] 2418 2426
  $ labels        : int [1:2] 2 1
  $ SV            : num [1:4844, 1:19] 0.754 -0.973 -0.812 -0.772 -0.651 ...
   ..- attr(*, "dimnames")=List of 2
   .. ..$ : chr [1:4844] "2" "5" "6" "10" ...
   .. ..$ : chr [1:19] "satisfaction_level" "last_evaluation" "number_project" "average_montly_hours" ...
  $ index         : int [1:4844] 1 2 3 4 5 6 7 8 9 10 ...
  $ rho           : num 1.98
  $ compprob      : logi FALSE
  $ probA         : NULL
  $ probB         : NULL
  $ sigma         : NULL
  $ coefs         : num [1:4844, 1] 1 1 1 1 1 1 1 1 1 1 ...
  $ na.action     : NULL
  $ fitted        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
   ..- attr(*, "names")= chr [1:10427] "2" "5" "6" "10" ...
  $ decision.values: num [1:10427, 1] -1.211 -0.183 -0.324 -0.376 -0.482 ...
   ..- attr(*, "dimnames")=List of 2
   .. ..$ : chr [1:10427] "2" "5" "6" "10" ...
   .. ..$ : chr "1/0"
  $ terms         :Classes 'terms', 'formula'  language left ~ satisfaction_level + last_evaluation + number_project + …
   .. ..- attr(*, "variables")= language list(left, satisfaction_level, last_evaluation, number_project, average_montly_…
   .. ..- attr(*, "factors")= int [1:10, 1:9] 0 1 0 0 0 0 0 0 0 0 ...
   .. .. ..- attr(*, "dimnames")=List of 2
   .. .. .. ..$ : chr [1:10] "left" "satisfaction_level" "last_evaluation" "number_project" ...
```

Figure 4- 13 Classifier detailed

- Classifier in nutshell

```
Call:
svm(formula = left ~ ., data = train, type = "C-classification", kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  4844
```

Figure 4- 14 Classifier in nutshell

Predicting the test set result

```
#predictive
y_pred = predict(classifier, newdata = test)
y_pred
```

The output is:

```
> y_pred = predict(classifier, newdata = test)
> y_pred
   1    3    4    7    8    9   12   20   26   30   32   35   38   39   40   47   50   53   55   56 
   0    1    0    1    0    0    1    0    0    0    0    0    0    1    0    0    0    0    1    0 
  57   59   61   65   66   75   77   78   79   80   85   87   91   93  100  101  103  105  106  108 
   1    0    0    1    0    1    1    0    0    0    1    1    0    0    0    0    0    0    0    0 
 109  112  113  115  122  127  129  131  132  141  147  150  152  155  156  160  161  163  166  167 
   0    1    0    0    0    0    1    1    0    1    0    0    0    0    1    0    1    0    1    0 
 168  170  175  180  182  189  190  199  210  211  212  214  226  233  239  240  242  243  249  251 
   0    0    0    1    1    0    0    1    0    0    1    0    0    1    0    1    0    0    0    0 
 252  265  266  270  277  280  283  285  291  298  301  304  310  311  313  317  318  324  325  326 
   0    0    0    0    1    0    0    0    0    0    0    0    1    1    0    0    0    1    0    0 
 329  331  339  340  342  348  352  359  366  373  374  375  377  379  380  381  383  390  391  402 
   1    0    0    1    0    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0 
 404  406  407  409  412  414  416  417  426  427  431  433  436  448  450  451  452  454  457  464 
   0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0 
 468  470  471  477  478  489  491  493  495  507  508  510  516  518  519  521  523  524  527  528 
   1    0    1    1    1    1    0    1    0    0    0    0    0    1    0    0    1    0    0    1 
 529  533  536  539  543  546  547  551  561  563  566  569  573  575  576  577  581  587  594  597 
   1    1    0    0    0    0    0    1    1    0    0    0    0    0    0    0    0    0    0    0 
 603  604  609  611  620  621  622  625  637  644  648  656  665  667  673  676  681  682  688  691 
   1    0    0    1    0    0    1    0    0    0    0    0    1    0    0    0    1    0    1    0 
 694  696  698  700  701  704  705  707  713  714  715  717  718  720  722  732  733  738  742  743 
   1    0    1    0    0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0 
 757  759  762  763  769  774  776  777  778  780  792  794  797  799  801  804  805  806  808  810 
   0    0    1    0    1    0    0    1    1    1    1    0    0    1    0    1    0    0    0    1 
 811  817  820  821  824  825  830  831  834  837  841  847  849  852  856  867  868  882  889  896 
   0    0    0    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    1    1 
 898  899  900  901  906  909  910  917  925  926  927  934  936  944  948  955  957  958  959  960 
   0    1    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    0    0    1 
 962  968  972  973  975  979  980  982  983  991  992  996 1001 1002 1006 1012 1014 1016 1018 1020 
   0    0    0    1    1    0    1    0    0    1    0    0    0    1    0    0    1    0    0    0 
1024 1026 1029 1036 1038 1047 1048 1053 1055 1061 1063 1065 1067 1069 1070 1076 1077 1079 1081 1082 
   0    0    0    1    1    0    0    0    1    0    0    0    0    0    1    0    1    0    0    0 
1086 1092 1095 1097 1112 1121 1122 1124 1127 1135 1145 1147 1149 1153 1154 1158 1166 1176 1180 1181 
   0    0    1    0    0    1    0    1    0    0    0    0    0    0    1    0    1    0    0    0 
1182 1188 1192 1193 1196 1198 1199 1200 1201 1213 1215 1217 1223 1229 1230 1231 1232 1234 1237 1238 
   0    1    1    0    0    0    1    0    0    0    1    0    0    1    0    0    1    0    0    0 
1245 1252 1254 1255 1258 1261 1270 1277 1280 1285 1290 1292 1299 1310 1323 1328 1330 1331 1332 1334 
   1    0    0    0    0    1    0    0    0    0    0    1    0    0    0    0    1    0    1    1 
1335 1336 1342 1347 1349 1352 1359 1362 1368 1370 1373 1375 1379 1381 1382 1386 1391 1394 1401 1404 
   0    0    0    0    0    1    1    0    0    1    0    0    0    0    0    0    0    0    0    0 
1408 1411 1413 1418 1419 1421 1425 1426 1432 1433 1436 1444 1454 1455 1457 1463 1469 1470 1473 1474 
   0    0    0    1    1    0    0    1    0    0    0    1    0    0    0    0    0    1    0    1 
1482 1487 1488 1489 1491 1493 1494 1497 1500 1504 1505 1510 1511 1512 1515 1517 1522 1525 1528 1530 
   1    1    0    1    0    0    0    1    0    0    0    0    0    0    1    0    0    0    0    0 
1532 1535 1536 1537 1539 1547 1550 1558 1564 1565 1570 1572 1576 1583 1584 1586 1594 1595 1608 1609 
   0    1    0    1    0    0    0    0    0    1    0    0    0    1    0    0    0    0    0    1 
1611 1614 1620 1621 1633 1640 1641 1643 1649 1653 1655 1657 1660 1667 1668 1670 1671 1673 1678 1679 
   0    0    0    0    0    0    0    1    1    0    1    0    0    0    0    0    0    0    0    0 
1682 1684 1686 1690 1693 1701 1702 1706 1707 1711 1715 1717 1719 1721 1722 1723 1727 1729 1730 1732 
   0    0    0    0    0    0    1    1    0    0    0    1    0    0    1    0    0    1    0    0 
1734 1742 1743 1747 1752 1759 1761 1762 1763 1766 1769 1788 1789 1790 1791 1792 1793 1794 1798 1799 
   0    0    1    1    1    0    0    1    0    0    0    0    0    0    0    1    0    0    0    1 
1800 1807 1809 1811 1812 1814 1817 1820 1821 1832 1833 1844 1846 1851 1853 1863 1866 1873 1875 1877 
   0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    1    1    1    0    1 
1878 1880 1886 1888 1896 1904 1906 1907 1910 1912 1913 1914 1916 1930 1931 1932 1935 1940 1942 1944 
```

```
1946 1947 1950 1954 1959 1960 1965 1968 1971 1973 1976 1978 1980 1982 1984 1988 1990 1994 1998 1999
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2002 2009 2010 2014 2015 2016 2017 2020 2022 2027 2033 2034 2038 2042 2044 2048 2051 2052 2057 2062
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0
2064 2065 2066 2067 2068 2082 2087 2088 2092 2093 2100 2103 2104 2114 2115 2118 2119 2120 2122 2123
   0    0    1    0    0    0    1    0    0    1    0    1    0    0    0    0    0    0    0    0
2133 2134 2138 2139 2140 2142 2143 2145 2147 2149 2150 2153 2155 2163 2171 2174 2175 2177 2180 2189
   0    0    0    0    0    0    0    0    1    0    0    0    1    0    0    0    0    0    0    1
2194 2195 2196 2200 2208 2213 2215 2216 2220 2221 2223 2224 2230 2234 2237 2238 2241 2246 2251 2253
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
2255 2256 2259 2268 2270 2272 2274 2275 2280 2281 2282 2286 2288 2289 2292 2295 2299 2301 2303 2308
   0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2311 2317 2321 2323 2325 2328 2332 2335 2342 2347 2348 2352 2354 2356 2357 2360 2365 2366 2371 2379
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
2381 2382 2391 2392 2393 2394 2396 2399 2400 2402 2403 2404 2405 2410 2411 2412 2413 2417 2418 2420
   0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0
2422 2427 2428 2430 2431 2435 2437 2439 2445 2446 2448 2449 2450 2455 2466 2471 2473 2475 2476 2485
   0    0    0    0    0    0    0    0    0    0    0    1    0    0    1    0    0    0    0    0
2490 2491 2501 2503 2506 2509 2512 2513 2515 2517 2519 2530 2538 2543 2548 2558 2559 2560 2563 2566
   0    0    1    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
2574 2575 2580 2585 2588 2591 2594 2613 2617 2628 2631 2632 2633 2637 2640 2642 2644 2648 2650 2651
   0    1    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
2654 2656 2657 2659 2664 2669 2670 2671 2683 2689 2696 2698 2700 2702 2703 2704 2706 2707 2708 2720
   1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2721 2724 2725 2729 2730 2734 2737 2739 2741 2742 2744 2745 2746 2750 2755 2757 2760 2761 2762 2766
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2768 2779 2785 2788 2794 2797 2800 2802 2803 2807 2812 2815 2817 2821 2824 2830 2833 2835 2836 2840
   0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2842 2843 2844 2848 2852 2853 2865 2867 2869 2870 2872 2873 2877 2878 2879 2884 2885 2886 2893 2899
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
2912 2913 2915 2916 2918 2921 2925 2926 2927 2933 2934 2941 2942 2945 2950 2952 2959 2960 2964 2965
   1    0    0    0    0    0    0    0    0    1    0    0    0    0    1    0    0    0    0    0
2968 2970 2983 2985 2986 2987 2988 2991 2992 2996 2997 3001 3006 3008 3016 3018 3025 3030 3037 3040
   0    1    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
3042 3043 3048 3054 3056 3057 3058 3066 3069 3070 3075 3079 3082 3084 3096 3097 3098 3103 3106 3107
   0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
3112 3113 3118 3119 3123 3127 3132 3142 3145 3149 3150 3151 3156 3157 3158 3162 3173 3174 3183 3185
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
3188 3192 3195 3198 3199 3204 3217 3221 3224 3225 3230 3232 3237 3239 3240 3246 3247 3251 3252 3261
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0
3265 3267 3270 3274 3276 3277 3278 3282 3286 3287 3289 3292 3295 3297 3300 3307 3308 3310 3313 3314
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
3319 3320 3322 3326 3331 3332 3333 3335 3337 3338 3345 3347 3352 3353 3357 3358 3359 3360 3362 3365
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
 [ reached getOption("max.print") -- omitted 3572 entries ]
Levels: 0 1
>
```

Figure 4- 15 Predicting the test set result

Here these are the parameters that we got using SVM Classifier and these parameters are used to know the nature and behavior of the model. The accuracy that we got using the SVM Classifier is 0.7743 .

```
               Accuracy : 0.7743
                 95% CI : (0.7619, 0.7863)
    No Information Rate : 0.7651
    P-Value [Acc > NIR] : 0.07336

                  Kappa : 0.2188

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.9394
            Specificity : 0.2365
         Pos Pred Value : 0.8003
         Neg Pred Value : 0.5451
             Prevalence : 0.7651
         Detection Rate : 0.7187
   Detection Prevalence : 0.8981
      Balanced Accuracy : 0.5879
```

4.5.5.Comparison between the three Classifiers:

Here we are using three classifiers named:

1. Logistic Regression Classifier
2. Random Forest Classifier
3. SVM Classifier.

By comparing the accuracy of the three Classifiers, in Logistic Regression we got accuracy as "**0.765**" and in Random Forest Classifier we got accuracy as "**0.994**" and finally in SVM Classifier we got accuracy as "**0.7743**". Here, by comparing all the three Classifier we can observe that Random Forest Classifier is having more accuracy. So, here we can conclude that for our project Employee Churn Prediction Random Forest Classifier is the best Classifier method.

4.6. Confusion Matrix:

Confusion matrix is used for performance Measurement of the Classification Problem and where the output can be in two or more classes. Confusion matrix is a table with 4 different combinations of predicted and actual values.

The 4 different Combinations are:

True Positives: Predictived positive and it's true

True Negatives: Predictived Negative and it is true

False Positive: Predictived positive and it's false

False Negatives: Predictived Negative and it is false

- **Logistic Regression Classifier**

```
             Predicted_Value
Actual_Value FALSE TRUE
           0  7352  603
           1  1857  643
> 
```

Figure 4- 16 Logistic Regression Classifier

Evaluating model accuracy using confusion matrix. We can see the predicted values versus the actual values. It’s important here to know if it was predicted false, and it was false, or if it was predicted true, and it was true.

In this case the “0” and “1” in the rows represent whether employees churn or not. The “FALSE” and “TRUE” in the columns represent whether we predicted employees churn or not.

- **Random Forest Classifier**

Let's determine the misclassification rate. First, build a confusion matrix. Each column of the matrix represents the number of predictions of each class, while each row represents the instances in the actual class.

```
rf_confmatrix <- table(rfPred, test$left)

rfPred    0    1
     0 3469   23
     1    4 1048
```

Figure 4- 17 Random Forest Classifier

- **SVM Classifier.**

```
y_pred    0    1
     0 3286  820
     1  212  254
```

Figure 4- 18 SVM Classifier

+ True Positives : 3286

+ False Positives: 820

+ True Negatives: 254

+ False Negatives : 212

CHAPTER 05: RESULT

Identifying the number of people left:

```
> table(data$left)

    0     1
11428  3571
```

Figure 5- 1 number of people left

Identifying the percentage of people left:

```
> table1 = as.table(table(data$left))
> prop.table(table1)

        0         1
0.7619175 0.2380825
```

Figure 5- 2 percentage of people left

Identifying the some of the factors responsible for an employee leaving the company:

satisfaction_level,last_evaluation:

```
+   group_by(left) %>%
+   summarise_at(vars(satisfaction_level), list(" " = mean))
# A tibble: 2 x 2
   left    ` `
  <int> <dbl>
1     0 0.667
2     1 0.440
```

number_project:

```
+   group_by(left) %>%
+   summarise_at(vars(number_project), list(" " = mean))
# A tibble: 2 x 2
   left   ` `
  <int> <dbl>
1     0  3.79
2     1  3.86
```

average_montly_hours:

```
+   group_by(left) %>%
+   summarise_at(vars(average_montly_hours), list(" " = mean))
# A tibble: 2 x 2
   left   ` `
  <int> <dbl>
1     0  199.
2     1  207.
```

time_spend_company:

```
+   group_by(left) %>%
+   summarise_at(vars(time_spend_company), list(" " = mean))
# A tibble: 2 x 2
   left   ` `
  <int> <dbl>
1     0  3.38
2     1  3.88
```

Work_accident:

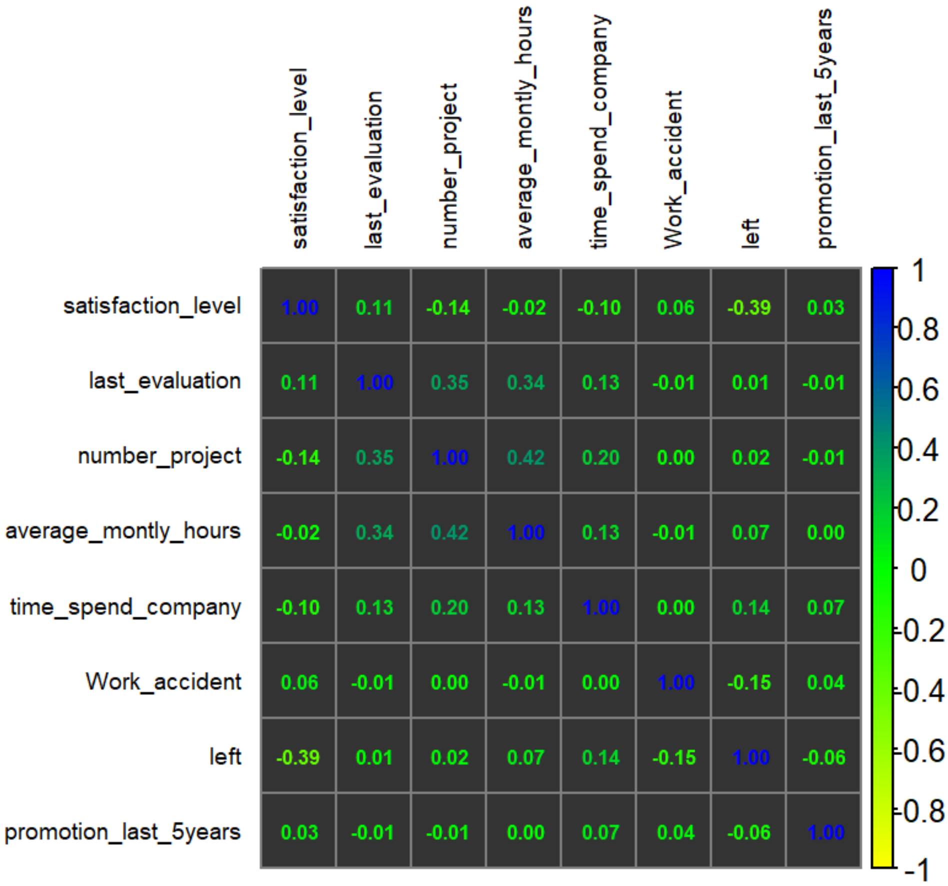
```
+   group_by(left) %>%
+   summarise_at(vars(Work_accident), list(" " = mean))
# A tibble: 2 x 2
   left    ` `
  <int>  <dbl>
1     0 0.175
2     1 0.0473
```

promotion_last_5years:

```
+   group_by(left) %>%
+   summarise_at(vars(promotion_last_5years), list(" " = mean))
# A tibble: 2 x 2
   left     ` `
  <int>   <dbl>
1     0 0.0263
2     1 0.00532
```

Figure 5- 3 Factors responsible for an employee leaving the company

Here we can observe that the employees who left the company has low satisfaction level, worked more hours and low promotion rate.



Identifying the main factor responsible for an employee to leave a company:

In, this we can observe that the satisfaction level is the main factor responsible for an employee leaving the company.

CHAPTER 06: CONCLUSION

Employee churn can affect an organization in many ways like goodwill, revenues and cost in terms of both time and money. The predictive churn model helps in not only taking preventive measure, but also making better hiring decisions. In this study implementation of various classification method helps in predicting whether a particular employee might leave the organization in the near future by deriving trends in the employee's past data. It was intuited that salary or other financial aspect like promotions are not the sole reasons behind the attrition of employees. These models can help us in prioritizing the features with higher impact in attrition of an employee and the possible reasons behind it so that HR can take appropriate decision for the retention process. The main purpose of this research is to build reliable and accurate models which can optimize the hiring and retention cost of quality employees. This could be done by determining the attrition status of employee under consideration by using the appropriate data mining techniques.

Reference

---

[1] K. Coussement and D. Van den Poel, “Integrating the voice of customers through call center emails into a decision support system for churn prediction,” Information & Management, vol. 45, no. 3, pp. 164–174, 2008.

[2] C.-P. Wei and I.-T. Chiu, “Turning telecommunications call details to churn prediction: a data mining approach,” Expert systems with applications, vol. 23, no. 2, pp. 103–112, 2002.

[3] K. Coussement and D. Van den Poel, “Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques,” Expert systems with applications, vol. 34, no. 1, pp. 313–327, 2008.

[4] J. Burez and D. Van den Poel, “Handling class imbalance in customer churn prediction,” Expert Systems with Applications, vol. 36, no. 3, pp. 4626–4636, 2009.

[5] C.-F. Tsai and M.-Y. Chen, “Variable selection by association rules for customer churn prediction of multimedia on demand,” Expert Systems with Applications, vol. 37, no. 3, pp. 2006–2015, 2010.

[6] B. Huang, M. T. Kechadi, and B. Buckley, “Customer churn prediction in telecommunications,” Expert Systems with Applications, vol. 39, no. 1, pp. 1414–1425, 2012.

[7] https://www.youtube.com/watch?v=Z5WKQr4H4Xk

[8] https://www.youtube.com/watch?v=XycruVLySDg

[9] https://uc-r.github.io/logistic_regression

[10] https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/logistic-regression-analysis-r/tutorial/

[11] https://www.r-bloggers.com/2015/09/how-to-perform-a-logistic-regression-in-r/

[12] https://www.geeksforgeeks.org/logistic-regression-in-r-programming/

[13] https://www.youtube.com/watch?v=HeTT73WxKIc

[14] https://www.youtube.com/watch?v=dJclNIN-TPo&t=928s

[15] https://www.edureka.co/blog/random-forest-classifier/

[16] Random Forest Approach in R Programming, 2020 random-forest approach-in-r-programming

[17] https://en.proft.me/2017/01/24/classification-using-random-forest-r/

[18] https://www.guru99.com/r-random-forest-tutorial.html