GRENZE Scientific Society

Analytic Approach of Predicting Employee Attrition using Naive Bayes and SVM Model

M.Kanagaraj, M.Anish Sharma, T.Mohamed Zebril,
Department of CSE, A.V.C College of Engineering, Mayiladuthurai
M.Kavitha, M.E.,
Asst. Professor, Department of CSE, A.V.C College of Engineering, Mayiladuthurai
Email: Kanagu1468@gmail.com

***Abstract*— Employee turnover has become a big challenge for Information Technology companies. The departure of key software developers might cause an enormous loss to an IT company since they also depart with important business knowledge and important technical skills. Understanding developer turnover is very important for IT companies to retain talented developers and reduce the loss due to developer's departure. In this Project, monthly self-report of the software developers includes developer's activities, working hours, no of projects they have been assigned, etc., will be been taken into account for analysis for doing the prediction with the help of Data Science algorithm. By using, Naive Bayes classification algorithm and SVM algorithm, the Classification and Prediction model will be created. The performance of both the models will be compared on an experimental basis and deliver the result of which algorithm is performing better. Then, this model will predict whether the employee will leave the company or not.**

Keywords – **Employee Attrition, Naïve Bayes, SVM Model, Comparison, Accuracy Output..**

I. INTRODUCTION

Retaining the simplest employees ensures customer satisfaction, increased revenues, and satisfied colleagues and staff. Organizations invest tons of cash on training, giving employees the onsite opportunity, offering compensations above market level to retain employees. However, currently, these methods are being generically applied to regulate employee attrition. Building data-driven predictive models for attrition, to predict future attrition over time, both at aggregate levels as well as for identifying individuals with a high risk of attrition. This is highly useful to HR where they can focus only on the main root causes when dealing with the employee. The insights, along side data-driven predictive models, are often wont to design effective plans for reducing attrition, improving retention, reducing attrition costs, and mitigating attrition effects. Employee Attrition is when an employee leaves a corporation thanks to normal means, (loss of consumers, retirement, and resignation), and there's not someone to fill the vacancy. A company with a high employee rate of attrition may be a good sign of underlying problems and may affect a corporation during a very negative way. Nowadays, employee attrition became a significant issue regarding a company's competitive advantage. It's very expensive to seek out, hire and train new talents. It's less expensive to stay the workers a corporation already has. A company must maintain a pleasing working

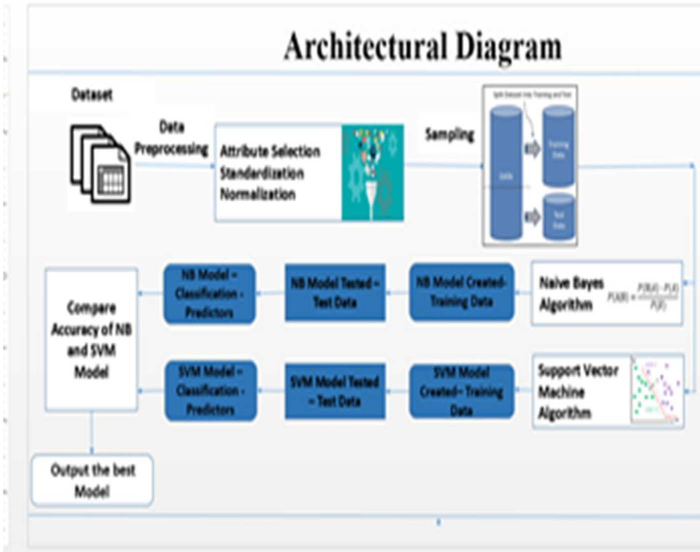
Grenze ID: 02. ICIEICE.2021.1.1

atmosphere to form their employees stay therein company for a extended period. Now, a company's HR Department uses some data analytics tool to identify which areas to be modified to make most of its employees stay. Retaining the simplest employees ensures customer satisfaction, increased revenues, and satisfied colleagues and staff. Organizations invest tons of cash on training, giving employees the onsite opportunity, offering compensations above market level to retain employees. However, currently, these methods are being generically applied to regulate employee attrition. Building data-driven predictive models for attrition, to predict future attrition over time, both at aggregate levels as well as for identifying individuals with a high risk of attrition. This is highly useful to HR where they can focus only on the main root causes when dealing with the employee. The insights, along side data-driven predictive models, are often wont to design effective plans for reducing attrition, improving retention, reducing attrition costs, and mitigating attrition effects

II. Leterature Survey

Lingfeng Bao, Zhenchang Xing, et. Al - Who Will Leave the Company. The critics of the project are low accuracy level of the output is obtained.
Andry Alamsyah; Nisrina Salma - A Comparative Study of Employee Churn Prediction Model, IEEE International Conference on Science and Technology, 2018. The precise output is not obtained.
Mohammad Nayeem Hasan - A Comparison of Logistic Regression and Linear Discriminant Analysis in Predicting of Female Students Attrition from School in Bangladesh, IEEE International Conference on Electrical Information and Communication Technology, 2019. This paper also gives not precise output.
Praphula Kumar Jain, etal - Explaining and predicting employees' attrition: a machine learning approach Springer Nature Applied Sciences, 2020. In this paper, data science is not used but the concept is similar.
Xiao-Li Qu - A decision tree applied to the grass-roots staffs' turnover problem IEEE International Conference on Grey Systems and Intelligent Services, 2015. In this paper, a decision tree algorithm was used. We use the high-level algorithm

III. Architecture


Architectural Diagram
Dataset
Data Preprocessing
Attribute Selection
Standardization
Normalization
Sampling
Naive Bayes Algorithm
Support Vector Machine Algorithm
NB Model Created-Training Data
NB Model Tested - Test Data
NB Model - Classification - Predictors
SVM Model Created- Training Data
SVM Model Tested - Test Data
SVM Model - Classification - Predictors
Compare Accuracy of NB and SVM Model
Output the best Model

In our project, we take the oracle dataset. First, we need to import the dataset into our environment. Next, we do data preprocessing which is used to removing the unwanted data in the dataset. After completing the process we divided the dataset into two formats. one is training data and another one is test data. Sampling deals with Naïve Bayes and SVM algorithm. At first, the Naïve Bayes and SVM model was created by using training data. Next, these two algorithms are tested using tested data. Both models work classification and prediction concepts and then compare two model outputs. Finally, a high accuracy output is obtained.

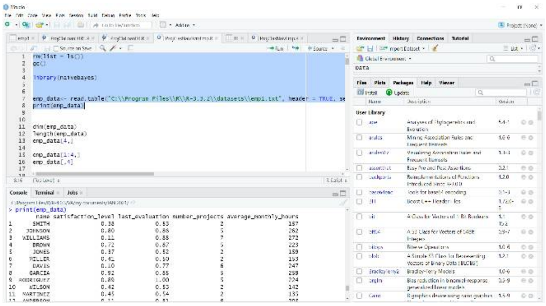
IV. METHDOLOGY

1. Data Importing and Preprocessing

Data have to be imported into the R environment for analysis. The Data can be any format like txt, .csv, .xlsx, .SPSS etc.Package necessary for Naive Bayes and Support Vector Machine algorithm has to be installed and loaded into the program.   NB –naivebayes, SVM – e1071

*Project Attributes:*name, satisfaction_level, last_evaluation, number_projects, average_monthly_hours, time_spent_company, work_accident, left, promotion_last_5_years, department, salary, salary_level

Eg: SMITH,0.38,0.53,2,157,3,0,1,0,sales,low,1



```
rm(list = ls())
gc()

library(naivebayes)


emp_data<- read.table("C:\\Program Files\\R\\R-3.3.2\\datasets\\emp1.txt", header = TRUE, se
print(emp_data)


dim(emp_data)
length(emp_data)
emp_data[4,]

emp_data[1:4,]
emp_data[,4]
```

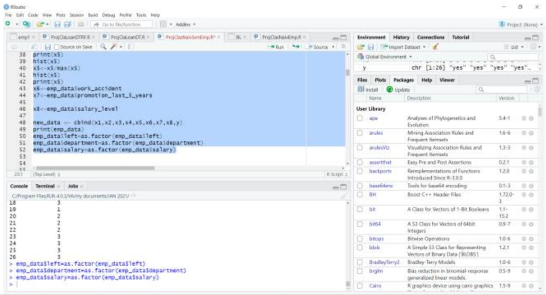
```
> print(emp_data)
        name satisfaction_level last_evaluation number_projects average_monthly_hours
1      SMITH               0.38            0.53               2                   157
2    JOHNSON               0.80            0.86               5                   262
3   WILLIAMS               0.11            0.88               7                   272
4      BROWN               0.72            0.87               5                   223
5      JONES               0.37            0.52               2                   159
6     MILLER               0.41            0.50               2                   153
7      DAVIS               0.10            0.77               6                   247
8     GARCIA               0.92            0.85               5                   259
9  RODRIGUEZ               0.89            1.00               5                   224
10    WILSON               0.42            0.53               2                   142
11  MARTINEZ               0.45            0.54               2                   135
```

Fig. 1. Data importing

In Data pre processing, attribute selection, standardization, and normalization functions will be applied. In standardization, raw data is transformed into a common, understandable format. In attribute selection, hold only the attributes which are affecting the analysis and it is not necessary to hold all the attributes for doing the analysis. In Normalization, the mean of

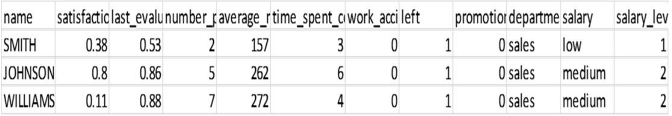
$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

the attribute will be 0 and the standard deviation will be 1.



```
print(x5)
hist(x5)
x5<-x5/max(x5)
hist(x5)
print(x5)
x6<-emp_data$work_accident
x7<-emp_data$promotion_last_5_years

x8<-emp_data$salary_level

new_data <- cbind(x1,x2,x3,x4,x5,x6,x7,x8,y)
print(emp_data)
emp_data$left=as.factor(emp_data$left)
emp_data$department=as.factor(emp_data$department)
emp_data$salary=as.factor(emp_data$salary)
```

```
18            3
19            1
20            2
21            2
22            2
23            3
24            3
25            3
26            3
> emp_data$left=as.factor(emp_data$left)
> emp_data$department=as.factor(emp_data$department)
> emp_data$salary=as.factor(emp_data$salary)
>
```



name	satisfactic	last_evalu	number_p	average_r	time_spent_c	work_acci	left	promotio	departme	salary	salary_lev
SMITH	0.38	0.53	2	157	3	0	1	0	sales	low	1
JOHNSON	0.8	0.86	5	262	6	0	1	0	sales	medium	2
WILLIAMS	0.11	0.88	7	272	4	0	1	0	sales	medium	2

Datasets

Fig. 2. Finished wearable device

2. Model Generation Using Naive Bayes Algorithm

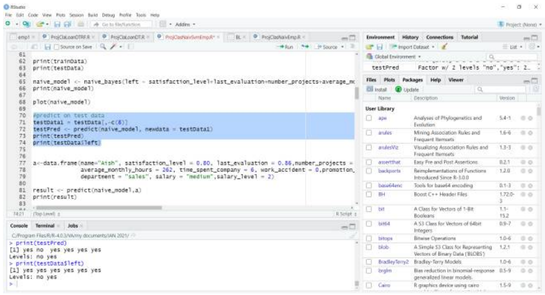
P(A|B) - Probability of occurrence of event A given the event B is true - Posterior Probability. P(A) - Probabilities of the occurrence of event A-Class prior Probability. P(B) - Probabilities of the occurrence of event B – Predictor Prior Probability. P(B|A) - Probability of the occurrence of event B given the event A is true - Likelihood.

Steps in Naive Bayes algorithm:

Step 1: Convert the data set into a frequency table.

Step 2: Create a Likelihood table by finding the probabilities values of each attribute.

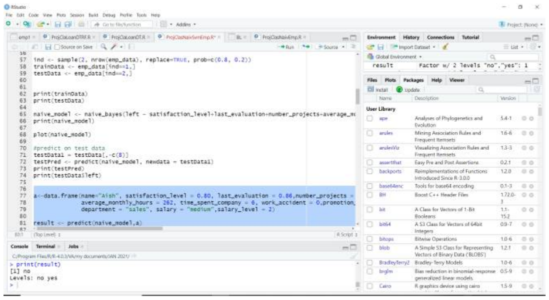
Step 3: Now, use a Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the the prediction.



```
print(trainData)
print(testData)

naive_model <- naive_bayes(left ~ satisfaction_level+last_evaluation+number_projects+average_mc
print(naive_model)

plot(naive_model)

#predict on test data
testData1 = testData[,-c(8)]
testPred <- predict(naive_model, newdata = testData1)
print(testPred)
print(testData$left)


a<-data.frame(name="Aish", satisfaction_level = 0.80, last_evaluation = 0.86,number_projects =
              average_monthly_hours = 262, time_spent_company = 6, work_accident = 0,promotion_
              department = "sales", salary = "medium",salary_level = 2)

result <- predict(naive_model,a)
print(result)
```

```
> print(testPred)
[1] yes no  yes yes yes yes
Levels: no yes
> print(testData$left)
[1] yes yes yes yes yes yes
Levels: no yes
>
```

Fig.3 NB model Testing Using Test Data



```
ind <- sample(2, nrow(emp_data), replace=TRUE, prob=c(0.8, 0.2))
trainData <- emp_data[ind==1,]
testData <- emp_data[ind==2,]


print(trainData)
print(testData)

naive_model <- naive_bayes(left ~ satisfaction_level+last_evaluation+number_projects+average_mc
print(naive_model)

plot(naive_model)

#predict on test data
testData1 = testData[,-c(8)]
testPred <- predict(naive_model, newdata = testData1)
print(testPred)
print(testData$left)


a<-data.frame(name="Aish", satisfaction_level = 0.80, last_evaluation = 0.86,number_projects =
              average_monthly_hours = 262, time_spent_company = 6, work_accident = 0,promotion_
              department = "sales", salary = "medium",salary_level = 2)

result <- predict(naive_model,a)
```

```
> print(result)
[1] no
Levels: no yes
>
```

Fig.4 NB model Classification Using Predictors

3.Model Generation Using SVM Algorithm

Classification is a process to determine the extent to which a thing will or will not be a part of a category or type. Regression is used to find the relationship between the variables and predict the future. SVM stands for Support Vector Machine. It is a machine learning approach used for classification and regression analysis. It depends on supervised learning models and is trained by learning algorithms. They analyze a large amount of data to identify patterns from them. An SVM generates parallel partitions by generating two parallel lines.

Each category of data in a high-dimensional space and uses almost all attributes. It separates the space in a single pass to generate flat and linear partitions. Divide the 2 categories by a clear gap that should be as wide as possible. Do this partitioning by a plane called a hyperplane. An SVM creates hyperplanes that have the largest margin in high-dimensional space to separate given data into classes.

The margin between the 2 classes represents the longest distance between the closest data points of those classes. The larger the margin, the lower is the generalization error of the classifier. After the training map, the new data to same space to predict which category they belong to. Categorize the new data into different partitions and achieve it by training data.

Classification can be performed by finding the hyper-plane that differentiates two classes which are shown in fig 4.2. Select the hyper-plane which segregates the two classes better. have three hyper-planes (A, B, and C) and all are segregating the classes well. Identify the right hyper-plane.

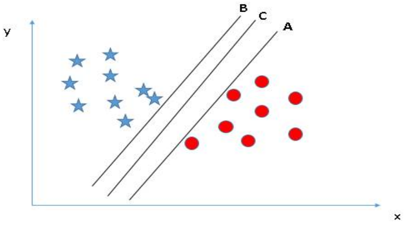
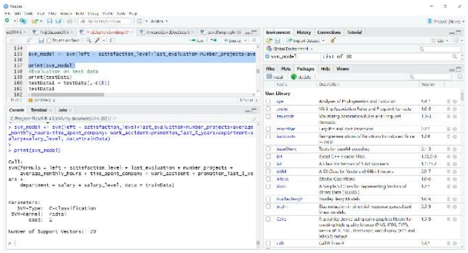


Fig.5 Classification using hyperplane

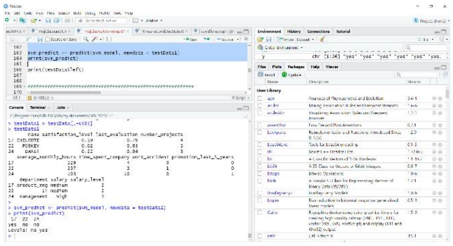
Maximizing the distances between the nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called a Margin. The margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with a higher margin is robustness. If we select a hyper-plane having a low margin, then there is a high chance of miss-classification. The following Support Vector Machine algorithm is faster than other classification and prediction algorithms. It is very simple and easy to implement. It needs less training data and able to handle continuous and discrete data. It is used for both binary and multiclass classification. Once the preprocessed data is applied to the Support Vector Machine algorithm, Support Vector Machine Model has been created.



```
svm_model <- svm(left ~ satisfaction_level+last_evaluation+number_projects+ave

print(svm_model)
#Evaluation on test data
print(testData)
testData1 = testData[,-c(8)]
testData1
```

```
> svm_model <- svm(left ~ satisfaction_level+last_evaluation+number_projects+average
_monthly_hours+time_spent_company+ work_accident+promotion_last_5_years+department+s
alary+salary_level, data=trainData)
>
> print(svm_model)

Call:
svm(formula = left ~ satisfaction_level + last_evaluation + number_projects +
    average_monthly_hours + time_spent_company + work_accident + promotion_last_5_ye
ars +
    department + salary + salary_level, data = trainData)


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

Number of Support Vectors:  20

>
```

Fig.6 SVM Model



```
svm_predict <- predict(svm_model, newdata = testData1)
print(svm_predict)

print(testData$left)


##########################################################
```

```
> testData1 = testData[,-c(8)]
> testData1
       name satisfaction_level last_evaluation number_projects
17 CHILCOTE               0.19            0.79               4
22   FOSKEY               0.62            0.85               3
24    DARST               0.22            0.94               3
   average_monthly_hours time_spent_company work_accident promotion_last_5_years
17                   228                  4             0                      0
22                   237                  3             1                      0
24                   103                 10             0                      1
    department salary salary_level
17 product_mng medium            2
22          IT medium            2
24  management   high            3
>
> svm_predict <- predict(svm_model, newdata = testData1)
> print(svm_predict)
 17  22  24
yes  no  no
Levels: no yes
>
```

Fig.7 SVM Model Testing using Test Dataset

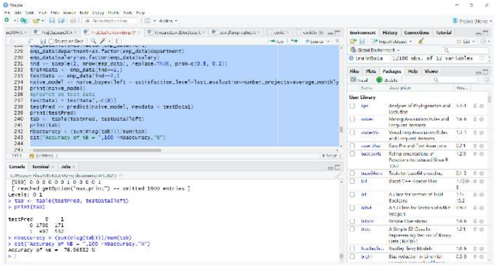
4. Comparison of NB and SVM Algorithm **-**

Naive Bayes algorithm and Support Vector Machine algorithm are used to classify and predict the developer turnover using factors, such as satisfaction_level, last_evaluation, number_projects, average_monthly_hours, time_ spent_company, etc. The performance of both the Naive Bayes algorithm and Support Vector Machine has been compared in terms of accuracy.

IV. Results And Discussion:

Naive Bayes Algorithm predicted the developer turnover with the accuracy of 76% and Support Vector Machine with the accuracy of 96%. Performance of Support Vector Machine is more accurate when compared to Naive Bayes Algorithm for this dataset. So, the SVM Algorithm model will predict and classify whether the developer will leave the company or not.

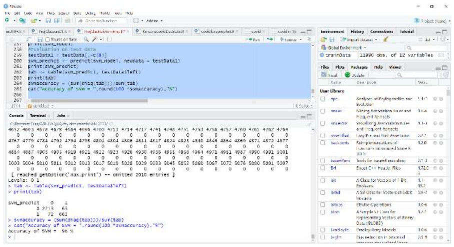
Performance of NB algorithm



```
emp_data$department=as.factor(emp_data$department)
emp_data$salary=as.factor(emp_data$salary)
ind <- sample(2, nrow(emp_data), replace=TRUE, prob=c(0.8, 0.2))
trainData <- emp_data[ind==1,]
testData <- emp_data[ind==2,]
naive_model <- naive_bayes(left ~ satisfaction_level+last_evaluation+number_projects+average_monthly
print(naive_model)
#predict on test data
testData1 = testData[,-c(8)]
testPred <- predict(naive_model, newdata = testData1)
print(testPred)
tab <- table(testPred, testData$left)
print(tab)
nbaccuracy = (sum(diag(tab)))/sum(tab)
cat("Accuracy of NB = ",100 *nbaccuracy,"%")
```

```
 [989] 0 0 0 0 0 0 0 1 0 0 0 0 1
 [ reached getOption("max.print") -- omitted 1800 entries ]
Levels: 0 1
> tab <- table(testPred, testData$left)
> print(tab)

testPred    0    1
       0 1700  171
       1  407  532
> nbaccuracy = (sum(diag(tab)))/sum(tab)
> cat("Accuracy of NB = ",100 *nbaccuracy,"%")
Accuracy of NB =  76.98552 %
>
```

Performance of SVM algorithm



```
print(svm_model)
#evaluation on test data
testData1 = testData[,-c(8)]
svm_predict <- predict(svm_model, newdata = testData1)
print(svm_predict)
tab <- table(svm_predict, testData$left)
print(tab)
svmaccuracy = (sum(diag(tab)))/sum(tab)
cat("Accuracy of SVM = ",round(100 *svmaccuracy),"%")
```

```
4652 4663 4678 4679 4684 4698 4700 4713 4714 4717 4741 4748 4751 4753 4756 4757 4760 4761 4762 4764
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
4767 4779 4784 4792 4794 4795 4801 4804 4806 4811 4817 4824 4825 4830 4849 4864 4869 4871 4872 4877
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
4884 4887 4903 4905 4918 4921 4922 4923 4926 4930 4936 4955 4958 4964 4971 4981 4987 4990 4991 5001
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
5003 5004 5010 5011 5012 5013 5017 5019 5028 5029 5033 5045 5052 5060 5067 5072 5076 5090 5091 5097
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 [ reached getOption("max.print") -- omitted 2010 entries ]
Levels: 0 1
> tab <- table(svm_predict, testData$left)
> print(tab)

svm_predict    0    1
          0 2213   63
          1   72  662
> svmaccuracy = (sum(diag(tab)))/sum(tab)
> cat("Accuracy of SVM = ",round(100 *svmaccuracy),"%")
Accuracy of SVM =  96 %
> 
```

V. CONCLUSION

This project addresses the Naive Bayes and SVM algorithm for Classifying and predicting, whether the employee will leave the company or not. The performance of both SVM and Naive Bayes Algorithms has been compared in terms of accuracy. Naive Bayes Algorithm predicted the employee turnover with the accuracy of 76% and SVM with the accuracy of 96%. This project concludes performance of the SVM algorithm is more excellent than the accuracy of the Naive Bayes algorithm. This SVM model can potentially help a company to predict the departure of key software developers and they can be retained in the company, by taking proactive action such as providing salary hikes or flexible timing or by better managing workload variance among project members, etc. to avoid a huge loss to the company.

REFERENCES

[1] Andry Alamsyah; Nisrina Salma, “A Comparative Study of Employee Churn Prediction Model", IEEE International Conference on Science and Technology, 2018.

[2] Anusha Garlapati et,al , "Predicting Employees under Stress for Pre-emptive Remediation using Machine learning Algorithm", IEEE International Conference on Recent Trends on Electronics, Information, Communication & Technology, 2020.

[3] Dilip Singh Sisodia, “Evaluation of Machine Learning Models for Employee Churn Prediction ", IEEE International Conference on Inventive Computing and Informatics, 2017.

[4] Francesca Fallucchi, “Predicting Employee Attrition Using Machine Learning Techniques", Mdpi Journal of computers, 2020.

[5] Kuei-Chen Chiu; Tsai-Wei Huang; Shulan Hsieh; Abdul Hameed Pitafi, " An employee assistance program by analyzing the correlation between work stress and dreams for Chinese employees", IEEE First International Conference on System Analysis & Intelligent Computing, 2014

[6] Lingfeng Bao, Zhenchang Xing, “Who Will Leave the Company? A Large-Scale Industry Study of Developer Turnover by Mining Monthly Work Report", IEEE/ACM International Conference on Mining Software Repositories, 2017.

[7] Mohammad Nayeem Hasan, “A Comparison of Logistic Regression and Linear Discriminant Analysis in Predicting of Female Students Attrition from School in Bangladesh”, International Conference on Electrical Information and Communication Technology, 2019

[8] Sundus Younis, " Know Your Stars Before They Fall Apart: A Social Network Analysis of Telecom Industry to Foster Employee Retention Using Data Mining Technique", IEEE Access Volume: 9, 2021.

[9] Xinjun Cai, "DBGE: Employee Turnover Prediction Based on Dynamic Bipartite Graph Embedding", IEEE Access, Volume: 8, 2020. https://ieeexplore.ieee.org/document/89557880

[10] Xiao-Li Qu, “A decision tree applied to the grass-roots staffs' turnover problem", IEEE International Conference on Grey Systems and Intelligent Services, 2015.