HR Analytics Using Machine Learning

Samrudh Nagesh
Stevens Institute of Technology
Hoboken, New Jersey
snagesh@stevens.edu

Ajay Malik
Stevens Institute of Technology
Hoboken, New Jersey
amalik8@stevens.edu

Alison Sorkenn
Stevens Institute of Technology
Hoboken, New Jersey
asorkenn@stevens.edu

Abstract—**In any business organization, there are many positions in the organization for their employees. These positions are sometimes based on hierarchy where employees who are in top level positions are more experienced and have higher skill level than the employees who work under them. Employees who are in the lower levels, however, can get promoted to a higher position by the organization if their work efforts are recognized by the organization. The role of analyzing, screening, recruiting and promoting workers in a company is done by the company's Human Resources (HR) manager. This project is developed to assist a HR manager in the tasks mentioned above by creating a model using different machine learning algorithms such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Forest Regressor and Decision Trees.**

Index Terms—**Machine Learning, Artificial Neural Networks, SVM, Random Forest Regressor, Decision Trees.**

I. Introduction

Every business and corporations hire employees who are assigned to work in their company's various departments based on the employee's skillsets. These employees utilize their skills to accomplish the tasks and objectives set by their company to fulfil their company's goal. By committing their time and energy to their company, an employee can be rewarded with a promotion to a higher rank within the company. A company's job positions are usually based in a hierarchy where top-level jobs are usually reserved for employees with high level skills and high knowledge on their respective job. They should also have many years of experience in their field of work. Finding suitable and deserving employees in a company for promotion could be difficult as there could be thousands of employees are many of them could be in same contention for a promotion. This process of promoting an employee based on their skill and dedication is overseen by the company's HR manager.

Human resources specialists are responsible for recruiting, screening, interviewing and placing workers. They may also handle employee relations, payroll, benefits, and training. Human resources managers plan, direct and coordinate the administrative functions of an organization. They oversee specialists in their duties; consult with executives on strategic planning, and link a company's management with its employees. HR specialists tend to focus on a single area, such as recruiting or training. HR generalists handle a number of areas and tasks simultaneously. Small companies will typically have one or two HR generalists on staff, while larger ones may have many devoted to particular areas and services.

The purpose of the project is to develop a system which analyzes the dataset of all employees in a company and to display weather they are eligible to be promoted by the company

The dataset for this model should consist of all employees in the company. The company will have various departments; therefore, the dataset can be further be divided by the various departments the employees are assigned to. This can help the user to analyse employee data based on performance in each department. The dataset should also consist of the professional data of the employees. Data, such as number of years of experience, education, number of promotions, etc. Professional data as mentioned is very important as they are the parameters that weigh the most when it comes to deciding whether to promote an employee or not. The dataset also consists of personal data such as age which could also be a factor in determining the result. These parameters in the dataset are entered into different machine learning algorithms and tested for their accuracy in order to determine which algorithm provides the most accurate results so that it can be used to predict the results any existing employee or new employee in the company.

For this project we use different machine learning algorithms such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Forest Regressor and Decision Trees.

- *Support Vector Machines*: SVM's are supervised learning models with associated learning algorithm that analyse data for classification and regression analysis.
- *Artificial neural networks*: ANN's are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold.
- *Random Forest Classifier*: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- *Decision Tree Classifier*: Decision tree classifier is a supervised machine learning algorithm that uses a set of rules to design a decision tree. We use decision trees to make observations about an object (or data) or make conclusions about that data target value.

II. Related Work

While researching for this project, we have found that there already exist various other models for HR analytics using machine learning. Each model use different machine learning models and they also predict different results using their models. But every model has the same motive which is the computerise HR analytics in order to assist a company's HR manager.

D. S. Sisodia, S. Vishwakarma and A. Pujahari in their work showed how important employees are to any organisation and how difficult it is in terms of cost for that organisation of employees quit unexpectedly. They showed that hiring new employees will consume not only money and time but also the freshly hired employees take time to make the respective organization profitable. In their model, they predict employee churn rate based on HR analytics dataset obtained from Kaggle website. The correlation matrix and heatmap is generated to show the relation between the attributes they have chosen for their model. For prediction, they have used five different machine learning algorithms such as linear support vector machine, C 5.0 Decision Tree classifier, Random Forest, k-nearest neighbor and Naïve Bayes classifier [1]. Sisodia, in their model, make use of a histogram between the employees who have left vs their salaries to see their job satisfaction. This model is useful for us as they have used multiple ML algorithms in their model. But they are analysing the reason why employees unexpectedly leave a company, which is not our objective.

Franchesca Fallucchi, Marco Coladangelo, Romeo Giuliano and Ernesto William De Luca in their paper 'Predicting Employee Attrition using Machine Learning Techniques' show how there is more attention given to Human Resources (HR) by the company since worker quality skills represent growth in production and are also real competitive advantage for that company. They also explain how Artificial Intelligence and machine learning techniques are used to guide employee related decisions within HR management. Similar to D.S. Sisodia's model [1], they are trying to analyse how objective factors try to influence employee attrition, in order to figure out the main reasons and factors behind an employee's decision to leave the company. They train the model and test it on a dataset provided by IBM analytics which contains 35 features of about 1500 samples. The results they obtained are expressed in terms of classical metrics and the algorithm that produced the best results is Gaussian Naïve Bayes Classifier [2].

V. Kakulapati, Kalluri Krishna Chaitanya, Kolli Vamsi Guru Chaitanya and Ponugoti Akshay, in their model apply machine learning techniques to analyse the employee information for improving his/her position in the organization. In their model, they use random forest classification, which facilitates employee classification based on their monthly income and informal way to execute analytics on data. Further, we use clustering techniques based on the performance metrics similarity to analyze employee performance [3]. Although this is similar to the model we are trying to build, we use more than one machine learning algorithm to analyze employee information and generate output.

III. Our Solution

A. Description of Dataset

For this project, since we are dealing with a corporation's employee data, we have a large dataset of about 58,000 employees. The dataset consists of the professional data of these employees to analyse whether they are eligible for a promotion or not. The professional data for the employees consists of their employee ID, the department whey work under, their education, gender and age.

The dataset also consists of number of training sessions undergone by the employee and their average training score. There is also data on how they were recruited by the company, their length of service in that

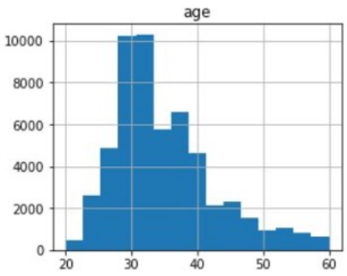


Fig. 1. Histogram of Employee Age

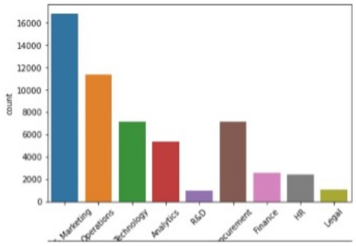


Fig. 2. Departments in the Organization with Employee Count

company, the number of times they were promoted in the past and the number of awards won by the employee.

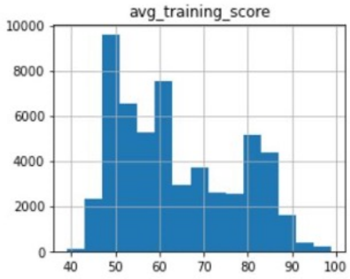


Fig. 3. Average Training Score Graph

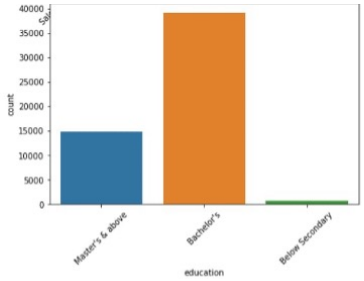


Fig. 4. Employee Education Qualification

The company also keeps track of the employee's key performance indicator (KPI) and how many times that statistic was greater than 80% after every year. With these parameters, we take the data of each employee and analyze them to check whether they have met the criteria to be promoted.

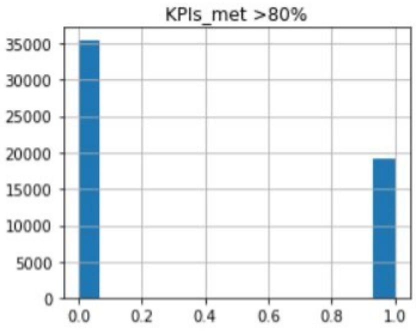


Fig. 5. Employees With KPI $>$ 80%

Initially, the dataset consisted of many NULL values in fields such as previous_years_ratings. Since these parameters should not be NULL, we use the mean to fill these null values using the fillna() function in python.

B. *Machine Learning Algorithms*

For this project, we plan to use 4 different machine learning algorithms such as SVM, Random Forest classifier, artificial neural networks and decision tree.

Initially we started off with Random Forest Classifier. We first chose the parameters for the Random Forest Classifier as n_estimators=100, max_depth= 10, random_state=0. We then use GridSearchCV to find the best parameters for n_estimators and max_depth for the classifier. This will be further explained in section C. We decided on Random forest classifier as one of the machine learning algorithms as it uses multiple decision tree classifiers to predict the output.

Another algorithm we have tested for accuracy was the MLPClassifier algorithm. Using the 'sgd' solver and hidden_layer_sizes as 2, we get the accuracy of 49.98%, which is very low. Using the 'adam' solver and hidden_layer_sizes as 100, we get the accuracy of 82.36%. We can further try out other various parameters and find the best parameters for highest accuracy for our model using GridSearchCV.

Decision Tree Classifier can also be used in our model. Since decision trees are used for regression problems, we can implement this algorithm to our model. By setting the parameter max_depth as 10 we get the accuracy as 82.72%. By increasing the max_depth to 20 we improve our models accuracy to 91.72%. Setting max_depth to 40 we get the accuracy of 96.09% which is very high.

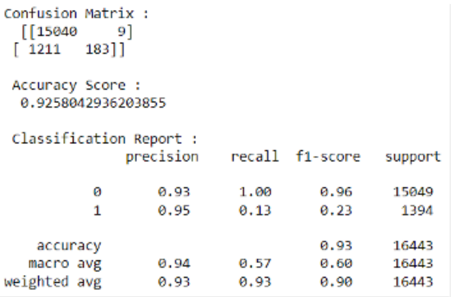
We try to increase this accuracy by pruning the tree. We again use GridSearchCV to find the best parameters for max_leaf_nodes to prune our decision tree to analyze if we get higher accuracy.

Support Vector Machines are supervised learning models with associated learning algorithm that analyse data for classification and regression analysis. For the algorithm, we use SVC (Support Vector Classification) with a linear kernel.

C. Implementation Details

As mentioned in the previous section, we first split the dataset into training and testing data. We then use the training data to predict the results and compare that result with our testing data.

1) Random Forest Classifier: Using random forest classifier, using n_estimators as 50 and max_depth as 5, we found that the accuracy is 92.58%. We also calculate the precision, recall and f-1 score of the algorithm. This is shown in Fig. 6 below.

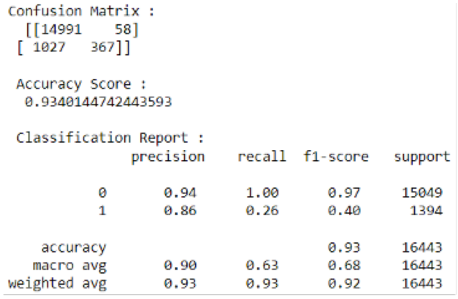


```
Confusion Matrix :
 [[15040     9]
[ 1211   183]]

 Accuracy Score :
 0.9258042936203855

 Classification Report :
               precision    recall  f1-score   support

           0       0.93      1.00      0.96     15049
           1       0.95      0.13      0.23      1394

    accuracy                           0.93     16443
   macro avg       0.94      0.57      0.60     16443
weighted avg       0.93      0.93      0.90     16443
```

Fig. 6. Classification Report for n_estimators=50, max_depth=5

If we need this algorithm to provide results with higher accuracy, we need to find the best parameters of n_estimators and max_depth. For this we use GridSearchCV. Using GridSearchCV, we find that the best parameters for n_estimators is 300 and max_depth is 20. By using these parameters in our random forest classifier algorithm, we get an accuracy of 93.40%.

Since we use various parameters in our dataset as mentioned in section III.A, we need to give weightage to these parameters as some data of an employee is more important to the organisation than others. For example, average_training_score has more value to an employee and organization than the employee's gender. Therefore, we use feature_importances_ and plot a graph to show which parameters are of more importance than the others.



```
Confusion Matrix :
 [[14991    58]
[ 1027   367]]

 Accuracy Score :
 0.9340144742443593

 Classification Report :
               precision    recall  f1-score   support

           0       0.94      1.00      0.97     15049
           1       0.86      0.26      0.40      1394

    accuracy                           0.93     16443
   macro avg       0.90      0.63      0.68     16443
weighted avg       0.93      0.93      0.92     16443
```

Fig. 7. Classification Report for n_estimators=300, max_depth=20

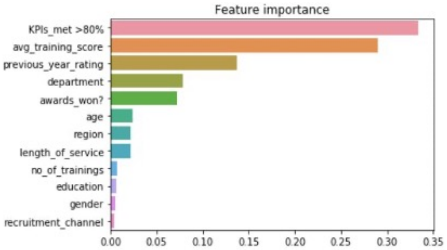


Fig. 8. Feature Importance of Dataset

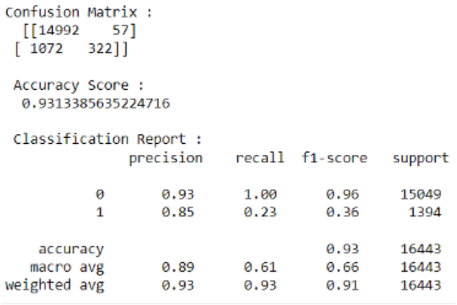
From the plot, we observe that 'KPIs_met>80%' has most importance to an organization.

2) Artificial Neural Networks: To implement Artificial Neural Networks (ANN) into our model, we can use the algorithm MLPClassifier. To use this algorithm, we need to find the best suitable parameters for this algorithm to generate the best accuracy possible. We need to find the number of hidden layers and number of neurons in each layer. We should also find the activation function for the hidden layer. We have used two different activation functions-'logistic' and 'tanh' which is the hyperbolic tan function. We should also find the solver for weight optimization. For our model we have 'adam' over 'sgd'(stochastic gradient descent. The solver 'adam' is just another stochastic gradient-based optimizer but it works better for very large datasets for both training time and validation score and therefore we use this parameter.

By using GridSearchCV, we find the number of hidden layers we use is 20. By using these parameters in MLPClassifier, for the activation function tanh we get the accuracy of 93.13%.

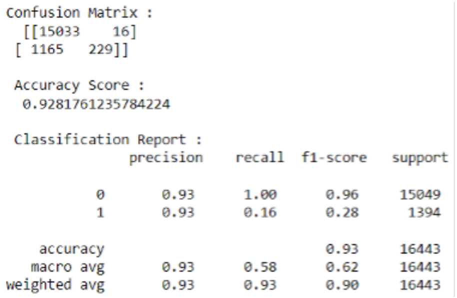
For the activation function 'logistic' we get the accuracy of 92.81%.

Although the difference in accuracy is very low using the activation function tanh gave the highest accuracy for MLPClassifier algorithm.



```
Confusion Matrix :
 [[14992    57]
 [ 1072   322]]

 Accuracy Score :
  0.9313385635224716

 Classification Report :
               precision    recall  f1-score   support

           0       0.93      1.00      0.96     15049
           1       0.85      0.23      0.36      1394

    accuracy                           0.93     16443
   macro avg       0.89      0.61      0.66     16443
weighted avg       0.93      0.93      0.91     16443
```

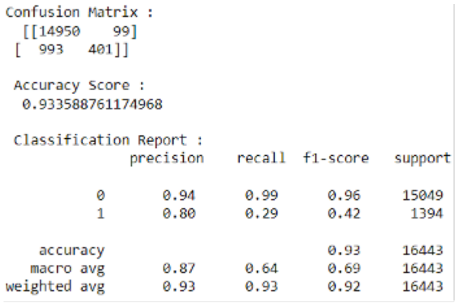
Fig. 9. Classification Report and Accuracy for ‘tanh’



```
Confusion Matrix :
 [[15033    16]
 [ 1165   229]]

 Accuracy Score :
  0.9281761235784224

 Classification Report :
               precision    recall  f1-score   support

           0       0.93      1.00      0.96     15049
           1       0.93      0.16      0.28      1394

    accuracy                           0.93     16443
   macro avg       0.93      0.58      0.62     16443
weighted avg       0.93      0.93      0.90     16443
```

Fig. 10. Classification Report and Accuracy for ’logistic’

3) Decision Tree Classifier: To obtain our decision tree, we use the DecisionTreeClassifier algorithm. By setting the random_state to 0, we trained the data with max_depth of 10 ,20 and 40. We find the highest accuracy with max_depth of 10 with 93.35%.



```
Confusion Matrix :
 [[14950    99]
 [  993   401]]

 Accuracy Score :
  0.933588761174968

 Classification Report :
               precision    recall  f1-score   support

           0       0.94      0.99      0.96     15049
           1       0.80      0.29      0.42      1394

    accuracy                           0.93     16443
   macro avg       0.87      0.64      0.69     16443
weighted avg       0.93      0.93      0.92     16443
```

Fig. 11. Classification Report and Accuracy for Decision Tree

To find out if we can get better accuracy, we try to prune the decision tree. We, again, use GridSearchCV to find the best parameter for max_leaf_nodes. We try to limit the maximum number of leaf nodes to 10. Using GridSearchCV, we find the best value for max_leaf_node to be 9. By pruning the tree, we get the tree shown in Fig.12.

Although we obtain a pruned tree, our accuracy does not improve. We find the accuracy of the pruned tree to be 77.42%.

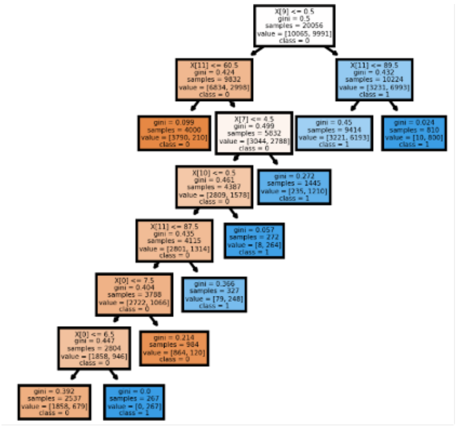
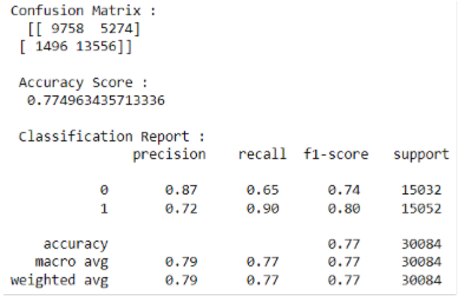


Fig. 12. Pruned Decision Tree



```
Confusion Matrix :
 [[ 9758  5274]
 [ 1496 13556]]

 Accuracy Score :
  0.774963435713336

 Classification Report :
               precision    recall  f1-score   support

           0       0.87      0.65      0.74     15032
           1       0.72      0.90      0.80     15052

    accuracy                           0.77     30084
   macro avg       0.79      0.77      0.77     30084
weighted avg       0.79      0.77      0.77     30084
```

Fig. 13. Classification Report and Accuracy for Pruned Tree

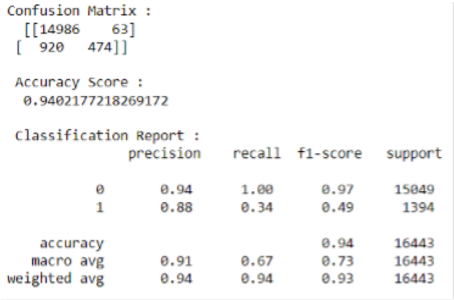
Since the accuracy did not increase after pruning the tree, we conclude that the decision tree should not be pruned to get accurate results.

4) Support Vector Machines: We tried to implement SVM’s into our model, but it is impractical to use SVM for a dataset as large as ours. By using a linear kernel, we could obtain an accuracy of 73.01%. We could get higher accuracy by HyperParameter Tuning using GridSearchCV but for our dataset with over 54,000 samples and with the hardware resources we currently have, the search will consume hours of time to complete.

5) XGBoost: XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning. We tried the XGBoost algorithm to compare the accuracy with our previous algorithms. We obtained a high accuracy of 94.02%.

IV. Comparison

Out of the 5 different machine learning algorithms we have used, we have observed that Random forest classifier and XGBoost give the highest accuracy while



```
Confusion Matrix :
 [[14986    63]
[   920   474]]

 Accuracy Score :
  0.9402177218269172

 Classification Report :
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     15049
           1       0.88      0.34      0.49      1394

    accuracy                           0.94     16443
   macro avg       0.91      0.67      0.73     16443
weighted avg       0.94      0.94      0.93     16443
```

Fig. 14. Classification Report and Accuracy for XGBoost

comparing the testing dataset and predicted target value. Since random values are used in some algorithms, the accuracy varies every time we restart the kernel. For this particular test, we find the accuracy of XGBoost to be the highest, followed by random forest and decision tree. SVM is the least accurate as we have not used the best possible parameters for the algorithm. Searching for the best parameters for SVM will consume a lot of time for the large dataset we are using. XGBoost produced the highest accuracy as has built in Lasso Regression and Ridge Regression to prevent overfitting. It also uses multiple CPU cores to execute the model.

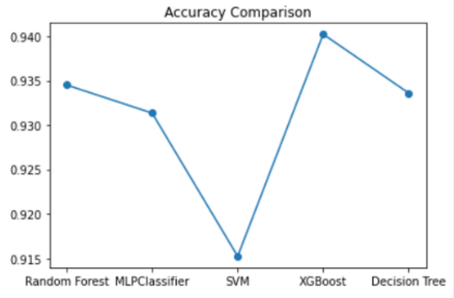


Fig. 15. Plot to Compare Accuracy of Algorithms Used

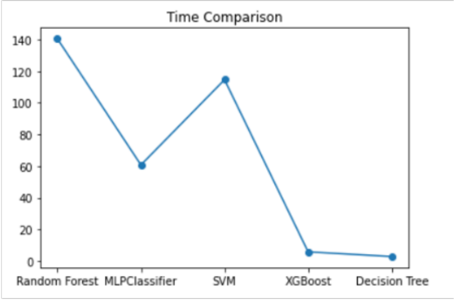


Fig. 16. Plot to Compare Runtime of Algorithms Used

As seen in Fig. 16 above, in addition to determining the accuracy of the five algorithms, we also ran a comparison on the runtimes of each algorithm in order to see which one was the fasted and most efficient to be used. After we determined the accuracies and time of the machine learning algorithms, we were able to compile the results into a table as shown in Table 1. After comparing each of the five algorithms we ran, we were able to see that the XGBoost algorithm was the fasted and had the highest accuracy.

TABLE I
COMPARISON OF ACCURACY VS TIME

Algorithm	Accuracy	Time (Seconds)
Random Forest	93.43%	140.58
MLP Classifier	93.13%	60.72
SVM	91.52%	114.63
XGBoost	94.02%	5.71
Decision Tree	93.35%	2.78

V. FUTURE RESEARCH DIRECTIONS

This project has a lot of potential for future expansion. We can design a very interactive and simple user interface and also use database management to develop a computer application that can assist a company's HR manager in their work.

Since, in this project we are only focusing on whether an employee can be promoted or not, we can also add other functions for this application. Functions such as employee screening data storage and visualization, employee payroll management and training analysis.

VI. CONCLUSION

The purpose of this project is to develop a system where the user, who is the HR manager of a company can determine whether an employee can get promoted by the company. By using machine learning algorithms, we find that in our dataset of over 58,000 employees, approximately 4,600 employees are eligible for a promotion. Our model uses 5 different ML algorithms and compares the accuracy to find the most accurate algorithm for the given dataset and uses this algorithm to predict the output. In our model, we have found that XGBoostClassifier gave us the best results not only in terms of accuracy but also time. By implementing this model to our test data, we can predict the employees who are eligible for promotion.

References

[1] D. S. Sisodia, S. Vishwakarma and A. Pujahari, ”Evaluation of machine learning models for employee churn prediction,” 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 1016-1020, doi: 10.1109/ICICI.2017.8365293.

[2] Fallucchi, F.; Coladangelo, M.; Giuliano, R.; William De Luca, E. Predicting Employee Attrition Using Machine Learning Techniques. Computers 2020, 9,86.https://doi.org/10.3390/computers9040086

[3] V. Kakulapati, Kalluri Krishna Chaitanya, Kolli Vamsi Guru Chaitanya and Ponugoti Akshay (2020) Predictive analytics of HR - A machine learning approach, Journal of Statistics and Management Systems, 23:6, 959-969, DOI: 10.1080/09720510.2020.1799497