MACHINE LEARNING REPORT

IBM EMPLOYEE ATTRITION ANALYSIS AND PREDICTION

Module Leader: Dr. Alessandro Di Stefano

Module Code: CIS4035-N

Name: Rohith Vaddepally

Student ID: W9529921

IBM EMPLOYEE ATTRITION ANALYSIS AND PREDICTION

ABSTRACT

In this paper, an employee attrition of the company IBM is analysed and predicted with the implementation of seven different supervised machine learning algorithms to find out the factors and reasons behind an employee leaving the organization. Attrition means departure of an employee from an organization. In this fast-moving world, it is their responsibility of the organization's Data Scientist and Data Analyst to find out the different factors and reasons why an employee is leaving the organization. By this organization can improve and try to stop the employee to leave.

Firstly, after loading the employee attrition dataset a basic analysis is done. It is found that there are no null values in the data and it contains 1470 rows and 35 columns. Secondly, Exploratory Data Analysis is performed on the data where Univariate analysis is done on few selected important columns and Multivariate Analysis is done to the columns against the Attrition column. The count of the employees who had left are 237 and employees who stayed are 1233. After the EDA, feature engineering is done the columns like removing the unwanted columns, identified and removed the highly correlated columns from the data. Label encoding and Numerical encoding is performed on the remaining columns before balancing the target training values. As the target values are balanced now, 7 different supervised machine learning algorithms are applied to predict the attrition rate of the employees from the organization. In comparative analysis, the algorithm which produces the best results with Accuracy, Precision, Recall, ROC AUC scores will be considered as the best for this data.

INTRODUCTION

Attrition is defined as departure of an employee from an organization. The reasons behind this are pressure from the managers, Employee salary could be low, travelling distance from the home to office could be high, not getting any recognition in the team etc. There will always be various personal and professional reasons other than few mentioned above. There will be a problem when these attrition rates are higher and the organization should take few precautions to stop the employees from leaving. It is always the duty of the leaders, managers and portfolio leaders of the company to analyse these situations and implement preventive measures to improve their business performance. Machine learning has become a vital part of the human's day to day life. Including machine learning to analyse the data and predict the churn or attrition rate will be helpful to the organization.

Objectives

This paper discusses the implementation of different machine learning algorithms to find out the factors and predict the attrition of the employees in the organization IBM.

Literature Review

As the problem of the research is discussed above, a Literature Review is performed on the research topic to gather more information about this. The databases chosen for the literature review is SCOPUS and IEEE XPLORE. A few journals or conference papers are read and are discussed below.

Keywords – Machine Learning, ML Techniques and Algorithms, Attrition Analysis, Attrition Prediction, Employee Churn.

(Mansor, Sani and Aliff, 2021) conducted a study on the IBM attrition prediction with applying machine learning algorithms like Decision Trees, Support Vector Machines and Artificial Neural Networks. Initially, there was an imbalance in the target trained values. To resolve the imbalance nature in the data, SMOTE (synthetic minority oversampling technique) is used. Finally, in their paper it is concluded that SVM stood out with best accuracy, RMSE and speed value after parameter tuning and regularization.

In a conference paper, Study and Predictive Analysis of the Employee Turnover Using Machine Learning Approaches (R. Chakraborty *et al.*, 2021) after pre-processing and data validation techniques the two best algorithms like Random Forest and Naïve Bayes Classifier. Exploratory data analysis is done and then few missing values and categorical variables were handling in the data. The Random Forest algorithm is found to be best algorithm with precision of 0.99 and F-1 score of 0.59. In this referred conference paper, a detailed architecture diagram is given which is shown below.

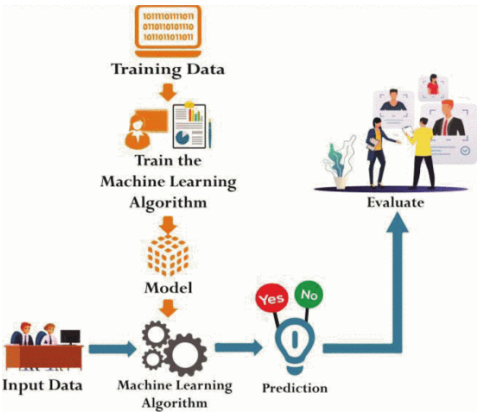

Figure 1: Architecture diagram for predicting the employee attrition. (R. Chakraborty *et al.*, 2021).

(Saradhi and Palshikar, 2011) conducted a study on employee churn prediction and developed machine learning models like Support vector Machine, Random Forests and Naïve Bayes and concluded that SVM model can be used to build reliable and accurate predictive models for employee attrition.

DATA EXPLORATION

The dataset IBM HR Analytics Employee Attrition & Performance is taken from www.kaggle.com. This is the fictional dataset created by IBM. In the selected data, there are a total of 35 columns and 1470 observations in total. These all uncovers the factors that lead to employee leaving the organization and also answer some important questions such as Business Travel vs Attrition, is overtime effecting the attrition rate? Etc.

Importing the libraries

There are many libraries in python. So, the libraries used here are

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Sklearn

```
185]: #importing numerical python and pandas libraries
      import numpy as np
      import pandas as pd

      #Numpy is used to perform mathematical operations with the data and it also adds extra support arrays
      #pandas is used to perform data analysis and manipulation within the data.

[194]: #importing matplot library into the notebook. Matplotlib is to create static, interactive and animat
       import matplotlib.pyplot as plt
       #%matplotlib inline is basically which happens back-end to work correctly with matplotlib library
       %matplotlib inline

       #importing seaborn library into the notebook. Seaborn library is related and based to matplotlib whi
       import seaborn as sns

242]: #importing label encoder and standard scaler libraries
      from sklearn.preprocessing import LabelEncoder
      from sklearn.preprocessing import StandardScaler

248]: #Importing train_test_split from sklearn library
      from sklearn.model_selection import train_test_split
      X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1) #Here

60]: #importing the accuracy score, confusion matrix, roc auc score, classification report from sklearn
     from sklearn.metrics import accuracy_score
     from sklearn.metrics import confusion_matrix
     from sklearn.metrics import roc_auc_score
     from sklearn.metrics import classification_report
```

Figure 2: Screenshots of libraries that are imported in the application.

Loading the dataset

The raw dataset which is available from the website Kaggle is in the form of csv file. Used the library Pandas to read the csv file and load it into the notebook.

1. DATA LOADING INTO THE JUPYTER NOTEBOOK

```
In [185]: #importing numerical python and pandas libraries
          import numpy as np
          import pandas as pd

          #Numpy is used to perform mathematical operations with the data and it also adds extra sup
          #pandas is used to perform data analysis and manipulation within the data.

In [186]: empatt = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

Using pandas, we have read the data into the notebook and stored into the variable empatt

Figure 3: Screenshot of loading the dataset into the notebook.

FEATURE ENGINEERING

Before applying the machine learning models directly on the dataset, feature engineering is done. Feature pre-processing is must because without this model could lead to incorrect precisions and accuracy.

Removing the unwanted columns

In the data, there are few unwanted columns which contains one data level where they are of no use for modelling and does not help our research. Those columns are Employee Count, Standard Hours, Over18 and Employee Number. These unwanted columns are removed from the data.

Identifying correlation between the columns

Next, identified the highly correlated columns in the data by plotting a heatmap. Heatmap is plotted using a seaborn library and the screenshot is shown below (Kumar, 2022).

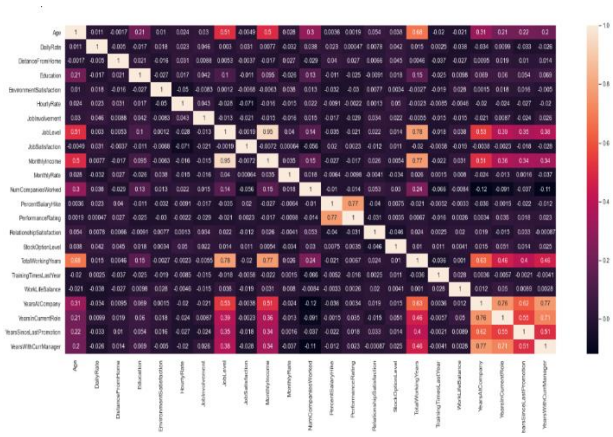

Figure 4: Heatmap showing the highly correlated coefficients between the columns.

After plotting a heatmap, considering the correlation coefficients 0.70 to 0.95 as highly correlated. There are few columns which are highly correlated other than themselves.

1. Job level<--->Monthly Income are correlated with 0.95
2. Performance Rating<--->Percent Salary hike are correlated with 0.77
3. Total Working Years<--->Job Level are correlated with 0.78
4. Total Working Years<--->Monthly Income are correlated with 0.77
5. Years in Current Role<--->Years at Company are correlated with 0.76
6. Years with Current Manager<--->Years at company are correlated with 0.77
7. Years with Current Manger<--->years in Current Role are correlated with 0.71

The columns removed are Job Level, Total Working Years, Performance Rating, Years with Current Manager and Years in Current Role as they have a high correlation coefficient value. A new heatmap is shown below

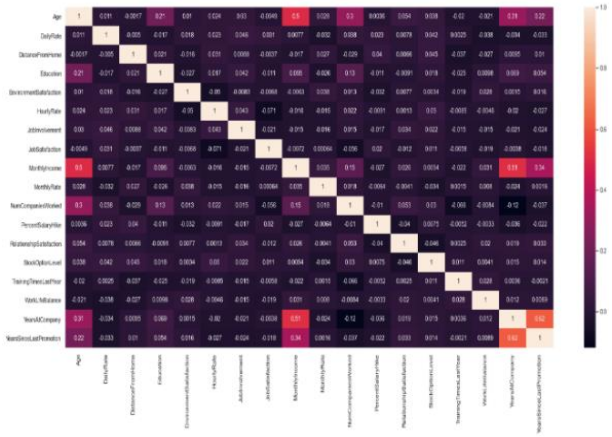

Figure 5: Heatmap after removing the highly correlated columns.

Feature Scaling

After correlation, encoding is done to the data. **Label Encoder** is applied for the categorical features and coverts them in to 0 or 1 and **Standard Scaler** is applied for numerical features to resize the distribution of values so that mean of the observed value is 0 and standard deviation is 1.

```
In [243]: le = LabelEncoder() #Label encoder is stored in le
          for feature in categorical_features:
              empatt[feature] = le.fit_transform(empatt[feature]) #fitting the le in Categorical features

In [244]: sc = StandardScaler() #standard scaler is stored in sc
          for feature in numerical_features:
              empatt[feature] = sc.fit_transform(np.array(empatt[feature]).reshape(-1,1)) #fitting the sc in num
```

Figure 6: Screenshot of Jupyter notebook performing feature scaling.

Values in the data will look like below after encoding the data.

```
246]: empatt.head() #printing the first 5 rows after encoding
```

246]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	MonthlyRate
0	0.446350	1	2	0.742527	2	-1.010909	1	1	1	0	...	0.726020
1	1.322365	0	1	-1.297775	1	-0.147150	0	1	2	1	...	1.488876
2	0.008343	1	2	1.414363	1	-0.887515	1	4	3	1	...	-1.674841
3	-0.429664	0	1	1.461466	1	-0.764121	3	1	3	0	...	1.243211
4	-1.086676	0	2	-0.524295	1	-0.887515	0	3	0	1	...	0.325900

5 rows × 26 columns

Figure 7: Screenshot of data after encoding.

Splitting the data into Training and Testing data

6.8 Train and test data split

```
[247]: X = empatt.drop("Attrition",axis=1) #independent variable
       y = empatt["Attrition"] #dependent variable

[248]: #Importing train_test_split from sklearn library
       from sklearn.model_selection import train_test_split
       X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1) #Here data is split into 70% train data and 3

[249]: #printing the shape of the variables X_test, y_test, X_test, y_test
       print(X_train.shape)
       print(y_train.shape)
       print("----------------------")
       print(X_test.shape)
       print(y_test.shape)

       (1029, 25)
       (1029,)
       ----------------------
       (441, 25)
       (441,)
```

Figure 8: Screenshot of splitting into train and test data.

Here, the data is split into 70% train data and 30% test data. After splitting the data, we must ensure that the dependent target train data is balanced. To identify that printed all the values '0' & '1' in the y_train data and the result is '0': 869 & '1': 160. This is now taken as imbalanced data. With the imbalanced data the models could predict incorrect accuracies and that will lead the models to a failure side.

Resolving the imbalance problem with SMOTE

To resolve the imbalance problem in the data, SMOTE (Synthetic Minority Oversampling Technique) is used. It is clearly shown in the application. After oversampling the y_train data, the values '0' & '1' are now balanced.

```
In [255]: #printing the values i.e; 0 and 1 of the oversampling_y_train after the oversampling
          print("After OverSampling, counts of label '1': {}".format(sum(oversampled_y_train == 1)))
          print("After OverSampling, counts of label '0': {}".format(sum(oversampled_y_train == 0)))

          After OverSampling, counts of label '1': 869
          After OverSampling, counts of label '0': 869

          Now, the oversampled_y_train data is balanced.
```

Figure 9: Screenshot of the data after solving the imbalance problem.

A pie chart is also plotted to show the balanced trained target values. It is shown below

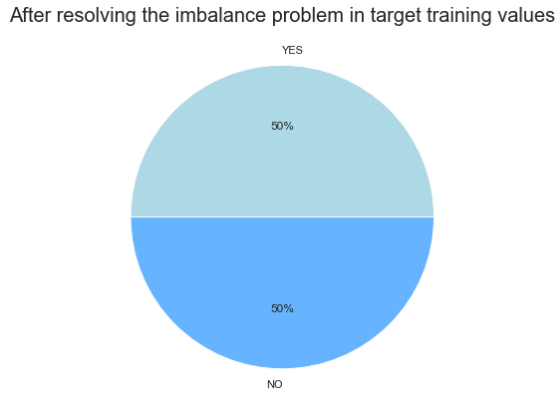

Figure 10: Screenshot of pie chart after using the SMOTE.

EXPERIMENTS

For the analysis and prediction of IBM employee attrition considered 7 classification algorithms which include Logistic Regression, XGBoost Classifier, Decision Tree, Support Vector Machine, AdaBoost Classifier, Random Forest, Gradient Boosting. These are like straightforward algorithms which are easy to implement. These are chosen because they usually give us the best accuracy results with categorical variables and the dependent variable 'Attrition' contains only categorical values.

LOGISTIC REGRESSION

The method of modelling the probability of a discrete result given an input variable is known as Logistic Regression. Logistic Regression is great entry level model which is used on binary and categorical variables. In this research used this model to predict the dependent variable "Attrition" i.e; whether the employee will leave the organization or not?

XGBOOST CLASSIFIER

XGBoost is a scalable and highly accurate variant of gradient boosting that pushes the limits of computing power for boosted tree algorithms. It was designed primarily to increase machine learning model performance and computational speed. XGBoost has a built in distributed weighted quantile sketch algorithm which makes the algorithm easier and effectively work on the data. This model is applied on the data expecting the best accuracy results.

DECISION TREE

Decision Tree is a supervised machine learning model where the model predicts the value of a target value where the data is continuously split according to certain parameter. It incrementally cuts down a dataset into smaller and smaller sections while also developing an associated decision tree. A tree with decision nodes and leaf nodes is the end result. Decision tree doesn't promise accuracy but it promises interpretability.

SUPPORT VECTOR MACHINE

Support is a supervised machine learning algorithm used for classification problem. SVM categorises data points by mapping it to a high-dimensional feature space, even while the input is not otherwise linearly separable. Then it will be able to train the labelled data which is provided to the algorithm. SVM is effective in high dimensional spaces and it is memory efficient.

ADABOOST CLASSIFIER

AdaBoost is also called as Adaptive Boosting. Adaptive Boosting is a Machine Learning technique that is used as part of an Ensemble Method. It builds a model and make predictions and it again assigns some weights to it and builds a model again. It will repeat these steps and at last it uses the weighted average of individual models.

RANDOM FOREST

Random Forest is an unsupervised machine learning which is used for classification problems. A random forest algorithm consists of many decision trees. It is also one of the most as it is easy to use and a diversity in it. More number of trees in the forest leads to higher accuracy and prevents from overfitting in the model. It takes less training time than any other model. Results will come with better accuracy than decision tree.

GRADIENT BOSSTING CLASSIFIER

Gradient Boosting is used for predicting of a categorical target variable. Gradient Boosting is based on the assumption that when the best potential next model is coupled with prior models, the overall prediction error is minimised. Gradient Boosting Models will keep improving in order to reduce all errors. This can lead to overfitting by exaggerating outliers.

RESULTS

After applying the above machine learning models on the data, performance metrics of each algorithm are analysed. The model performance is determined based on these performance metrics. The metrics used in this research are:

Accuracy: It is a measurement which is used to determine which model is best at identifying the relationships between the train and test samples.

Precision: Precision is a value in machine learning which measures the quality if the product. It is formulated by number of true positives divided by number of true positives + number of false positives.

Recall: Recall is a metric that quantifies the number of true positive predictions made out of all positive ones.

F1-Score: This is the mean of precision and recall values.

Classification results are presented below in a table:

Accuracy and ROC AUC Scores

MODELS	Accuracy	ROC AUC Score
Logistic Regression	0.764172	0.7035464535464536
XGBoost Classifier	0.841270	0.6683316683316683
Decision Tree	0.727891	0.6098901098901099
Support vector Machines	0.759637	0.6291208791208791
AdaBoost Classifier	0.807256	0.7450049950049951
Random Forest	0.820862	0.6559690309690309
Gradient Boosting Classifier	0.807256	0.7450049950049951

Figure 11: Table showing the accuracy and ROC AUC scores of all the models.

ROC curve is plotted after calculating the fpr, tpr values of the models individually and the figure is can be seen beside.

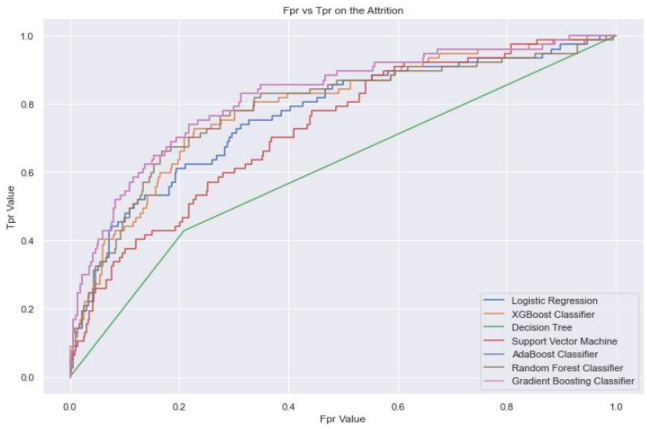

Figure 12: Screenshot of ROC curve of all the models.

AdaBoost Classifier Gradient Boosting classifier gave us the similar ROCAUC score of 0.7450049950049951. The curve which is closed to top left is determined as best one. Gradient boosting and AdaBoost classifier is considered as the best one.

Visualizing the confusion matrix

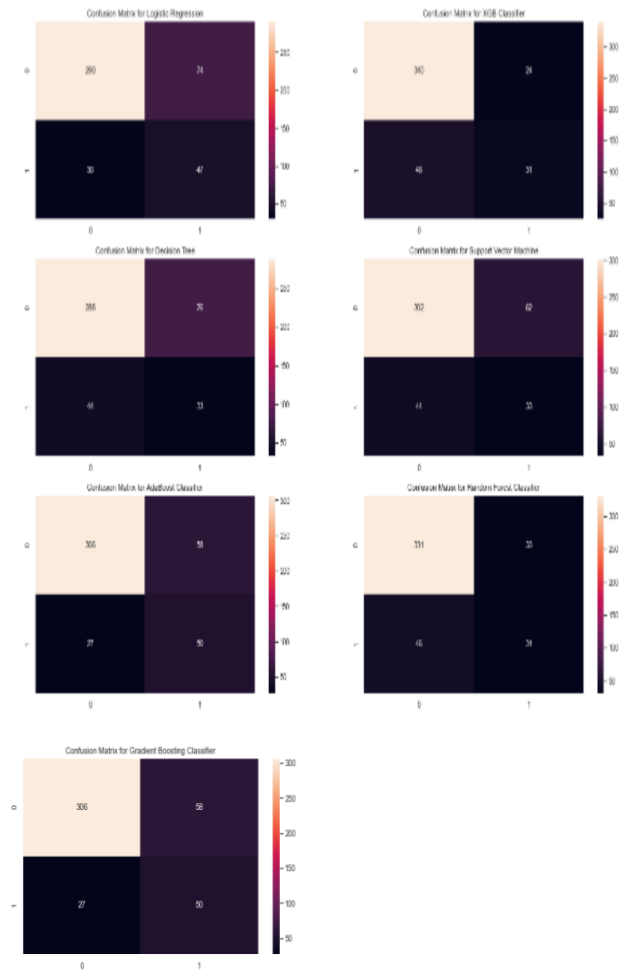

Figure 13: Screenshot of confusion matrix of all the algorithms.

Note: From the above, models it is observed that XGBoost Classifier, Random Forest, AdaBoost and Gradient Boosting classifier are well performed, based on the accuracy scores. The highest accuracy is obtained from XGBoost classifier of 0.841270. But when it comes to ROC curve Gradient and AdaBoost classifiers gave us the best scores in that.

DISCUSSIONS, CONCLUSIONS AND FUTURE WORK

This research paper discusses the analysis and attrition of the employees of company IBM. A literature review is done in this paper which discusses topic related journals and conference papers and what they achieved. This paper studied the importance of using ML algorithms in predicting the employee attrition in the company IBM. This research found that the XGBoost Classifier is the best one as its accuracy is 84%. Along, with this Random Forest, AdaBoost classifier and Gradient Boosting classifier also performed very well. Hence it is suggested that the company should make some interesting offers and incentives to the employees who want to leave and prevent them from leaving the organization. In the future, this research study can be used by Multi-National Companies in order to predict the attrition of their employees.

REFERENCES

IBM HR Analytics Employee Attrition & Performance in www.kaggle.com https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Budhwar P S, Bhatnagar J. Talent management strategy of employee engagement in Indian ITES employees: key to retention[J]. Employee relations, 2007.

What is an Employee Attrition? Reasons and factors behind it? https://www.toolbox.com/hr/engagement-retention/articles/what-is-attrition-complete-guide/

Mansor, N., Sani, N.S. and Aliff, M. (2021) 'Machine Learning for Predicting Employee Attrition', *International Journal of Advanced Computer Science and Applications*, 12(11), pp. 435-445. doi: 10.14569/IJACSA.2021.0121149.

R. Chakraborty *et al.* (2021) 'Study and Prediction Analysis of the Employee Turnover using Machine Learning Approaches', - *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON).* doi: 10.1109/GUCON50781.2021.9573759.

Saradhi, V.V. and Palshikar, G.K. (2011) 'Employee churn prediction', *Expert Systems with Applications*, 38(3), pp. 1999-2006. doi: https://doi-org.ezproxy.tees.ac.uk/10.1016/j.eswa.2010.07.134.

Ajitesh Kumar (2022), Correlation Concepts, Matrix and Heatmap using Seaborn, https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/#:~:text=A%20correlation%20heatmap%20is%20a,necessarily%20imply%20a%20causal%20relationship.