Correlation of employee attrition rate and its interpretability using workforce analysis

ECE-579- Intelligent systems - Instructor. Yi Lu Murphey

by
Pooja Alumalla (apooja@umich.edu) – MS Data Science (CIS)

Sathya Narayanan Thothathri (sathyasn@umich.edu) – MS Data Science (CIS)

Harsha Vardhan Sai Machineni (hvm@umich.edu) – MS Data Science (CIS)

Abstract

This study addresses employee attrition using machine learning models, comparing eight techniques and designing a custom model. With the IBM HR dataset and an open-source dataset, preprocessing and parameter tuning were applied. The ensemble XGBoost-Random Forest model emerged as the best predictor of employee attrition among the tested classifiers.

Keywords: — Predictive analysis, employee attrition, machine learning, ensemble model

Introduction The pandemic has transformed the IT industry, with industries struggling to retain and hire talent due to market volatility and changing work-life balance ideologies. This project aims to identify these trends, helping companies retain employees amidst predictions of a recession. By studying selective job features across multiple machine learning models, we can better understand their functionality. The project benefits HR departments by projecting trends and suggesting strategies for employee retention and hiring. Our 4400-record dataset combines IBM HR analytics and a Kaggle open-source dataset with 29 features. We compared eight machine learning models and created a custom ensemble model of XGBoost and random forest, which proved to be the best performer.

Problem Statement

The objective is to predict the attrition at a workplace and analyze the key factors which would help reduce the attrition rate using analytics.

Related Work

- Springer Switzerland study uses SVM, Decision Tree, and Random Forest algorithms on a dataset with 10 features and 14,000+ records to predict employee attrition. Metrics used are precision, recall, F1 score, and accuracy. Decision Tree and Random Forest perform best.

- arXiv paper explores voluntary attrition and uses Naïve Bayes, KNN, Multi-layer Perceptron, and Logistic Regression on a dataset with 7 numerical, 2 categorical features, and 14,999 records. Evaluation methods include ROC-AUC curve and F1 score. KNN classifier proves superior.

- ResearchGate – MDPI analyzes objective factors influencing attrition using Gaussian Naive Bayes, Naive Bayes, Logistic Regression, KNN, Decision Tree, Random Forest, SVM, and LSVM on an IBM dataset with 35 features and 1,500 samples. Metrics include precision, recall, accuracy, F1 score, ROC curve, and AUC. Naïve Bayes has the best recall rate, highlighting monthly income, age, overtime, and distance from home as key attrition variables.

Background Concepts(/Knowledge) (/Preliminary) (Optional)

Attrition refers to the departure of employees from the workplace due to resignation, retirement, death, or any other reasons. There are two types of attrition namely – Voluntary attrition and Involuntary attrition.

Voluntary attrition is when an employee voluntarily departs from the firm due to personal reasons or resignation.

Involuntary attrition is when the employee departs from the firm due to reasons such as retirement, death or if the employee was fired.

Data

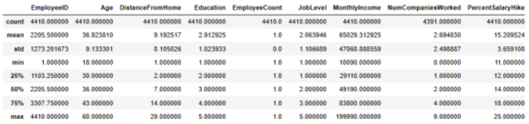
The dataset used is a combination of the IBM HR analytics dataset and a dataset from Kaggle

It contains about 4400 records with 29 features in it.

Data exploration

We applied the below data exploration methodologies to gain insights:

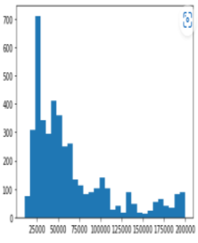
First, we found out some of the basic metrics associated with the dataset as seen.



	EmployeeID	Age	DistanceFromHome	Education	EmployeeCount	JobLevel	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike
count	4410.000000	4410.000000	4410.000000	4410.000000	4410.0	4410.000000	4410.000000	4391.000000	4410.000000
mean	2205.500000	36.923810	9.192517	2.912925	1.0	2.063946	65029.312925	2.694830	15.209524
std	1273.201673	9.133301	8.105026	1.023933	0.0	1.106689	47068.888559	2.498887	3.659108
min	1.000000	18.000000	1.000000	1.000000	1.0	1.000000	10090.000000	0.000000	11.000000
25%	1103.250000	30.000000	2.000000	2.000000	1.0	1.000000	29110.000000	1.000000	12.000000
50%	2205.500000	36.000000	7.000000	3.000000	1.0	2.000000	49190.000000	2.000000	14.000000
75%	3307.750000	43.000000	14.000000	4.000000	1.0	3.000000	83800.000000	4.000000	18.000000
max	4410.000000	60.000000	29.000000	5.000000	1.0	5.000000	199990.000000	9.000000	25.000000

Fig1-Basic metrics obtained from the raw dataset

we performed Univariate and Bivariate analysis to gain more insights.



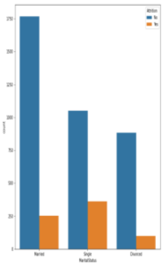


Fig2a- Univariate analysis of the correlation of Monthly Income and their bins

Fig2b- Bivariate analysis of the correlation of Marital Status and Attrition

Later, we went on to analyze each feature and its possible relationship with the most relevant feature which could possibly correlated.

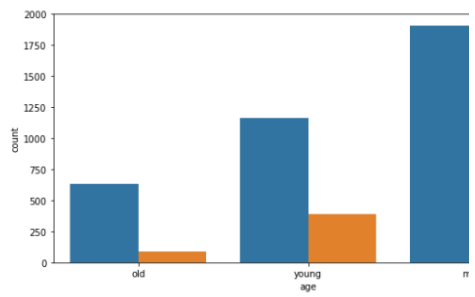


Fig2c- Bivariate analysis of the correlation of Age and Attrition

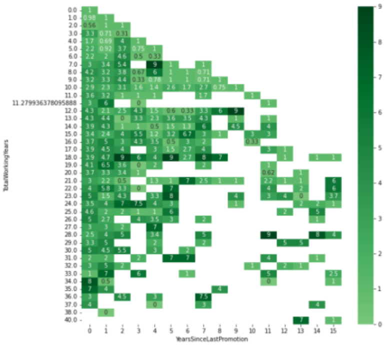


Fig3- Heatmap of the correlation between Total Working Hours and Years since Promotion

Data preprocess

For our dataset, we firstly performed data cleansing to handle missing values, null values. Post that, performed data imputation by imputing the missing values with the mean of the dataset and dropped unnecessary columns.

We then used the 'Binning' technique for the features 'Age' and 'Income' to convert them to categorical features.

age

Bining Age into 3 categories

- (17.958, 32.0] : young
- (32.0, 46.0] : middle
- (46.0, 60.0] : old

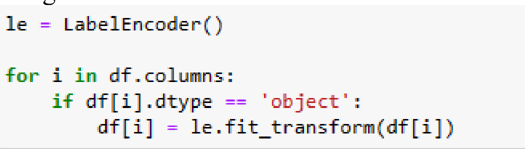
Encoding

Converting Gender to binary values for the ease of processing

- 'Female' : 0
- 'Male' : 1

Data preparation for the machine learning model

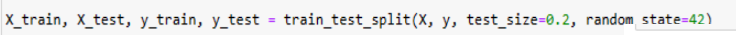
We used the Label Encoder to encode the categorical features in the dataset.



```
le = LabelEncoder()

for i in df.columns:
    if df[i].dtype == 'object':
        df[i] = le.fit_transform(df[i])
```

Fig4- Label Encoding

The dataset was then split into two parts – training and testing set each with a ratio of 80 and 20 respectively.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig5-Dataset Split

Methodology (/Proposed Methods/ Approach/ Procedure)

We used two methods for obtaining the solution –
1. Comparative study of 8 machine learning models
and 2. Custom model using the ensemble technique.

A. *Method 1 - Comparative study of 8 machine learning models*

Below is Brief summary of each model studied:

This study examined eight models:

a. XGBoost - an efficient open-source gradient boosted trees algorithm implementation, often used for accurate prediction in supervised learning.

b. Random Forest - a supervised algorithm for classification and regression, building decision trees on samples and using majority vote or average.

c. Decision Tree - a non-parametric supervised method for classification and regression, predicting target variables through simple decision rules.

d. Logistic Regression - a statistical method predicting binary outcomes through linear relationships.

e. Support Vector Machine - a linear model for classification and regression, creating lines or hyperplanes separating data into classes.

f. Perceptron - a linear binary classifier dividing data into two parts.

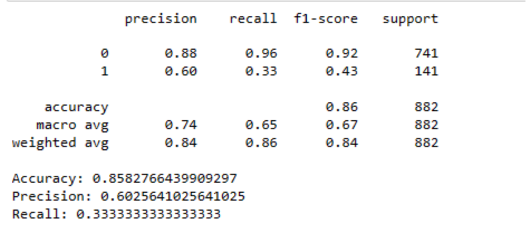
g. Naïve Bayes - a probabilistic classifier predicting based on object probabilities.

h. KNN - a non-parametric supervised algorithm classifying objects according to their neighbors' classes.

Post the comparative study of the models, we inferred that the top performing models were XGBoost, Random Forest and Decision Tree Classifier. The boosting algorithms performed better than the classification algorithms.

B. *Method 2 – Custom Ensemble models*

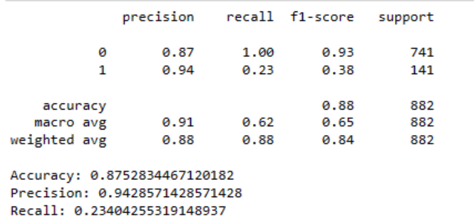
Adaboost with Grid Search CV and k-fold cross validation - adaptive boosting adjusts eights for misclassified cases.



```
              precision    recall  f1-score   support

           0       0.88      0.96      0.92       741
           1       0.60      0.33      0.43       141

    accuracy                           0.86       882
   macro avg       0.74      0.65      0.67       882
weighted avg       0.84      0.86      0.84       882

Accuracy: 0.8582766439909297
Precision: 0.6025641025641025
Recall: 0.3333333333333333
```

Fig6-Metrics of the Adaboost model

XGBoost + Random Forest (XGBRF Classifier) with Grid Search CV and k-fold cross validation - combines top-performing models for improved accuracy (87%).



```
              precision    recall  f1-score   support

           0       0.87      1.00      0.93       741
           1       0.94      0.23      0.38       141

    accuracy                           0.88       882
   macro avg       0.91      0.62      0.65       882
weighted avg       0.88      0.88      0.84       882

Accuracy: 0.8752834467120182
Precision: 0.9428571428571428
Recall: 0.23404255319148937
```

Fig7-Metrics of XGBRF Classifier

Experiment

We experimented with ensemble models, using 80% of the data for training and 20% for testing. Models were validated with k-fold cross-validation and underwent hyperparameter tuning using Grid Search CV. Of the ten prediction models designed, the XGBRF ensemble model performed the best.
In total, we designed 10 prediction models and inferred that the ensemble model of XGBRF performed the best amongst them.

Model Evaluation

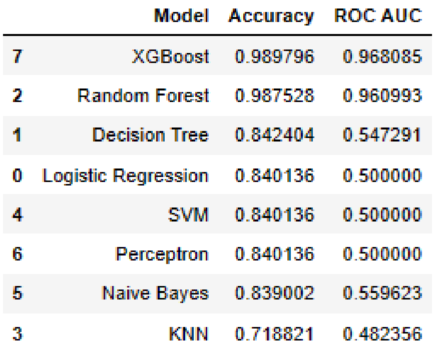
We used several model evaluation techniques such as cross validation, F1 measure from the confusion matrix and ROC-AUC curve.

Cross validation -it is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.

F1 measure- An F-score is the harmonic mean of a system's precision and recall values.

ROC-AUC curve - The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.
The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

Model comparison of the 8 models is seen below:



	Model	Accuracy	ROC AUC
7	XGBoost	0.989796	0.968085
2	Random Forest	0.987528	0.960993
1	Decision Tree	0.842404	0.547291
0	Logistic Regression	0.840136	0.500000
4	SVM	0.840136	0.500000
6	Perceptron	0.840136	0.500000
5	Naive Bayes	0.839002	0.559623
3	KNN	0.718821	0.482356

Fig8a-Model result comparison

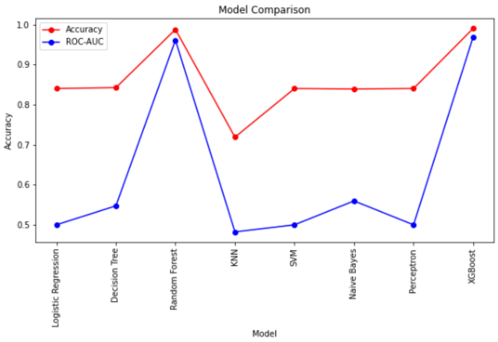


Fig 8b – Model comparison graph

Result comparison of the ensemble models:

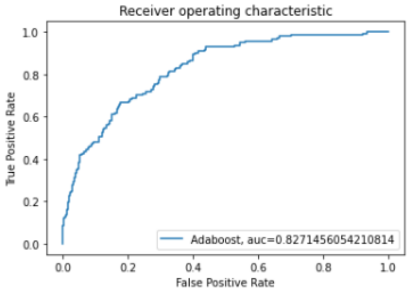


Fig9a-ROC curve for Adaboost

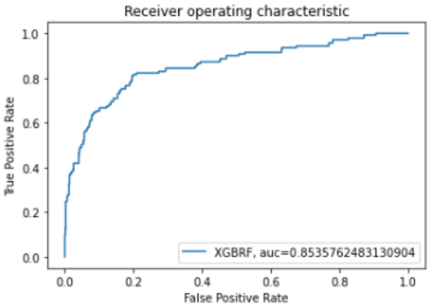


Fig9b-ROC curve for XGBRF

In Fig9a we can see that the curve for Adaboost is leaning more towards the center of the graph which shows it has room for improvement. It's AUC value is 0.82 which is slightly less closer to the value 1.

Whereas in Fig18b we can see that the curve for XGBRF is hugging the Y axis which shows that it performs better. Its AUC value is 0.85 which is closer to the value 1 hence making it a better model.

Conclusion

This study demonstrated the XBRF classifier's superior accuracy and predictive effectiveness using the ROC curve. Despite dataset noise, the optimally configured model delivers precise results, aiding HR in employee retention efforts. The dataset represents the general workforce well, and results from various classifiers confirm the relevance of selected features. However, the dataset may not apply to all industries, as certain features might not be relevant for specific tasks. High dimensionality presents scalability challenges, which can be addressed using techniques like PCA. Future work could focus on identifying key attrition factors and refining hyperparameter tuning to enhance model efficiency and scalability.

References

[1] Praphula Kumar Jain, Madhur Jain, Rajendra Pamula "Explaining and predicting employees' attrition: a machine learning approach" , Springer Nature Switzerland AG 2020

[2] Francesca Fallucchi , Marco Coladangelo ,Romeo Giuliano ,and Ernesto William De Luca " Predicting Employee Attrition Using Machine Learning Techniques" , MDPI

[3] Norsuhada Mansor , Nor Samsiah Sani, Mohd Aliff "Machine Learning for Predicting Employee Attrition", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 11, 2021

[4] Rahul Yedida, Rahul Reddy , Rakshit Vahi, Rahul J, Abhilash, Deepti Kulkarni, "Employee Attrition Prediction"

[5] Cotton, J.L. and Tuttle, J.M., 1986. "Employee turnover: A meta analysis and review with implications for research" Academy of management review, pp.55-70.

[6]. Deshpande, V., & Golhar, D. Y. (1994). Human resource management systems: a structured review. The International Journal of Human Resource Management, 5(1), 69-90.

[7]. Kakar, U., & Pandey, V. (2017). Employee attrition prediction using machine learning: A systematic literature review. 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 1-5.

[8]. Liu, W., Zeng, Q., Chen, Z., & He, Q. (2021). A Comparative Study of Machine Learning Algorithms for Employee Attrition Prediction in High-Tech Industries. IEEE Access, 9, 107468-107480.

[9]. Sabir, M. A., & Arshad, S. (2019). Predicting employee attrition: a machine learning approach. International Journal of Advanced Computer Science and Applications, 10(4), 153-159.

[10]. Wu, C. H., & Lee, Y. H. (2020). The role of HR analytics in enhancing organizational performance: The mediating role of employee retention. Technological Forecasting and Social Change, 158, 120158.

Responsibilities

Pooja Alumalla (data gathering and analysis, data cleansing and structuring)
Email: apooja@umich.edu

Sathya Narayanan Thothathri (data modelling, evaluation, data preparation ML algorithms and vizualization)
Email: sathyasn@umich.edu

Harsha Vardhan Sai Machineni (UI building, Deployment)
Email: hvm@umich.edu