DATA 1030 Midterm Report
Subtitle

Zhangqi Liu
Brown Data Science Initiative
October 20, 2020
https://github.com/zhangqi-liu

INTRODUCTION

Attrition (the rate at which employees leave a company) has a substantial impact on a company's productivity, efficiency, and operation rate. Nowadays it is more common for people to change their job more frequently, especially during and following the COVID-19 pandemic. This results in a lot of pressure on start-up companies since it puts them at risk of losing vital data and revenue. Therefore, it is necessary to conduct analysis on employee attrition to understand what causes employees to search for new job opportunities. The dataset used in the project is from Kaggle, and we will use the third edition maintained by IBM. This dataset takes into account work-related measures regarding staff who quit their company. By analyzing the data, this project aims to identify the factors most related to employee attrition, as well as determine whether it can be predicted that an employee will leave based on their information.

This dataset consists of a total of 1470 data points and 32 features (columns). There were 35 columns in the data, but 3 columns were deleted due to the zero-variance check. The target variable 'Attrition' is marked as either 'Yes' or 'No'. As a result, this is a classification problem. Most features in this dataset are ordinal variables. The features consist with employee's personal information, their specific information in their company, and their work-life balance condition. There is a more specific features overview in my github. Many of the variables in this dataset have a range from 1-5. In this case, the lower the ordinal variable, the worse it is. For Example, Job Satisfaction 1 = "Low", while 4 = "Very High".

In the literature research, all publications use classification to predict the attrition of the employee. Since the data is imbalanced, some of the publications use SMOTE to oversample the data, while others use Stratified Sampling. One project uses Random Forest and a Gradient Boosting classifier, taking less than a minute to run and returning an 89% accuracy and 51% f1-score on its predictions. Another project, which is Kaggle's best competition prediction project, also uses Random Forest. Their results of this dataset has accuracy=85%, precision=57%, recall=49%, and f1-score=53%. In my project, f1-score will be used as the evaluation metric because of the imbalance of the data, and I expect to have a f1-score greater than 53%. By using Random Forest, these two projects found employee's age, income, tenure and job history with the company were the most influential predictors of employee attrition.

Exploratory Data Analysis

For the exploratory data analysis process, I checked the overall distribution, conducted zero-variance Checking, Missing and Duplicate Value Checking, and Bivariate Analysis by using the summary statistics and visualization. Three columns were deleted due to the zero-variance.

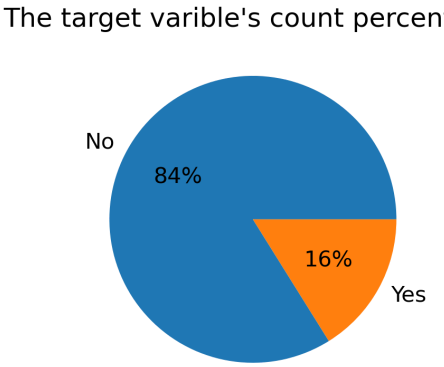

Figure1: Target variable count percentage

By calculating the target variable's value count, we could find out that 1237 (84%) observations choose not to leave the company, showing that the dataset is imbalanced.

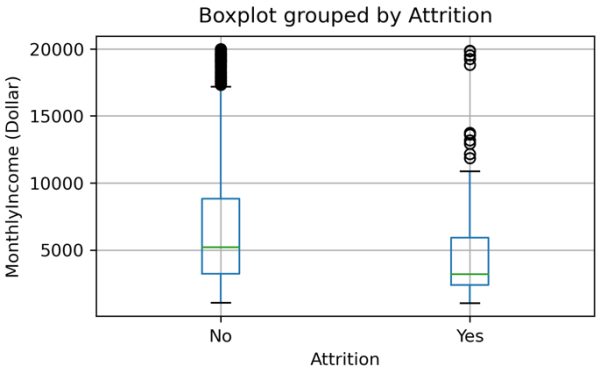

Figure 2: Boxploy grouped by Attrition

The above box plot shows the distribution of different attrition choices with their monthly income. By observing the plot, we found that the monthly incomes of the employees who chooses to quit have a lower mean, 25% quantile, 75% quantile and maximum value. From the above plot, we could make the assumption that the employee who choose to leave their job have an overall lower monthly income.

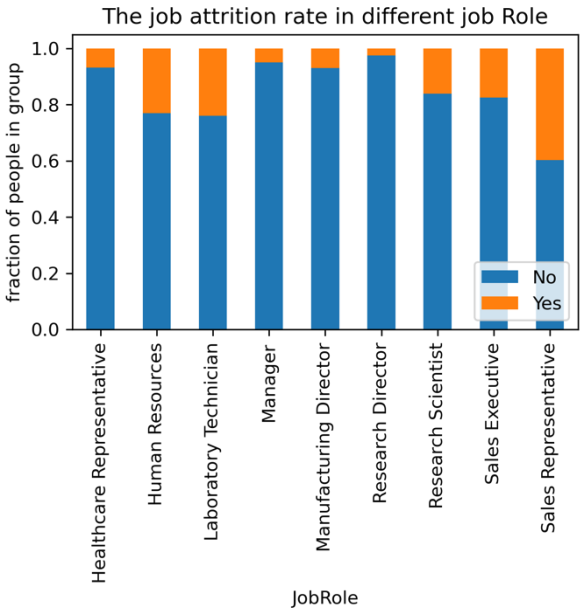

Figure 3: The job attrition rate in different job role

The above box plot shows the job attrition rate in different job roles. By observing the plot, we found that different job roles have different attitudes toward attrition. The roles of 'Sales Representative', 'Human Resources' and 'Laboratory Technician' are the three positions with the highest turnover rate.

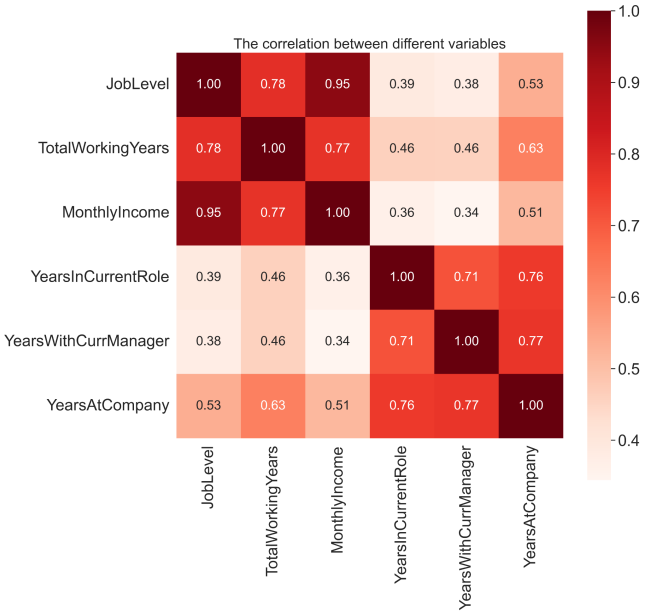

Figure 4: The correlation map

This correlation matrix heat map shows us the linear relationship between predictor variables that are of continuous type. Because the original image is too large, I picked out the feature with high correlation to plot this graph. In this graph, the darker the red is, the stronger the correlation between each variable. This shows that 'YearWith currentManager', 'YearsInCurrentRole', and 'YearsAtCompany' are all highly positively correlated, but they do not have a significant influence on other variables. Similarly, 'JobLevel', 'MonthlyIncome', and 'TotalWorkingYears' are also highly correlated with each other.

3. Methods:

3.1 Data Splitting and Pre-Processing:

During the EDA, we deleted several useless features including constant variables (’EmployeeCount’, ’Over18’ and ’StandardHours’), and we do not have any missing value,

The dataset is Independent and Identically Distributed (IID) data due to the dataset consisting of independent data points which are identically distributed. There is no group structure in this data set, and it is also not a time-series data set.

The original dataset is imbalanced with respect to the target variable Attrition, and there are only 1470 data. As a result, the stratified K-fold splits were conducted. I first split 20% for test set, then use 5 fold to split the training set and cross-validation set.

A pipeline that consisted of StandardScaler, OneHotEncoder, and OrdinalEncoder was conducted during the encoding process. Since there are several categorical features such as ‘JobRole’, I needed to use the OneHot encoding technique to transform them into dummy variables. I used the StandardScaler for the numerical features that are normally distributed and upper or lower bounder are not well known. The OrdinalEncoder was used for the ordinal features that are a categorical variable for which the possible values are ordered, such as ‘Education’, ‘JobInvolvement, and ‘JobSatisfaction’. After this process, there were 46 features and 1470 data points left.

3.2 Model Training & Hyperparameter Tuning:

Six different random states were used in the development of my machine learning pipeline, because train-test splits are always non-deterministic. From the figure below, we could determine that Random Forest, Ridge, and SVC have the highest standard deviation scores, which means that they have larger uncertainty when going through the splitting process. After preprocessing the data, the input models will be trained by using the training data, and then GridSearchCV used for identifying the optimal parameter values from a given set of parameters in a grid with 5-fold cross validation. After obtaining the best models, a transformed test set is used to calculate the F-score. The reason for choosing F_1-score is that the dataset is imbalanced, and F_1 is appropriate here because it is calculated as the harmonic mean of both precision and recall for the minority positive class.

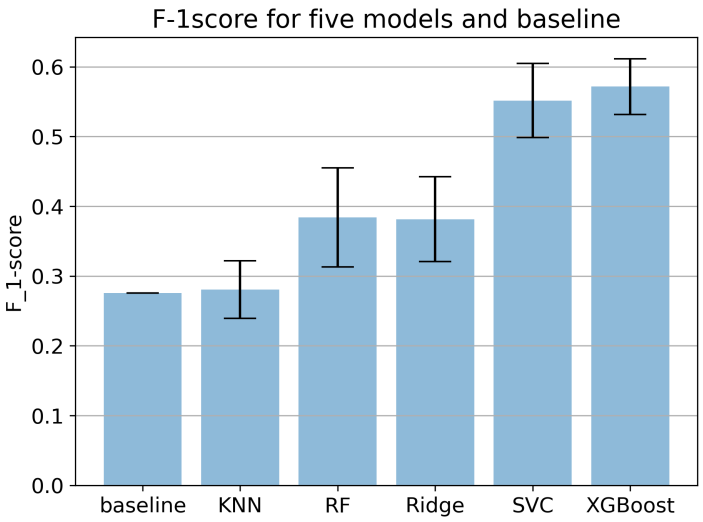

Figure 6: F1 score for five models and baseline

By using the designed machine learning pipeline, five machine learning models were trained and tested together with a baseline model for comparison: a logistic regression with L2 regularization (Ridge), a RandomForestClassifier, an XGBoost classifier, a K-nearest neighbors classifier, and a support vector machine classifier. Below are the parameters and values which were tested.

Model	Parameter
a logistic regression with L2 regularization (Ridge)	alpha: [1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2,1e-3]
RandomForestClassifier	max_depth: [15,30,50], max_features: np.arange(0.1, 1, 0.2)
XGBoost classifier	learning_rate: [0.03,0.1,0.3], seed: [0], max_depth: [1,2], colsample_bytree: [0.4,0.9], subsample: [0.66], gamma:[0.4, 0.75,1], eval_metric: ['logloss']
K-nearest neighbors classifier	n_neighbors: np.arange(1,5,1), weights: ["uniform", "distance"]
Support vector machine classifier	C: [1 , 10 , 20 , 30 , 40 , 50], kernel: ['linear', 'poly', 'rbf','sigmoid'], degree: [1,2,3,4,5]

Figure 5: Table showing parameters and values for parameter tuning

4. Results:

4.1 Model Performance Comparison and Selection:

The model with the optimal parameter value is fitted on the test set and used to calculate the test score for each random state. The average F_1-score and standard deviation for each round of hyperparameter tuning is calculated as shown in Figure 6, which indicates that the XGBoost model has the highest F_1-score. The best learning_rate=0.03, n_estimators=5000, max_depth=1, subsample=0.66, and gamma=0.75, along with other default parameter settings.

The baseline F_1 score in this dataset is 0.276. The F_1 score of the best model classifier (XGBoost) is 0.558, with standard deviation 0.034.The XGBoost classifier is 8.12 standard deviation above the baseline.

4.2 Feature Importance and Interpretation:

In order to verify test set predictions and inspect the model, the feature importance is introduced. The first global importance is the permutations importance. From the figure below, we could observe that ‘OverTime’, ‘MonthlyIncome’, and ‘EnvironmentSatisfaction’ are the three factors which are most relevant to the prediction.

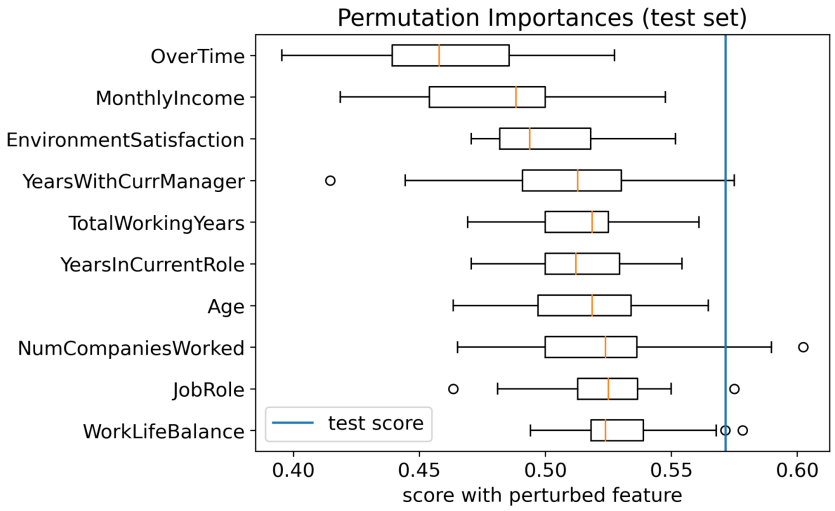

Figure 7: Top 10 permutation importance

Another global feature importance which could be used is the important features that are generated by the built-in feature importance method for tree-based algorithms. By using the ‘Weight’, the feature importance is generated based on the number of times a feature is used to split the data across all trees. In this case, ‘std_MonthlyIncome’,

‘std_DailyRate’, and ‘std_NumCompaniesWorked’ have the three highest feature importance.

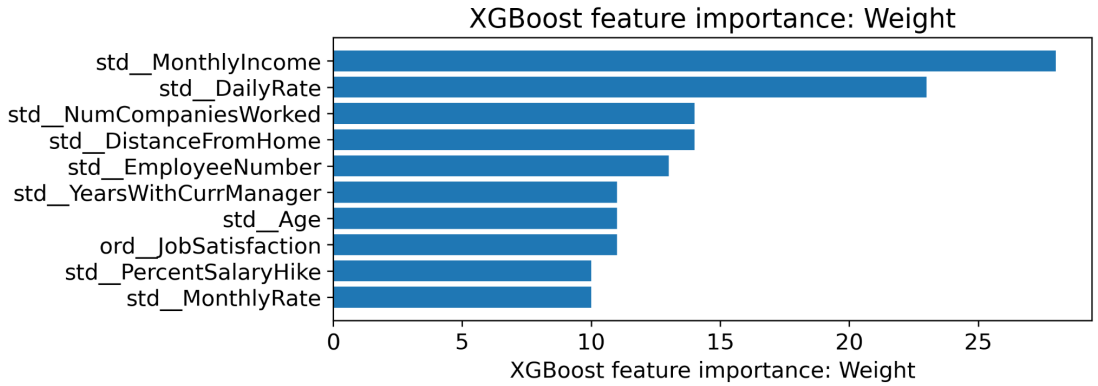

Figure 8: XGBoost feature importance: Weight

The global feature importance can also be shown in Figure 9 by applying SHAP values. The most relevant features are ‘onehot_OverTime_Yes’, ‘std_MonthlyIncome’, and ‘std_NumCompaniesWorked’. We could also find that people who work overtime are more likely to quit their job, and people who do not work overtime are more likely to stay in their job.

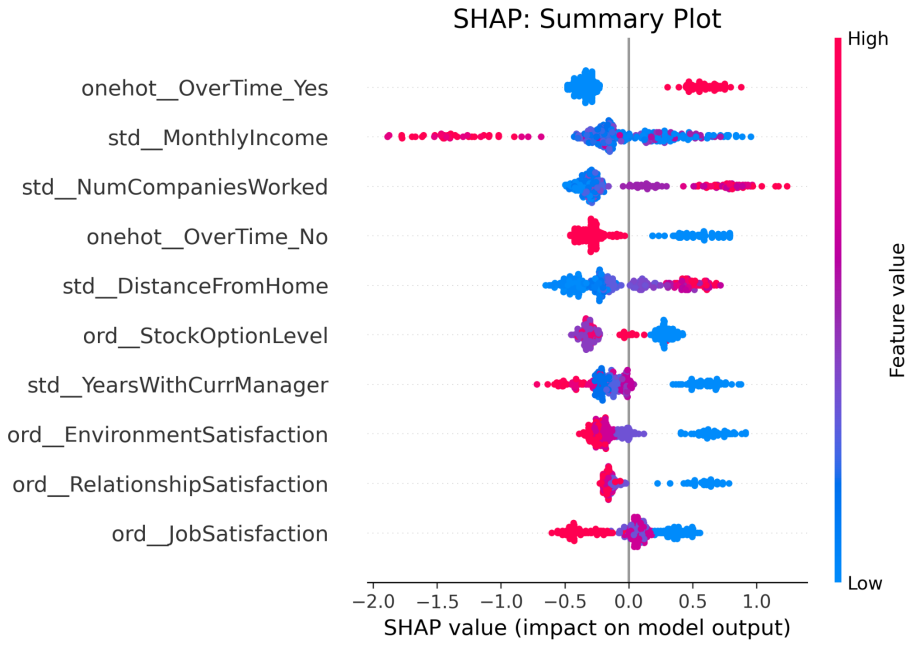

Figure 9: SHAP Global feature importance

In the XGBoost feature importance, we could find out that ‘onehot_EducationField_Medical’ , ‘onehot_JobRole_Sales Executive’, and ‘onehot_MaritalStatus_Married’ are the least important features.

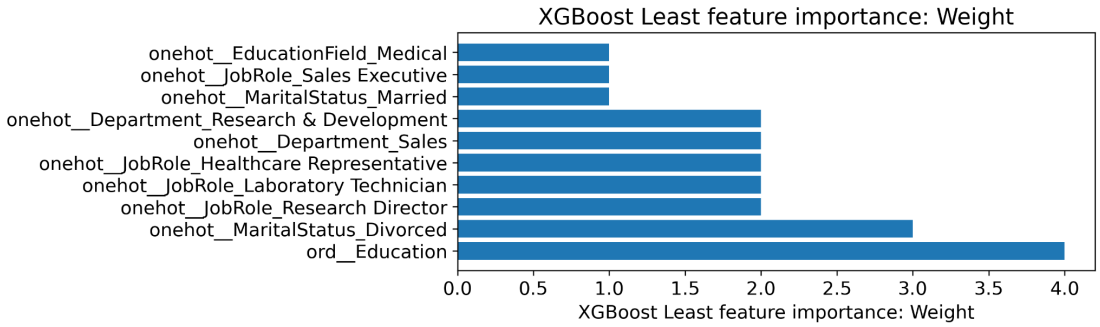

Figure 10: XGBoost feature importance: Least Important

Moreover, the SHAP is also used to calculate the local feature importance. The Force plot shows the influence of each feature on the current prediction. In Figure 11, The base value is 0.16, which is the average predicted probability across all samples. The red arrows represent the values of these features that have a positive influence on the prediction. When 'ord__EnvironmentSatisfaction' = 0 and 'ord__JobSatisfaction' = 0 would push the prediction higher. The blue arrows representing the values of these features have a negative influence on the prediction('onehot_overtime_yes' = 0). The bold value 0.29 is the actual prediction for this sample.

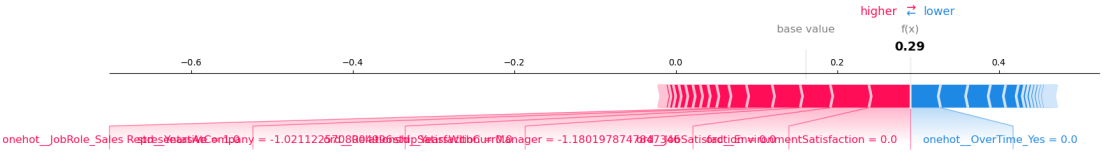

Figure 11: SHAP local feature importance

By comparing the different feature importance plot, we could find out their important features are quite similar. 'onehot_OverTime', 'std_MonthlyIncome', 'std_NumCompaniesWorked', and 'ord_EnvironmentSatisfaction' are always important features. The overall important features are quite consistent with what I expected. The high income, good working conditions, and less workload are the key to increase employee's happiness, and thus avoid their turnover. The surprising thing about feature importance is that I found Job Roles have very low feature importance, as shown below. I expected that Job Roles and Education Level would be more decisive to employee's attrition, however the opposite is the truth.

5. Outlook:

A weak spot of my project's dataset was created 6 years ago, however there are some changes happening in people's working style like working remotely, thus the dataset should be updated. Since the dataset is imbalanced, SMOTE could be tried for the XGB classifier. There are only 1470 data points in this dataset, which is not very many. As a result, capturing more data points would be helpful to increase the F_1 score of the model. Moreover, we could try more models like LightGBM to get a better prediction. This project could be used by companies to increase their benefit.

By identifying the features which make employees more likely to quit their job, companies could lower employee turnover rate.

Reference:

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

[2]: JANIO MARTINEZ BACHMANN. Attrition in an Organization, Why Workers Quit?
https://www.kaggle.com/code/janiobachmann/attrition-in-an-organization-why-workers-quit

[3]: KELLI BELCHER. HR Analytics and Prediction of Employee Attrition
https://www.kaggle.com/code/kellibelcher/hr-analytics-and-prediction-of-employee-attrition/data