

# Duke CS671 Fall 2022 Kaggle Competition

Shucheng Zhang sz255

December 10, 2022

## 1 Exploratory Analysis

### 1.1 Dataset Description and Clean

The dataset used in this competition contains 35 features for 1470 samples of two classes (Attrition and No Attrition). In these features, 26 features are numbers and 9 are categories. After encoding, the feature details can be shown in Figure 1.



Figure 1: Dataset

From Figure 1, we can see this dataset is relatively clean, so any further clean will be done in this project is in order to improve the prediction performance [1-4]. There are 5 points we noticed should be improved.



**Age** From the Figure 3, the *Age* feature shows a normal distribution but with a slightly positive skew. Also, the attrition is related to the *Age*. Employee in early 20s are most likely to leave. After that, the turnover rate tends to decline until it reaches its lowest point in the 40s, and then, slightly increase until 60s.

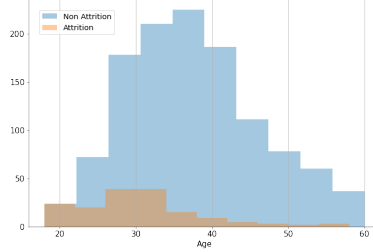


Figure 3: Age Distribution

**Monthly Income** From Figure 4, basically, higher paid jobs averages the lower attrition, while more employee with low income will leave. In the correlation matrix, we can see there are several features are related to the *MonthlyIncome*. Their relationship can be described as Figure 5.

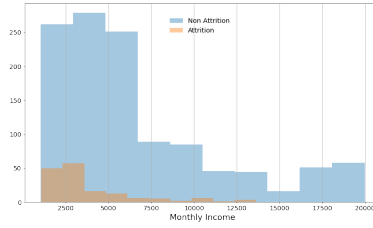


Figure 4: Monthly Income Distribution

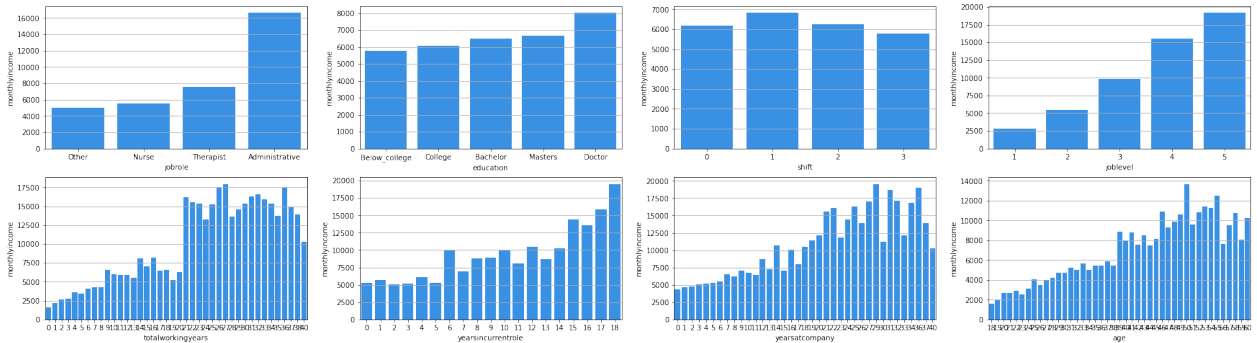


Figure 5: Monthly Income with its Related Features

**OverTime** Moreover, people who work overtime are more likely to quit.

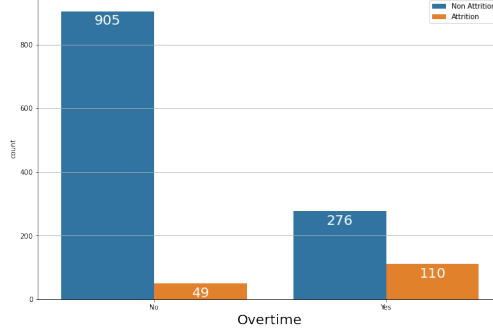


Figure 6: OverTime Distribution

Therefore, to take advantage of the relationship between the features, we multiply them together and get  $48 * 48$  size of features (after minus *Attrition*). Then, in order to reduce dimension of input and prevent overfitting, we select more related features by Lasso Regression [8]. By decreasing some of the regression coefficients to zero, the LASSO approach regularizes model parameters. After that, the features corresponding to every non-zero weight is important here and will be used in the models later. Here, after lasso feature selection, we only left 319 features.

**Attrition** The predicted target *Attrition* shows that the samples in this dataset is not balanced. In another word, the original dataset contains 1470 employee records. Only 237 employees have left the organization, whereas 1233 employees still working, which bias the dataset towards the working employees. Due to this imbalance, the prediction model performs relatively poorly. To improve the model’s performance, we use the Adaptive synthetic (ADASYN) sampling approach [7] to transform the dataset into its balanced version.

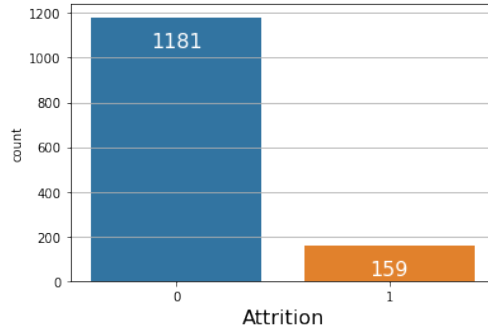


Figure 7: Attrition Distribution

Thus, the size of final training dataset is (2382, 319), including 1191 attrition and 1191 non-attrition.

## 2 Models

In this project, we choose Adaboost [5] and Deep Neural Network [6]. The reasons are described in this section.

### Adaboost

- AdaBoost has high accuracy and is easy to use with less need for adjusting parameters.
- AdaBoost is not prone to overfitting because of its joint optimization
- Compared with bagging algorithm and Random Forest algorithm, AdaBoost fully considers the weight of each classifier and can use different classification algorithms as weak classifiers.

### Deep Neural Network

- Compared with traditional machine learning algorithm, DNN can fit almost any function, because the nonlinear fitting ability of DNN is very strong.
- Typically, DNN has a good performance on classification problem.
- There are many methods like *dropout* and *NormalizationLayer* can be used to prevent overfitting.
- Because of the development tool named Pytorch, DNN is much easy to implement and can be trained with a high computational efficiency.

## 3 Training

### Adaboost

Here, we use choose *Grid Search* method to select the optimal parameters. The base estimator is Decision Tree due to the great performance of it in classification. The number of estimator will be set as large as possible to overcome overfitting. As for the learning rate will be set according to the number of estimator, because a higher learning rate increases the contribution of each classifier and there is a trade-off between the learning rate and the number of estimator. One Adaboost model can be trained in two minutes.

### Deep Neural Network

For DNN, there are many parameters can be optimized because of the complex construct of it. The key elements affecting the accuracy includes training epoch, learning rate, the number of hidden layer, the number of neurons and the activation function. Among them, the activation functions of hidden layers are softplus, which is a curvy version of ReLU, while the activation functions of input and output layers are sigmoid. The reason why we use sigmoid function in output layer is that it exists between  $(0, 1)$ , so it is especially used for models where we have to predict the probability as an output. Other parameters will be chosen by grid search. Moreover, the Dropout layer is used to prevent overfitting. Typically, the dropout probability is 0.2 to balance the accuracy and the generalization. Finally, we choose Adam and MSELoss as optimizer and loss function respectively. Because of the Cuda, one DNN model can be trained less than one minutes if the training epoch is less than 100.

## 4 Hyperparameter Selection

As described before, most of the hyperparameters are selected by grid search in a reasonable range.

**Adaboost** The parameters chosen ranges are shown in this Table 1.

Table 1: **Adaboost Training Details**

Learning rate	[0.01, 0.1, 1]
Number of estimator	[500, 600, 700, 800, 900, 1000, 1100, 1200]

From Figure 8, we choose  $learn\_rate = 0.01$  and  $n\_estimator = 1100$ .

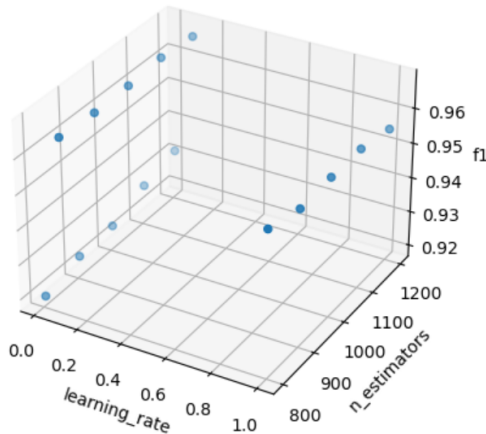


Figure 8: Adaboost Grid Search Result

**Deep Neural Network** For DNN, we does not use Lasso to select features, because the features will be multiply together in the network.

Table 2: **DNN Training Details**

Epoch $e$	[40, 60, 80]
Learning rate $r$	[0.0001, 0.001, 0.01, 0.1]
Batch size $b$	128
Decay step	[20, 40]
Decay rate	0.1
Number of hidden layer	[3, 5, 7]
Number of neurons	[50, 70, 100, 200]

From huge number of experiments (part of the results shown in the Figure 9), we finally choose  $epoch = 80$ ,  $learn\_rate = 0.01$ ,  $decay\_step = 20$ ,  $Number\_of\_hidden\_layer = 3$  and  $Number\_of\_neurons = 100$ . The training results with these parameters are shown in Figure 10.

	50 neurons	70 neurons	100 neurons	200 neurons
3 layers	0.0643	0.0660	0.0630	0.0641
5 layers	0.0652	0.0642	0.0667	0.4703
7 layers	0.0657	0.0649	0.0680	0.4703

Figure 9: DNN Grid Search Result

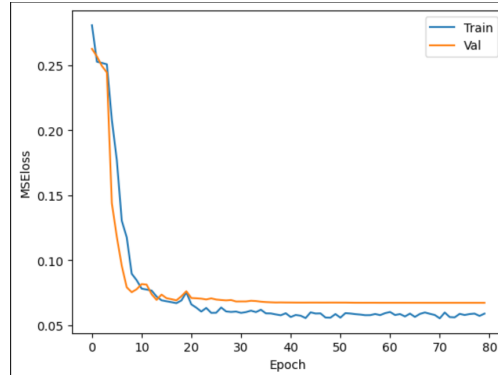


Figure 10: DNN Training Result

## 5 Data Splits

We use 5-fold CV to evaluate the Adaboost model's performance and prevent overfitting. The results shown in the Figure prove our model is robust. Also, we did not use the whole

training set to train the model, while split 10% samples as test set. The results are also shown here.

	Train					Test
CV	1	2	3	4	5	
F1 Score	0.87	0.93	0.94	0.95	0.94	0.69

Figure 11: Adaboost CV Result

## 6 Errors and Mistakes

The hardest part is feature engineering, because as a student who doesn't have data science background, I need to study from beginning. As for the mistakes, at the beginning of the project, I used a inner function in *pandas* to encode the training data and the test data respectively, which leads to the encoding rules are different in training and testing, so all models performed worse and it takes a long time for me to find this bug.

## 7 Predictive Accuracy

### Kaggle Name: Shucheng Zhang

The adaboost model with Lasso got a 0.68421 in both private score and public score. However, surprisingly, the adaboost model with label encoding show a great performance: 0.63157 in public score and 0.82051 in private score. The DNN model show 0.56521 in public score and 0.64 in private score. The training result of adaboost is shown in Figure 11 and the training result of DNN is shown in Figure 10.




Submission and Description	Private Score	Public Score	Selected
 <b>pred_ada_lasso.csv</b> Complete · 17h ago	<b>0.68421</b>	<b>0.68421</b>	<input checked="" type="checkbox"/>
 <b>pred_ada_labelcoder_all.csv</b> Complete · 18h ago	<b>0.82051</b>	<b>0.63157</b>	<input type="checkbox"/>
 <b>pred_dnn_all.csv</b> Complete · 2d ago	<b>0.64</b>	<b>0.56521</b>	<input type="checkbox"/>

Figure 12: Kaggle Result

## 8 Reference

[1] <https://www.kaggle.com/learn/feature-engineering>

[2] <https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/>



- [3] <https://www.knime.com/blog/predicting-employee-attrition-with-machine-learning>
- [4] <https://www.enjoyalgorithms.com/blog/attrition-rate-prediction-using-ml>
- [5] Schapire, Robert E. "Explaining adaboost." Empirical inference. Springer, Berlin, Heidelberg, 2013. 37-52.
- [6] Al-Darraj, Salah, et al. "Employee Attrition Prediction Using Deep Neural Networks." Computers 10.11 (2021): 141.
- [7] He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
- [8] Fonti, Valeria, and Eduard Belitser. "Feature selection using lasso." VU Amsterdam research paper in business analytics 30 (2017): 1-25.

## 9 Code

Shown in next pages.