Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230147

Comparison of machine learning models for employee turnover prediction

Qing Yin

Department of Information Engineering, Hebei Agricultural University, Qinhuangdao, Hebei, 066000, China

yinqing@STUDENT.WUST.EDU.PL

Abstract. In the era of big data, companies are able to more easily record, analyze, and utilize data, making it feasible to predict employee turnover. In the current economic situation, how to predict employee turnover has become particularly important. The core of this article is the establishment of a forecasting model based on real data, and then through the application to reflect the use and importance of the forecasting model. This paper uses XGBoost, LGBM, random forest and decision tree models, uses IBM's real data to construct prediction data, and evaluates the models to select the model that is most suitable for the lost data. Additionally, the role of Synthetic Minority Over-sampling Technique (SMOTE) in improving classification accuracy was explored. The results of this work indicate that: 1) balancing the data set using SMOTE oversampling technique is more effective than using the original data set; and 2) The XGBoost algorithm performs better, with higher precision and greater flexibility.

Keywords: XGBoost algorithm, SMOTE, employee turnover prediction, machine learning.

1. Introduction

In 2020, COVID-19 spread all over the world, putting immense pressure on China's social and economic systems within just a few months. Enterprises across all sectors faced a funding crisis, and some even went bankrupt. To cope with the epidemic's impact on businesses, many companies had to reduce employee wages. Some employees also choose to leave voluntarily. However, if the needs and emotions of employees are not met, the loss of a significant number of employees could cause serious damage to the company, making it difficult to recover after the epidemic. Talent loss has been a common phenomenon for various companies worldwide for many years. With the development of the economy and technology, the turnover rate in most companies is also increasing. In the current economic situation, predicting employee turnover has become particularly important. On the one hand, it can provide an additional retention plan for some core employees at IBM. On the other hand, it can consider the necessity of retaining employees through decision-making, enabling small and medium-sized enterprises to take corresponding measures to reduce core employee loss. In this paper, the dataset is predicted based on the LGBM and XGBoost algorithms, evaluate the model, and find that the XGBoost model performs best.

DOI: 10.54254/2755-2721/8/20230147

2. Related work

Machine learning is commonly used to predict employee turnover, with different approaches such as using a single model, improved models, or a combination of multiple models. Li Qiang and Xu Liang [1] combined Adaboost and Random Forest algorithms through the Stacking integrated learning algorithm to create a new model called "LRA," which achieved a correct rate of 89.09%. Wang Guanpeng and Qin Shuangyan [2] used the random forest algorithm to predict an accuracy rate of 95.43%, but their results were biased and had limited practical significance. Zhang Tianyi [3] proposed the SMOTE balance algorithm to address this issue.

3. Method

3.1. Dataset

This article uses the IBM dataset from a certain branch of the company's human resources department published on the Kaggle platform. The dataset contains only English and numerical values and has 35 variables horizontally with a total of 1470 rows of data. The variables primarily include five types of information: general employee information, employee job information, employee-company information, company information, and funds.

3.1.1. Feature rebalancing. In practical binary classification problems, unbalanced data that follow natural laws are common. This dataset has a positive sample-to-negative sample ratio of 5.2:1, making it highly unbalanced. Direct modelling using this dataset would seriously affect how accurate the model is. Therefore, appropriate processing is crucial. Because of the limited data used in this paper, the SMOTE oversampling and unprocessed models are compared. The processing result is: Counter ({0: 1233, 1: 1048}).

3.1.2. Feature screening. The Employee Number is categorized as an ID attribute, and several single-valued features, including Employee Count, Over 18, Standard Hours, and Attrition result variables, were identified as irrelevant and removed from the dataset using the 'df.drop' function to eliminate redundant data. Observing the correlation between each feature and 'Attrition' through chi-square test and A NOVA test, and delete the weakly correlated features.

3.2. Models

3.2.1. XGBoost. It is an algorithm based on Gradient Boosting Decision Trees (GBDT), whic was formally proposed by Chen Tianqi [4] in 2016 in the paper. The XGBoost algorithm is based on Boosting, and its process is similar to that of Boosting. The weak learners of XGBoost are obtained through multiple iterations, and a more powerful learner is formed through a certain combination. However, compared to Boosting, XGBoost uses a CART (Classification and Regression Tree) as the basic learner. Initially, a basic learner is formed, and a new learner is subsequently formed to compensate for the residual error of the previous learner. By repeatedly forming T learners, the residual error becomes smaller and smaller in continuous iterations, ultimately forming a learner with minimal error and high precision. Then, learners are combined to further enhance the accurate prediction of the ultimate classifier by reducing the error during the training process. This allows XGBoost to perform better on the training set. Once the weak learners are found, XGBoost combines them by constructing an additive model.

$$\widehat{y_i^{(t)}} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y_i^{(t-1)}} + f_t(x_i) \#(1)$$

Among them, the variable x_i represents the feature value of sample i, while $f_k(x_i)$ is called leaf weight, which represents its corresponding output of the k-th tree. The ultimate output of XGBoost could be obtained by summing all leaf weights from k trees.

DOI: 10.54254/2755-2721/8/20230147

3.2.2. Random forest. This model could be enhanced and strengthened on the basis of Bagging [5, 6]. Although it, like Bagging, requires random sampling of the training set to form T sampling sets, there are two essential differences between random forest and Bagging: one is The model trained by the random forest using the sampling set is a decision tree, so it is called "forest" vividly; the second is that Bagging takes all the features into account when building the tree, and the random forest is in every split of each decision tree At every point, k features are randomly sampled from the feature set p of the initial training set, where k is much smaller than p, and then when the decision tree is split according to the feature with the best performance, this feature is selected from the k features. Since the random forest has the double randomness of the random sampling set and the random split feature set, the word "random" is included in the name. This two randomness greatly improve the prediction effect of random forest. The main reason is the integration effect mentioned above. Having two layers of randomness makes the difference between weak learners larger and classifiers more abundant. Diversity, which satisfies the "different" requirements of integrated learning, enables the model to improve the generalization ability while improving the accuracy. On the other hand, since it calculates the best features from the k features at the split point of the decision tree, this greatly improves the efficiency of the model and makes the model faster than extracting features from p features.

3.2.3. Decision tree. The algorithm steps of the decision tree: First, the priority of each decision attribute in the dataset is selected. Second, the data and labels of the training set are used to generate a decision tree with each priority of each attribute. Finally, input the test set into the decision tree, compare the attributes of each decision tree, and finally use the label of the traversed leaf node as the result of the test data. If the traversed leaf node is empty, count the sibling nodes of the empty node The label with the most occurrences is used as the output result of the test data [7, 8].

3.2.4. LGBM. A major feature of LightGBM [9] is the introduction of two new technologies and an improvement based on the traditional GBDT:

In order to reduce the impact of the low gradient and long tail, the Gradient-based One-Side Sampling (Goss) algorithm abandons a significant portion of the data with a modest shaving and merely leverages the remainders to measure the information gain. Because data which propagates large gradients is more important during the measurement of the information gain, Goss is still robust and could achieve high accurate performances with insufficient amount of data.

Exclusive Feature Bundling (EFB) technology is developed to minimize the number of supporting features by bundling mutually exclusive features together. However, searching the optimal bundling strategy is difficult. To solve this problem, greedy algorithm is leveraged and is observed could achieve outstanding performances.

In traditional GBDT algorithms, finding the optimal division point is slow. The traditional method, called Pre-Sorted, involves enumerating all possible feature points on the sorted eigenvalues. However, LightGBM leverages the histogram method instead. Firstly, the continuous eigenvalues will be discretized and meanwhile a k-width histogram will be constructed. At the beginning, the discrete value is taken as accumulative index for constructing the histogram. Then, after the first traversing of data, the statistics are accumulated for discretization. When the node is split later, it can be based on the discretization on the histogram value, find the best split point from these k buckets, so that the optimal split point can be found faster. The histogram algorithm does not record pre-sorted results like Pre-Sorted, it merely saves the discrete process of the features, so using the histogram can reduce memory consumption.

4. Experiment and result

4.1. Experimental setting

The study uses real data provided by IBM. The dataset obtained through SMOKE technology has a positive to negative sample ratio of 0.85:1, with 85% for training and 15% for testing. To reduce the

DOI: 10.54254/2755-2721/8/20230147

difference between predicted and actual model performance, the cross-validation [10] is employed to separate the training set, with 10 splits, and repeat three times. Four methods are compared in the experiment: XGBoost with parameters learning_rate=0.01, max_depth=3, n_estimators=1000; random forest with parameters max_depth=4, random_state=0; LGBM with parameters learning_rate=0.01, max_depth=3, n_estimators=1000; decision tree with parameters max_depth=3, random_state=0.

4.2. Result

4.2.1. Effectiveness of SMOTE. Table 1 shows the results of one of them. Through extensive testing, it was found that SMOTE was used. The integrated results of the processed training set are better than the unprocessed results. Although random forests indicate that there is no need to deal with unbalanced data, the results show that it will be better to deal with it. Therefore, this paper uses sampled data training for other algorithms.

Table 1. Comparison of models before and after unbalanced treatment.

	AUC value	F1score
Unsampled training set	0.706	0.57
Training set after SMOTE	0.887	0.88

4.2.2. Effectiveness of SMOTE. Table 2 demonstrate the comparison of the machine learning algorithms.

Table 2. Comparison results of each model.

ML Algorithm	Accuracy	Cross Validation Score	ROC AUC Score	F1 Score with Attrition	F1 Score without Attrition
XGB Classifier	88%	91.92%	88.25%	87%	89%
LGBM Regression	87%	92.14%	88.44%	88%	90%
Decision Tree Classifier	79%	80.22%	78.68%	77%	81%
RandomForest Classifier	82%	87.62%	81.72%	80%	84%

5. Conclusion

To enable churn prediction, this work first analyzed the attributes of the original data provided by IBM, processed the attributes of different data types, and integrated the data sets as input for the model. Then model parameters are calculated and verified using the training and validation set data. Finally, the model's performance is evaluated on the test data set to enable churn prediction.

In summary, it is finally proved that the X GBoost model and the L GBM model have the best effect, followed by the random forest model, and the worst performance is the decision tree model, so the X GBoost and L GBM models can be used for modelling prediction in daily life, to increase the value.

DOI: 10.54254/2755-2721/8/20230147

XGBoost has greatly improved the generalization capacity under the double function of the regular term and the loss function. In addition, when XGBoost is looking for parameters, it uses the cross-validation method in the training iteration. It could prevent overfitting and output the number of iterations accurately and simply. Using the XGBoost model to predict problems can provide a lot of convenience, and it can also make the prediction effect of the model better.

References

[1] Li, Q., & Zhai, L. (2019). Analysis and research on employee turnover prediction based on stacking algorithm. Journal of Chongqing Technology and Business University (Natural Science Edition), 36(01), 117-123.

[2] Duan, Y. (2022). Statistical analysis and prediction of employee turnover propensity based on data mining. In 2022 International Conference on Big Data, Information and Computer Network, 235-238.

[3] Sağlam, F., & Cengiz, M. A. (2022). A novel SMOTE-based resampling technique trough noise detection and the boosting procedure. Expert Systems with Applications, 200, 117023.

[4] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 785-794.

[5] Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.

[6] Breiman, L. (2001). Random forest. Machine Learning, 45, 5-32.

[7] Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

[8] Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.

[9] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., et, al., (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[10] Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The'K'in K-fold Cross Validation. In ESANN, 441-446.