A Predictive Model for Employee Voluntary Turnover: An Empirical Study of a Manufacturing Company in Taiwan

Lynn Chiao

Graduate Institute of Human Resource Management,

National Central University

Abstract

This study collects data from a manufacturing company in Taiwan to build a reliable predictive model of employee voluntary turnover. The results suggest that the random forest and extreme gradient boosting algorithm perform the best. Moreover, the results of a variable importance investigation indicate that the elementary level of managerial training hours, professional training hours, job tenure, the average number of promotions, and age contribute the most in predicting employee voluntary turnover outcomes. Imbalanced classification, feature selection, and K-fold cross-validation are introduced and tested in this study.

Keywords: Machine learning; Employee Voluntary Turnover; Feature Selection

Introduction

In this study, we use data from a Taiwan manufacturing company to examine whether employee voluntary turnover can be accurately predicted using machine learning algorithms. The study focuses on employee voluntary turnover because it is considered to be costlier and more disruptive than employee involuntary turnover (Lambert, Hogan, & Altheimer, 2010), and practitioners are more concerned about the attributes that significantly affect employee voluntary turnover. This study will identify the variables that affect voluntary turnover by calculating and ranking variable importance.

Research Methodology

For this study, SVM, decision tree, logistic regression, random forest, and XGBoost algorithms was implemented to build predictive models for employee voluntary turnover. The synthetic minority oversampling technique (SMOTE), feature selection, and K-fold cross-validation were implemented to balance the data between the turnover and non-turnover classes and optimize the models. The following flowchart (Figure 1) describes the procedure in this study.

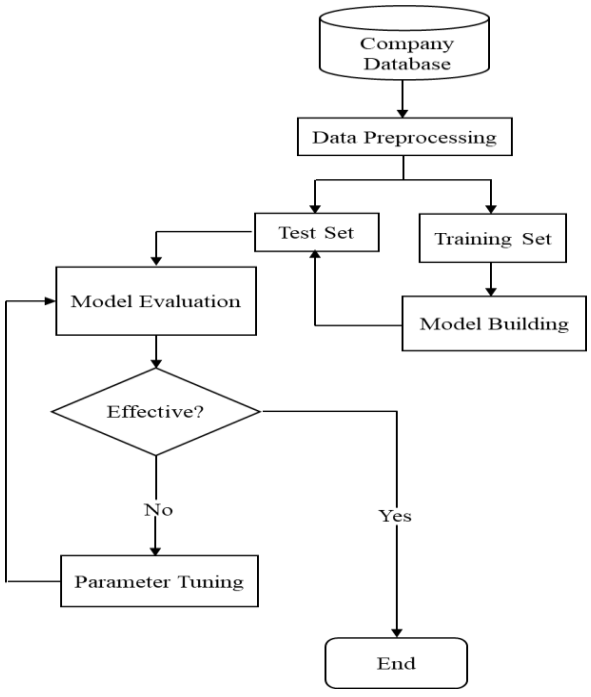

Figure 1: Flowchart of the machine learning procedure

Human Resource Dataset

Data were collected from 2012 to 2019. The identities were removed from all information to protect employee privacy. The dataset contained a sample of 2,169 individuals with 91 variables. After excluding individuals marked as involuntary turnover, 1,944 individuals remained. Out of the 91 variables, 49 were selected based on factors illustrated in Table 1.

Table 1. *Factor Description*

Variable No.	Category
No. 1	Target Class
No. 2 – No. 15	Background Information
No. 16 – No. 20	Promotion Records
No. 21 – No. 38	Aptitude Test
No. 39 – No. 45	Training Records
No. 46 – No. 48	Mentoring Program
No. 49	Leave Records

The study used a simple random sampling method to divide the data into training and test sets. Normally, the rule of thumb of data partitioning is to split the data into a 70% training set and 30% test set. However, considering that the data we obtained are scarce, it was preferable to use 80% of the data for algorithm training to build a more reliable classification model.

Data Pre-Processing

Basic data cleaning procedures were implemented. Misspellings were corrected and irrelevant or duplicate data were removed. The advanced pre-processing techniques that were performed are described as follows.

A. Imbalanced Classification

Imbalanced classification occurs when samples contained in one class is overly outnumber samples contained in other classes. In this binary classification case, the turnover class is the majority class. The imbalanced proportion of samples between those employees in the turnover and non-turnover classes was a problem that needed to be addressed. For the imbalance between these classes employees, consider that the dataset contained only 1,944 samples; if an under-sampling technique was conducted, the samples would be down to 1,468. Therefore, it was not wise to use the under-sampling technique because it discarded too much valuable information. Hence, we used SMOTE to oversample the dataset so as to balance the two classes.

In general, when dealing with imbalanced data, common techniques include under-sampling, over-sampling, and hybrid sampling. Each technique has its advantages and drawbacks. These three techniques are easy to implement; however, under-sampling may result in the loss of potentially useful information due to sample reduction, and over-sampling could lead to the over-fitting of a model because of repeated sampling (He & Ma, 2013). When a hybrid sampling method is used, the problems of both under-sampling and over-sampling could occur. To solve this, Chawla, Bowyer, Hall, and Kegelmeyer (2002) developed an advanced technique called SMOTE. Unlike traditional over-sampling techniques that use replication to increase the number of samples in the minority class, SMOTE creates synthetic samples to increase the sample size. With the application of SMOTE, the overfitting of a model is less likely to happen.

B. Hyper-parameter Tuning

This study combined K-folds cross-validation (Figure 2) with grid search to find the optimal hyper-parameter tuning estimates by testing various combinations of hyper-parameters. K-folds cross-validation is the procedure of splitting training data into K parts and using K-1 parts for training and one part for testing. It continues to select different parts as test data and calculates the result at each fold. Once the process is complete, the results from each fold are compared, and the set of hyper-

parameters that produce the best performance result for a classifier is determined. To train the classifiers, this study used a 5-fold cross-validation to optimize the set of parameters for the classification algorithms.

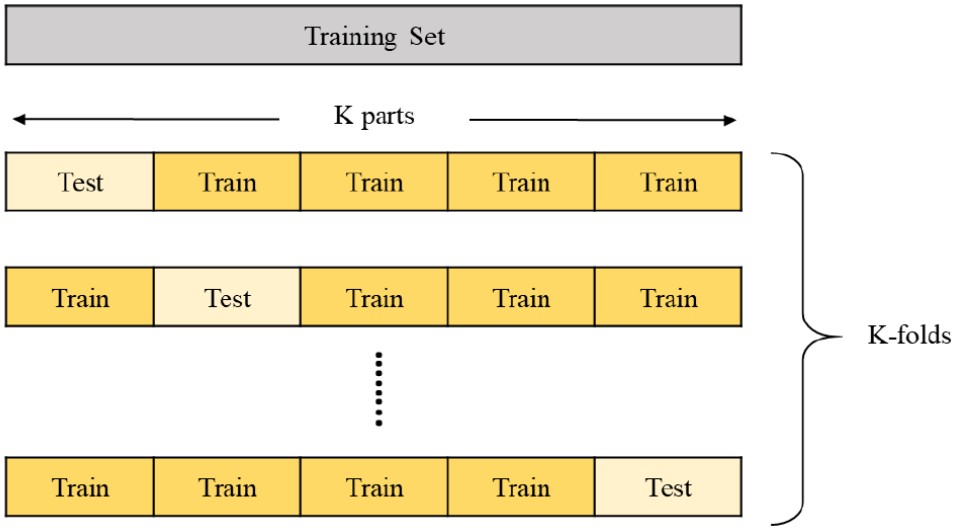

Figure 2: K-cross validation

C. Feature Selection

Feature selection is implemented to test whether the performance result would be the same if unimportant variables were removed using a machine learning algorithm. We used a feature selection wrapper algorithm called Boruta, which performs a top-down search for variables by randomly comparing their importance. The variables that the algorithm deem as irrelevant are excluded from the dataset.

D. Evaluation Metrics for Classification

To evaluate the performance of a model, there are several ways to calculate different metrics based on a confusion matrix. Accuracy, precision, recall (also known as sensitivity), and F-score were chosen to evaluate the model performance.

In this study, we calculated the accuracy, F-score, recall, and precision to test the models' performance. However, the primary metric we used for classifier evaluating was the F-score instead

of the accuracy because of the imbalanced data problem. When the accuracy is used as the primary metric for imbalanced data, an accuracy paradox will often result. This occurs when the class proportion is imbalanced; it leads to machine overlearning and the entire sample being classified into the majority class. When this situation occurs, calculating the performance of a model using the accuracy would produce a bias. Therefore, the F-score is preferred over accuracy as it balances the values of recall and precision.

E. ROC-AUC

The area under the curve(AUC) of the receiver operating characteristic curve(ROC) is calculated using the true and false positive rates. It can also be considered when evaluating the performance of classifiers.

An AUC value under 0.5 indicates that the model prediction is worse than random guessing. An AUC value between 0.6 and 0.7 is a fair classifier. An AUC value between 0.7 and 0.8 indicates that the test quality of the model is moderate. A model with an AUC value between 0.8 and 0.9 has a good test quality. The higher the AUC value, the greater the test quality of the classifier. By the same token, when an AUC value is above 0.9, the classifier is seen as having an excellent test quality. We call a model that has an AUC value equal to one a perfect classifier. It should be noted that although an AUC value equal to one indicates that the model has a zero percent false positive rate and a hundred percent true positive rate, one should be cautious of a data contamination problem in which test data is incorporated into training data by accident, resulting in a machine simultaneously learning test data, hence, a perfect classifier.

Results

Classifier Performance

The performance of the classifiers is summarized in Table 2.

Table 2: Comparison of the Classifiers

Algorithms	Accuracy	F-Score	Recall	Precision	AUC
Logistic Regression	0.799	0.834	0.810	0.860	0.795
SVM (Linear)	0.796	0.836	0.835	0.838	0.784
SVM (Polynomial)	0.716	0.809	0.963	0.698	0.636
SVM (RBF)	0.802	0.841	0.843	0.840	0.788
SVM (Sigmoid)	0.807	0.844	0.839	0.849	0.796
Decision Tree (C4.5)	0.802	0.833	0.793	0.877	0.804
Decision Tree (C5.0)	0.781	0.820	0.798	0.843	0.776
Decision Tree (CART)	0.794	0.821	0.760	0.893	0.805
Random Forest	0.832	0.865	0.864	0.867	0.822
XGBoost	0.843	0.873	0.864	0.882	0.836

From the results in Table 2, it can be seen that the SVM with a polynomial kernel has the lowest performance. The random forest and XGBoost clearly outperformed all of the other machine learning algorithms presented in this study. The results show that the random forest has a slightly higher accuracy than does XGBoost. Nevertheless, our study used the F-score as the primary metric for the comparison. In this case, the performance of the XGBoost classifier is better than that of the random forest classifier. To further compare the performance between the random forest and XGBoost, we conducted a K-fold cross-validation, SMOTE, and feature selection for testing.

K-fold Cross-validation

This study used grid search with a 5-fold cross-validation to search for the optimal hyper-parameters for the random forest and XGBoost classifiers. For the random forest, we tuned the hyper-parameters of “mtry”, which is the number of predictors sampled for splitting at each node. Figure 3

suggests that for a 5-fold cross-validation, the accuracy is the highest when mtry is set to 28. The accuracy is 0.922, and the Kappa statistic is 0.844.

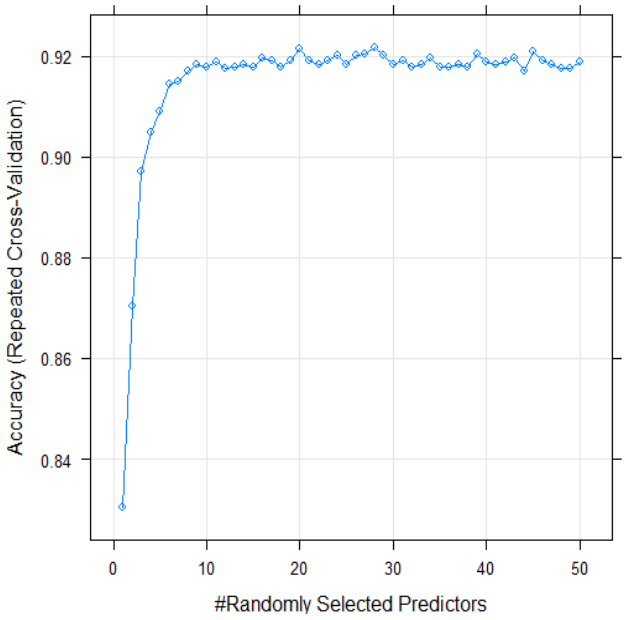

Figure 3: 5-fold cross-validation on random forest

For XGBoost, three primary hyper-parameters were tuned in this study: gamma, learning rate, and max tree depth. Gamma specifies the minimum loss reduction required to make a further partition on a leaf node of the tree. When gamma is specified, XGBoost will grow the tree to the max depth specified and remove splits that do not meet the specified gamma. The learning rate determines the contribution of each tree to the final outcome and controls how quickly the algorithm proceeds down the gradient descent. A high max tree depth allows the algorithm to capture unique interactions from the data but also increases the risk of over-fitting (Boehmke & Greenwell, 2019).

Figure 4 shows the hyper-parameter tuning result with a 5-fold cross-validation. The results indicate that if a set of hyper-parameters has the values of gamma equal to 0.25, max depth equal to 4, and shrinkage (learning rate) equal to 0.1, the highest accuracy is produced. The accuracy is 0.830, and the Kappa statistic is 0.642.

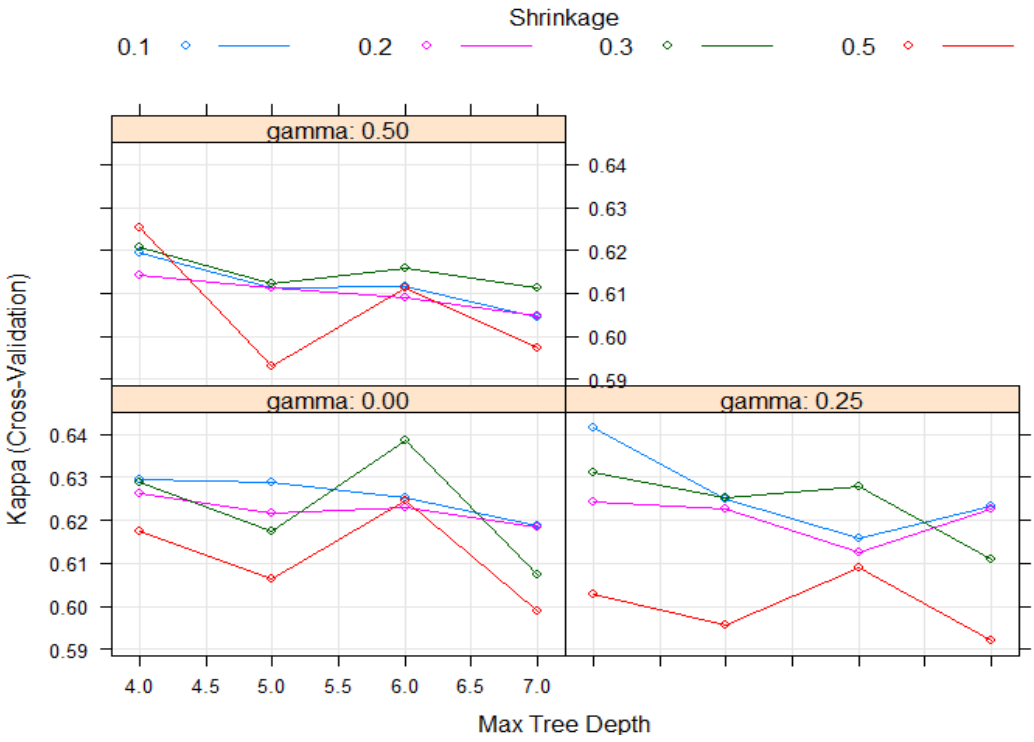

Figure 4: 5-fold cross-validation on XGBoost

We inputted the newly found hyper-parameters into XGBoost and the random forest and ran the classifiers once again. As Table 3 shows, the performance of the XGBoost and random forest classifiers improved significantly.

Table 3: Hyper-parameter Tuning Results

	Accuracy	F-Score	Recall	Precision	AUC
XGBoost	0.858	0.884	0.864	0.905	0.857
Random Forest	0.845	0.876	0.876	0.876	0.835

SMOTE

Next, to tackle the imbalanced data problem between two classes, we conduct a SMOTE to determine if this technique can handle the imbalance well and improve the performance of the random forest and XGBoost classifiers. Results are as follows (Table 4).

Table 4: Results using SMOTE

	Accuracy	F-Score	Recall	Precision	AUC
XGBoost	0.848	0.874	0.847	0.903	0.848
Random Forest	0.863	0.886	0.847	0.928	0.869

It appears that the random forest classifier performs exceptionally well with SMOTE, whereas the performance of the XGBoost classifier decreased.

Feature Selection

Table 5 presents the performances after using the Boruta algorithm for feature selection. The algorithm deemed as unimportant the following three variables: gender, job related to major, major type. Therefore, we removed these four variables and input the remaining variables into the random forest and XGBoost. As the results indicate, the values of the F-score increased slightly with XGBoost, whereas the F-score of the random forest declined.

Table 5: Results using Feature Selection

	Accuracy	F-Score	Recall	Precision	AUC
XGBoost	0.848	0.878	0.880	0.877	0.837
Random Forest	0.835	0.868	0.868	0.868	0.824

SMOTE with Feature Selection

We then combined SMOTE with the feature selection to examine whether the performance would change in the last tryout. From the results shown in Table 6, it can be seen that XGBoost once again did not respond well to SMOTE. Its precision and F-score were slightly lower than those using only the feature selection. The performance of the random forest, on the other hand, improved when SMOTE was used.

Table 6: SMOTE with Feature Selection

	Accuracy	F-Score	Recall	Precision	AUC
XGBoost	0.848	0.875	0.855	0.896	0.846
Random Forest	0.858	0.881	0.839	0.927	0.865

Variable Importance

The best performing random forest model was the one that included SMOTE. Thus, we calculated the variable importance of this model to determine the most prominent variables for predicting turnover. The variable importance of the random forest was determined based on the mean decrease Gini. The higher the mean decrease Gini, the greater the importance of a variable. Table 7 shows the three most important variables.

Table 7: Variable Importance in the Random Forest

Variable	Importance (Mean Decrease Gini)
Average Number of Promotions	270.581
Managerial Elementary Training Hours	127.405
Professional Training Hours	108.362

The best performing model of XGBoost was the one with the 5-folds cross-validation. The calculation of the variable importance for XGBoost was different. We used Gain to calculate the best XGBoost model; the higher the Gain value, the more important a variable is for generating a prediction. The three most important variables are presented in Table 8.

Table 8: Variable Importance in XGBoost

Variable	Importance (Gain)
Managerial Elementary Training Hours	0.151
Job Tenure	0.151
Age	0.082

Conclusion

Discussion

Several types of machine learning algorithms were tested for predicting employee voluntary turnover. According to the initial comparisons of the results of different types of machine learning classifiers, the random forest and XGBoost easily outperformed others. When the effect of implementing the K-fold cross-validation for hyper-parameter tuning was considered, it appeared that tuning the hyper-parameters can greatly improve the performance of the XGBoost and random forest. Based on the performance results, XGBoost seemed to be the ideal classifier with or without hyper-parameter tuning. However, when using SMOTE to handle the imbalanced classes problem, our findings suggested that SMOTE could be an effective method to improve the performance of the random forest. When using SMOTE, the random forest classifier exceeded the XGBoost classifier. However, XGBoost did not work well with SMOTE. To conclude, the results indicated that XGBoost itself can handle an imbalanced dataset better than the random forest. On the other hand, the random

forest required further pre-processing as a means to achieve better prediction; if the imbalanced problem is properly handled, the random forest could outperform XGBoost.

With regard to feature selection, the process allows researchers to automatically remove unimportant variables. Results from the random forest and XGBoost classifiers suggested that the performance did not decline much even when four variables were removed. Therefore, feature selection could be a useful tool if a dataset has too many variables and requires variable reduction.

We calculated various metrics to evaluate the performance of machine learning classifiers in this study. Both recall and precision are crucial for evaluating the performance of classifiers; therefore, we used the F-score as the primary metric for selecting the best model. However, comparing F-scores is not always the best solution for selecting models. There is a trade-off between recall and precision. The higher the recall, the lower the precision, and vice versa. To decide which metric is our foremost concern, we should first consider the goal we plan to achieve. In this study, we aimed to predict the voluntary turnover of employees in an organization. Recall measures the proportion of actual turnovers that have been identified correctly, while precision measures the proportion of employees classified as turnovers that are actually correct (See Eqs. 1 and 2).

$$\text{Recall} = \frac{turnover\ correctly\ identified}{turnover\ correctly\ identified + turnover\ incorrectly\ labeled\ as\ not\ turnover} \quad (1)$$

$$\text{Precision} = \frac{turnover\ correctly\ identified}{turnover\ correctly\ identified + employees\ incorrectly\ labeled\ as\ turnover} \quad (2)$$

Because the risk of classifying employees who end up leaving as “non-turnover” is more severe than that of classifying employees who stay as “turnover”, in this study, recall was the most important metric. If a company invests a great amount of effort and resources in employees who plan on leaving, the cost to the company will be high. Low recall was what we wished most to avoid, whereas low precision could be endured in exchange for a higher recall. Under this premise, we found the performance of XGBoost with feature selection was preferable. It should be noted that although the polynomial SVM had the highest recall, it is not a good classifier considering that all the other metrics

except recall were low. XGBoost with feature selection not only had a recall of 0.88 but also had a decent F-score. Therefore, this classifier was preferable if a company deemed recall to be crucial.

Overall, the random forest combined with SMOTE along with a 5-fold cross-validation yielded the best result if judged by the F-score. However, if we want to decrease the risk of a classifier mistaking turnover as “non-turnover”, then XGBoost with feature selection together with a 5-fold cross-validation would be the preferable choice for predicting employee voluntary turnover. With regard to variable importance, the average number of promotions contributed the most in the random forest classifier to predict employee voluntary turnover, while the elementary-level of managerial training hours was the primary attribute for the XGBoost classifier to predict employee outcomes. The findings were consistent with previous studies that claim the numbers of promotions and training hours play a significant role in affecting employee turnover (Nyberg, 2010; Choi & Dickson, 2009).

The results showed that the elementary level of managerial training hours, professional training hours, job tenure, average number of promotions, and age were contributed the most to the prediction of employee voluntary turnover outcome. XGBoost did not respond well when SMOTE and feature selection were used. In contrary, the random forest reacted well to these methods.

Limitations of the Study

The samples we collected for this study were insufficient. Furthermore, previous works often included variables of compensation, overtime, and performance records. Unfortunately, the dataset we obtained for this study did not include these variables. Despite not having these variables, we obtained great prediction results. Future works could include data such as employee salary, working time records, and performance as well as training records, job tenure, and promotion records to determine whether the predictions would improve.

Future Research

The potential for generalizing the research findings of this study across companies is unknown. Future works could consider using the same machine learning algorithms and pre-processing techniques to determine if the results from different datasets are consistent with the results of this study. Additionally, other advanced machine learning algorithms such as the neural network, adaptive boosting, or deep learning could be considered in future work. Moreover, the technique of applying a K-fold cross-validation with grid search could be explored more thoroughly. It is advisable to conduct 10-fold, 100-fold, or 500-fold cross-validations to examine whether different numbers of folds would generate different combinations of hyper-parameters and improve the performance of the classifiers.

References

Boehmke, B., & Greenwell, B. M. (2019). Gradient boosting. *Hands-on machine learning with R* (pp. 238 - 245).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Choi, Y., & Dickson, D. R. (2009). A case study into the benefits of management training programs: Impacts on hotel employee turnover and satisfaction level. *Journal of Human Resources in Hospitality & Tourism*, *9*(1), 103–116.

He, H., & Ma, Y. (2013). *Imbalanced learning: Foundations, algorithms, and applications*: John Wiley & Sons.

Lambert, E. G., Hogan, N. L., & Altheimer, I. (2010). An exploratory examination of the consequences of burnout in terms of life satisfaction, turnover intent, and absenteeism among private correctional staff. *The Prison Journal, 90*(1), 94–114.

Nyberg, A. (2010). Retaining your high performers: Moderators of the performance–job satisfaction–voluntary turnover relationship. *Journal of Applied Psychology*, *95*(3), 440–453.