Application of Xgboost Model to Predict Bank Loan

Yihan Fang *

School of Data Science, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China, 518172

* Corresponding Author Email: 122040050@link.cuhk.edu.cn

Abstract. Nowadays the bank loan has been a high-frequent behavior in the field of finance. Predicting the deposits paid by the lenders can effectively reduce the risk of credit fraud and avoid the loss of bank. Traditional methods like classification regressons and support vector machines are difficult to deal with large sets of data and with relatively low accuracy. In this paper, a novel and robust method called XGBoost is applied in the bank loan and achieves a high accuracy. Also, proving the accessibility of XGBoost model in the business analysis field. The data set used in this study comes from a real bank customer database, which contains 16 variables. Regarding the categorical variables, this paper used one-hot encoding to make these variables can be identified by the model. Following by two steps feature selection, firstly used LVQ to filter variables with low importance, then used RFE to further select 11 variables. After that, this paper used these variables to build a XGBoost model to predict the bank loan deposit. Through hyperparameters optimization, this paper achieved a final accuracy of 80.62%, which shows an excellent model performance. In the future, this model can be used in the bank loan decision systems to determine the deposits that paid by lenders.

Keywords: XGBoost, Learning Vector Quantization, Recursive Feature Elimination, Bank Loan.

1. Introduction

Machine learning is a cross subject applied a lot in the business marketing field. In financial field, machine learning has an increasing tendency to be used by helping predict the stock tendency, label different customers, accomplish the recommendation system and so on[1]. The use of machine learning and AI in business analysis allows organizations to process large volumes of data efficiently, uncover hidden patterns, and make accurate predictions[2].

Usually, linear regression, logistic regression, support vector machine and k-means clustering are several traditional ways in the marketing field. In terms of the bank loan,BayesNet and NaiveBayes which are classification algorithms were tested in dealing with loan data sets, but only achieved around 75% accuracy.According to past authors,multilayer perceptron was the best classification algorithm that was used in coping with the bank loan prediction. However,the multilayer perceptron is doomed to have overwhelming parameters when it comes to a large data set[3].

In this paper, an efficient method called XGBoost, which is a recursive gradient way based on decision trees, is used to solve the loan problems in banks. By now, this model has been applied in weather forecasting, water resources engineering , gold prices prediction and so on[4][5]. With the mighty XGBoost model, we can predict whether a man is able to pay back his loan according to his personal information accurately.

Also, unlike other researches that directly applied XGBoost model in the business field. Before training the XGBoost model, we used LVQ and RFE to eliminate unimportant features first. Thus, the accuracy will be higher and the whole training time will be cost. Apart from the model itself, those bankers can make policies based on this research. For example, when they calculate a man's credit, they can set more weights to features that are maintained in this paper and neglect other unnecessary features------features pruned by the LVQ and RFE.

2. Data Description

The data we use is the classic marketing bank dataset uploaded originally in the UCI Machine Learning Repository. We got the data from the kaggle website. Here is the link: https://www.kaggle.com/datasets/janiobachmann/bank-marketing-dataset?resource=download

Given a data set from the bank, there are 11162 samples in total which each has 16 characteristics, such as age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome. And the target value is deposit. We are going to predict whether a person has the deposit based on those 16 characteristics.

3. Data reprocessing

3.1. Data Normalization

Data Normalization is a way to mitigate the data bias and adjust the data range. Normalization of the data reduces the difficulty of the method for processing it, since there is typically a very high disparity between the dataset's greatest and lowest values[6]. It is conducive to those models which are sensitive to the size of inputs, such as XGBoost. In XGBoost model, we need the gradient descent method to reduce the cost, thus normalizing data in advance improving the convergence speed.

There are many normalization methods, like Z-score normalization,min-max scaling,robust scaling and log scaling. Min-Max normalization is used here for its simplicity. The formula is as below:

$$x = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

3.2. One-hot encoding

Among 16 features, there are 10 categorical features which cannot be discerned by the XGBoost model. We use one-hot encoder to transform categorical features into numerical features.

One-hot encoder can achieve this goal by turning each category value into a new binary column and assigning a 1 or 0 (notation for true/false) across these columns. Although this will add more feature columns resulting in the high dimension, it can meanwhile cause the data sparsity due to many zeroes, which facilitates the model computation.

4. Featue selection

4.1. LVQ

In this part, we want to reduce the amount of features, in order to simplify our model training and avoid the overfitting problem. We apply the LVQ and the recursive feature elimination(RFE) to do the feature selection.

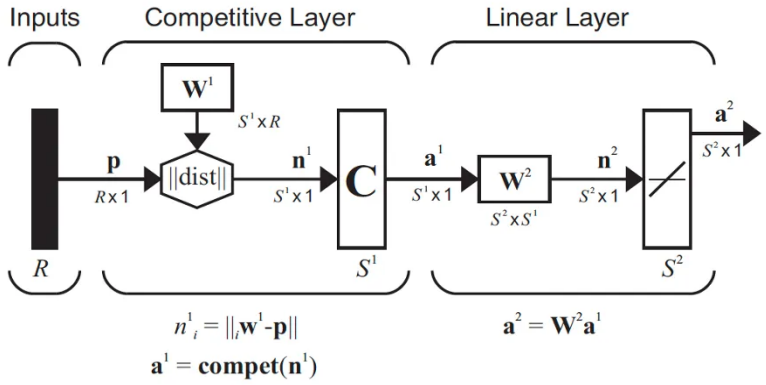


Figure 1: the general process of Learning Vector Quantization.

Theoritically, LVQ is used to classify, like K-means clustering[7]. LVQ is used here in order to rank the importance of each feature. LVQ network is a two-layered network. According to the figure 1, the first layer can be called the competitive layer and the second layer can be called the linear layer. The names of the layers are the result of the activation function used in that layer.

In the first fully connected layer, the goal is to find the nearest Euclidean distance between each weight and the input. And the activation function of this layer is called competition function, which will set the largest number of one vector to be 1 and set others to be 0. Thus the output of this layer is a vector containing only one and zeroes. In the graph ı, S^1 is the row number of the first weight matrix W^1, whereas R is the column number of the input.

In the second partly connected layer, the activation function is linear function, calculating the product of the second weight matrix W^2 whose size is $S^2 \times S^1$. W^2 is human designed according to the data and is unchanged during the whole training process.

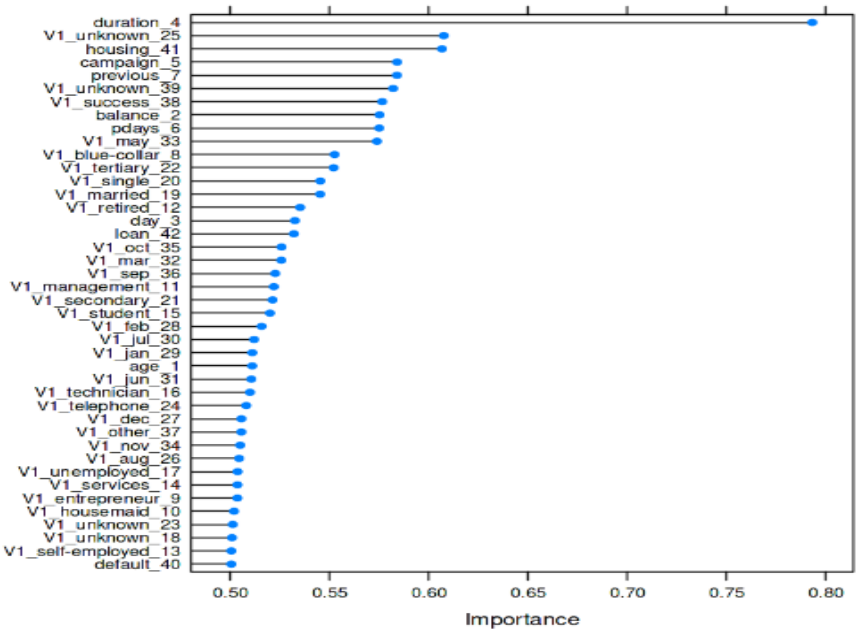


Figure 2: the order of each feature's importance by LVQ

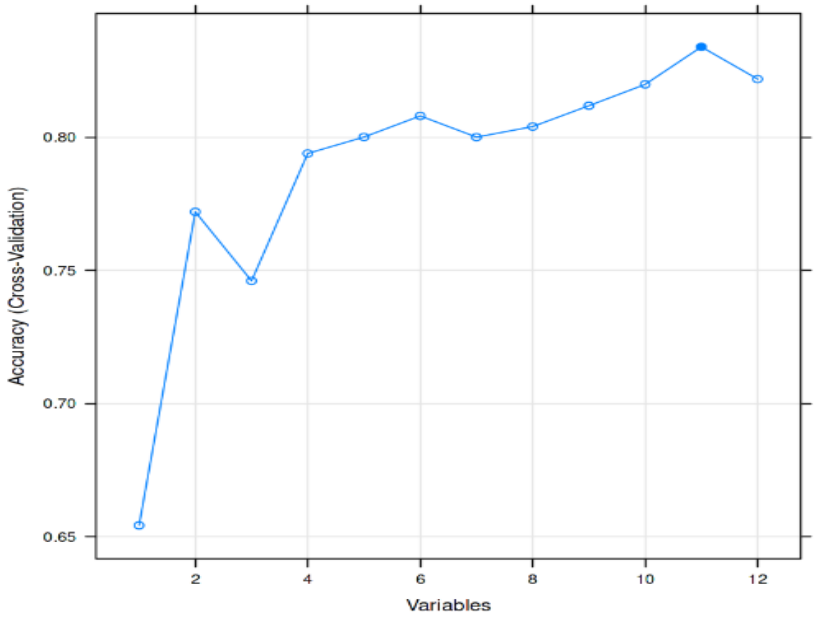


Figure 3: the result of Cross Validation

From the perspective of the clustering, neurons in the first layer are different clusters and the first weight W^1 is the location of the cluster center. The second weight W^2 is the class type of each cluster.

4.2. RFE

After the LVQ to rank features' importance, we get the figure 2 showing the order of each feature's importance. However, it is hard to solely select those important features. RFE is introduced here to do the feature selection.

RFE is a recursive way consisting of two steps. First, rank features according to their importance through a base model. Second, prune the least important feature. Iteratively do the process until it converges to a certain accuracy, which can be shown in a cross validation graph. Random forest model is chosen as a base model and choose 11features at last as the figure 3 shows, which are 'duration_4', 'V1_success_38', 'V1_unknown_25', 'pdays_6', 'housing_41', 'V1_may_33', 'V1_unknown_39', 'previous_7', 'balance_2', 'campaign_5', 'V1_tertiary_22'.

5. Related Works to XGBoost Model

5.1. Introduction to XGBoost

XGBoost is a variant algorithm of Boost. Compared with GBDT which only uses the first order Taylor expansion, XGBoost uses the second order Taylor expansion to derive the objective function and the parameter. In addition, the design of the objective function includes regular items in order to avoid overfiting by using the hyperparameter[8]. Especially in classification problems, in contrast to scikit-learn and R's GBM, XGBoost has a higher accuracy at the same time lowering the time[9].

In general, XGBoost continuously improves the prediction performance of the model by iteratively training the decision tree model and optimizing the loss function.And it has the advantages of handling high-dimensional sparse data, being robust to missing values, and evaluating feature importance[10].

5.2. Algorithms in XGBoost

First, start with the input and output of the XGBoost.

$$\{D\} = \{(x1, y1), (x2, y2), \ldots\ldots, (xn, yn)\} \tag{2}$$

This is the initial data set containing n samples which each has m eigenvalues. And the y'(the predict of y) is formulated as below:

$$y'_i = F(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{3}$$

f_k represents the k-th regression tree. When all eigenvalues are inputted in the k-th regression trees, a corresponding $z_k$will be outputted, which means the predicted value of the k-th regression trees. As all predicted values of all regression trees are summed up, the predicted value of y appears.

Second, define the objective function of the XGBoost. According to Sun.et, the formula is:

$$L(F(x_i)) = \sum_i l(y_i, y'_i) + \sum_k \Omega(f_k) \tag{4}$$

$l(y_i, y'_i)$ is the cost function, calculating the bias of the predicted value and the real value[11]. $\sum_k \Omega(f_k)$ is the so called regularized term.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||\omega||^2 \tag{5}$$

This formula includes two hyperparameter to avoid the overfiting. T is the amount of leaf nodes of each regression tree. $||\omega||^2$ is the square of the norm of a vector consisting of the number of each leaf nodes. Combining all formulas, a more detailed formula is derived.

$$\text{Obj}^{(t)} = \sum_i l(yi, y'_i{}^{(t)}) + \gamma T + \frac{1}{2}\lambda||\omega||^2$$

$$= \sum_i [\, l(yi, y'_i{}^{(t-1)} + f_t(x_i)] + \gamma T + \tfrac{1}{2}\lambda||\omega||^2 \quad (6)$$

Notice that we do a simple transformation according to the hierarchical regression trees. The formation of each regression trees is based on the previous regression trees. Thus, $y'_i{}^{(t)} = y'_i{}^{(t-1)} + f_t(x_i)$. (The superscript (t) means in the t-th regression trees). Then, the second order Taylor expansion is:

$$f(x) \sim f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \quad (7)$$

$$l(y_i, y'_i{}^{(t-1)} + f_t(x_i)) \approx l(y_i, y'_i{}^{(t-1)}) + l'(y_i, y'_i{}^{(t-1)}) f_t(x_i) + \frac{1}{2}l''(y_i, y'_i{}^{(t-1)}) f_t^2(x_i) \quad (8)$$

Combining the (6) and (8), the objective function can be approximated as the following (assume $g_i = l'(y_i, y'_i{}^{(t-1)}), h_i = l''(y_i, y'_i{}^{(t-1)})$):

$$\text{Obj}^{(t)} \approx \sum_{i=1}^{n} [l(y_i, y'_i{}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda||\omega||^2$$

$$= \sum_{j=1}^{T} [l(y_i, y'_i{}^{(t-1)}) + (\sum_{i \in I_j} g_i) w_j + \tfrac{1}{2}(\sum_{i \in I_j} h) f_t^2(x_i)] + \gamma T + \tfrac{1}{2}\lambda||\omega||^2 \quad (9)$$

($I_j = (i|\text{the ith sample is on the jth leave node})$)

Removing the constant part because it does nothing to the optimization, and make the formula more compact(w_j *is the weight of the jth leave node*):

$$\text{Obj}^{(t)} \approx \sum_{j=1}^{T} (G_j w_j + \frac{1}{2} H_j w_j^2) + \gamma T + \frac{1}{2}\lambda||\omega||^2 \qquad (G_j = \sum_i g_i, H_j = \sum_i h_i) \quad (10)$$

By now, we've already known how the XGBoost uses the second order Taylor expansion and the regularized term to make the cost function more precise.

6. Results

From all the samples, 70 percent of them were chosen as training samples and the others were as testing samples.

Table 1: the arguments of the XGBoost function

Parameter	Number Set
Max Depth	2
Eta	1
Nthread	2
Nrounds	50
Objective	Binary:logistic
Verbose	2

Table 2: the confusion matrix when the threshold is 0.5

Positive(predicted)	Negative(predicted)
Positive(actual)	1403
355	
Negative(actual)	298
1333	

Then we set a XGBoost model whose parameters are specifically written in the table 1. Max depth means the maximum depth of each decision tree, and a lower depth can avoid overfitting. Eta is the

learning rate. Nthread controls the number of threads used during training, affecting computational speed and nrounds controls the number of boosting rounds or iterations, affecting the model's complexity and performance. Also, we chose the logistic loss function as the loss function which is a common loss function used for binary classification problems. Such parameter can be fine-tuned in future works.

After training a XGBoost model, we began to predict the test samples, finding

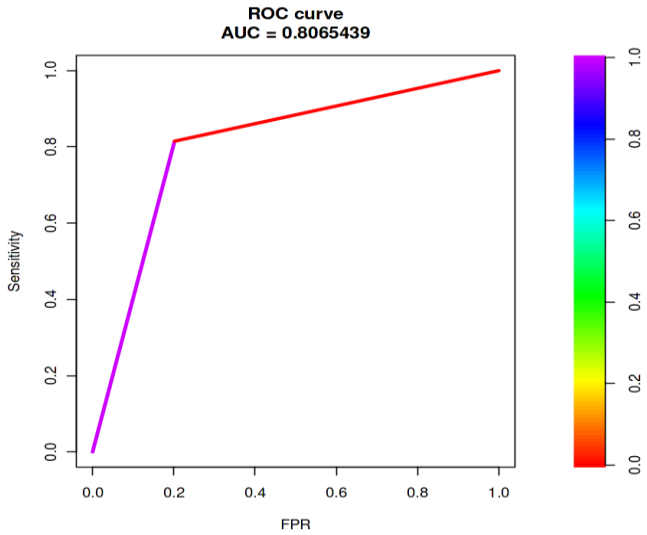


Figure 4: the ROC curve

that the accuracy is 80.62% and the confusion matrix shown in table 2 is quite satisfying. A confusion matrix is a table used to evaluate the performance of a classification model, typically in supervised learning. It allows you to see not only the errors made by the model but also the types of errors (false positives and false negatives). According to the table 2, true positive samples are 1403;true negative samples are 1333; false positive samples are 298;false negative samples are 355. This is the confusion matrix when the threshold is assigned as 0.5. When we set different thresholds between 0 to 1, different confusion matrixes will be produced. With these different confusion matrixes, the ROC curve can be drawn with the x-axis is false positive rate and the y-axis is true positive rate, like figure 4. AUC is the area under the ROC curve.

7. Conclusion

In this study, we implemented XGBoost model to predict bank loan approvals with a final accuracy of 80.62%. The model was trained using a bank load dataset, with features selected based on their relationship with loan approval outcomes. By optimizing hyperparameters such as 'max depth', 'learning rate', and 'the number of boosting rounds', we were able to achieve a robust predictive model. The results demonstrate the efficacy of XGBoost in handling complex datasets with high dimensionality and non-linear relationships. This paper contributes a novel approach to bank data analysis, distinct from traditional business analytics methods. The XGBoost model offers a more efficient and accurate solution of prediction. Researchers in this field can further fine-tune the XGBoost parameters to achieve even higher accuracy. Future work may focus on enhancing model accuracy through feature engineering and incorporating additional data sources.

References

[1] Al-Anqoudi, Y., Al-Hamdani, A., Al-Badawi, M., & Hedjam, R. Using machine learning in business process re-engineering[J]. Big Data and Cognitive Computing, 2021,5(4): 61.

[2] Bharadiya, J. P. Machine learning and AI in business intelligence: Trends and opportunities[J]. International Journal of Computer (IJC), 2023,48(1):123-134.

[3] Alsaleem, M. Y., & Hasoon, S. O. Predicting bank loan risks using machine learning algorithms[J]. AL-Rafidain Journal of Computer Sciences and Mathematics, 2020,14(1): 149-158.

[4] Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., Dhawan, P., & Righetti, M. Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023) [J]. Environmental Modelling & Software,2024,174: 105971.

[5] Frifra, A., Maanan, M., Maanan, M., & Rhinane, H. Harnessing LSTM and XGBoost algorithms for storm prediction[J]. Scientific Reports, 2024,14(1): 11381.

[6] Khang, A., Gupta, S. K., Dixit, C. K., & Somani, P. Data-driven application of human capital management databases, big data, and data mining[M]. CRC Press,2023: 105-120

[7] Shaffiee Haghshenas, S., Guido, G., & Astarita, V. Predicting Number of Vehicles Involved in Rural Crashes Using Learning Vector Quantization Algorithm[J]. AI, 2024, 5(3): 1095-1110.

[8] Jabeur, S. B., Mefteh-Wali, S., & Viviani, J. L. Forecasting gold price with the XGBoost algorithm and SHAP interaction values[J]. Annals of Operations Research, 2024, 334(1): 679-699.

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

[10] Yuan, Y., Du, J., Luo, J., Zhu, Y., Huang, Q., & Zhang, M. Discrimination of missing data types in metabolomics data based on particle swarm optimization algorithm and XGBoost model[J]. Scientific Reports, 2024,14(1): 152.

[11] Sun, Z., Li, Y., Yang, Y., Su, L., & Xie, S. Splitting tensile strength of basalt fiber reinforced coral aggregate concrete: Optimized XGBoost models and experimental validation[J]. Construction and Building Materials, 2024,416:135133.