Python Project
Loan Default Prediction Model
Comparison of KNN and Random Forest methods and Model Update Notification

Yixin Fan

Dec 15, 2021

Introduction

The credit scoring is a statistical analysis performed by lenders or financial institutions to determine whether they should lend money to borrowers. These lenders will collect data describing features of borrowers and use relative techniques such as various algorithm to make loan decision. This credit scoring project will perform a complete process of training credit scoring model including:

- Data preprocessing
- Construct models using K-Nearest Neighbour(KNN) and Random Forest and evaluate performance based on real life condition and make comparison.
- Model update Notification based on Credit Scoring background

Data Preparation

The dataset was collected from Bondora online lending platform with 179,235 samples and 112 features (Bondora, 2021). Features of borrowers include income, age, credit record and the target variable is 'Status', which contains three classes: 'Repaid', 'Late' and 'Current'. Several special points in data preparation are described as follow:

Target Variable The class 'Current' means that the loan is in progress and cannot be used to train the model. Hence, delete the sample with 'Current' status.

Variable Selections After deleting the variables containing more than 50% of the NA values, variables which contains the information that cannot be acquired before loan decision should be removed. This operation is significant since it can helps to avoid overfitting problem.

Pre-processing variables by data type The weight of evidence(WOE) is used while dealing with category variables. It is defined as the contribution of each class in category variable to target variable and has been widely used in credit risk. In this loan decision case, WOE of value x for variable X is calculated as below:

$$woe(x) = log(\frac{P(X = x|Y = 0)}{P(X = x|Y = 1)}) \tag{1}$$

However, there may be cases when $P(X = x|Y = 1) = 0$. Hence the well-defined woe is:

$$\hat{w}(x) = log(\frac{\hat{P}(X = x|Y = 0)}{\hat{P}(X = x|Y = 1)}) \tag{2}$$

$$= log(\frac{max(0.5, occurrence\ of\ (X = x, Y = 0))/occurrence\ of\ (Y = 0)}{max(0.5, occurrence\ of\ (X = x, Y = 1))/occurrence\ of\ (Y = 1)}) \tag{3}$$

In this project, WOE has applied to some of the category variables which contains too many classes, such as date variable.

Model Construction and Evaluation

This project will focus on two algorithms: KNN and Random Forest. The model performance evaluation will based on confusion matrix and cross validation. Both construction and evaluation will use *sklearn* package. Specially, considering the credit scoring background, it is more proper to use F1 score rather than precision rate, recall rate and accuracy. This is because that both Type I error and Type II error will result in loss to lenders and F1 score can better balance this two error. Below is confusion matrix:

	Observed Positive	Observed Negative
Predicted Positive	True Positive(TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Table 1: Confusion Matrix of Classifier Performance Outcome

The False Positive, also called as Type I error, means rejecting the good customers by mistake. The False Negative is regarded as Type II error, which accept the bad customer by mistake.

Model Construction

The K-Nearest Neighbour(KNN) A new data sample will be classified based on its n closest neighbour. As Figure 1 shown, the sample will be classified as class 1 since there are 3 of them among the 5 neighbours.

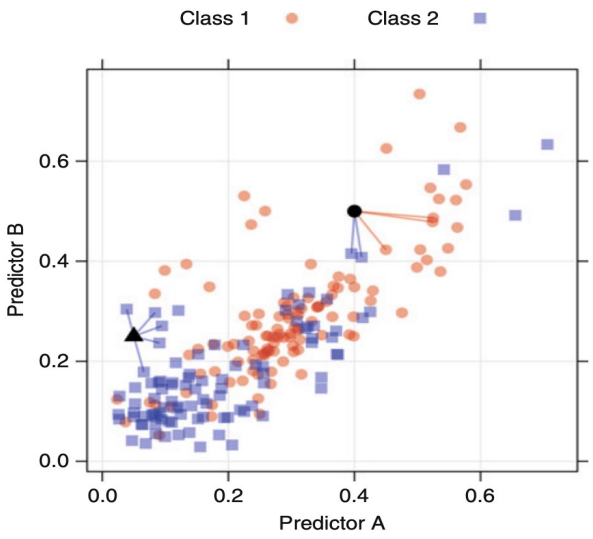

Figure 1: KNN Algorithm for Binary Classification (Kuhn and Johnson, 2013)

- **Problem:** The F1 score of KNN algorithm with 5 neighbour is only 0.61 approximately. Comparing to the F1 score of random forest with 0.91, there seems to be some problems in data and parameter choosing
- **Solution:** The probable reason for this result is because random forest to do need scaling the dataset because it is a tree-based model. However, model like KNN is distance-based and hence need to be normalized. Hence, perform outlier deletion and zero-mean normalisation to dataset.
- **Optimal Hyperparameter:** Try a range of hyperparameter values and found F1 score reaches a maximum of 87.06% when number of neighbors is 7. Plot F1 score against hyperparameter(neighbours):

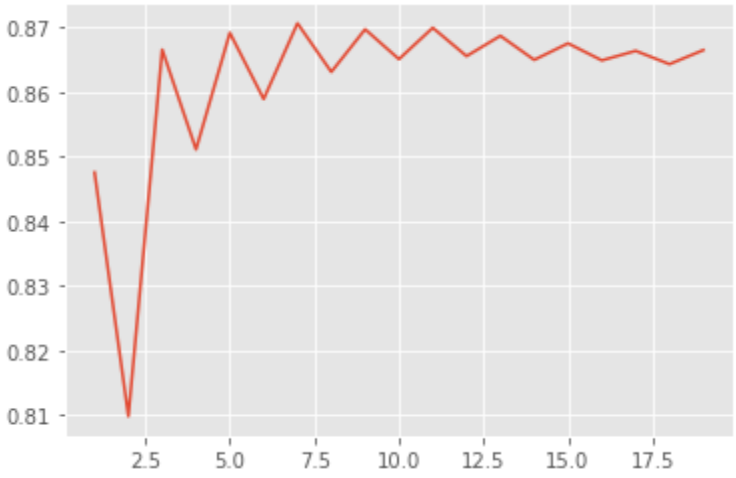

Figure 2: F1 Score vs Hyperparameter Choice

Random Forest The algorithm is described as below(Kuhn and Johnson, 2013):

Select the number of trees built, m
for i=1 to m **do**
Generate a bootstrap sample of the original data
Train a tree model on this sample
for each split **do**
Randomly select k features
Select the best one among the k feature and decide the best threshold. Split the data based on the best feature and its threshold
end for
Use typical decision tree model stopping criteria to determine when a tree is completed
end for

General Model Evaluation

KNN

	Observed Positive	Observed Negative
Predicted Positive	10119	1886
Predicted Negative	1455	12354

Table 2: Confusion Matrix for KNN

- Cross-Validation: The average F1 score of KNN is 0.8621.

0.7524	0.7206	0.8288	0.8832	0.9032	0.9074	0.9043	0.9014	0.9032	0.9160

Table 3: F1 score in 10-fold cross validation for KNN

Random Forest

- Confusion Matrix for Random Forest:

	Observed Positive	Observed Negative
Predicted Positive	13897	2226
Predicted Negative	708	16617

Table 4: Confusion Matrix for KNN

- Cross-Validation: The average F1 score of Random Forest is 0.9115.

0.8818	0.8582	0.8606	0.8877	0.9207	0.9328	0.9271	0.9359	0.9445	0.9658

Table 5: F1 score in 10-fold cross validation for Random Forest

It could be summarized that random forest is a better choice in this project:

1. Random forest has better F1 score than KNN.
2. Its data preparation process is much simpler, normalisation not necessary.
3. It do not need to find the optimal hyperparameter. During KNN training, it took a long time to find the optimal hyperparameter.

Model Update Notification based on Credit Scoring background

In credit risk, the change of economic environment will affect the efficiency of the model and hence KS-test and PSI are commonly used to test whether the model is still efficient to distinct customers. This section will demonstrate the theory of KS-test and PSI and test the three models were created in model construction section: one knn model with disappointing performance and two models with well performance.

Kolmogorov–Smirnov test (KS-test) KS-test helps to measure the ability of this model distinct 'good' and 'bad' customer. If the KS-statistic<0.25, meaning that this model do not have enough ability to make loan decision and hence need to use more recent data to update the model.

- Method: KS-statistic is calculated directly by using ks_2samp function in Scipy package.
- Result:
  1. The KS Statistics for KNN with optimal hyperparameter is: 0.7375 > 0.25
     The model is good enough to distinct customers and no need to be updated.
  2. The KS Statistics for KNN with 5 neighbours is: 0.0833 < 0.25
     The model need to be updated. It do NOT have enough ability to distinct customers.
  3. The KS Statistics for Random Forest is: 0.8211 > 0.25
     The model is good enough to distinct customers and no need to be updated.

Population Stability Index(PSI) PSI can describe how much change of a variable change. It could be used in monitor changes in features and measure the potential model performance. The formula of calculating PSI is:

$$PSI = \sum_{i=1}^{n} (Actual\% - Prediction\%) * log(\frac{Actual\%}{Prediction\%}) \tag{4}$$

When PSI>0.1, it means model is unstable and need to update the model by using more recent data.

- Method: PSI is calculated by writing self-defining function based on the formula mentioned(eq.4)
- Result:
  1. The PSI for KNN with optimal parameter is: 0.0011 < 0.1
     The model is stable and no need to be updated

2. The PSI for KNN with 5 neighbours is: 0.5547 > 0.1
   The model need to be updated since it is not stable enough.

3. The PSI for Random Forest is: 0.0083 < 0.1
   The model is stable and no need to be updated

Conclusion

In summary, this project has display the whole process on training and modify credit scoring model. Even though Random forest has shown better performance in this case, it could not be concluded that Random forest is the only choice. In reality, the algorithm choice is not only based on theoretical performance measure(such as accuracy or F1 score), but also rely on the reality consideration(such as insufficient dataset and low data quality).

References

[1] Bondora (2021). *Bondora Loan Book*. Bondora.com. URL: `https://www.bondora.com/en/public-reports` (visited on 08/22/2021).

[2] Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. URL: `http://appliedpredictivemodeling.com` (visited on 2021).