Martinez P. M June 11th, 2021

Modeling the Credit One database

The problem to resolve is that there is an increase in customer default rates of Credit Ones company. The goal is to build a model that predict what credit limit should be assigned to a customer. We are addressing the following questions:

1. How ensure that customers can/will pay their loans?
2. Is it possible to approve customers with high certainty?

The cleaned Credit One database contains 24 columns and 30,000 non-null rows (observations, see Table 1). From the previous data exploration, I identified the column "LIMIT_BAL" as the dependent variable and the rest as the features. Using this database, I built a pipeline to model the data. The pipeline includes three machine learning regression algorithms. They are, Random Forest Regressor, Linear Regression, and Support Vector Regression.

I tried the three models assuming a minimal number of parameters, most of them with the values by default, and compared the coefficient of determination (namely score) of the model regression (R2 or r2_score function in skitlearn). Then, I increased and decreased the complexity of the models, by trying several grids of parameters, and compared the scores. In all cases I noted that the scores didn't improve more than ~0.4 (note that a good score that indicate a good performance and prediction of the model should be around 1).

In order to improve the model performance, I investigated the feature importance for the Random Forest Regressor algorithm (which give the less bad score, R2~0.4). However, after dropped the less important features and trained again, the model score did not improve.

Due to the difficulties in improving the model performance, I explore some classification models. Sine with the current analysis, it is impossible to answer the questions.

Figure 1: features importance from the Random Forest Regressor algorithm

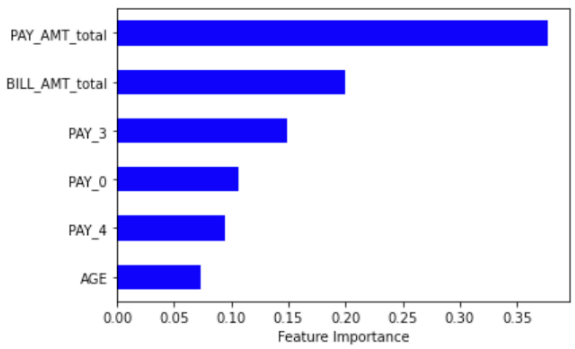

Table 1: basic info of the data

#	Column	Non-Null Count	Dtype
0	LIMIT_BAL	30000 non-null	int64
1	DPNM	30000 non-null	int64
2	SEX	30000 non-null	int64
3	EDUCATION	30000 non-null	int64
4	MARRIAGE	30000 non-null	int64
5	AGE	30000 non-null	int64
6	PAY_0	30000 non-null	int64
7	PAY_2	30000 non-null	int64
8	PAY_3	30000 non-null	int64
9	PAY_4	30000 non-null	int64
10	PAY_5	30000 non-null	int64
11	PAY_6	30000 non-null	int64
12	BILL_AMT1	30000 non-null	int64
13	BILL_AMT2	30000 non-null	int64
14	BILL_AMT3	30000 non-null	int64
15	BILL_AMT4	30000 non-null	int64
16	BILL_AMT5	30000 non-null	int64
17	BILL_AMT6	30000 non-null	int64
18	PAY_AMT1	30000 non-null	int64
19	PAY_AMT2	30000 non-null	int64
20	PAY_AMT3	30000 non-null	int64
21	PAY_AMT4	30000 non-null	int64
22	PAY_AMT5	30000 non-null	int64
23	PAY_AMT6	30000 non-null	int64

For the classification approach, we assumed three algorithms, RandomForestClassifier, GradientBoostingClassifier, and DecisionTreeClassifier.
I started discretizing some variables, like, for example, the AGE and LIMIT_BAL. I also took into account all features and used the LIMIT_BAL as the dependent variable. Then, I ran the pipeline and compared the scores. I found that all models give a poor accuracy of 40%. Therefore, to improve the models accuracy, I dropped the less important features, used the DPNM as the dependent variable, and tried different data splits (e.g., 20% and 30% for the test sample), sample sizes (e.g., 1000, 10000 and, 300000), and bins (e.g. 10 and 5) for the discretization of the variables.
It is important to mention that using another target as the dependent variable (e.g., the AGE or PAY_0) resulted in a poorer accuracy, while a sample three times smaller than the original produced similar results. Additionally, discretizing the AGE and LIMIT_BAL did not improve the models performance.

I used the important features listed in Table 2 and the DPNM target as the dependent variable. Then, to find the combination of parameters that produce the best model, I explored a grid of parameters for each model—using the cross_val_score function with the key_word scoring as 'accuracy'. I found that the three models give similar accuracy (82%).
On the other hand, I trained the model, assuming 20% for the test data (80% for the train data), and inspected the classification report for each model. I noted that the GradientBoostingClassifier algorithm produces the best combination of accuracy and precision on each class (see Table 3), becoming the algorithm in predicting the largest number of individuals that default or not, see the confusion matrix in Table 4 and the Jupyter notebook.

Table 2: features used to train the models

	LIMIT_BAL	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT_total	PAY_AMT_total
25665	0	-1	0	0	0	-1	0	129703	35353
16464	1	0	0	0	0	0	0	349621	12830
22386	2	2	2	2	2	2	2	972809	32800
10149	3	-2	-2	-2	-2	-2	-2	24305	742
8729	0	0	0	0	0	0	0	53551	3648

Table 3: classification report for the GradientBoostingClassifier

```
print(classification_report(y_test, y_predict_gradientb))
```

```
              precision    recall  f1-score   support

           0       0.87      0.95      0.91       158
           1       0.71      0.48      0.57        42

    accuracy                           0.85       200
   macro avg       0.79      0.71      0.74       200
weighted avg       0.84      0.85      0.84       200
```

Table 4: confusion matrix for the GradientBoostingClassifier

Actual / Predicted	Not Default	Default
Not Default	4484	181
Default	912	423

Finally, we found that the model predicts that people with default have, on average, a limit balance lower than 500,000$ and delay payments since the first month, for one, and until eight months (see Jupyter notebook. file). Additionally, the model predicts that total payment six months later is nearly ten times lower than those without default (see Figures below). Therefore, considering that the group with default has a limit credit similar (~1.4 times lower) to the group without default, plus the difficulties of the former group in making payments, we suggest reducing the amount of credit by, for example, a factor of five.

In summary, reducing the balance limit will improve the possibility that customers can pay their loans. While, it is no possible to predict when a customer will make default.