Credit Card Default Predictive Modeling

Background:

Predicting credit card payment default is critical for the successful business model of a credit card company. An accurate predictive model can help the company identify customers who might default their payment in the future so that the company can get involved earlier to manage risk and reduce loss. It is even better if a model can assist the company on credit card application approval to minimize the risk at upfront. However, credit card default prediction is never an easy task. It is dynamic. A customer who paid his/her payment on time in the last few months may suddenly default his/her next payment. It is also unbalanced given the fact that default payment is rare compared to non-default payments. Unbalanced dataset will easily fail using most machine learning techniques if the dataset is not treated properly.

Data:

In this demo, we showcase our solution by using a public dataset which contains 30,000 credit card accounts. Among the total 30,000 accounts, 6636 accounts (22%) are cardholders with default payments. The response variable of this study is ***default payment*** (Yes=1, No=0). Our goal is to build an accurate classifier to predict if a credit card account will default or not. Below you can see the first 5 records of this dataset:

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6
1	20000	2	2	1	24	2	2	-1	-1	-2	-2
2	120000	2	2	2	26	-1	2	0	0	0	2
3	90000	2	2	2	34	0	0	0	0	0	0
4	50000	2	2	1	37	0	0	0	0	0	0
5	50000	1	2	1	57	-1	0	-1	0	0	0

	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1
1	3913	3102	689	0	0	0	0
2	2682	1725	2682	3272	3455	3261	0
3	29239	14027	13559	14331	14948	15549	1518
4	46990	48233	49291	28314	28959	29547	2000
5	8617	5670	35835	20940	19146	19131	2000

	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default.payment.next.month
1	689	0	0	0	0	1
2	1000	1000	1000	0	2000	1
3	1500	1000	1000	1000	5000	0
4	2019	1200	1100	1069	1000	0
5	36681	10000	9000	689	679	0

Figure 1. First 5 records of the dataset

Specifically, this dataset contains 23 explanatory variables (X1 – X23) as follows:

- X1: *Amount of the given credit (dollar): it includes both the individual consumer credit and his/her family (supplementary) credit*
- X2: *Gender (1 = male; 2 = female)*
- X3: *Education (1 = graduate school; 2 = university; 3= high school; 4 = others)*
- X4: *Marital status (1 = married; 2 = single; 3 = others)*
- X5: *Age (year)*

- X6–X11: *History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .; X11 = the repayment status in April, 2005.*
- X12–X17: *Amount of bill statement (dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005*
- X18–X23: *Amount of previous payment (dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .; X23 = amount paid in April, 2005*

More information about this dataset can be found in Yeh's paper [1].

Benchmark:

As discussed in [1], Yeh and Lien reviewed 6 popular machine learning models and tested their performance on predicting default payment. Since this is an unbalanced dataset (22% payment default vs. 78% non-default), error rate is not an appropriate measurement of model performance. A dummy model which classifies all account to non-default will easily achieve 22% error rate, although the model can be completely useless.

To fairly compare model performance, this study randomly split the dataset into 50% training set and 50% validation set. The most critical measure will be the lift area ratio on the validation set. Figure 2 shows a lift curve. X axis of Fig. 2 represents the percentage of credit card accounts we examined. Y axis represents the percentage of default accounts we found. In an ideal case, each credit card account that we review is default, so that we only need to review 22% of all credit card accounts to find all the default cases, which is the left line of the shaded area. For a random classifier, we will need to review all credit card accounts to find all the default cases, which is the diagonal of Fig. 2. The curve in between represents the performance of a predictive model. Lift area ratio is calculated by the area between the curve and the diagonal over the entire shaded area. The larger the ratio, the better the performance.

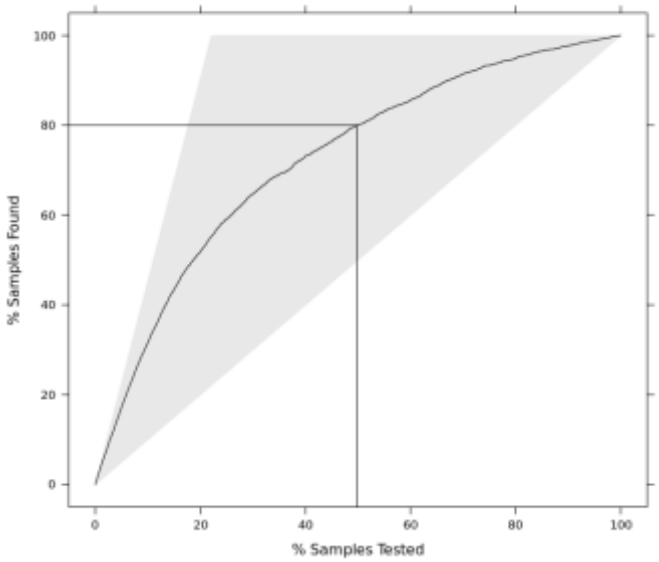


Figure 2. Lift Curve

Table 1 below summarizes the performance of 6 models on this dataset [1]. Among them, Neural Networks achieved the highest area ratio (0.54) on the validation set.

Table 1
Classification accuracy

Method	Error rate		Area ratio	
	Training	Validation	Training	Validation
K-nearest neighbor	0.18	0.16	0.68	0.45
Logistic regression	0.20	0.18	0.41	0.44
Discriminant analysis	0.29	0.26	0.40	0.43
Naïve Bayesian	0.21	0.21	0.47	0.53
Neural networks	0.19	0.17	0.55	0.54
Classification trees	0.18	0.17	0.48	0.536

In the next section, we will introduce how the data scientists in Vista Analytics build a predictive model which significantly outperform Neural Networks on the validation set.

Solution:

To build a better model, we spent our effort on **feature engineering** and **ensemble models**.

In predictive modeling, designing good features is a critical step to build a solid model, which prevents the garbage in, garbage out phenomenon from happening. In this study, we derived a few new features which we believe (and confirmed by experiments) will help to boost the performance of the predictive model. For example, we introduced a new feature called BILL_AMT_1_OVER_2, which is BILL_AMT1 divided by BILL_AMT2. BILL_AMT1 is the amount of bill statement (dollar) of the current month, while BILL_AMT2 is the amount of bill statement of the previous month. This new feature captures the ratio of these two variables and indicates the short-term billing trend of the current month. Similarly, we introduced another feature called BILL_AMT_1_OVER_ALL, which is BILL_AMT1 divided by the mean of BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT4 and BILL_AMT5.

In additional to feature engineering, we also leveraged a better machine learning technique, **Gradient Boosting Machine (GBM)**, to build our predictive model. GBM is a machine learning technique for both classification and regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

In our setting, we trained a GBM model based on the training dataset with new features, and then measured the area ratio of the trained model on the validation set. Figure 3 shows the performance of the GBM model on training dataset under different settings. We used a 10-fold cross validation to select the best parameters combination that produce the GBM model with the largest AUC (area under the ROC curve). Specifically, the highest 10-fold cross validation AUC, 0.774, can be achieved when max tree depth is set to 9, minimum observation inside each leaf node is set to 10, boosting iteration is set to 50.

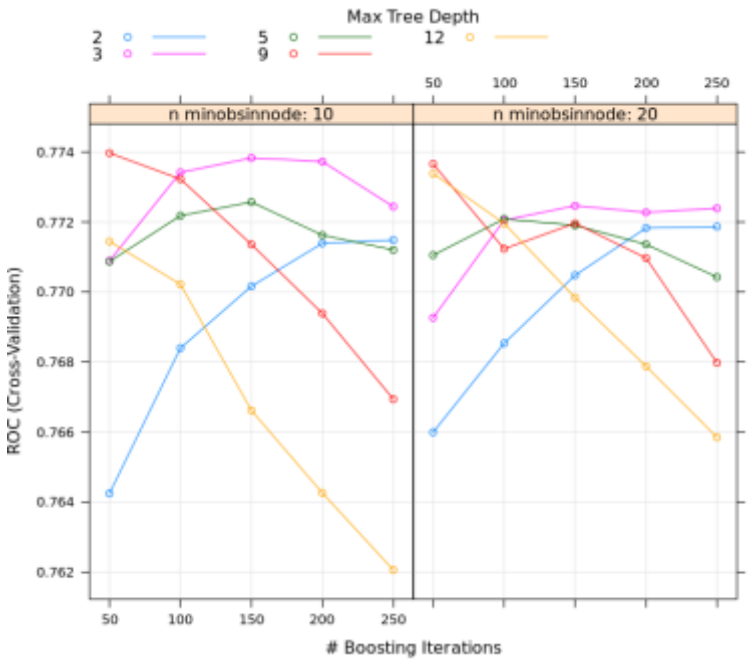


Figure 3. AUC under different parameter settings

We then used the best GBM model to predict payment default on the validation dataset. Area ratio is calculated and benchmarked with the methods mentioned in [1]. Figure 4. showed that by using GBM and our new features, we built a predictive model outperformed the neural network by 8.8%.

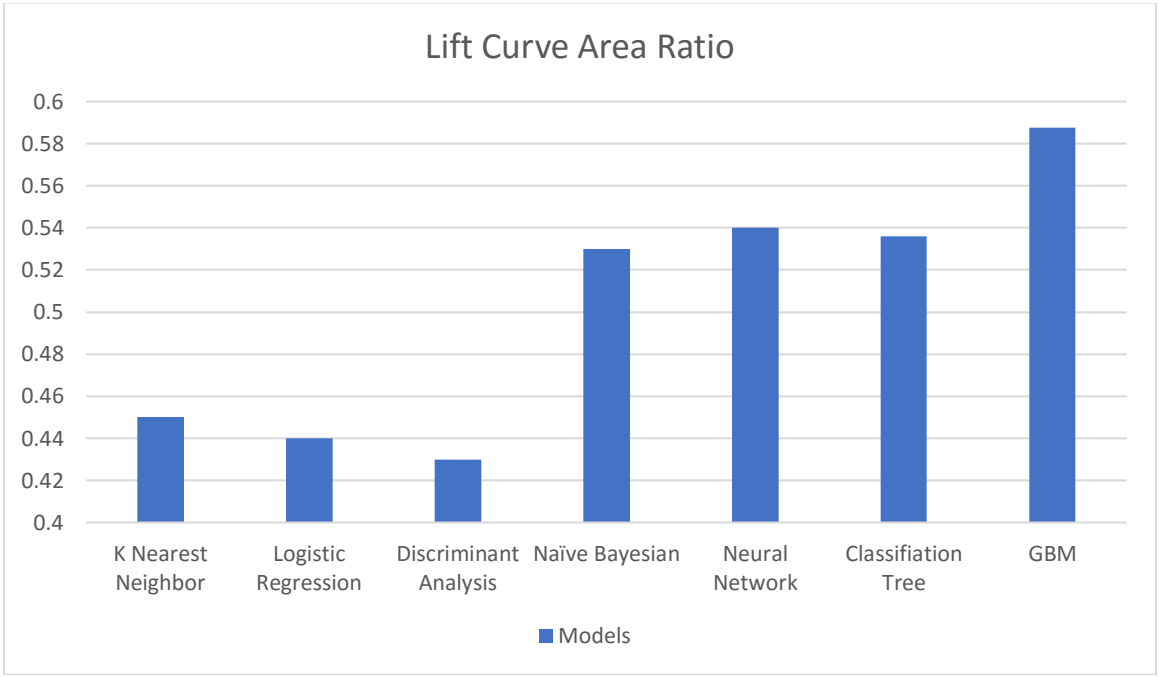
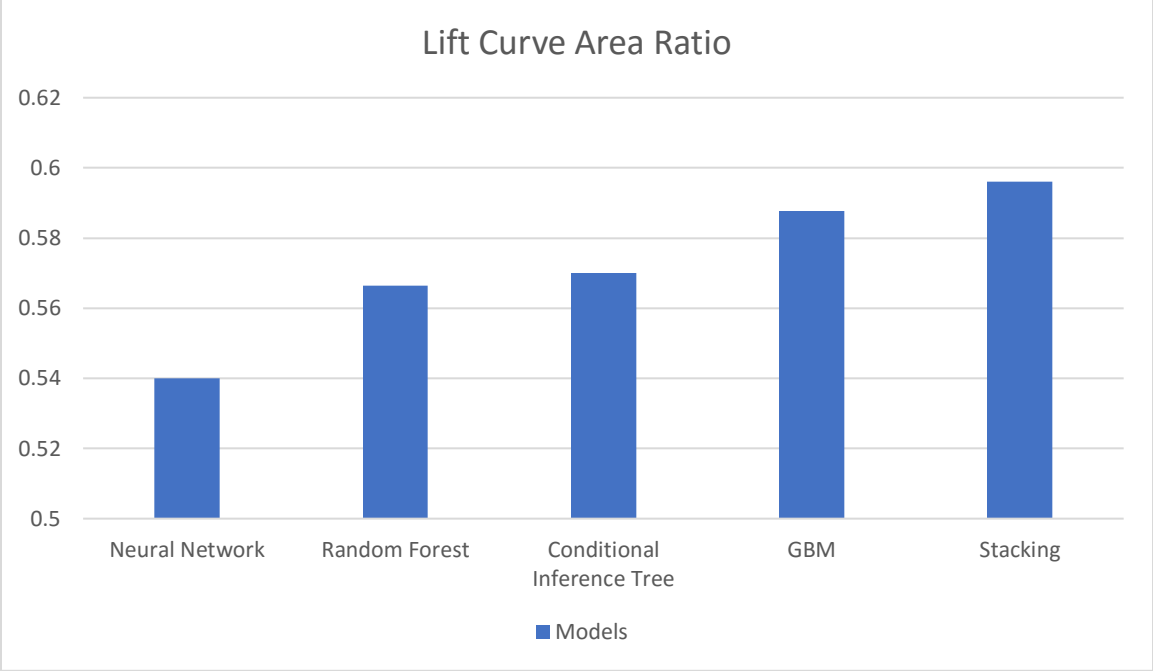


Figure 4. Models performance on validation dataset

GBM as a single model outperformed all other techniques mentioned in [1]. But, can we build a better model to beat GBM? As you will imagine, every little improvement in the area ratio translates to millions of dollars. The answer is Yes! To achieve this goal, we leverage **ensemble models**. General speaking, besides GBM, we trained another two predictive models using random forest and conditional inference tree. Like what we did in training GBM, we also fine-tuned random forest and conditional inference tree in a 10-fold cross validation, and searched for the parameters that produce the best models. After that, we used a technique in machine learning called stacking which involves training a learning algorithm to combine the predictions of GBM, random forest, and conditional inference tree. Theoretically, this ensemble model will outperform the individual models and further boost the result, especially when the underlying models do not usually agree with each other. As shown in Figure 5, stacking achieved 0.596 in lift curve area ratio which is 10.3% higher than the best model in [1] and 1.5% higher than GBM.



Reference:

[1] Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.