**KDD CUP 2009 Analysis Report**

Xin Yang, Hui Ge, Aidong Huang, Ziyan Zou

University of Texas at Dallas

1. **INTRODUCTION**

Customer Relationship Management (CRM) is an important element of modern marketing strategies. The data sets are from the French Telecom company Orange. It will be used to predict the probability of customers to switch provider (churn), buy new products or services (appetency), or buy upgrades or add-ons proposed to them to make the sale more profitable (up-selling).

Producing scores is the most practical method in a CRM system to build information of customers. The model extracts numbers of features from customers’ records and provides an output as a score of evaluation, to predict the interest of customers, i.e. churn, appetency or up-selling. Then the information system (IS) uses the scores to personalize the customer relationship. A robust detection of the most predictive variables is a significant factor for marketing application. Orange marketing is using Orange Labs to build predictive models for big datasets which have a large number of input variables and instances. An automatic process becomes the key requirement. A subset of informative variables and instances are extracted from great amount of variables and an accurate classifier is efficiently built. This platform implements several processing methods for instances and variables selection, prediction and indexation based on an efficient model combined with variable selection regularization and model averaging method.

This project is to beat the in-house system developed by Orange Labs. We handle a very large database, including heterogeneous noisy data (categorical and numerical variables), and unbalanced class distributions. A smaller database is then is used to enter the project with limited computer resources.

1. **BACKGROUND AND MOTIVATIONS**

In this project, important marketing problems are used to benchmark classification approaches in a situation with typical of large industrial applications. The unusually large dataset with a large amount of variables brings more difficulties to traditional machine learning methods. We use masked customer records provided and the goal was to predict whether a customer will switch provider (churn), buy the main service (appetency) and/or buy additional extras (up-selling). It is a three binary classification problems, where churn is the propensity of customers to switch between service providers, appetency is the propensity of customers to buy a service, and upselling is the success in selling additional good or services to make a sale more profitable. It is technically difficult to scale up the existing algorithms. And the dataset is heterogeneous, with numerical and categorical variables, including noisy data and missing values. The distribution of predictive variables is unbalanced. And only 1 to 7 percent of the examples belong to the positive class. All of these contribute to the difficulty of the project.

1. **MODELING PURPOSE**

The goal of this project is to find the most effective model and the set of best parameters of that model to predict a future customer’s likelihood of churn, appentency or up-selling using the test or available dataset. As each dataset has binary output, models used are those applicable to binary classification, such as Naive Bayes, Logistic Regression, Random Forest, Decision tree and some ensembled technology like adaboosting and bagging.

1. **DATASET**

There are two dataset packages. The small dataset package contains one training set, one test set, and three target sets (churn, appetency, up-selling). The large dataset package contains 5 training sets, 5 test sets and three target sets (churn, appetency, up-selling). For the small dataset, the first 190 variables are numerical and the last 40 are categorical. For the large dataset, the first 14,740 variables are numerical and the last 260 are categorical. The small data set is designed for personal computers, the large data set is designed for more powerful computing systems. Due to the limitation of computing power of laptop, we choose to use the small dataset for training purpose.

1. **PREPROCESSING**

We performed different data clean-up strategies for numerical variables and categorical variables.

**5.1 Missing Values**

We firstly checked the number of missing values in the dataset. We found that there are lots of features having more than 90% missing values. We choose to delete those features with 90% or more missing values, since we cannot get enough information from limited data points of those features. For those numerical variables who have a reasonable percentage of missing values, we chose to fill the missing value with the mean value of that feature firstly and then perform standardization in a feature-wise manner.

**5.2 Categorical values**

There are many categorical variables in the dataset. A smart strategy is required to deal with those categorical variables. First, we checked how many categories each categorical variable has. We noticed that some categorical variables have over 500 categories. We decided to delete those categorical variables with more than 500 categories. Secondly, missing values also exist in some categorical variables. Our strategy is to create another category named as “Missing” to represent the missing values if there is any in each categorical variables. Thirdly, some categorical variables has over 100 categories, with each categories only have few instance in the dataset. We believe that too few instances will not have enough power to explain variation in the dataset, we, therefore, merge those categories with few instances in the dataset into one category, ‘OTHER’.

**5.3 Feature Engineering**

In case some of the features are highly linear related, we use PCA method to combine features with high linear relationship to avoid overfitting. We tested a series of number of principal components to explain most variations of the dataset with a reasonable amount of features.

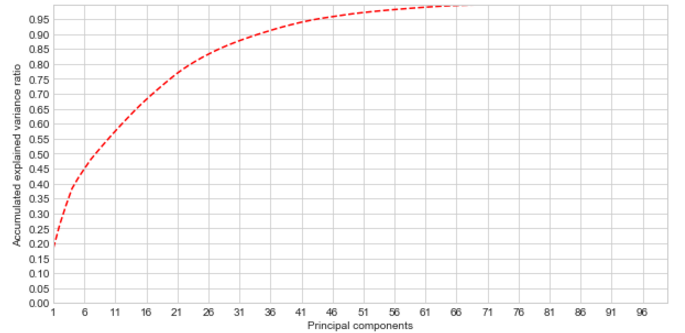


Fig.1. PCA results

As we can see from Figure 1, choosing 32 principal components for PCA is able to explain most of variation, over 85%, of the training dataset. Even though choosing more than 32 principal components can explain more variation, for the purpose of reducing dimensionality, we stick with our decision to use 32 principal components.

1. **PROCESSING**

After the data is cleaned up, we apply our processing method to it.

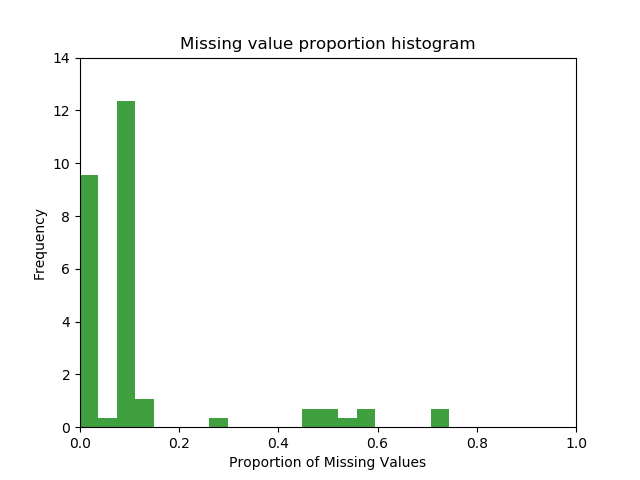
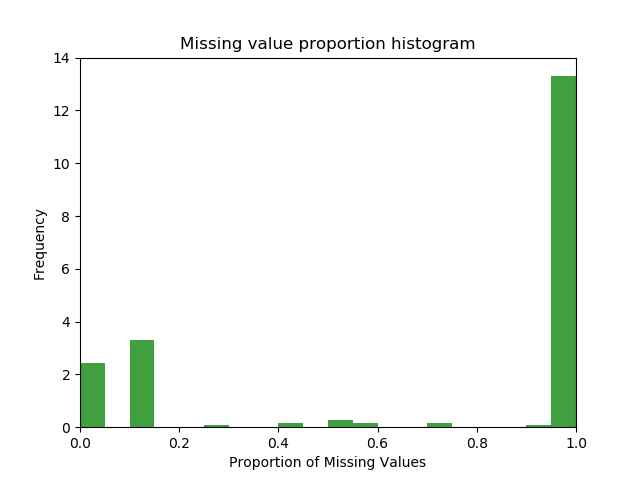
The training and validation data sets are generated randomly from the original train data sets. The label-files are the output files.

Multiple models such as Naive Bayes, Logistic Regression, Random Forest, Decision tree and some ensembled technology like ada boosting and bagging are trained by the training sets and validated by the validation sets. GridSearch API is applied to find the best parameters for those models which have multiple parameters.

Then all the models with best parameters are tested by the test datasets. Models are evaluated by accuracy and ROC curve at the end. ROC curves assess predictive behavior independent of error costs or class distributions.

1. **RESULTS**

The same preprocessing and model training process was applied to churn, appentency and up-selling datasets. The following figures show the frequency values based on missing value proportion before and after removing missing values.



(a) (b)

Fig.2. (a) Before removing missing values; (b) After removing missing values

Table 1 shows the basic statistics for those three datasets after data cleaning.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Target Bias  Yes/No | # of Feature Remain | # of Input | Variance |
| Churn | 3672/46328 | 52 | 32 | 0.8851 |
| Appetency | 890/49110 | 52 | 32 | 0.8851 |
| Up-Selling | 3682/46318 | 52 | 32 | 0.8851 |

Table 1: Basic Statistic of Preprocessed Dataset

**7.1 Results in Accuracy**

Figure 3 shows the statistic results for models on dataset churn. The score represents the accuracy of each classifier, and training and testing time are also shown. From Fig.1 the accuracy of all six classifiers are similarly around 90%, with slightly higher score in Random Forest, Logistic Regression and Voting classifier, and slightly lower accuracy in Gaussian Naive Bayes. Taking running time into consideration, Adaboost Classifier uses most training time, while Voting Classifier uses most testing time. Logistic Regression cost least time in general.

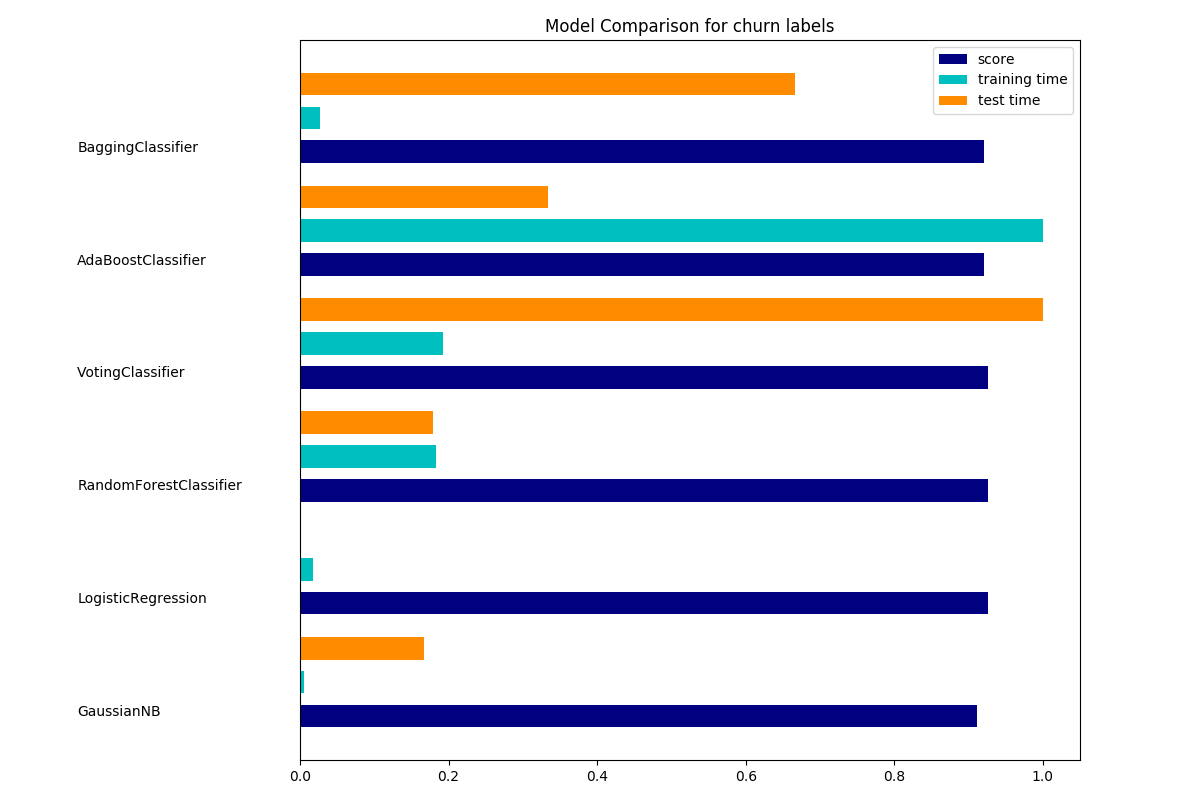


Fig.3. Accuracy & Time Cost of all the Classifier for Dataset Churn

Figure 4 shows the statistic results for models on dataset appentency. Again, the accuracy of all six classifiers are similarly around 96%. Although with slight difference, Logistic Regression and Voting Classifier works best and Gaussian Naive Bayes works relatively worst. The training time and testing time difference is similar to dataset churn.

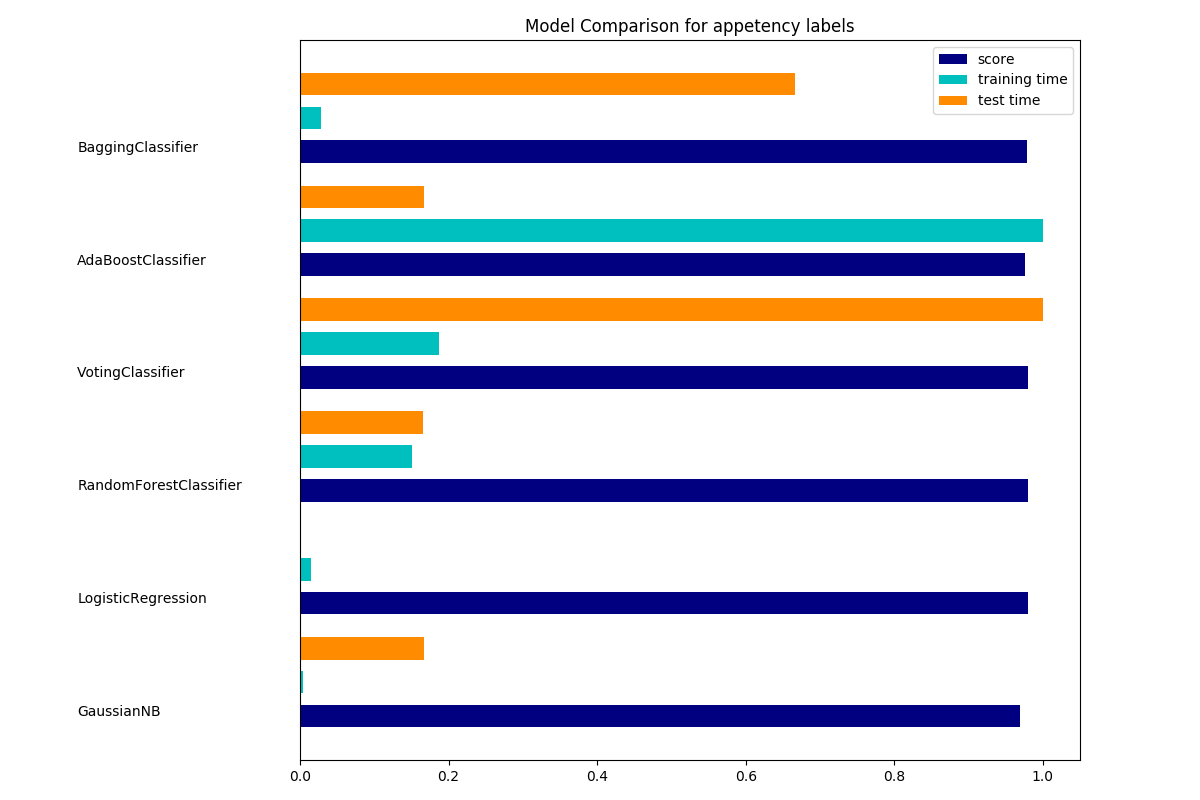


Fig.4. Accuracy & Time Cost of all the Classifier for Dataset Appentency

Figure 5 shows the statistic results for models on dataset up-selling. Here the simplest model, Gaussian Naive Bayes, provides an accuracy of 85%, while the other classifiers improve the accuracy to around 92%. AdaBoost classifier works slightly worse than other classifier. Combining accuracy and running time, Logistic Regression shows the best performance.

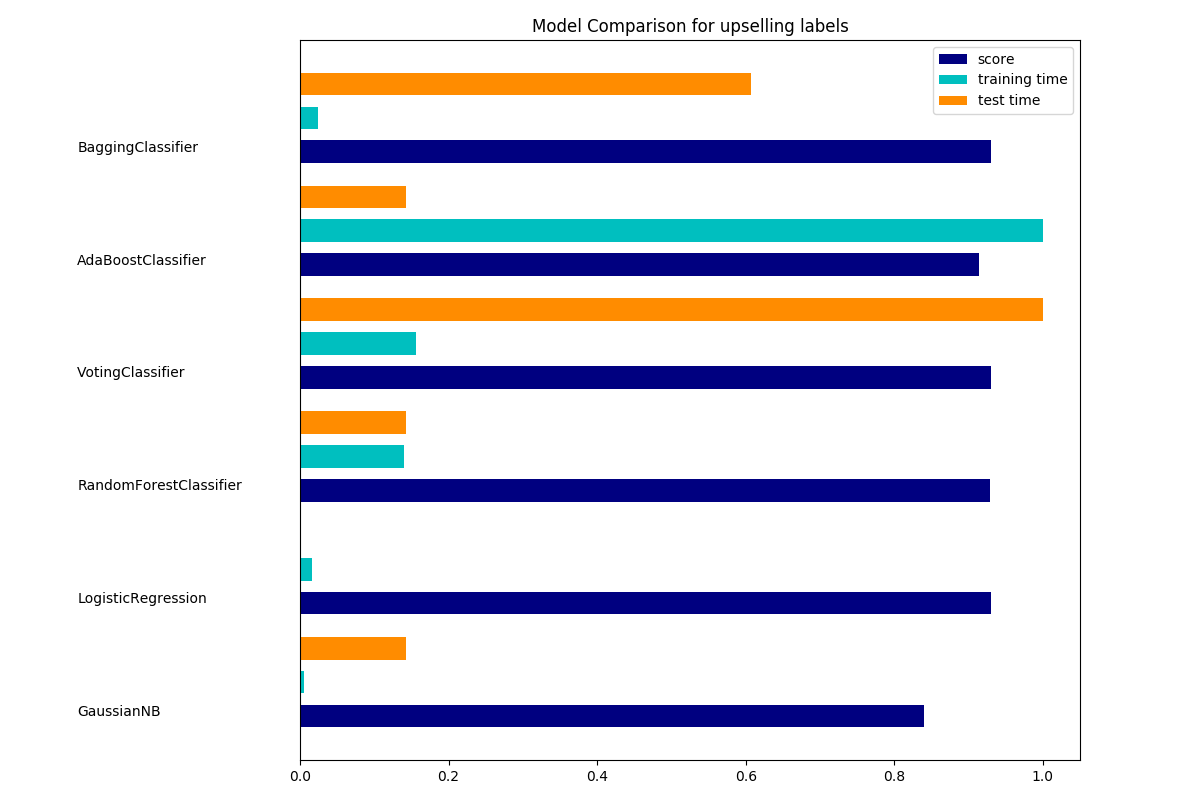


Fig.5. Accuracy & Time Cost of all the Classifier for Dataset Up-selling

**7.2 Results in ROC curves.**

Figure 6 shows the ROC curves for models on dataset churn. The blue diagonal line means the random selection, which provides a base for evaluation. For dataset churn, all the classifiers give similar ROC, which are around random line. The AdaBoost Classifier is slightly lower than others.

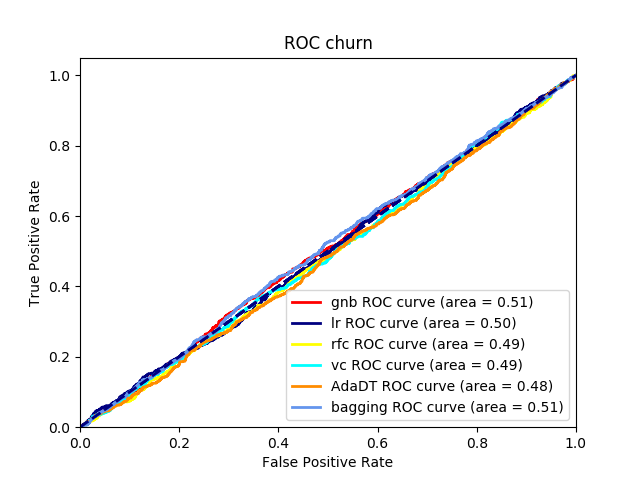


Fig.6. ROC Curves for Dataset Churn

Figure 7 shows the ROC curves for models on dataset appetency. For this, all the classifiers except that of the AdaBoost classifier, have their ROC curves higher than that of the random selection i.e the blue diagonal line. Bagging and lr classifiers have the highest ROC curves. The AdaBoost classifier is still lower than others.

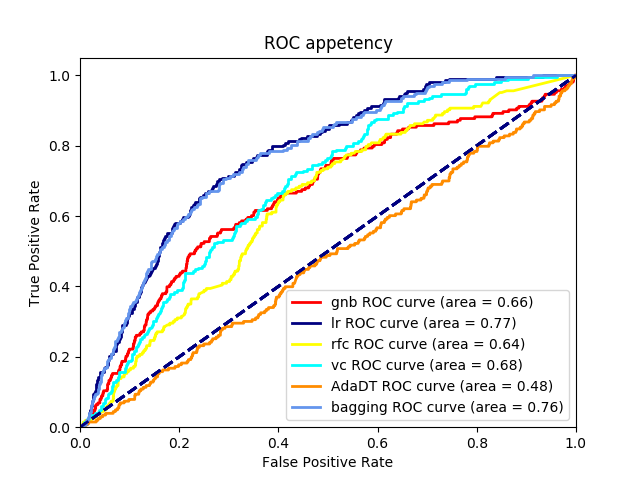


Fig.7. ROC Curves for Dataset Appetency

Figure 8 shows the ROC curves for models on dataset upselling, where all the classifiers have higher ROC curves when compared to the blue line which is the base for our evaluation.Bagging and Linear Regression classifiers have the pretty high ROC curves when compared to other classifiers.

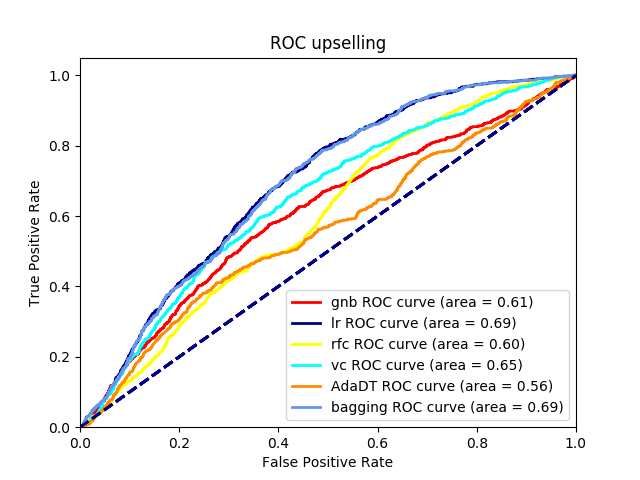


Fig.8. ROC Curves for Dataset Up-selling

1. **ANALYSIS**

From the above result, it was clear that all the ensembled classification (Bagging, Random Forest, AdaBoost) were strong classification with about 95% accuracy, both the Bagging and Random Forests had the highest accuracy but Bagging also had higher precision. Thus, the Bagging was the best classifier in this experiment.

Logistic Regression and Voting were also good with accuracy about 90%. The rest classifications were weak methods, since the accuracy or precision was less than 90%. Especially, the Naive Bayes was the weakest classifier here.

In our case, the dataset had 230 inputs/features and 32 features were used. Those inputs are not linear related to each other. It is a very complicated model. Simple structure classifiers like Naïve Bayes is not suitable to this case. Logistic Regression classifier has better performance because they can deal with more complicated cases. For ensemble classifiers with bootstrapping, their error will be less than that of simple classifiers. That is the reason they have higher accuracy. Bagging is a bag of logistic regression which has the highest accuracy within those non-ensembled classifiers. That is the reason Bagging has the highest accuracy.

The ROC results are out of our expectation. The Logistic Regression has the highest score. The bagging of Logistic Regression does not improve the score a lot. Voting mode has the second highest score since it is a combination of good and bad models. The ROC scores of Random Forest and Ada boosting are lower than those of no-ensemble models. The most likely reason is that the best parameters of those ensemble models are not found.

Our model is running relatively well on the appetency and upselling data. Highest ROC score is at about 0.70. The best team which won the KDD Cup 2009 got the ROC score at about 0.89. But they tested the large dataset which had more data point. We think our result is relatively good. For the low score of the churn dataset, we think it is due to some issues we did not pay attention to during the preprocessing process, such as the data distribution or data bias. We will do future work to improve it.

**9. FUTURE IMPROVEMENT**

From the results and analysis above, it is clear that we need to improve our data cleaning process and hyperparameter tuning. After applying PCA to the raw dataset, we still need to apply feature selection for each target, because PCA only explores the correlations among inputs, feature selection makes more sense to find out the relation between inputs and targets.

The ROC curve for churn dataset was strange. It might be due to the bias of labeling. The class ‘no’ accounts for 92.6% in the training dataset, which means a severe imbalanced class distribution. To solve this problem, we will apply down-sampling technology to improve it.

Due to the time limitation of this project, the best parameter for ensemble classifiers, such as Ada boosting, random forest and bagging, were not found. Grid Search API can not find the best parameter of ensemble classifier and its basic model’s parameter at the same time. Also, it is very time-consuming when the classifier has plenty of hyperparameters to tune. We need to find a better method to find the best parameters.