Machine Learning, Spring 2018
Project1 SunnyBridge Data Science Case Study

Chen Xin 30074765
Lin Daquan 85610653
Xu Yue 61643984

Abstract

Observing the data set, we found it has a lot of missing values and class imbalance, so we used the existing values to predict the missing values and resample the minority class data to train model. Our project contains a classification problem and a linear regression problem. The classifier was trained by the whole training data set with label as responded. The linear regression model was trained by the reponded customers(their profits are existing). Then, we utilized the classifier to predict the probability of responded and the linear regression model to predict the profit for every customer in predict data set. Finally, we calculated the estimated profit according to the probability of responded and the profit for every customer who needs to predict. If estimated profit of a customer is more than zero, it is supposed we should market to him or her.

1 Introduction

In this project, we were asked to use a historical data sets(training data set) to predict whether or not to marketing for potential customers in the list (testing data set). The data set is almost same with Bank Marketing Data Set[7]. and the paper has detailed introduction about the background of data set. Therefore, we skipped this section.

2 Data preparation

First of all, we can find column *id* in data set is useless, and column *pmonths* and column *pdays* are redundancy. So, we ignored one of them. Then, non-numerical data in the data set are transformed into int value data for subsequent processing, such as 'yes' to '1' and 'no' to '0', and supposed *unknown* is equal to NA except column $default$.

Since there are many missing values in the initial data, we use the existing data to predict the missing values except column $default$. In there, we suppose value *unknown* in column $default$ is a variable, because the column $default$ just has values *no* and *unknown*. We used different models to predict the missing values, and select the best performance model to supplement the missing values by the existing data.

Finally, we split the pre-processed train data set into two parts at a ratio of $4 : 1$, as a training data set and a validation data set.

We use the "pandas"[1] data analysis toolkit to complete these process.

2.1 Model selection

In the experiment of supplementing missing values and predict the probability of responded, we tried the following models: "RandomForest Classifier", "AdaBoost Classifier", "Bagging Classifier", "ExtraTrees Classifier", "GradientBoosting Classifier".

AdaBoost is the abbreviation of "Adaptive Boosting" (Adaptive Boosting)[2]. Its adaptation consists in that the weights of the samples of the previous basic classifier that are incorrectly classified will increase, while the weights of the correctly classified samples will decrease. And again used to train the next basic classifier. At the same time, in each round of iterations, a new weak classifier is added until a certain sufficiently small error rate is reached or a predetermined maximum number of iterations is reached before the final strong classifier is determined.

Bagging[3] is a parallel ensemble learning method based on bootstrap sampling. The method first samples T sample sets containing m training samples, then trains a basic learner based on each sample set, and then combines these learners.

Random forest[4] is an extended variant of bagging. It bases on the bagging integration of decision tree-based learners, and further adds random attributes to the decision tree training process.

ExtraTrees[5] is the abbreviation of"Extremely Randomized Trees". Different from the random forest, the extreme random tree adopts all the training samples. The training sample set of each tree is the same. In addition, the branch node is directly selected randomly instead of selecting the optimal branch node.

Gradient Boosting[6] use the gradient descent method, moving to the direction of the smallest loss function at each iteration reduces the loss function. The gradient boosting decision tree concatenates each decision tree, and each tree's learning target is the residual of the previous N-1 tree.

2.2 Evaluation

For the above-mentioned model, we selected the following to evaluate the performance of the method model.

1. Roc auc score

   ROC is the abbreviation of "Receiver Operating Characteristic Curve", AUC (Area Under Curve) is defined as the area under the ROC curve. Obviously the value of this area will not be greater than 1. Because the ROC curve is generally above the line $y = x$, the AUC value generally ranges between 0.5 and 1. The use of the AUC value as an evaluation criterion is because many times the ROC curve does not clearly indicate which classifier is better, but as a numerical value, the classifier having a larger AUC is more effective.

In Table 1, 'ROC AUC score original' means the responded is a probability, which range from 0 to 1. 'ROC AUC score' means the responded is a binary value $\{0, 1\}$, which divide the probability of responded by 'cutoff'.

2. Best cutoff

   It means we can get maximal profit, if we divide the probability of responded in the value.

3. Best profit

   It means the maximal profit we get in validation data set by sum the profit of responded customers.

4. Accuracy

   It means the accuracy of responded in validation data set.

The results are shown in the Table.1 and from which we can draw that best model is GradientBoosting Classifier.

Table 1: Result of Different Classifiers

Classifier	ROC AUC score original	Best cutoff	Best profit	Accuracy	ROC AUC score
RandomForest	0.75665	0.23	14991.0	0.84337	0.73355
AdaBoost	0.78854	0.5	13341.0	0.77948	0.73304
Bagging	0.75449	0.37	11940.0	0.89312	0.7008
ExtraTrees	0.73322	0.18	11601.0	0.83968	0.67726
GradientBoosting	0.77793	0.59	17016.0	0.85319	0.75052

It seems *AdaBoost* has more stable cutoff and the highest ROC AUC score original, which may the best choice. But, if use *AdaBoost*, the responded probability doesn't reasonable, they almost close to 0.5, which is bad for us to estimated profit. So we chose *GradientBoosting*. It has more reasonable probability distribution.

3 Experiment

Observing the data set, we found it has a lot of missing values and class imbalance, so we used the existing values to predict the missing values and resample the minority class data to train model(split the training and test data set,then resample the minority class of data in training data set). Our project contains a classification problem and a linear regression problem. The classifier was trained by the whole training data set with label as responded. The linear regression model was trained by the responded customers(their profits are existing). Then, we utilized the classifier to predict the probability of responded and the linear regression model to predict the profit for every customer in predict data set. Finally, we calculated the estimated profit according to the formula:

$$E_{profit^{(i)}} = Pr^{(i)}[responded] \times pred_profit + (-30) \times (1 - Pr^{(i)}[responded]),\ i \in N \tag{1}$$

where $E_{profit^{(i)}}$ represents the estimated profit, $Pr^{(i)}[responded]$ represents the probability of responded, $pred_profit$ is obtained by linear regression model, N is the number of test sample. If estimated profit of a customer is more than zero, it is supposed we should market to him or her.

4 Result

The predict total profit is 23205$ in test data set. See more details in file *insurance.ipynb* and result in *testingCandidate_output.csv*.

References

[1] http://pandas.pydata.org/pandas-docs/stable/index.html

[2] Freund, Y. and R.E.Schapire A decision-theoretic generalization of on-line learning and an application to boosting,*Journalof Computer and System Science*, 55(1):119-139,1997

[3] Breiman, Bagging predictors.*Machine Learning*, 24(2):123-140, 1996

[4] Breiman, Random forests.*Machine Learning*, 45(1):5-32, 2001

[5] Geurts, Pierre and Ernst, Damien and Wehenkel, Louis, Extremely randomized trees.*Machine learning*, 63(1):3-42, 2006

[6] Friedman, Jerome H, Greedy function approximation: a gradient boosting machine.*Annals of statistics*, 1189–1232, 2001

[7] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014