Minh D. Vo

Technical report

1. Problem description:

As a media company, helping local small and medium sized business through digital advertising is one of our main revenue streams. The advertising system is operated on its platforms such as Google, Facebook … The objective is to predict if a client will stop running advertising campaigns (churn).

2. Dataset:

The data (907 KB) in csv file includes 10,000 observations which cover some information (10 variables). They include cost per lead with respect to business category, client location, duration of advertising campaign, number of distinct advertising products, average monthly budge … The description and name of each variables are provided in two text files (dict.txt and header.txt).

3. Methodology

I have treated this study as a supervised learning classification problem. The binary variable (churn) indicates whether a client will stop using advertisement (0 for retention and 1 for churn). All codes were written in python 3 and run in Jupyter notebook. Additional libraries (such as pandas, numpy, scikit-learn and matplotlib) were also used.

The procedure of my study is as follows:

- Data wrangling
- Model fitting
- Model evaluation

a. Data wrangling

The data were loaded into a data frame (10,000 rows and 10 collumns). There are 8 variables in numerical type (integer and float type) and two categorical variables. It is found that there are 1092 missing data in CPL_wrt_self variable. Then, I filled these missing data by zero value. I assumed that there is no change in client's cost per leat in past three months is zero if this variable is NaN. Because I don't know the reason of these missing values and it also makes sense if the client's cost is not changed. Note that mean and standard deviation of CPL_wrt_self are 0.6 and 11.1, respectively.

For variable of client_state, it included 50 states and DC. I grouped all state have number of total clients are smaller than 100 together. Data show that 30 states were grouped into a new group. It is call Other_states. This is a simple way to reduce number of variables in data file for prediction.

To deal with categorical data, dummy variables are created for each category. So, our data become a data frame with 10,000 rows and 61 columns. Then, I defined independent variables as x (60 features) and target variable as y (churn variable).

After that, data were splitted into train and test set with test ratio of 0.2. Besides, test and training of x data were also scaled by using MinMaxscaler.

b. Building a model

Models were fit using GridSearchCV, which searches through a grid of parameters for each model, returning the model that gives best score. I tried to run with four classification models (logistic regression, decision tree, random forest and gradient boosting classifier).

c. Model Evaluation

Model performance can be displayed with different metrics (like accuracy, recall, precision…). Choice of metric depends on your business objective. In this problem, I think we focus on the probability of the client will stop running advertisement (equivalent to class 1 for chunk variable). So, it expected to decrease the false negative (FN) and false positive (FP) in prediction. Note that false positives (clients are retention that are flagged to be ‘churn’) are more acceptable than false negatives (clients will stop running advertising that are not detected). Eventually, accuracy does not turn out to be a very useful way of evaluating the models. Instead, precision and recall are much more important. The goal is to find a model with high value of recall and precision.

To do the evaluation, I used confusion matrices to analyze precision and recall for each model. I also adjusted the classification threshold to improve the model performance.

4. Results

Based on confusion matrix, it is found that:

- Logistic regression has a poor performance.
- Decision tree is good model with smallest false negative (type II errors).
- Random forest is only good at prediction for class 0.
- Gradient boosting works fairy well. It has a little high value of FN.

Because dataset is unbalanced (80% of targe variable is for class 0). It leads to the problem that most classification models are highly specific, not sensitive. They tend to have very high recall and precision for class 0 and low recal, precision for class 1.

To improve the model performance, classification threshold was used. It is found that gradient boosting with threshold of 0.25 shows the good recall and precision as well as high accuracy.

In addition, it also presented that duration, CPL_wrt_self, CPL_wrt_BC and avg_budget are important features. They are strongly influenced on the target variable.

5. Further improvement

The performance of the classification model is not optimized yet. I think there are some ways to improve it:

- It should be carried out the outlier analysis for some variables (like CPL_wrt_self, CPL_wrt_BC and avg_budget).
- All the important features can be examined for further correlations/patterns. Some features seem likely candidates for dimensionality reduction through PCA.