A Simple yet Efficient Method for a Credit Card Upselling Prediction

Peter Romov

Yandex Data Factory

1 Introduction

ECML/PKDD 2016 Discovery Challenge Upselling Prediction task was about predicting whether the user will apply for a credit card in the next 6 months. This is an important information for the bank because it can be used to initiate an upselling. Organizers provided one year of historic data on user attributes, activity events and geolocation. It was organized as a classification contest, evaluated with Area Under the ROC-Curve (AUC-ROC).

Our approach to the challenge problem consists of a predictive XGBoost machine learning model based on around 100 extracted features from user information and his events in the first 6 month of year. The work is focused mostly on feature engineering described in section 2. The process of building a model is described in section 3.

Reproducible version of the solution code is published on GitHub[1].

2 Feature Engineering

To represent the user with a feature vector, several logical group of features were designed. They are described below. Some features could not been computed for some users due to the lack of user information or events of some kind, in this case this feature had missing value.

Personal information. Categorical features from client profile:

- gender
- age category (35 and less, 36–65, 65 and more)
- location category (capital, city, village)
- income category (no income, low, medium, high)

The features were encoded in two ways: 1) with one-hot encoding (to eliminate order of categories) 2) with one integer number (to take the order into account).

[1] https://github.com/romovpa/ecmlpkdd2016-otp-bank-upselling

Cards and Wealth The following features are computed on the first six months of the year:

- Number of months when user had a card, the same aggregation for being wealthy
- Last month of the period when user had a card / was wealthy
- Number of indicator changes from having a card / being wealthy to a month without this indicator, and vice versa

Event counters. The first feature of this group was total number of events. Then for each categorical variable of an event two count vectors were computed: the first represents the exact number of events having specific value of categorical faatures; the second represent ratio of events having specific value in all events.

Event counter features were calculated for the following categorical features:

- Type of activity (point of sale, webshop, branch)
- Time rounded to three ranges (05-11h, 12-18h, 19-04h)
- Event location category (capital, city or village)
- Anonymized market category groups (7 unique categories)
- Type of card used (credit or debit)
- Amount of money spent in three categories (low, medium, high)
- Weekday

Number of unique shops. Total number of unique places of events and the number by channel type: number of unique branches of the bank, web shops and point of sale.

Client activity. Denote the client *active* in the specific period when he committed at least one transaction in the period. As for features describing client activity in the first 6 months of year, the following features were computed:

- Number of days / weeks the client was active
- Duration in days from the first event to the last
- Average number of active days / weeks
- Average number of inactive days / weeks
- Active days rate

Geolocation features. Coordinates of user home address as known client features.

The geographical coordinates of branches of the bank and points of sale were provided. To represent geographical statistics for events of each user, the idea was to compute distances and angles of event points to the client home and the capital city (Budapest) aggregated with several statistics: average, minimum, maximum, standard deviation, 20/50/80-quantiles

3 Prediction model

Decision tree gradient boosting machine (GBM)[1] is a widely applicable machine learning method that works out-of the box without complex hyperparameter tuning. XGBoost[2] is a well-known implementation of GBM and was successfully applied to many machine learning tasks and competitions.

To build an upselling classifier, the model of choice was XGBoost[2] with binary classification objective and default parameter settings: maximum tree depth = 3, number of iterations = 100 and learning rate = 0.1.

Tuning parameters such as maximum tree depth, learning rate and several schemes of filling missing values didn't provide any significant gain on AUC-ROC estimated with simple stratified cross-validation on the one year data. Moreover, fair cross-validation wasn't feasible due to lack of year-to-year overlaps between train and test. Hence the idea was to stick with the Occam's Razor and build a simple yet reasonable model to counter potential overfitting.

4 Conclusion

The proposed solution focused mostly on detailed feature engineering. When it comes to predictive modeling, standard XGBoost solution achieved a reasonable good result on the public leaderbord with AUC-ROC score of 0.7136. We also believe that finetuning this predictive model will likely not provide accuracy gains due to high risk of overfitting. Lack of historic data is a major obstacle to building more sophisticated models because one cannot properly validate results on the next year given only one year of history.

References

1. Friedman, Jerome H. Greedy function approximation: a gradient boosting machine. Annals of statistics (2001): 1189-1232.
2. Chen, Tianqi, and Carlos Guestrin. Xgboost: A scalable tree boosting system. arXiv preprint arXiv:1603.02754 (2016)