Purchasing Probability Prediction: Do Advertisements know you well?

Kehao Yao (ky392) Yuan Yao (yy874)

Abstract

As online purchasing becomes increasingly prevailing, online advertisements providers are seeking better ways to target ads at users. By predicting the purchasing probability of users after they click on the ads, online advertisements providers can optimize the way they display ads, resulting in higher sales and better user experience. In this project, we construct a combined model of gradient boost decision tree and logistic regression with real data from Taobao.com to predict users' purchasing probability. Feature engineering techniques such as one-hot encoding and correlation analysis and resampling techniques are utilized for model performance improvements. The model achieves an out-of-sample true positive rate of 74.2% after calibrating parameters.

Contents

1 Introduction

Online advertising allows advertisers to only bid and pay for measurable user responses, such as purchases after clicks on ads. By predicting the probability that customers purchase the advertising item after clicking on the ads, advertisement providers can optimize what kinds of ads to show, in order to boost the sale for merchants.

As a consequence, purchase prediction systems are essential to many online advertising platforms. On these platforms, the allocation of ads is dynamic, tailored to users' interests and based on their observed feedback. For the popular online shopping platform, Taobao.com, there are over 580 million daily active users and over 1 million active advertisers. Thus, predicting how target customers will respond to certain ads on Taobao.com is a challenging but rewarding data analytics task.

In this project, our goal is to predict the **conversion rate (marked as CVR)** -- the probability that the target users finally decide to buy the advertising item, given the information about the advertising items, target users and item providers. Using conditional probability we get the formula: CVR = P(conversion=1|query, user, ad, context, shop)

2 Data Analysis

2.1 Data Description

The dataset for this project was provided by Alibaba. It contains over 500,000 advertising data with 27 features, which were collected directly from Taobao.com. Our dataset mainly includes five different aspects, which are basic information, advertising items, target users, contexts and merchants.

2.2 Data Preprocessing

Since the dataset comes from real transactions, it has high overall quality, with a few missing entries marked with -1.

We process the missing data by various means. For merchants' rating, we replace them with median value, while for users' age, we drop the missing values, since users from different age groups can act very differently in online shopping behaviors.

The variables presented in the dataset are a mix of nominal, ordinal, discrete and continuous values. To get a general idea of how our dataset looks, we visualize some key feature of our dataset.

2.3 Data Visualization

Different groups of people behave very differently in the consumption habits. On online shopping platforms, uses' gender and age can result in very different behavior. Therefore, regarding the user features, we visualize the data of user age, gender and occupations. Since some the original data is encrypted by the data provider, we can only plot the data in group codes.

According to the graphs, female consumers make up a majority, over 75%, of all consumers.

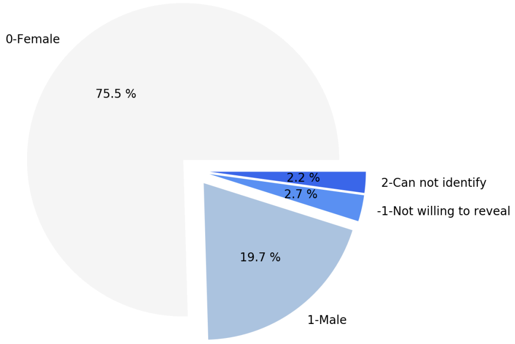

Fig 1. User gender

User occupation is generally divided into four categories, among which the category '2005' stands out and the category '2002' follows.

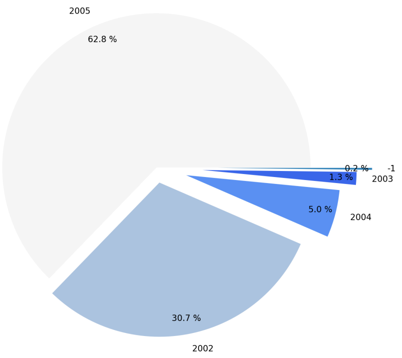

Fig 2. User occupation

According to the user age, the users of online shopping platform are mostly young people with high purchasing power, whose ages range from 25 to 40.

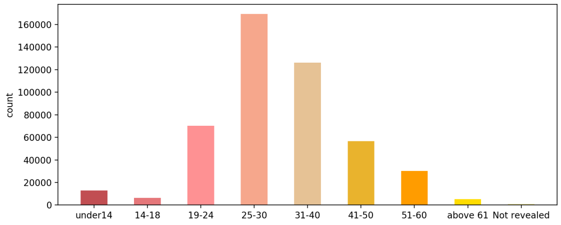

Fig 3. User age

When it comes to item features, item sales level is a deciding factor for online shopping since a high sales level usually indicates high quality. Most items have a sales level between 9 and 14. Items with a sales level below 4 are purchased with a very low probability.

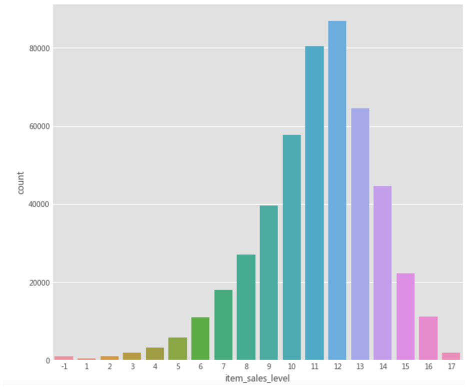

Fig 4. Item sales level

Regarding the timestamp, we plotted the conversion rate using heat map. In our dataset, we have the data across one week. By plotting the CVR in different hour of different day, it can be concluded that the CVR varies according to time. On weekdays, CVR is significantly lower than on weekends. Besides, CVR is higher from 20 pm to 3 am. Thus, the time factor matters in predicting whether a customer would buy an advertising item.

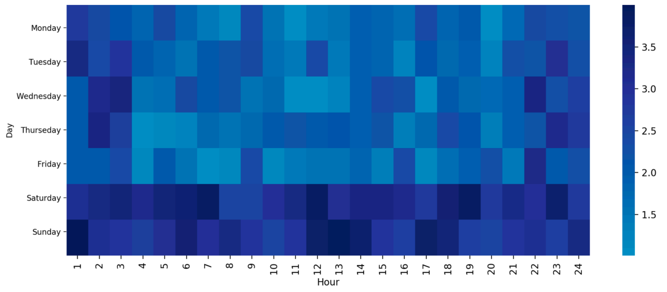

Fig 5. CVR Heat Map

2.4 Imbalanced Data

Although the amount of data is large, the average purchased rate is less than 2% (only a few data are marked with label "1", according to Figure 5), which brings a problem of unbalanced distribution of the samples. Due to the unevenness of the training set, linear regression models provide a high accuracy rate but actually perform badly, because almost all of the data are labeled as "0", but the true positive rate is very low. We tried multiple approaches to resolve this problem, including resampling techniques and different predicting algorithms, which will be discussed in great detail later.

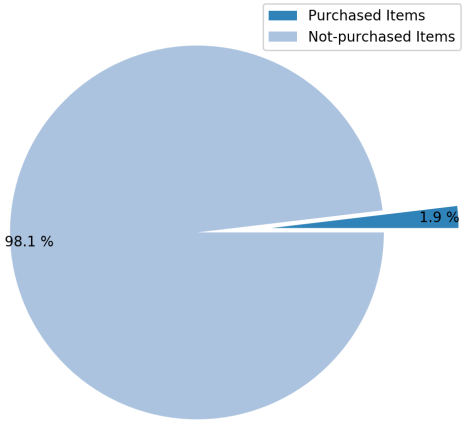

Fig 6. The proportion of purchased and not-purchased items

3 Feature Engineering

3.1 Correlation Analysis

Also, we conduct correlation analysis on numerous values like merchant score. It is obvious to see that, in the training set, the merchants' score on service is highly correlated with their score on delivery. Similar results are also observed in merchants' score on service and description of items. Therefore, we adopt a linear combination method with the help of linear regression, which integrates the highly correlative features into one feature, and receives a better result in the model performance.

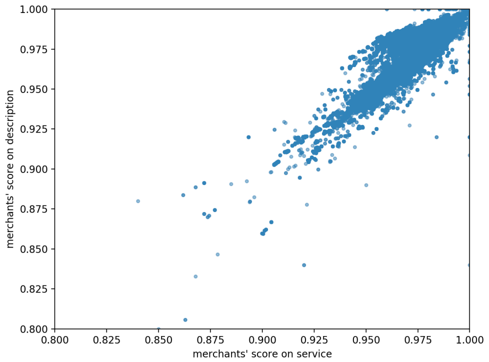

Fig 7. Correlation of merchants' score features

3.2 One-hot Encoding

One-hot encoding is a feature engineering technique frequently used when dealing with categorical data. For features with multiple classes, one-hot encoding replaces the initial feature with several binary features. In our model, we use one-hot encoding to pre-process features like user occupations and gender.

Also, we apply one-hot encoding to process some of the transformed features of GBDT and feed these features to the logistic regression model.

3.3 Other Pre-processes

In our dataset, the categories and properties of the advertising item and the keywords provided by the users are provided as set data. Obviously, how the categories and properties of advertising item match those of the items that users are searching has great effect on the probability of purchasing. The item is more likely to be purchased as their intersection increases. Therefore, we generate a new feature "intersection", which represents the similarity in categories and properties of the wanted item and the advertising item. With this new feature, how the advertising items match what users are searching is better represented.

Besides, we replace some ID features with their appearing frequencies. The intuition is that item IDs that appear more frequently in the dataset represent that the items are more popular than those that appear less frequently.

4 Model Construction

Essentially, in order to predict how likely is a customer to buy an advertising item, we are facing a supervised classification problem. Intuitively, we start with a logistic regression model. Resampling techniques are utilized to address the imbalanced data problem. To future improve our model, we combine GBDT (gradient boosting decision tree) and LR (Logistic Regression) and get a better accuracy.

As the dataset is imbalanced, we evaluate the performance of our models with logarithmic loss as below, where p_i is the estimated conversion rate of the i-th sample. We expect a lower logarithmic loss for a better model.

$$\log loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$$

Fig 8. Logarithmic loss function

4.1 Logistic Regression with Resampling Technique

4.1.1 Logistic Regression

Logistic regression is frequently used for binary classification. Different from linear regression, logistic regression analyzes the weight of independent variables by minimizing a logarithmic loss function and map the prediction result to [0, 1] with a sigmoid function as Fig 9 shows. Here is the loss function with regularization:

$$\mathrm{h(x)} = \log\left(1 + e^{-yw^T x}\right) + \lambda\left|\left|x\right|\right| \qquad (1)$$

Both L1 and L2 are tested when choosing regularization parameters. The L1 regularizer seems to be better than L2 in this dataset. The results are shown in 4.5.

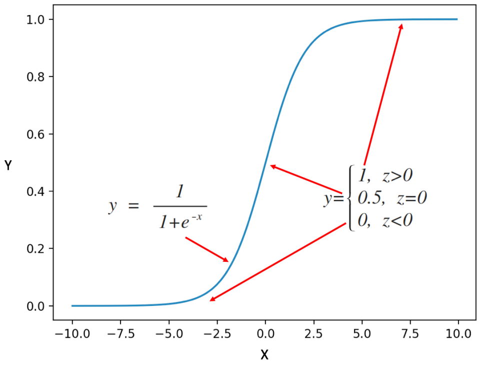

Fig 9. Sigmoid function

With logistic regression, we are able to predict the probability of users purchasing an advertising item.

4.1.2 Resampling Techniques

As stated before, in the dataset, over 98% of the samples are labeled as 0 and less than 2% of the samples are labeled as 1, thus creating a severe imbalanced data problem. In order to address this problem, we increase the weight of the minority class and decrease the weight of the majority class with resampling techniques.

For the minority class, SMOTE method are utilized to create more observations with label 1. SMOTE stands for Synthetic Minority Over-sampling Technique. Compared to simply replicate the minority observations, SMOTE provides a better solution to over-sampling. It first finds the k-nearest neighbors of the existing minority observations. Then according to the number of new observations is required, it creates new observations randomly on the line connecting observations and its neighbors.

For the majority class, we apply a random under sampler. A random under sampler delete the majority observations in a random and uniform manner. Considering that we have more than 490,000 relatively similar data with label 0, a simple under sampler performs considerably well under this circumstance.

To address the imbalanced data problem, we combine the two resampling techniques mentioned before and increase the proportion of the minority class from 2% to 33%.

4.1.3 Model performance

Before resampling, the training set contains about 430,000 rows of data and 77 features. The significant difference between the number of majority and minority classes makes the model label all of the samples as 0.

After applying resampling techniques, the training set contains about 630,000 rows of data and the performance of logistic regression model improves. The model achieves a true positive rate of 0.52, which is much higher than before, but is still to be future improved.

4.2 Gradient Boost Decision Tree

Gradient Boosting is a technique that produces a prediction model in the form of an ensemble of weak prediction models. Gradient Boosting Decision Trees use decision trees as weak prediction models. The iterative idea is that the strong learner obtained in the previous iteration is f_{t-1} (x) with the loss function L (y, f_{t-1} (x)). The goal of the iteration is to find a weak learner h (t) of the regression tree model, and minimize the loss L (y, ft) of this round (x) = L (y, f_{t-1} (x)) + h (x). The schematic diagram is as follows:

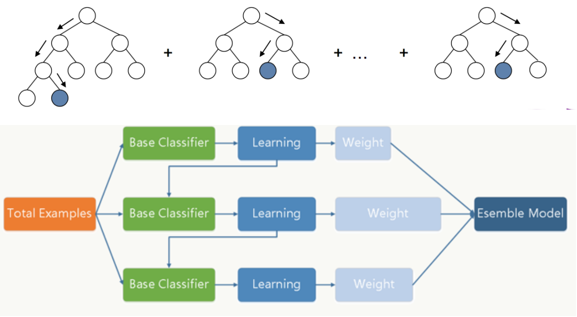

Fig 10. GBDT evolution

It includes two steps:

1. For each tree, partition sample space by growing n nodes.

2. Compute gradient and repeat.

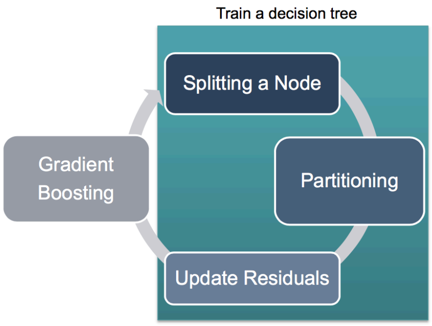

Fig 11. Training procedure

For more information about GBDT and decision tree, please refer to chapter 8 in the textbook, *An Introduction to Statistical Learning*

GBDT fits our model well for the following four reasons:

First, the decision tree model works well with imbalanced data and is able to handle both numerical and categorical data at the same time. Second, the boosting method improves performance of the decision tree with iterations. Third, GBDT models are relatively easy to interpret as we can output the leaf nodes and check the features. Finally, tree-based methods perform well on large datasets and produce results with reasonable time expenses.

4.3 GBDT + Logistic Regression

According to the paper: *Practical Lessons from Predicting Clicks on Ads at Facebook*, decision trees produce very powerful input feature transformations that can significantly increase the accuracy of probabilistic linear classifiers. Moreover, boosted decision trees provide a convenient way of doing feature selection by means of feature importance. One can aggressively reduce the number of active features when prediction accuracy is only moderately hurt. Thus, a logistic regression model with tree-transformed input features is worth trying.

First, we use GBDT to obtain a classifier, sending the leaf nodes of each tree as features to the LR training classifier.

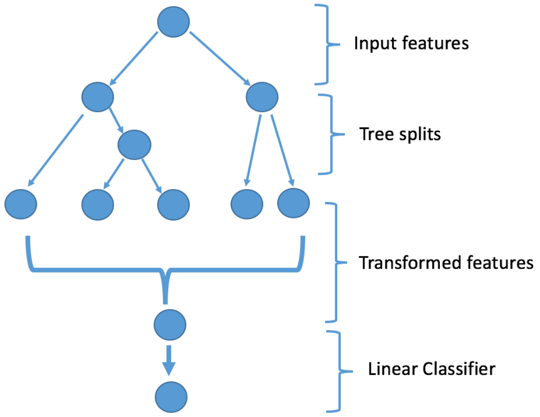

Fig 12. Model structure of GBDT and LR

Input features are transformed by means of boosted decision trees. The output of each individual tree is treated as categorical input features to a sparse linear classifier, Logistic Regression.

To further improve the algorithm, we only use GBDT to encode the continuous values, and use the original one-hot encoding for the rest. After that, two groups of features are spliced together and sent to the LR trainer. The new method reduces the GBDT training volume and improves the code speed by 14%.

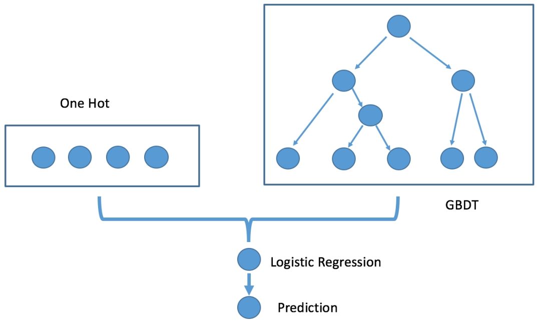

Fig 13. Further improvement of model structure

4.4 Decision Tree

For comparing purpose, we also tried the classical decision tree model, as well as the combination of decision tree and logistic regression. The results are shown in 4.5.

4.5 Model Validation

We valid our model by k folds method. Here we divided data into 10 folds (9 folds for training, one-fold for test set). The results are as follows.

Model	Logloss (Train)	Logloss (Test)
LR (L1)	0.08383	0.08402
LR (L2)	0.08496	0.08538
GBDT	0.08259	0.08342
GBDT+LR	0.08158	0.08259
DT	0.08845	0.08647
DT+LR	0.08560	0.08221
GBDT+LR (resampling)	0.08032	0.08250

Table 1. Model validation (Logloss)

From the table, it is clear that the LR and Tree models separately have comparable prediction accuracy (GBDT is a bit better). However, their combination yields an accuracy leap. The significant improvement from the combination of GBDT and LR makes sense as GBDT enhances the efficiency of feature engineering. What's more, the resampling method proves to be efficient, although the improvement is not prominent.

Among seven models, the decision tree model performs the worst. A reason may lie in the fact that the training error for decision tree is also high (close to 0.09) which implies that decision tree has a relatively high bias. Regarding regularization, the L1 regularizer seems to be better than L2 in this dataset. One reason is that the impute features of LR is a bit large and we need shrinkage of features.

We also try to compare our models with recall rate (also called the true positive rate), which measures the proportion of actual positives that are correctly identified as such. The result is similar and it again proves the good performance of our model.

Model	Recall (Train)	Recall (Test)
LR (L1)	0.769	0.661
LR (L2)	0.808	0.709
GBDT	0.782	0.713
GBDT+LR	0.805	0.737
DT	0.655	0.588
DT+LR	0.684	0.593
GBDT+LR (resampling)	0.883	0.742

Table 2. Model validation (recall)

5 Model Application

5.1 Weapon of Math Destruction

By analyzing the outcome and application scenarios, we are certain that our model is not a weapon of math destruction.

From the perspective of outcomes and feedbacks, the accuracy of prediction is easy to measure, and feedbacks can be utilized to improve the model. If the model is used by an online shopping platform, our model will determine what kind of items will be displayed to certain customers. Soon after the ads are displayed, we can gather the information about whether our target customers have bought the advertising items. By continuously collecting new information, we are able to find out whether we have over-predicted or under-predicted the probability of purchasing and revise our model accordingly.

With our model, ads will be shown to the target customers more precisely. Every item has its chance to be displayed to certain proper customers and the customers will have a higher probability to purchase an advertising item. Therefore, there is no negative consequence here. At most, it is the customers' wallets that are taking consequences!

5.2 Fairness

Fairness is another problem to consider in case of unintended (and sometimes unobservable) negative consequences.

Here fairness is not an important criterion for our model. Firstly, the prediction of our model does not have influential social effects compared to election or hiring process. Also, the given data is directly collected from real transaction process with little bias. Out-of-sample data should have the same distribution as our current data. Finally, removing some features like age and gender has little effect on the result (2% on average). Therefore, our model does not violate the principle of fairness.

5.3 Application Scenarios

Our model is supposed to be used by online ads providers. With a better prediction of how likely a customer will purchase an advertising item, the ads providers can better target the display of ads to different customers, according to their gender, age, occupation and items they are interested in.

From the perspective of ads providers, they can yield a better conversion rate and a higher efficiency of displaying effective information. Merchants will also get more exposure to their target customers and consequently increase their sales. For the items that are not frequently displayed, merchants are forced to improve in order to compete with their rivals.

From the perspective of users, the ads that are shown to them will seem more attractive and interesting and the user experience will be better.

When constructing the model, we take a lot of practical factor into consideration, such as the imbalanced data problem and time effect. As is shown below, the model can be updated according to new data and deliver prediction results almost simultaneously.

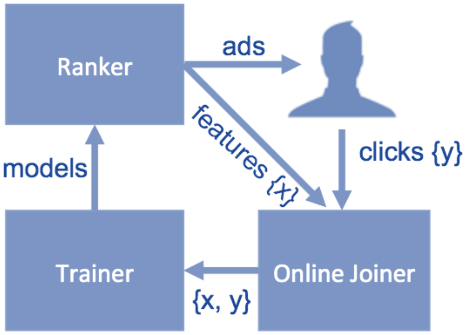

Fig14. Model Flows

In terms of efficiency, the model takes only a few minutes to train with CPU, which will be even faster with GPU or cloud computing. Thus, our model can perform well even with large datasets when put into use.

In all, our model predicts reasonably well and we are confident to include it into business scenarios.

6 Conclusion

In this project, after compassion, we introduce a model which combines Grand Boosting Decision Tree with logistic regression, outperforming either of these methods on its own or other models. The result shows that our GBDT-LR model performs well on the dataset and can be applied practically. By using four techniques that we learned in class including one-hot encoding, logistic regression, Linear regression, Feature Engineering and regularization, as well as other methods including GBDT and decision tree, we were able to predict the conversion rate with a minimum log loss of 0.0825.

We are fairly confident that our model will be able to generalize to new data well. Since a lot of practical factors are considered in our research process, we are willing to use them in production to change how our company (Alibaba or Amazon) or enterprise makes decisions.

In the future, for this model to be more informative and valuable for real advertisement targeting, further work needs to be performed. In this project, imbalanced data was a limitation and challenging problem. The team expects the model to perform better if the data is balanced. At the end, we would like to thank Alibaba for providing the data and hope this model can be beneficial to advertisement efficiency on Taobao.com and other e-commerce platforms.

Appendix: Reference

[1] McMahan, H. Brendan, et al. "Ad click prediction: a view from the trenches." *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013.

[2] Li, Cheng, et al. "Click-through prediction for advertising in twitter timeline." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.

[3] Zai Huang , Zhen Pan , Qi Liu , Bai Long , Haiping Ma , Enhong Chen, An Ad CTR Prediction Method Based on Feature Learning of Deep and Shallow Layers, Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, November 06-10, 2017, Singapore, Singapore

[4] Yap, Bee Wah, et al. "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets." *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*. Springer, Singapore, 2014.

[5] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.

[6] Chen, Junxuan, et al. "Deep ctr prediction in display advertising." *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016.

[7] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016.

[8] He, Xinran, et al. "Practical lessons from predicting clicks on ads at facebook." *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 2014.

[9] James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.