Repeat Buyers Prediction Based on User Online Shopping Logs

Xu Han
SID: 108592984, Email: xuha2442@colorado.edu, Course section: CSCI 5502-001

Yichen Wang
SID: 108588424, Email: yichen.wang@colorado.edu, Course section: CSCI 5502-001

Yawen Zhang
SID: 107121651, Email: yawen.zhang@colorado.edu, Course section: CSCI 5502-001

Xiaolan Cai
SID: 107100495, Email: xiaolan.cai@colorado.edu, Course section: CSCI 5502-001

ABSTRACT

In this project, we developed a framework to predict whether a customer promoted by a big promotion(like "Double 11") will become a repeat buyer or not based on the dataset provided by Tmall.com. Our framework is composed of four parts: data preprocessing, feature engineering, model training and evaluation. Comparing with prior work, our project brings up with innovable methods on filling the missing data and balancing data during data preprocessing, generating our self-designed features during feature engineering, deploying nerual networks and ensemble classifiers during model training and doing feature importance analysis during evaluation. The best performance of our framework has achieved an AUC score of 0.682731. In short, our work provides not only a great solution to this prediction task, but also some useful and valuable feature importance results for further feature engineering work.

KEYWORDS

user behavior, feature engineering, prediction

ACM Reference Format:
Xu Han, Yichen Wang, Yawen Zhang, and Xiaolan Cai. 2017. Repeat Buyers Prediction Based on User Online Shopping Logs. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

In this project, we use the dataset provided by Tmall.com (an online shopping mall in China) to predict whether a new customer promoted by a big promotion ("Double 11") will become a repeat buyer or not within the following 6 months. This prediction results will be quite crucial for merchants because a big promotion usually generates many one-time buyers. In this case, more competitive strategis could be made if accurate prediction results are available.

This task is actually very challenging. First, the raw data is extremely unbalanced and there are missing data. This requires us to come up with some suitable data preprocessing method. Second, there are too many features and it is very hard to find a balance between the model training time and model's prediction precision. In order to solve this problem, we need to first categorize all the features and evaluate the importance of each feature group. Third, the current classifiers cannot achieve very good performance. In this case, we need to select best classifiers and parameters to improve the prediction precision.

Our report will be organized mainly in four parts as follows: 1) data preprocessing (including data cleaning, aggregation), 2) feature engineering (with self designed features and a mix of feature engineering methods), 3) classification (including classic supervised classification methods and ensemble methods), 4) evaluation (using different metrics). We contribute by generating different features about users' purchasing behavior and applying advanced classification method to get good prediction results. At same time, we also provide some useful feature importance analysis results for future online buyer-related feature engineering.

2 RELATED WORK

User online purchasing behavior has been widely studied with aim of characterizing and predicting those behaviors. Feature selection and classification methods act as the two key components in the predicting task.

2.1 Feature Selection Methods

There are a lot of features related to user's online purchasing behavior, e.g. user's demographic information (gender, age, location, and income), online purchasing time and price history. All these features can be utilized to form effective features for the prediction task. Farshad et al. [9] present a quantitative analysis of these factors and find that temporal features plays the most important role in improving the prediction accuracy. Instead of directly using those features, Caroline et al. [11] divide all the purchasing signals into long-term and short-term categories and find that purchase intents can be mined through different purchasing categories. Another useful conclusion from them is that the purchasing signals in online behavior can exist weeks before a purchase is made and are amplified in the last three days before purchase. For this specific repeat buyers prediction, Guimei et al. [10] generate a large number of features including count/ratio features, aggregation features, recent activity features, age/gender related features and other complex features like PCA features, LDA features etc. and conduct feature ranking as to find the most effective features for prediction.

2.2 Classification Methods

For predicting user's behavior, different classifiers have been used [4, 10], e.g. Factorization Machine, Logistic Regression, Random Forest, GBM, XGBoost and Gradient Boosting Tree. Besides using

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

the classical models, Ye et al. [13] also designed a novel prediction model based on the negative binomial distribution (NBD), and the model is designed to deal with datasets which only contains information on the frequencies and timings of transactions. By utilizing the advanced deep learning algorithm, Armando [14] proposes a deep belief network for prediction user online behavior which is able to get better result than baseline methods.

2.3 Summary

For feature selection, it is important to pick effective features associated with user's online purchasing behavior and there are usually thousands of features that can be explored. For classification, although there are many classical methods to deploy, for this specific prediction task, it is important to pick the model with high accuracy and explore with advanced classification methods.

The work introduced before didn't explore enough features. For classification, they only focused on several single classifiers. Our work will explore more features and evaluate the effectiveness of several types of features. We will get a rank of the features. We will also add more classifiers, for example, artificial nerual network, and try to blend them together to get a higher prediciton accuracy.

3 DATA

The dataset is provided by Tmall (an online shopping mall in China). The data contains details of customers' shopping history in the past 6 months before and on the day of "Double 11" (a big promotion day on Nov. 11) in 2014, and the label information of whether the customers are repeat buyers for the merchants. The three dataset and their attributes are listed below.

1) User behavior logs within last 6 months before "Double 11":

User_id: A unique id for the shopper (Numeric)
Item_id: A unique id for the item (Numeric)
Cat_id: A unique id for the category that the item belongs to (Numeric)
Merchant_id: A unique id for the merchant (Numeric)
Brand_id: A unique id for the brand of the item (Numeric)
Time_tamp: Date the action took place (format: mmdd) (Numeric)
Action_type: 0 is for click, 1 is for add-to-cart, 2 is for purchase and 3 is for add-to-favourite (Enumerated)

2) User Profile:

User_id: A unique id for the shopper (Numeric)
Age_range: 1 for <18; 2 for [18,24]; 3 for [25,29]; 4 for [30,34]; 5 for [35,39]; 6 for [40,49]; 7 and 8 for >= 50; 0 and NULL for unknown (Enumerated)
Gender: 0 for female, 1 for male, 2 and NULL for unknown (Enumerated)

3) Training and testing data:

User_id: A unique id for the shopper (Numeric)
Merchant_id: A unique id for the merchant (Numeric)
Label: 1 means repeat buyer, 0 is for non-repeat buyer. This field is empty for test data (Enumerated).

The basic statistics of the above data is Table 1. In total, there are 212, 062 users, among which only 6.12% are repeat buyers of a merchant. In terms of user profile, there are 22.43% users without age information and 1.52% users without gender information. The missing data need to be filled before being used as valid features. There are 4, 995 merchants, and 2, 824, 241 user logs. The total number of labeled data is 260, 860, which indicate whether or not a user is the repeat buyer of a merchant. We use 80% and 20% separation for the training and testing data.

Table 1: Basic statistics of raw data

# users	# merchants	# user log	% of missing user info
212,062	4,995	2,824,241	age: 22.43%, gender: 1.52%

4 FRAMEWORK

This Repeat buyers prediction issue can be considered as a typical classification problem. Based on that point, our framework consists four aspects: data preprocessing, feature engineering, supervised classification (labeled data has been given) and evaluation, which is shown in Figure 1.

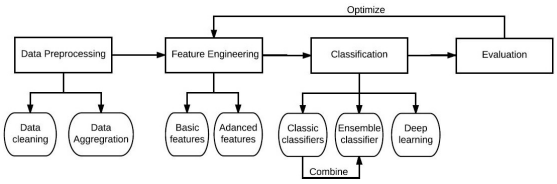

Figure 1: The framework of our proposed idea

5 DATA PREPROCESSING

Given the raw csv data, we firstly put all the data into PostgreSQL for better management as shown in Figure 2. Then, we combine different tables to generate the attributes we need. For data preprocessing, we deal with two major problems, one is the missing data in users' demographic information, e.g. age, gender, the other one is about the unbalanced data problem, as the percentage of repeat buyers is much smaller than non-repeat buyers in the training data.

5.1 Missing data

Based the statistics about missing data in Table 1, we need to fill in the missing value in age_range and gender. With the observation in Figure 3, we notice that females tend to have a larger range of total #purchases, and different age group also correspond to different ranges of total #purchases. Therefore, instead of filling a global value for both gender and age, we apply the local median instead. That is, we divide total #purchases into 16 groups, including $[0, 10]$, $[11, 20]$, $[21, 30]$, ..., $[141, 150]$, $[151, \infty)$. And when filling in the missing values, we used the median value of gender or age in the group a user belongs to. In this way, both gender and age information are filled in.

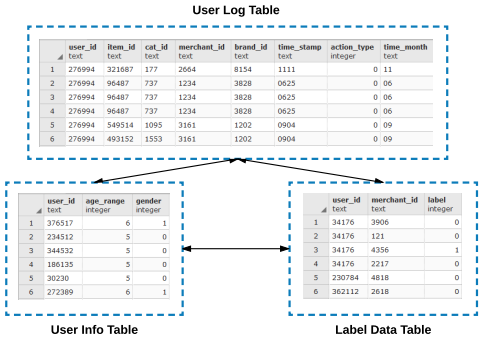

Figure 2: Data Management

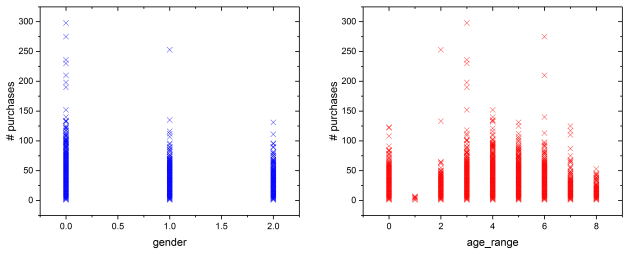

Figure 3: The relationship between #purchases and a user's gender/age_range (2.0 in gender: Unknown, 0 in age_range: Unknown)

5.2 Unbalanced data

According to the statistics, only 6.12% of the users are repeat buyers, which indicate the ratio of #non-repeat buyers and #repeat buyers is approximately 15 : 1 which is very unbalanced for the classification. Therefore, we apply two methods here to deal with the unbalancing problem: 1) Undersampling, the basic idea is to remove some non-repeat buyers from the training data to make the labeled data more balanced. Therefore, we conduct random sampling for non-repeat buyers and repeat buyers to make their ratio decreasing to 4 : 1, our results shows this ratio works good for most classifiers. 2) Threshold moving, the basic idea is instead of dealing with training set, we just assign a weight to the prediction result, and the weight is the original ratio of non-repeat buyers to repeat buyers. The second method will be used directly in classifiers by setting the "class_weight" to "balanced".

6 FEATURE ENGINEERING

Currently, we have extracted nine types of features: 1) user demographic features, 2) user purchasing behavior features, 3) merchant-related features, 4) user-merchant interaction features, 5) "Double 11" features, 6) slope features, 7) repeat buyer features, 8) similarity feature and 9) PCA features. In total, 2 user demographic features, 32 user purchasing behavior features, 32 merchant-related feature and 32 user merchant interaction features are extracted, 6 "Double 11" features, 12 slope features, 2 repeat buyer features, 1 similarity feature and 10 PCA features. We also conduct basic feature analysis to find how they relate to repeat buyers.

6.1 User demographic features

The user demographic features are from the original user profile data which provides information regarding user's age and gender. The basic statistics of these two features are illustrated in Figure 4. Obviously, the top 3 Tmall users age range are $3 : [25, 29]$, $4 : [30, 34]$ and $2 : [18, 24]$ and female users are much more than male users. Based on the statistics in Table 2, repeat buyers tend to have larger age than not repeat buyers and female users are more likely to become repeat users.

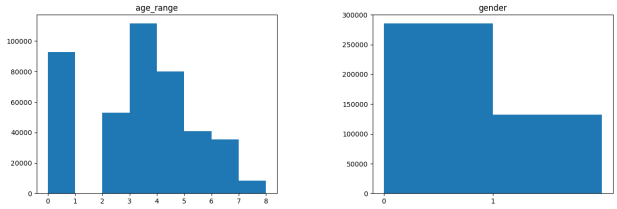

Figure 4: Histograms of users' age_range and gender.

Table 2: User demographic features statistics (repeat and not repeat buyers)

	age_range	gender
repeat	mean: 3.08, std: 1.95	mean: 0.31, std: 0.52
not repeat	mean: 2.95, std: 1.93	mean: 0.34, std: 0.53

6.2 User purchasing behavior features

The intuition is that users behavior on Tmall may influence whether or not the user will become repeat buyers, e.g. users that frequently purchase online tend to be more likely to buy something again. Therefore, we extract various features associated with user's purchasing behavior based on the user log data. First, as there are four different action types (click, add to cart, purchase and add to favorite), we firstly calculate the total count of each action type. Then, we distribute those actions into different months, e.g. May, June, July, August, September, October and November. The monthly count is an addition to the total count as to capture the user's monthly purchasing behavior, e.g. whether it is constant or increase sharply in certain period. Combining the total and monthly count of the four action types, there are totally 32 features associated with user's purchasing behavior.

The basic statistics of the four different action types for repeat and not repeat buyers are illustrated in Table 3. Based on the average total count of each action, users with more click, purchase and add to favorite tend to become repeat buyers, the only exception is add to cart. Repeat buyers tend to have less add to cart actions than not repeat buyers. In terms of the maximum value of the total count, not repeat buyers tend to have higher maximum values than repeat buyers for all types of actions (for purchase action, they are equal to 298), which indicates that the users with largest action count may not be a repeat buyer for certain merchant.

Table 3: User purchasing behavior features statistics (repeat and not repeat buyers)

	# click	# add to cart	# purchase	# add to favorite
repeat	mean: 151.39 std: 222.08 min: 0 max: 4040	mean: 0.15 std: 0.81 min: 0 max: 17	mean: 10.43 std: 10.91 min: 1 max: 298	mean: 9.53 std: 24.76 min: 0 max: 877
not repeat	mean: 124.07 std: 186.44 min: 0 max: 8005	mean: 0.19 std: 0.90 min: 0 max: 27	mean: 8.60 std: 8.76 min: 1 max: 298	mean: 7.71 std: 21.98 min: 0 max: 1706

6.3 Merchant-related features

We also extract features related to each merchant. The intuition is that popular merchant with frequent purchase tend to have more repeat buyers. Therefore, it will be helpful to extract features that can characterize the popularity of each merchant. Similarly to user related features, we calculate total count of each action associated with certain merchant, the actions can from any users. The four types of action, click, add to cart, purchase and add to favorite, are good metrics evaluating the popularity of a merchant. Also, we distribute the actions to monthly scale as to characterize the variation of the popularity for certain merchant. Combining the total and monthly count of the four action types, there are also totally 32 features associated with each merchant.

The basic statistics of the four different action types for repeat and not repeat buyers are illustrated in Table 4. Obviously, merchants with more click, add to cart, purchase and add to favorite tend to have repeat buyers, which indicates that the popularity of certain merchant can also influence whether it will have repeat buyers. However, for checking the maximum values of those action counts, we notice they are the same for both repeat and not repeat buyers, and the maximum values correspond to the same merchant (popular merchant). This indicates that even for the most popular merchant, some users may become its repeat buyers, some may not. Therefore, it will be important to extract features characterizing the interactions between users and certain merchants.

Table 4: Merchant-related features statistics (repeat and not repeat buyers)

	# click	# add to cart	# purchase	# add to favorite
repeat	mean: 69096.65 std: 137579.75 min: 380 max: 667956	mean: 104.21 std: 177.34 min: 0 max: 900	mean: 3454.92 std: 3908.60 min: 47 max: 18877	mean: 4330.68 std: 9099.75 min: 8 max: 42722
not repeat	mean: 51257.38 std: 111078.72 min: 207 max: 667956	mean: 82.38 std:145.71 min: 0 max: 900	mean: 2779.26 std: 3321.88 min: 13 max: 18877	mean: 3189.05 std: 7306.76 min: 8 max: 42722

6.4 User-merchant interaction features

User-merchant entity is the most important entity, because our task is to predict if a user will become a repeat buyer for a specific merchant. Each entity have 4 actions, click, add to cart, purchase and add to favorite. Action count means the number of the 4 actions in a specific period, for example, in each month. Using the action count, we can generate more complex features. Action ratio is a type of feature generated from action count. It is the proportion of a particular action type over all action types. Since add to cart action is pretty rare, we merge it with click action.

We calculate each month and the whole period 's action acount. The intuition behind this is that if some users buy a merchant's item regularly, or several times in the past period, another user who bought it once on "Double 11" has more possibility to become its regular buyer. Some statistics about these features among repeat and non-repeat buyers are shown in Table.5

Table 5: User-merchant behavior features statistics (repeat and not repeat buyers)

	click and add to cart	purchase	add to favorite
repeat	mean count: 14.846 max count: 3917	mean count: 1.623 max count: 10	mean count: 0.679 max count:104
not repeat	mean count: 8.727 max count: 1268	mean count: 1.321 max count: 10	mean count: 0.368 max count:107

6.5 Double 11 features

We also explore users' actions on "Double 11" day. For each user-merchant pair, we calculate each action's counts , as well as the ratio to the action counts for the whole period. Intuitively, if a user has a pretty high purchase ratio on the promotion day, he/she has a low possibility to be a repeat buyer in the following months without big promotion.

6.6 Slope features

The slope features are computed to represent the monthly variation in the four actions, for user, merchant and user merchant interactions. For example, if a user have a very high slope in monthly #purchases compared to the other users, then this user is very likely to be random user who just purchase things on Double 11 and also not likely to be a repeat buyer. The slope is calculated as below.

$$Slope = \frac{n\sum XY - \sum X \sum Y}{n\sum X^2 - (\sum X)^2}$$

Here, $X = 1, 2, 3, 4, 5, 6, 7$ which correspond to 7 months from May to December, and Y are the monthly counts of a feature, e.g. #purchases of a merchant as shown in Figure 12.

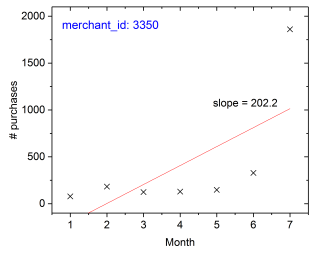

Figure 5: An example of the slope feature

6.7 Repeat buyer features

We calculate 2 repeat buyer features for merchants, repeat buyer number and repeat buyer ratio. For a merchant, repeat buyer number means the number of users who bought on at least 2 different days from the merchant, and repeat buyer ratio means the ratio of the repeat buyers to all (repeat and non-repeat) buyers for the merchant. A high repeat buyer number or ratio of a merchant indicates that the merchant is popular, which usually make a one-time buyer come again in the future.

6.8 Similarity features

Similarity between a user and a merchant is also extracted. It measures how similar between a user and a merchant on based the brands of the merchant bought by the user. For a merchant-brand pair, let N_B be the total number of purchases of the brand from all merchants, and N_{M-B} be the number of purchases of the brand from the merchant. Then the merchant's market share on that brand is defined as $\frac{N_{M-B}}{N_B}$. User-merchant similarity is defined as the inner product of the brand share vector of the merchant's brands and the vector of user's purchasing times of the merchant's brands. For example, if a merchant has 4 brands with market share (0.1,0.2,0.5,0.2), and a user's purchasing times of the 4 brands are (1,0,5,2), the similarity for that user-merchant pair is $0.1\times1+0.2\times0+0.5\times5+0.2\times2 = 3$. Higher similarity usually means a higher possibility the user will buy from the merchant again.

6.9 PCA features

PCA (Principle Component Analysis) features are extracted from the similarity between merchants. Similarity between 2 merchants is defined as the number of buyers who bought items from both the merchants. For 4995 merchants, the similarity matrix will be 4995×4995. PCA is applied on that matrix and the first 10 components are used as merchant features.

7 FEATURE RANKING

Totally we have generated 129 features and they are divideded into four basic feature groups according to their hypothesis, namely, user demographic info, user profile, merchant profile, user-merchant profile, and four complex feature groups, which are double 11 purchase features, slope, PCA, and similariy. In this section, we use the feature ranking function of XGBoost to analyze the importance of each feature group based on the AUC (area under curve) score and rank the features in each feature group based on F score.

7.1 Evaluate importance of feature groups

The result of importance of feature groups are listed in Table 6 and Table 7. The first row is the result of full feature set with 129 features.

Among the eight feature groups, the user demographic info group are from the original user profile data which provides information regarding user's age and gender. User profile feature group includes features such as user click counts, user add favourite counts , user purchase counts. and user behavior in 12 months etc. Merchant profile feature groups include features such as merchant click counts, merchat purchase counts, as well as merchant been added to cart counts for 12 months, etc. User-merchant groups includes features such as user click ratio, etc. And double 11 features includes the user's actions on "Double 11" day. PCA and slope are the complex features generated.

We use five-fold cross validation to evaluate and grid search to scan the parameters for XGBoost classifier. And the optiaml parameters that we use are as following: $nthread$: [3], $gamma$: [0.001], max_delta_step : [5], $objective$: [$'binary$: $logistic'$], $learning\_rate$: [0.04], max_depth : [7], min_child_weight : [200], $silent$: [1], $subsample$: [0.8], $colsample\_bytree$: [0.7], $n\_estimators$: [10], $missing$: [−999], $seed$: [1337]

The AUC score generated is the average AUC score of the five-fold cross validations. And we also evaluate the AUC score by excluding the feature groups from the full feature sets which is reflected in the excluded AUC column.

It is not hard to find from the table, that all features generate the highest AUC which is 0.659 and user-merchant interaction feature, PCA and merchant profile can also generate high AUC score which are 0.612, 0.6075 and 0.6074. If we exclude user-merchant interaction feature, PCA and merchant profile, we can find the AUC score are much higher. And we find if we remove double 11 purchase feature, the AUC score is as high as 0.66, which is higher than the al features result. It implies that the double 11 feature is not necessary feature and can be removed. From Figure 6, we can see ui feature group, namely user demographic info feature performs worst for AUC score, yet has second highest AUC score when being removed. It indicates that user demographic info feature is redundant and can't contribute to the performance of the classifier. And for other features, if we remove them, the AUC score will drop a little lowe than all feature scores, which means they are contributing to the result and thus can't be removed.

Table 6: Basic feature groups and the AUC score of them and excluding them

feature group	# features	AUC	excluded AUC
all features	129	0.659011522325	/
User demograph info	2	0.544381015702	0.658451598045
User profile	32	0.559563838359	0.656835731761
Merchant profile	34	0.607462937236	0.652726356603
User-merchant interactions	32	0.610887671133	0.651667163833

Table 7: Complex feature groups and the AUC score of them and excluding them

feature group	# features	AUC	excluded AUC
double 11 purchase	6	0.593725524037	0.660310950695
slope	12	0.593725524037	0.657355176676
pca	10	0.607552868314	0.658245286963

7.2 Feature Ranking

In this secion, we measure the ranking of the features in each feature group based on the importance score retrieved for each feature in a feature group. And we have combined user profile and user demographic inforation into one group. We use the feature importance method in XGBoost, which calculate the F score that

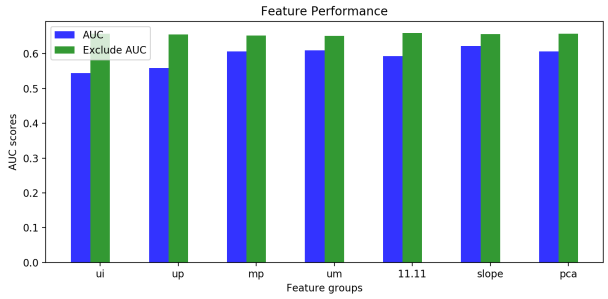

Figure 6: feature ranking in user information feature group

indicates how valueble and important in constructing the boosted decision trees in the classifier. The higher the more important.

The results for are shown in Figure: 7 8 9 10 11 12 13. In user profile group, the top features are *up_count_click_nov*, *up_count_click*, *up_count_purchase*, which implies the user click count matters ,especially that happens in November, and total purchase count is also a important factor. In merchant profile group, the top features are *mp_count_purchase_oct*, *mp_count_click_may*, which implies that the purchase count of the merchant and click count in may of this merchant matters the most among all the merchant related data. In user-merchant interaction feature group, the top features are *um_count_click*, *um_count_click_nov*. It means user click count of the merchant in total and in November are important attributes.

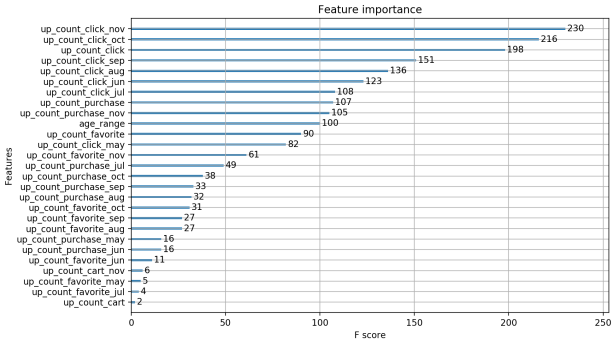

Figure 7: feature ranking in user information feature group

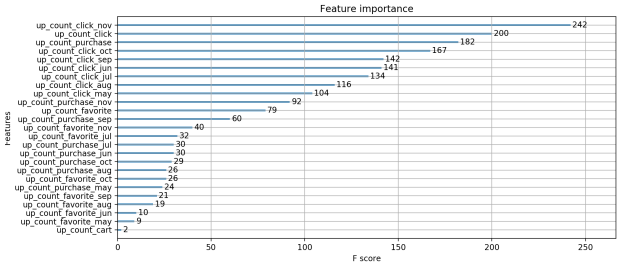

Figure 8: feature ranking in user profile feature group

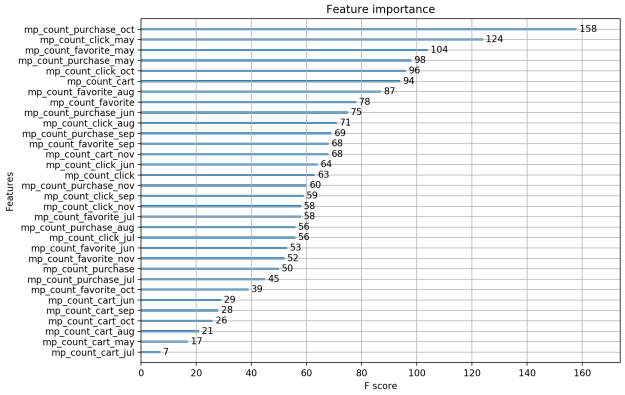

Figure 9: feature ranking in merchant profile feature group

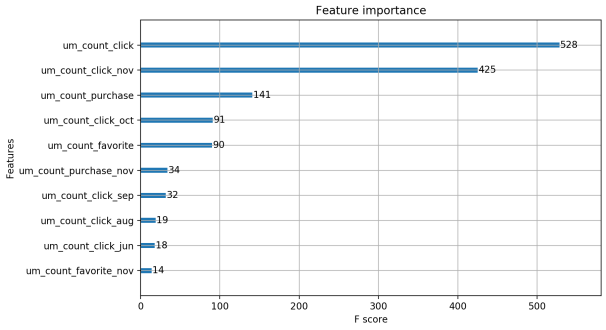

Figure 10: feature ranking in user-merchant interaction feature group

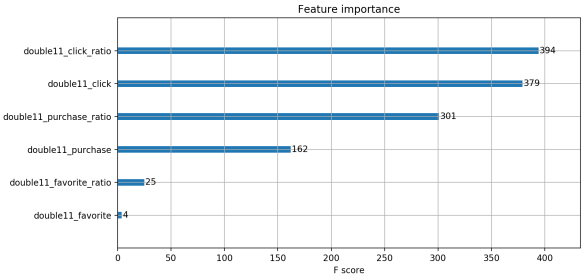

Figure 11: feature ranking in double 11 feature group

8 MODEL TRAINING

Based on all the extracted features we have trained multiple classification models, including Gradient Boosting Machines(GBM) [6], Random Forest [2], Support Vector Machine(SVM) [3], Adaboosting [5], Logistic Regression(LR) [12] and neural networks(NNs) [1]. Comparing with our results in checkpoint report, since we have applied complex features like PCA features, trend, similarity, the performance of our model training has greatly improved. In order to

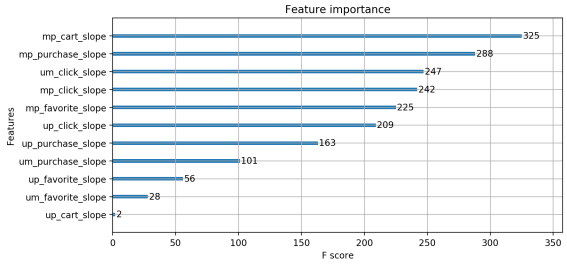

Figure 12: feature ranking in slope feature group

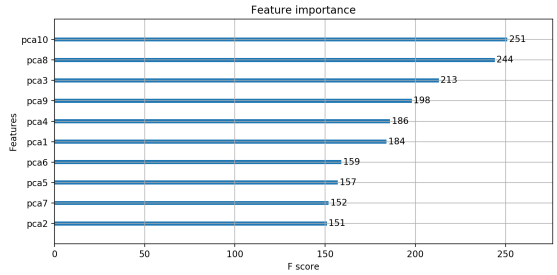

Figure 13: feature ranking in pca feature group

achieve better results, we also blend those afore-metioned models together.

In this section, the classification models we deployed will be described as well as their evaluation results.

8.1 Gradient Boosting Machines(GBM)

Gradient Boosting Machines(GBM) are widely used in regression and classification. They actually ensemble multiple prediction models(typically decision trees) together and generate the model in a stage-wise way. They allow optimization of an arbitrary differentiable loss function. There are many open-source projects based on this model and one of the most famous is Microsoft's lightGBM.

After feature engineering, we generated user demographic features(including age, gender and so on), user behavior features(like user activitiy), merchant-related features, user-merchant interaction features, double 11 related features and some complex features like PCA features, similarity and trend. The whole dataset includes around 260000 items and we randomly choose 200000 items as our training data and rest 50000(around 20% of the whole dataset) items as our testing data. In order to get more reliable results, we scale the data first. And then we use the grid search method [8] to optimize the parameters. We focus on adjusting the parameters of "num_leaves", "learning_rate", "n_estimators" and we use "max_depth", "min_data_in_leaf" to deal with over-fitting. The whole training process is very fast(within 5 seconds on OS X with 8 threads and 8G memory).

Table 8 and figure 14 show the evaluation results of our GBM model under different feature groups and overall features. Different feature groups have different effects on improving the AUC score,

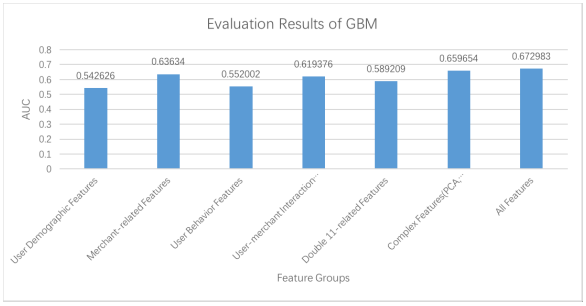

Figure 14: Evaluation Results of GBM

and when we use all the generated features together, we get the AUC score as 0.672983.

Table 8: Evaluation Results for GBM

	AUC
User Demographic Features	0.542626
Merchant-related Features	0.636340
User Behavior Features	0.552002
User-merchant Interaction Features	0.619376
Double 11-related Features	0.589209
Complex Features(PCA, Similarity, Slope)	0.659654
All Features	0.672983

8.2 Support Vectore Machine(SVM)

Support Vector Machine(SVM) is a very widely used regression and classification model. It could perform both linear and non-linear classification very well. The key strengths of SVM can be concluded as follows: first, SVM is different from traditional statistical methods since it generally does not involve the probaility measure and the law of large numbers – it is more simplified; second, the decision function of SVM is determined only by a few support vectors, in this case, the complexity of computation depends on the number of support vectors rather than the dimensions; third, SVM is robust and it currently could support the incremental data. However, there is an obvious limitation of this model, which is that when the size of training data is quite large(generally more than 10000 items), the training process will become very slow and less efficient.

In our experiment, we use different feature groups and overall features to train our SVM model. We scale the data in advance as well. While we were training, we deployed C-SVC as our SVM type and radial basis function(RBF) as our kernel function. After grid search method, we find our optimized parameters C and gamma(C is the parameter for C-SVC and gamma is the parameter for kernel function).

Table 9 and figure 15 show our training results. The performance of SVM classifier is when using AUC mertric.

8.3 Linear Regression(LR)

Linear Regression(LR) is a widely used classification which is based on linear model to analyzing a dataset in which there are one or

Table 9: Evaluation Results for SVM

	AUC
User Demographic Features	0.527624
Merchant-related Features	0.615308
User Behavior Features	0.554349
User-merchant Interaction Features	0.610259
Double 11-related Features	0.588276
Complex Features(PCA, Similarity, Slope)	0.630345
All Features	0.583437

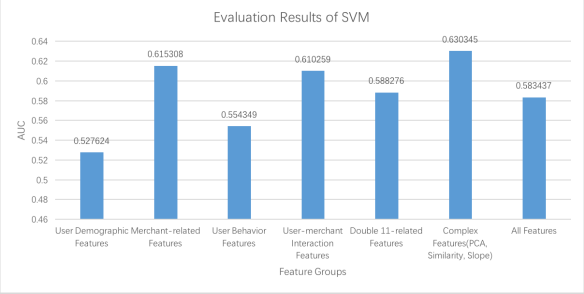

Figure 15: Evaluation Results of SVM

more independent variables that determine an outcome. It is a binary classification to classify two possible outcomes. In training the LR model, we scaled the features first and used different feature groups as well as overall features. The entire data set includes more than 260000 items and we select 20% of the data randomly as testing data, which is about 50000 items and the rest as training data set, which is about 200000 items. And while tuning the parameters of the classification, we found when class_weight="balanced", we are able to get the best result, which basically means replicating the smaller class until we have as many samples as in the larger one, but in an implicit way. Table 10 and figure 10 show the AUC score we can get is 0.672235.

Table 10: Evaluation Results for LR

	AUC
User Demographic Features	0.539039
Merchant-related Features	0.576665
User Behavior Features	0.562929
User-merchant Interaction Features	0.615636
Double 11-related Features	0.587280
Complex Features(PCA, Similarity, Slope)	0.609941
All Features	0.672235

8.4 Random Forest(RF)

Random Forest(RF) is an ensemble learning model for classification rather than a single decision tree. It grows many classification trees. In order to classify an object from an input vector, the classifier puts the input down through each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all

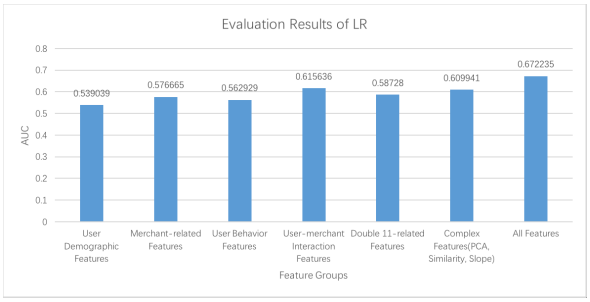

Figure 16: Evaluation Results of LR

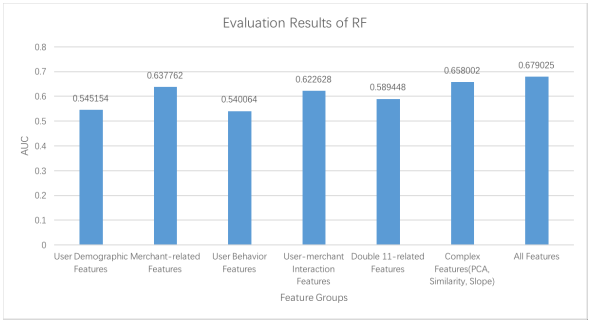

Figure 17: Evaluation Results of RF

the trees in the forest). In training the model, we scaled the features first and employed the grid search to choose the best parameters. We focus on the parameters of "n_estimators", "min_samples_split", "min_samples_leaf" and "max_depth". The features we use are the same as other classifers. The AUC scores we get are show in Table 11 and figure 17.

Table 11: Evaluation Results for RF

	AUC
User Demographic Features	0.545154
Merchant-related Features	0.637762
User Behavior Features	0.540064
User-merchant Interaction Features	0.622628
Double 11-related Features	0.589448
Complex Features(PCA, Similarity, Slope)	0.658002
All Features	0.679025

8.5 Adaboosting

Adaboosting algorithm combines the output of other weak learning algorithms into a weighted sum. At each iteration of the training process, the weight is assigned to each sample based on the current error on that sampele. These weights will be used to inform the training of the weak learning algorithms.

In our experiment, we use decision tree as our weak learners. After scaling the features, we used grid search method to find the

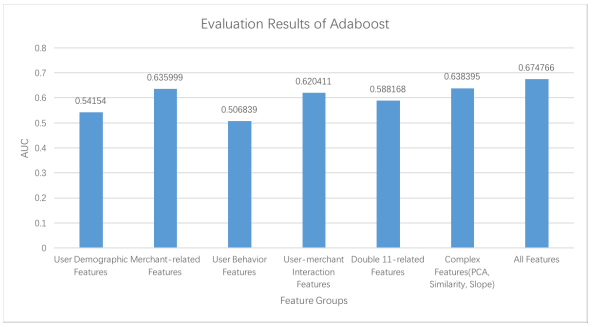

Figure 18: Evaluation Results of Adaboost

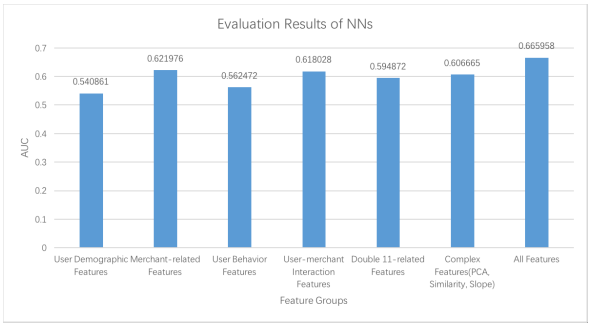

Figure 19: Evaluation Results of NNs

best parameters, including "n_estimators" and "learning_rate". Table 12 and figure 18 show the result of our training process.

Table 12: Evaluation Results for Adaboost

	AUC
User Demographic Features	0.541540
Merchant-related Features	0.635999
User Behavior Features	0.506839
User-merchant Interaction Features	0.620411
Double 11-related Features	0.588168
Complex Features(PCA, Similarity, Slope)	0.638395
All Features	0.674766

8.6 Nerual Networks(NNs)

Nerual Networks are learning systems inspired by the biological nerual networks. They contain multiple neurons which are orignized in layers and data travels from the input layer to output layer to generate the learning results. In our experiment, we deployed multi-layer perceptron(MLP) algorithm[1] to train our NNs. The weights are trained by using backpropagation[7] algorithm. Grid search method was used to find the best parameters like "alpha"(regularization term parameter), "learning_rate" and "hidden_layer_sizes". When finding the "hidden_layer_sizes", we also followed an empirical rule: the number of neurons in the first hidden layer equals to (number of inputs + number of outputs)*2/3. After scaling the features and finding the best parameters, we trained our NNs model and Table 13 and figure 19 show the results.

Table 13: Evaluation Results for NNs

	AUC
User Demographic Features	0.540861
Merchant-related Features	0.621976
User Behavior Features	0.562472
User-merchant Interaction Features	0.618028
Double 11-related Features	0.594872
Complex Features(PCA, Similarity, Slope)	0.606665
All Features	0.665958

8.7 Blending Model

To further improve the performance of prediction, we deploy a blending model to combine all the single classifiers together. The blending model is basically a weighted sum following the formula:

$$p = \sum_{i=1}^{k} w_i * p_i, \tag{1}$$

where p is the probability that a specific user will become a repeated buyer of a specific merchant after blending, p_i is the probability predicted by the ith classifier, w_i is the weight assigned to the ith classifier and k is the number of our models.

Here we manually assign weights to our single models, where single models with higher AUC score receive bigger weights. Table 14 and figure 20 indicate the results of our blending model.

Table 14: Evaluation Results for Blending Model

	AUC
User Demographic Features	0.544518
Merchant-related Features	0.641445
User Behavior Features	0.565093
User-merchant Interaction Features	0.624173
Double 11-related Features	0.594873
Complex Features(PCA, Similarity, Slope)	0.662468
All Features	0.682731

8.8 Discussion

From the figures of our evaluation results, we can find that when we put all of our generated features together, classifiers could achieve the best performance. What's more, complex features(including PCA, Similarity, Slope), merchant-related features and user-merchant interaction features contribute most to our model training process. Basically, the ranking of different feature groups is complex features contribute most, then merchant-related features, user-merchant interaction features, double 11-related features, user purchasing behavior features and user demographic features. This finding matches perfectly with the results of our feature importance analysis.

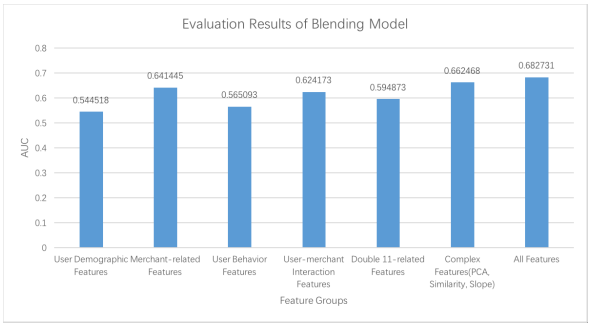

Figure 20: Evaluation Results of Blending Model

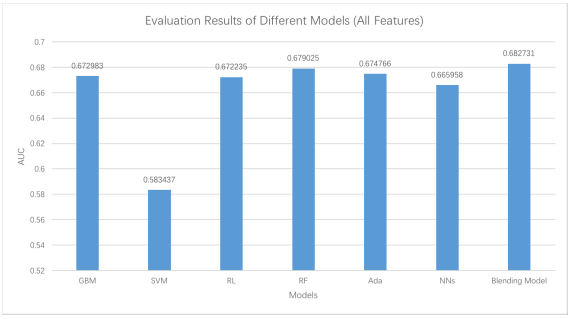

Figure 21: Evaluation Results of Each Model

In table 15, we also aggregate each model's evaluation results together. From figure 21 we can see each model's performance. Generally speaking, ensemble algorithms (Random Forest, Adaboost) perform better than linear algorithm and other single models. Among all the non-blending models, random forest has the best performance. Although other models' scores are not as high as random forest's, they still contribute a lot to the overall AUC score in the blending model.

Table 15: AUC scores of each model

	GBM	SVM
User Demographic Features	0.542626	0.527624
Merchant-related Features	0.636340	0.615308
User Behavior Features	0.552002	0.554349
User-merchant Interaction Features	0.619376	0.610259
Double 11-related Features	0.589209	0.588276
Complex Features(PCA, Similarity, Slope)	0.659654	0.630345
All Features	0.672983	0.583437

Table 16: AUC scores of each model (Continue)

LR	RF	Adaboost	NNs	Blending Model
0.539039	0.545154	0.541540	0.540861	0.544518
0.576665	0.637762	0.635999	0.621976	0.641445
0.562929	0.540064	0.506839	0.562472	0.565093
0.615636	0.622628	0.620411	0.618028	0.624173
0.587280	0.589448	0.588168	0.594872	0.594873
0.609941	0.658002	0.638395	0.606665	0.662468
0.672235	0.679025	0.674766	0.665958	0.682731

9 CONCLUSION

In this project, we proposed a novel solution to predict repeat buyers based on the datasets provided by Tmall.com. In order to achieve our goal, our project is basically divided into four steps: data preprocessing, feature engineering, classification and evaluation. Among these four parts, we believe that feature engineering is the key to success.

In the data preprocessing part, we focus on dealing with unbalanced data and missing data. In feature engineering, we not only generate 129 features, but also categorize them into different feature groups and evaluate the contribution of each feature group. After the analyzing process, we find that complex features, merchant-related features and user-merchant interaction features contribute most to our model training process. In model training, we innovatively deploy nerual networks and ensemble algorithms like random forest and Adaboost. At the same time, we also blend all the classifiers together by assigning weights to them and generate better results. At last, our framework's best performance has reached to the AUC score of 0.682731.

ACKNOWLEDGMENTS

We would like to thank Prof.Lv for providing us with such a good opportunity to deepen our understanding of data mining.

REFERENCES

[1] Yoshua Bengio et al. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.
[2] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[3] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 27.
[4] Zhanpeng Fang, Zhilin Yang, and Yutao Zhang. [n. d.]. Collaborative Embedding Features and Diversified Ensemble for E-Commerce Repeat Buyer Prediction. ([n. d.]).
[5] Yoav Freund and Robert E Schapire. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*. Springer, 23–37.
[6] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
[7] Yoshio Hirose, Koichi Yamashita, and Shimpei Hijiya. 1991. Back-propagation algorithm which varies the number of hidden units. *Neural Networks* 4, 1 (1991), 61–66.
[8] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification. (2003).
[9] Farshad Kooti, Kristina Lerman, Luca Maria Aiello, Mihajlo Grbovic, Nemanja Djuric, and Vladan Radosavljevic. 2016. Portrait of an online shopper: Understanding and predicting consumer behavior. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 205–214.
[10] Guimei Liu, Tam T Nguyen, Gang Zhao, Wei Zha, Jianbo Yang, Jianneng Cao, Min Wu, Peilin Zhao, and Wei Chen. 2016. Repeat buyer prediction for e-commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 155–164.
[11] Caroline Lo, Dan Frankowski, and Jure Leskovec. 2016. Understanding Behaviors that Lead to Purchasing: A Case Study of Pinterest.. In *KDD*. 531–540.
[12] John Neter, Michael H Kutner, Christopher J Nachtsheim, and William Wasserman. 1996. *Applied linear statistical models*. Vol. 4. Irwin Chicago.
[13] Ye Tian, Zuoliang Ye, Yufei Yan, and Miao Sun. 2015. A practical model to predict the repeat purchasing pattern of consumers in the C2C e-commerce. *Electronic*

Commerce Research 15, 4 (2015), 571–583.
[14] Armando Vieira. 2015. Predicting online user behaviour using deep learning algorithms. *CoRR* abs/1511.06247 (2015). http://arxiv.org/abs/1511.06247

A APPENDIX

A.1 Honor Code Pledge:

On my honor as a University of Colorado at Boulder student I have neither given nor received unauthorized assistance on this work.

A.2 Individual Contribution:

Xu Han: model training(GBM, LR, SVM, RF, Ada, NN, Xgboost), model blending, evaluation and analysis
Yichen Wang: feature extraction(user-merchant interaction , repeat buyer, pca and similarity features), help some bug fixing on model training
Yawen Zhang: data management in DB, data preprocessing (missing and unbalanced problem), feature generation
Xiaolan Cai: feature performance analysis, feature importance ranking