Repeat Buyer Prediction for E-Commerce

Youzheng Fang, Tianhao Lin, Yunxuan Yang

March 10, 2025

1 Abstract

Predicting repeat buyers is a critical challenge in the e-commerce sector, directly influencing customer retention strategies and long-term profitability. This study focuses on predicting repeat buyers on Tmall, one of the largest e-commerce platforms. Leveraging a robust feature engineering pipeline inspired by prior research, we analyze user behavior, merchant performance, and their interactions. We employ multiple machine learning models, including logistic regression, SVM, Random Forest, and XGBoost, to evaluate their performance. Our results reveal that merchant features and user-merchant interactions significantly contribute to prediction accuracy, while demographic information has a limited impact. Among the tested methods, XGBoost exhibits superior predictive capability but with higher variance. This paper offers actionable insights for feature selection and model optimization, laying a foundation for enhancing e-commerce customer retention strategies.

2 Introduction of the problem

Repeat purchase behavior is a cornerstone of success for e-commerce platforms, where customer retention is often more cost-effective than acquiring new users. Identifying repeat buyers allows platforms to implement personalized marketing strategies, optimize resource allocation, and foster long-term customer loyalty. Tmall, as a leading e-commerce platform, provides a compelling context to explore this challenge due to its vast user base and diverse product offerings.

In this project, we aim to predict the probability of a user making repeat purchases with a specific merchant. Our approach is grounded in a detailed feature engineering pipeline, inspired by Liu et al.'s work on repeat buyer prediction in e-commerce. By leveraging rich datasets, including user demographics, behavioral logs, and merchant performance metrics, we construct a comprehensive set of features to capture user behavior and their interactions with merchants.

To evaluate the effectiveness of our approach, we apply and compare several machine learning models, including logistic regression, SVM, Random Forest, and XGBoost. These models are assessed based on their accuracy, convergence speed, and Area Under the Curve (AUC) performance. Furthermore, we conduct ablation studies to identify key features, such as user activity patterns, merchant click-through rates, and user-merchant interaction metrics, which significantly influence model predictions.

Through this study, we aim to contribute to the ongoing development of predictive analytics in e-commerce, offering insights into effective feature engineering and model selection to enhance customer retention strategies.

3 Models applied

3.1 Features engineering

Feature engineering plays a crucial role in predicting repeat buyers in e-commerce. In this project, we implemented a comprehensive feature engineering pipeline inspired by the paper: Repeat Buyer Prediction for E-Commerce[1]. The features were designed to capture user behavior, merchant performance, and user-merchant interactions across multiple dimensions, ensuring a robust representation of the data for predictive modeling.

We are provided with four datasets: test_format1.csv, train_format1.csv, user_info_format1.csv and user_log_format1.csv. Here are samples in them:

	user_id	**merchant_id**	**Prob**
0	163968	4605	NaN
1	360576	1581	NaN

Table 1: Data in test_format1.csv

	user_id	**merchant_id**	**label**
0	34176	3906	0
1	34176	121	0

Table 2: Data in train_format1.csv

	user_id	**age_range**	**gender**
0	376517	6.0	1.0
1	234512	5.0	0.0

Table 3: Data in user_info_format1.csv

Table 1 and Table 2 show the testing data and training data. The label column in Table 2 can be either 0 or 1, where 1 indicates a repeat buyer and 0 indicates a non-repeat buyer. The Prob column in Table 1 is the probability that a given customer is a repeat buyer for a specific merchant, ranging from 0 to 1, which is what we want to predict. Table 3 shows users' age and gender, where the age is divided into seven ranges and the gender is divided into 3 kinds (0 is female, 1 is male, 2 is unknown). Table 4 shows users' log information. The action_type takes four values: 0 for click, 1 for add-to cart, 2 for purchase and 3 for add-to-favorite. Products sold in different merchants are assigned different item_ids even if the products are exactly the same. The time_tamp is in the form of "mmdd", which is the purchase time of an item.

Count/Ratio Features. Count and ratio features capture the overall behavior and preferences of users, merchants, and user-merchant interactions. These features include counts of actions such as clicks, purchases, adding items to the cart, and favorite items, as well as their ratios. Count and ratio features can be divided into five different kinds: action counts, action ratio, day counts, day ratio, and item/category/brand/merchant counts.

Action counts are number of actions like clicks, purchases, add-to-favorite actions. We use variables like total_interactions, action_0_count to describe these basic features.

Action ratio is the proportion of a particular action type over all action types. This is important because if there are two people and both of them bought 10 items in the past 6 months, one person did 10 purchases with 10 clicks while the other did 10 purchases with 1000 actions; their shopping behavior is obviously different, so we need to take ratio into consideration. We use variables like purchase_to_click_ratio to describe action ratio features.

Day counts are the number of days users are active, i.e. they do some actions. We use variables like total_active_days and active_days_0 to describe the number of active days a user is overall and the number of active days that correspond to a certain action.

	user_id	item_id	cat_id	seller_id	brand_id	time_stamp	action_type
0	328862	323294	833	2882	2661.0	829	0
1	328862	844400	1271	2882	2661.0	829	0

Table 4: Data in user_log_format1.csv

Day ratio is similar to action ratio, which is the proportion of of a particular active day over all active days. A person who did the click action every day is very likely to have a different shopping behavior with a person who did the click action once a month, though they did the same number of click action in the past 6 months.

Item/category/brand counts are the number of unique items, categories, brands, and merchants interacted with by the user. We use variables like unique_items and unique_categories to describe these features.

Age/Gender Related Features. Age/Gender Related Features leverage demographic information to identify behavioral patterns across different age groups and genders. Since the age range is divided into several groups and the gender is divided into three kinds, we use one-hot encoded age groups to represent different age ranges and one-hot encoded gender information to differentiate between male, female, and unknown gender. Table 5 is an example of dealing with a person whose gender is unknown:

gender_0	gender_1	gender_2
0	0	1

Table 5: One-hot code for a person whose gender is unknown

Repurchase Features. Repurchase features capture the tendency of users and merchants to engage in repeat transactions, highlighting loyalty patterns and effective interactions in the e-commerce environment. These features provide insights at user, merchant, and user-merchant levels.

User repurchase rate is calculated as the fraction of merchants with which the user made multiple purchases. First, we identify purchases (action_type == 2) and group by user_id and merchant_id to count unique purchase days. Second, we mark a merchant as a repeat purchase (repurchase_flag = 1) if the user has purchased on more than one unique day. Finally, we compute the repurchase rate for each user by averaging the repurchase_flag values across all merchants. This feature highlights user loyalty and the propensity for repeated engagement across merchants.

Merchant repurchase rate is calculated as the fraction of users who made repeat purchases with that merchant. We use the same repurchase_flag derived from user-level calculations and then we group by merchant_id and compute the average of repurchase_flag values across all users. This feature identifies merchants that successfully retain users and foster repeat purchases.

User-Merchant repurchase rate captures the repeat purchase relationship between specific users and merchants. We use the repurchase_flag to directly assign whether a user has made multiple purchases with a specific merchant and then merge this flag into the user-merchant interaction features. This feature emphasizes the strength and depth of the relationship between individual users and merchants, providing a granular view of loyalty.

3.2 Logistic Regression

First of all we construct linear function (w is the weight and f is the feature):

$$z = w_1 f_1 + w_2 f_2 + \cdot + w_n f_n + b$$

Then using sigmoid function mapping output to range [0, 1]:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Next we predict the probability and calculate the log-likelihood loss:

$$L(w) = -[ylog(\sigma(z)) + (1 - y)log(1 - \sigma(z))]$$

We use the model to iterate 1000 rounds and optimize the loss function using gradient descent to find the optimal parameters w and b.

3.3 SVM

The SVM method we used mainly involves soft interval and rbf kernel function methods, and the specific optimization objectives are as follows:

$$min\frac{1}{2}||w||^2 + C\sum \xi_i$$

$$subject\ to : y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

where $K(x, y) = \Phi(x)^T \Phi(y) = e^{(-\gamma||x-y||^2)}$

3.4 Random Forest

Random Forest is an ensemble learning method based on decision trees, which performs classification or regression by building multiple decision trees. Each decision tree is trained on different subsets of the data, and during the construction process, Random Forest randomly selects features to split, thus increasing the diversity and robustness of the model. Ultimately, Random Forest obtains the final result by voting for the predictions of all trees.

3.5 XGBoost

XGBoost is based on Gradient Boosting Decision Trees (GBDT) algorithm. GBDT is an ensemble learning approach that enhances model performance by training a series of decision trees. Each tree is trained on the residual of the previous tree.

XGBoost prevents model overfitting by introducing L1 (Lasso) and L2 (Ridge) regularization terms. Through regularization, XGBoost controls the complexity of the model and increases the ability to generalize.

4 Numerical results

To use relatively simple and familiar methods, we prioritized SVM and logistic regression for our experiments. Logistic regression performed well, converging quickly (in about two seconds) and achieving a decent prediction accuracy and AUC rate. However, SVM converged extremely slowly, forcing us to run it on a cluster. After one day and 13 hours, we finally obtained the results, but the AUC rate was very unsatisfactory. Considering that we had already used the rbf kernel function, which is effective for handling nonlinearity, we concluded that SVM was not feasible for this problem and abandoned this method.

Model	Train Accuracy	Test Accuracy	ROC AUC
Logistic Regression	0.93857	0.93832	0.66821
SVM	0.93918	0.93976	0.54404

Table 6: Comparison of Initial Methods

We then tested the above models and compared their performance (AUC) on the features we used. We ran ten sets of experiments with each model and visualized the results with bar charts and box statistics.

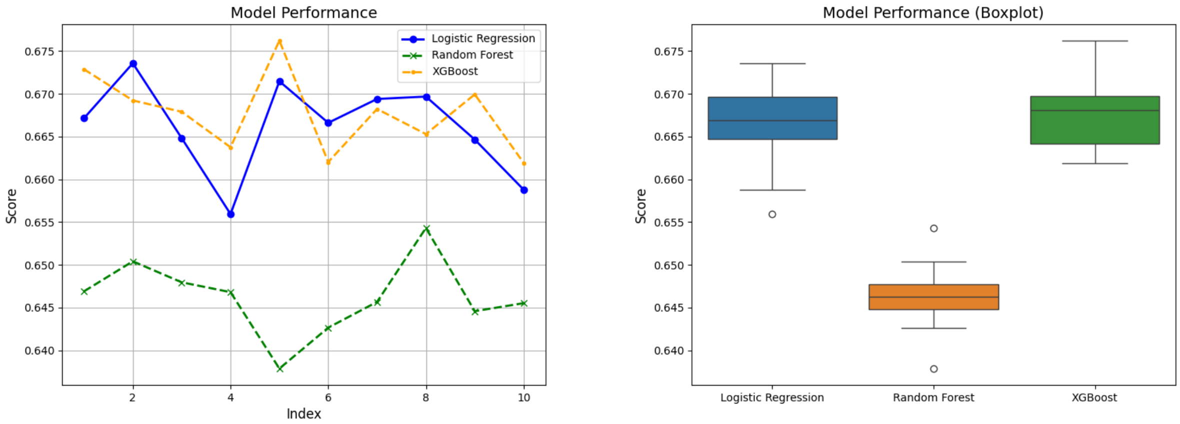

Figure 1: Comparison of model effect

Model Comparison. It can be seen that random forest has a small variance but a low mean performance, while XGBoost has the largest mean but also a large variance. Relatively speaking, logistic regression and XGBoost have the best performance in processing this feature group.

For logistic regression, it is easy to understand and implement, and it has higher interpretability. Also, logistic regression shows consistent results across different runs. However, it may not capture complex relationships between features as well as more advanced models like Random Forest and XGBoost. It assumes linearity between features and the target variable, which may not always hold true in our case.

For random forest, it handles non-linearity well as it can capture non-linear relationships between features and the target. Also, random forest is less sensitive to hyperparameters, which means it can perform reasonably well even without extensive hyperparameter tuning. However, it's harder to understand or explain why specific decisions were made, especially when compared to models like logistic regression. Additionally, random forest can be computationally expensive, especially with large datasets and many trees.

For XGBoost, it is designed to be highly efficient and fast, handling large datasets better. It is capable of capturing complex patterns in the data, handling both linear and non-linear relationships. However, XGBoost requires more tuning and can be complex to set up compared to logistic regression and random forest. While feature importance can be derived, understanding the underlying model behavior is more challenging than logistic regression.

Explanation of the Results. Logistic Regression is a linear model, meaning it assumes that the relationship between the features and the outcome (repeat buyer vs. non-repeat buyer) can be described by a straight line or hyperplane. If the data has complex, non-linear relationships between features, LR may not capture those effectively, leading to moderate performance.

Random Forest excels at capturing complex, non-linear interactions between features, but it requires the right data structure. If the data contains features that are not relevant or if there is a high amount of noise, RF might struggle to separate the classes properly. The fact that RF performed worse than LR and XGBoost suggests that the feature set we make in the feature engineering part might not be ideal for RF's decision tree-based approach.

XGBoost is based on gradient boosting, a technique that builds trees iteratively, where each new tree tries to correct the errors of the previous one. This approach allows XGBoost to better capture complex, non-linear relationships between features, leading to higher predictive power than both LR and RF.

Feature comparison. Next, we will focus on adjusting the feature to determine which indicators have the most obvious impact on the model performance.

First of all, age and gender are obscured. The feature generated in this step does not contain any information about age,gender or a thermal code. Under this condition, we use two methods, logistic regression and XGBoost, to conduct a comparative experiment. We mainly determine the performance effect of the AUC.

From Figure 2, we can see that masking gender and age has no significant effect on our AUC, with a slight decrease (0.001) in logistic regression and a slight increase (0.002) in XGBoost. Therefore, we believe that user gender and age are not the key feature.

Next, we consider the user feature, which mainly includes the features of the user's purchase information, such as the records of user clicks, favorites, visits and the ratio to purchase, as well as the active days and the proportion of each behavior type. By masking such data, we get a new set of features. The results of this set of feature training are shown in Figure 2. Similarly, logistic regression has a slight decline (0.008) while XGBoost has a certain improvement (0.004). Therefore, we believe that user feature is not the key feature either.

Next, we mask the merchant feature, which mainly includes the click-through rate, purchase rate and collection rate of each merchant. The results are shown in Figure 2. This time, there are obvious changes, and the AUC rate of the two categories of classifiers is significantly decreased (0.3∼0.4) compared with before, so it can be shown that the merchant feature is a more important feature.

Finally, we comprehensively consider the interaction information between users and merchants, which is a group of user and merchant information pairs, including user-merchant purchase click ratio and other data. After masking such data, we conduct model training, and the results are similar to that of masking merchant feature. AUC of both models decreases to a certain extent (0.2). It shows that this kind of feature is also the key feature.

Next, we conducted a comparative experiment on different subcategories of these two categories. Based on different subcategories, we took additional consideration of the scale information and divided the information of merchants of different sizes into five categories according to how much they were in each subcategory. Then, only the data of this category was used for AUC calculation, and the calculation results were shown in the figure

The first line represents the information of the first category, the merchant information, There are five subclasses: $merchant\_total\_interactions, merchant\_unique\_items, merchant\_unique\_categories,$
$merchant\_unique\_users, andmerchant\_total\_active\_days$, and X axis represents data types of different scales, ranging from small-scale merchant information to large-scale merchant information. The second row represents the information of the second major category, that is, the information of user interaction with merchants. There are five subclasses: $um\_total\_interactions, um\_total\_active\_days, um\_unique\_items,$
$um\_unique\_categories, um\_unique\_brands$. The same is true for X axis information. As can be seen from the figure, most of the information of large-scale merchants has a good positive rate, and the interaction information between users and merchants is the same. To some extent, this conforms to probability and logic: large merchants have relatively stable user profiles, while the data volume is large, the variance of the data is small, and the prediction is relatively stable.

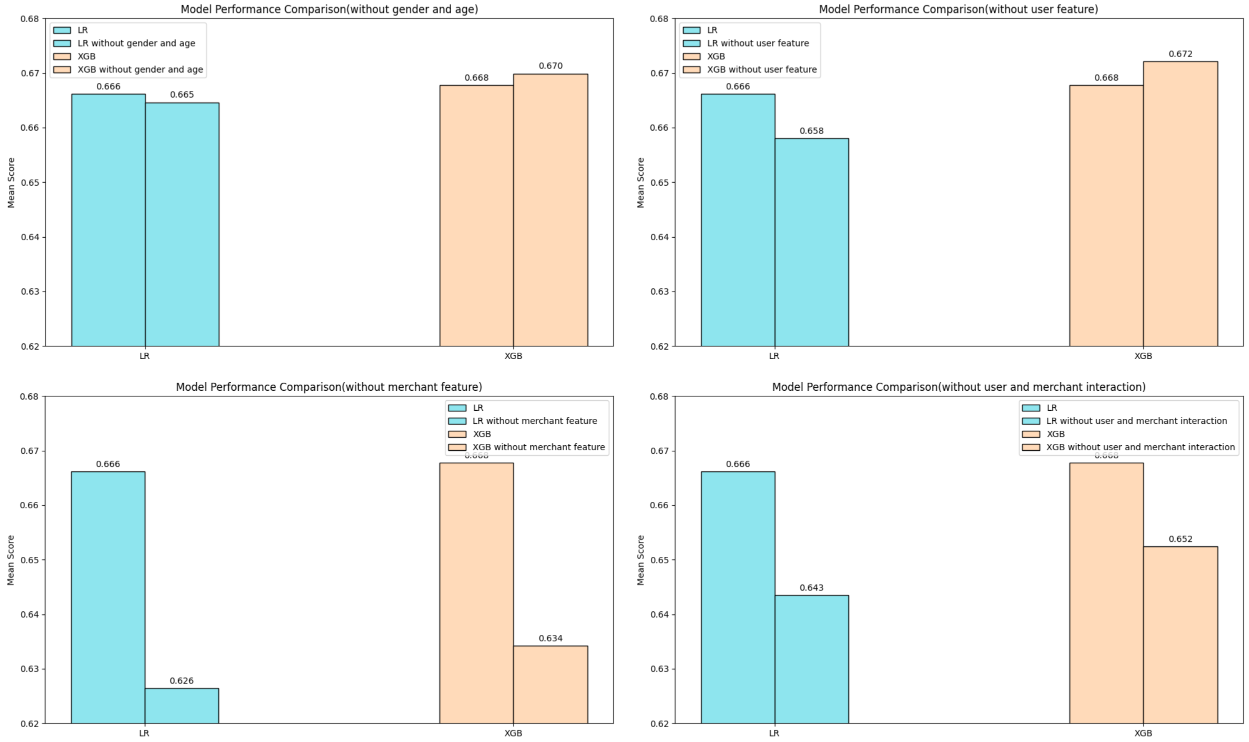

Figure 2: Feature Importance Analysis

5 Conclusions

This study focuses on predicting repeat purchase behavior in the e-commerce sector, specifically targeting Tmall's extensive platform. By utilizing a well-structured feature engineering pipeline, we analyzed diverse datasets, including user transaction records, demographic data, and behavioral logs, to construct a robust set of features. These features effectively capture user behavior, merchant performance, and the dynamics of user-merchant interactions.

Several machine learning models were implemented and compared, including logistic regression, SVM, Random Forest, and XGBoost. Among these, logistic regression demonstrated fast convergence and reliable performance, making it a practical choice for simpler scenarios. XGBoost emerged as the most effective model in terms of prediction accuracy and AUC performance, albeit with slightly higher computational demands. SVM, on the other hand, was deemed unsuitable due to its inefficiency and suboptimal results on large-scale datasets.

The feature importance analysis revealed key insights into the factors driving repeat purchase behavior. While demographic features such as age and gender were found to have minimal impact, merchant-related features and user-merchant interaction metrics, such as purchase-to-click ratios and merchant click-through rates, proved to be highly influential. These findings emphasize the importance of focusing on actionable features to enhance predictive performance.

Our experiments further highlighted the importance of evaluating models not only on accuracy but also on computational efficiency and generalizability. For instance, XGBoost's superior predictive power makes it ideal for tasks requiring high accuracy, whereas logistic regression may be preferred for applications prioritizing speed and interpretability.

This research offers practical recommendations for e-commerce platforms seeking to optimize customer retention strategies. By leveraging predictive analytics, businesses can identify high-potential repeat buyers and impleme

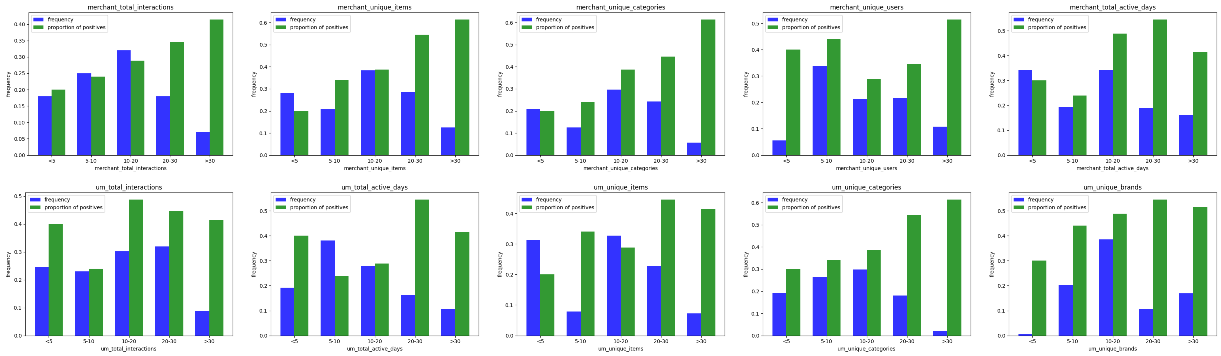

Figure 3: Subfeature Importance Analysis

6 References

[1] Liu G , Nguyen T T , Zhao G ,et al.Repeat Buyer Prediction for E-Commerce[C]//the 22nd ACM SIGKDD International Conference.ACM, 2016.DOI:10.1145/2939672.2939674.