CALIFORNIA STATE UNIVERSITY
LONG BEACH

Final Report
On
Repeat Buyer Prediction for E-Commerce

Submitted By:

Diksha Patil	030849765
Pratik Jadhav	030880471
Anthony Martinez	012500759
Sudarshan Powar	030845787
Pavan More	030873555

Submitted To:
Department of Computer Science and Computer Engineering
Long Beach
CA, United States

Under the Supervision of
Prof. Mahshid Fardadi

May, 2023

DECLARATION

We hereby declare that the report of the project entitled "Repeat Buyer Prediction for E-Commerce" which is being submitted to the Department of Computer Science and Computer Engineering, CSULB, in the partial fulfillment of the requirements for the award of the Degree of Master of Science in Computer Science is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work

Diksha Patil	030849765	_____
Pratik Jadhav	030880471	_____
Anthony Martinez	012500759	_____
Sudarshan Powar	030845787	_____
Pavan More	030873555	_____

DATE: May, 2023

CONTENTS

Repeat Buyers Prediction

Abstract

In this report, we describe a solution to the Repeat Buyers Prediction problem. We solved the machine learning problem as part of our final project for a Pattern Recognition course at CSU Long Beach. Our code performed well on the data provided.

1 Introduction

Promotions are a common strategy employed by both large and small vendors to attract new customers. Discounts are frequently offered during special shopping events, such as Double 11 Day in China and Black Friday in the US, with the goal of converting some of these new buyers into loyal customers who will return even when prices return to normal levels. This allows the store owners to recoup the revenue lost due to the discounts and ultimately increase their profits. However, not all new customers will become loyal customers, so vendors need to target their promotion campaigns towards those most likely to develop loyalty in order to maximize their return on investment (ROI).

To optimize the ROI of a promotion campaign, vendors can leverage machine learning tools if they have sufficient data. While this may be difficult for small physical stores, large e-commerce platforms like Amazon can use their extensive transaction and user data to create datasets with customer characteristics or transaction details over time.

A dataset was provided in two formats (section 2), which consisted of a set of customer-vendor pairs with labels indicating whether the customer became a repeat buyer at the vendor's store. To solve this problem, teams were required to submit a CSV file with their predictions for the labeled pairs, obtained using their machine learning model. The model evaluated the accuracy of these classifications using a ROC AUC score.

To generate their predictions, we first cleaned and preprocessed the data (section 3) before constructing features expected to have the strongest correlation with the output labels. Then we employed a combination of gradient boosting (section 4.1) and ensemble methods (section 4.2) to perform classification.

2 The Dataset

We were provided with a dataset containing two main types of information.

1. Customer demographic information, such as age and gender

2. Customer-merchant interaction data:

- Label indicating whether the customer is a repeated buyer (training dataset)
- Activity log: one record (with timestamp, category, brand and item number, plus the action type) for each item that was clicked, added to cart, purchased or added to favorite

To protect the buyers' and the vendors' privacy, the data was anonymized, and further it was also sampled in a biased way.

The data is offered in two formats. The first format (file `data_format1.zip`), divides the data in 4 tables, and is structured in a way that makes feature engineering easier. The second format (file `data_format2.zip`) is more compact, as it consists of a single table, and minimizes the redundancy of information. Because our goal was to extract features, we picked format 1 (section 2.1).

2.1 The feature-engineering-ready dataset format

The dataset from `data_format1.zip` is organized in the following 3 tables: the User Profile Logs (Table 1), the User Behaviour Logs (Table 2), and the Training and Testing Data (Table 3)

Data Field	Description	Data Type
`user_id`	The unique ID identifying each buyer	Integer
`age_range`	The user's age range, encoded as follows: 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 or NULL for unknown age,	Non-negative Integer or NULL
`gender`	The buyer's gender, encoded with 0 for female, 1 for male, 2 and NULL for unknown.	Integer or NULL

Table 1: The User Profile Table

Data Field	Description	Data Type
`user_id`	The unique ID identifying each buyer	Integer
`item_id`	The unique ID identifying each possible item that can be bought	Integer
`cat_id`	The unique ID identifying each possible category that an item can belong to	Integer
`merchant_id`	The unique ID identifying each vendor	Integer
`brand_id`	The unique ID identifying each brand an item can belong to	Integer
`time_stamp`	The date (`mm`: month and `dd`: day) when an action took place	String in `mmdd` format
`action_type`	The action taken by the buyer with respect to a vendor and an item. Encoded as follows: 0 for a click, 1 for add-to-cart, 2 for purchase and 3 for add-to-favorite.	Integer

Table 2: The User Behavior Logs Table

Data Field	Description	Data Type
user_id	The unique ID identifying each buyer	Integer
merchant_id	The unique ID identifying each vendor	Integer
label	A binary value indicating whether user_id became a repeated buyer at merchant_id. Encoded as: 1 for repeat buyer, 0 for non-repeat buyer. The label is only available for the training portion of the data	Binary number (train- ing), or empty (test- ing)

Table 3: The Training and Testing Data Table

3 Data Visualization

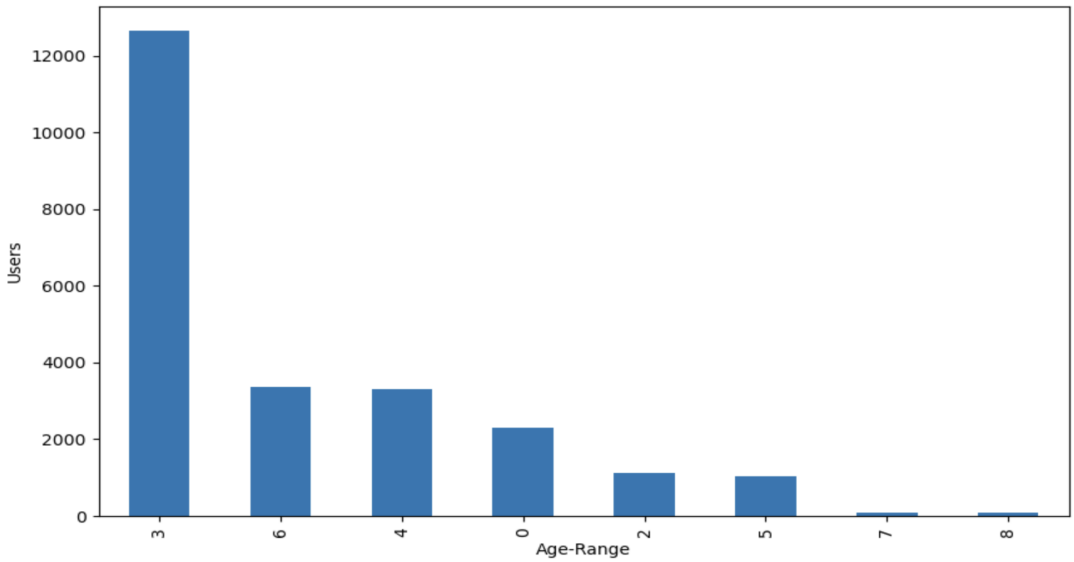


Figure 1: Users from different age-range

The graph above (Figure 1) indicates that more than 50% of users belong to the age category of 25 to 29 years old, which is labeled as category 3 on the x-axis. This suggests that this age group is the most prevalent among the users in the dataset, and therefore, it is likely to have a significant impact on the prediction task.

In the below graph (Figure 2), it shows that the highest number of clicks and purchases were made during the double 11 sale, which is a major shopping event in many countries, particularly in China. This event is also known as Singles' Day or 11.11, and it occurs on November 11th every year.
Merchants can use this information to determine which products are most popular during this time and offer discounts or promotions on those items. They can also adjust their advertising and promotional campaigns to target customers during the period leading up to and during the sale. By doing so, merchants can potentially increase their sales and revenue during this period.

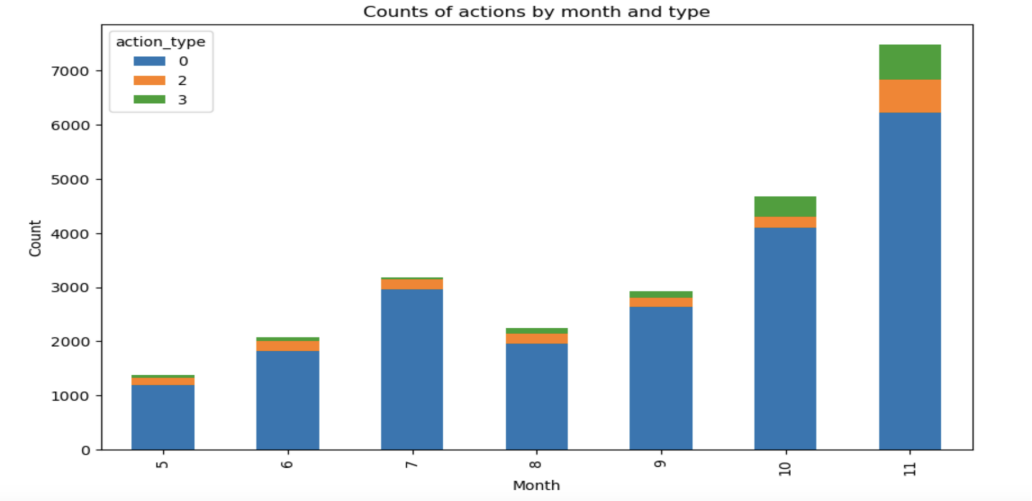


Figure 2: Counts of actions by month and type

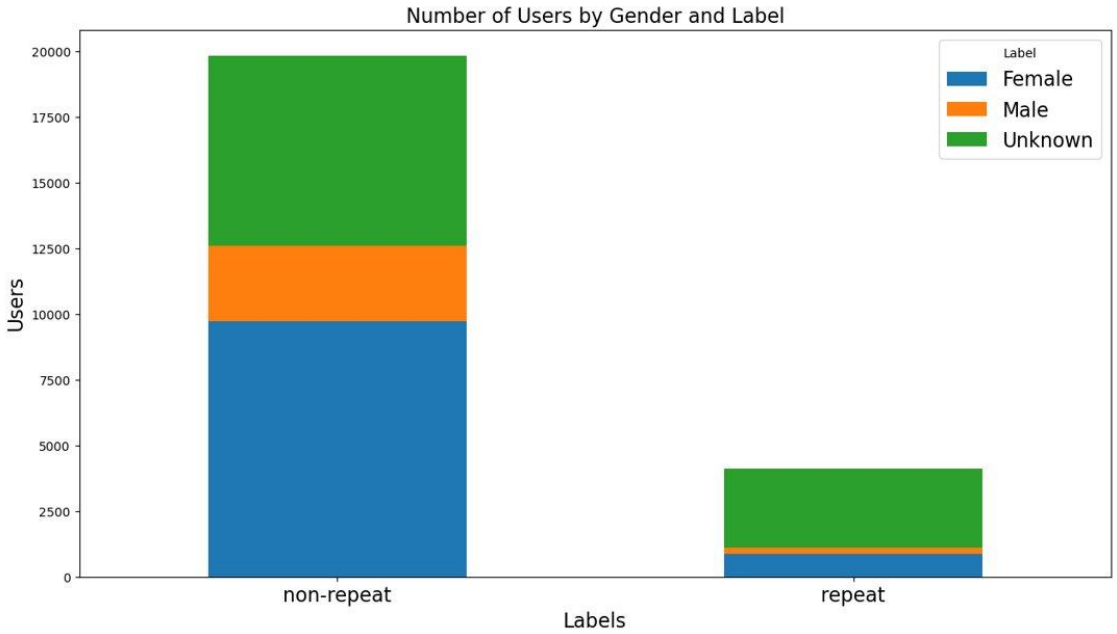


Figure 3: Number of users by Gender and Label

In the above graph (Figure 3), it shows that the number of non-repeat users is very high compared to the number of users who were repeat customers.

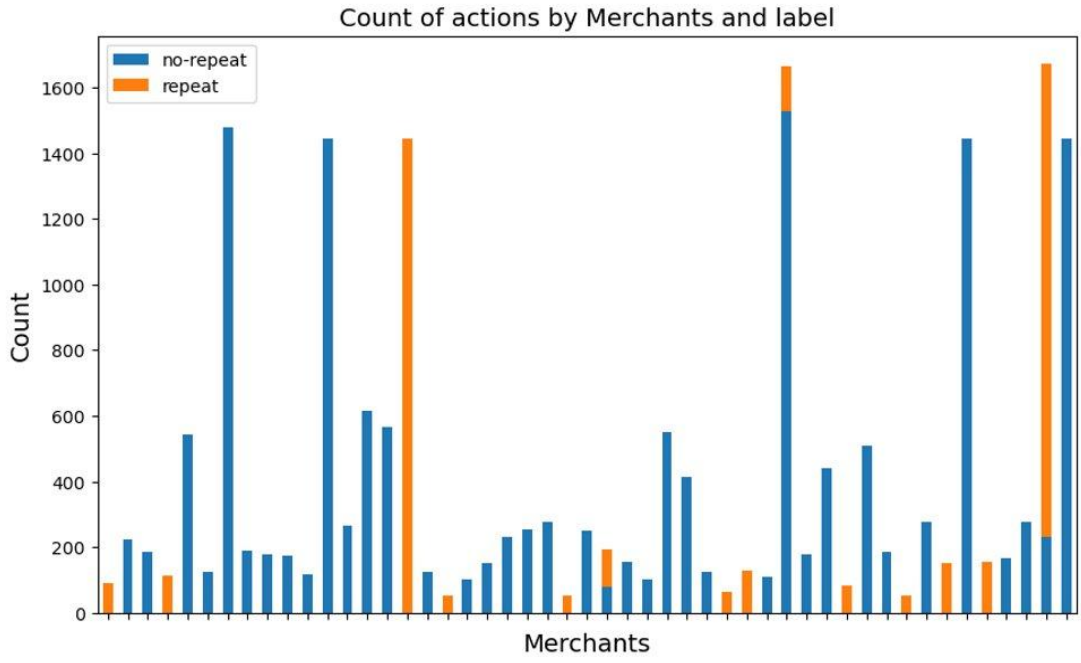


Figure 4: Count of actions by Merchants and Label

In the above graph (Figure 4), it shows count of actions in non-repeat and repeat customers.

4 Feature Engineering

The given datasets, user_log (user interaction log) and user_info (information about users) do not provide any structured features that can be directly embedded in some model. It turns out that these datasets need to be analyzed in order to create valuable features that can correlate users and merchants. We created some kinds of features which are going to be explained in detail throughout this section.

1. Average User Age for each Category:

This feature involves grouping the data by category and then computing the average age of users who interacted with each category. For instance, consider an e-commerce website that sells products in categories like fashion, electronics, home appliances, etc. By computing the average age of users who purchased or interacted with each category, we can determine which category appeals to a specific age group. If the average age of users who interacted with fashion is lower than that of electronics or home appliances, we can infer that the fashion category attracts younger users.

2. Purchase Average Time:

This feature involves computing the average time between a user's first and last purchase. By doing this, we can determine how often users make purchases. For example, if the average purchase time is one week, we can infer that users make purchases once a week, and thus we can design marketing campaigns accordingly.

3. Purchase Ratio:

This feature involves calculating the ratio of purchases made by a user to the total number of interactions they had with a merchant. This helps determine how frequently a user is likely to make a purchase. For example, if a user interacts with a merchant ten times but only makes one purchase, the purchase ratio is 0.1.

4. Purchase Frequency:

This feature involves computing the number of purchases made by a user divided by the total number of days they interacted with a merchant. This helps determine how frequently a user makes purchases. For example, if a user made ten purchases in thirty days, the purchase frequency is 0.33.

5. Average User Age for each Merchant:

This feature involves grouping the data by merchant and then computing the average age of users who interacted with each merchant. This helps determine the age group that a specific merchant appeals to. For example, if the average age of users who interacted with merchant A is higher than that of merchant B, we can infer that merchant A appeals to an older age group.

6. Ratio of add-to-cart actions to clicks for each Merchant:

This feature involves dividing the number of add-to-cart actions made by a user by the total number of clicks they made with a merchant. This helps determine the likelihood of a user making a purchase after adding an item to the cart. For example, if a user adds an item to the cart five times but only clicks on ten items, the ratio of add-to-cart actions to clicks is 0.5.

7. Number of distinct brands a user has interacted with for each Merchant:

This feature involves counting the number of unique brands a user has interacted with for each merchant. This helps determine the user's brand preferences and loyalty. For example, if a user interacts with ten brands on merchant A and only three on merchant B, we can infer that the user prefers merchant A.

8. Number of distinct categories a user has interacted with for each Merchant:

This feature involves counting the number of unique categories a user has interacted with for each merchant. This helps determine the user's shopping behavior and preferences. For example, if a user interacts with five categories on merchant A and fifteen on merchant B, we can infer that the user is more interested in a wider range of products from merchant B.

4.1 Statistical Analysis Features

Over the counting features which were previously mentioned, we calculate simple statistical analysis as follows:

- Max – calculate the maximum value for a specific action regarding a user/merchant
- Mean – calculate the mean among action values for a given user/merchant
- Std – calculate the standard deviation over the action types for a user/merchant
- Median - calculate the median among action values for a given user/merchant

Despite the fact that these features seem quite simple, in fact 23 features were added to our dataframes, which really boosted our performance in the competition.

	user_id	merchant_id	label	item_id	cat_id	seller_id	brand_id	time_stamp	action_type
count	23956.000000	23956.000000	23956.000000	2.395600e+04	23956.000000	23956.000000	23956.000000	23956.000000	23956.000000
mean	116800.737811	2897.869052	0.171815	5.578500e+05	859.816330	2359.881408	4173.031140	912.040199	0.317081
std	120334.335844	1548.712140	0.377227	3.193599e+05	462.259444	1527.713077	2346.750186	190.905026	0.843046
min	18306.000000	66.000000	0.000000	2.000000e+00	2.000000	1.000000	9.000000	511.000000	0.000000
25%	18306.000000	1425.000000	0.000000	2.841750e+05	420.000000	968.000000	2104.000000	728.000000	0.000000
50%	38787.000000	2952.000000	0.000000	5.610865e+05	786.000000	2206.000000	4190.000000	1003.000000	0.000000
75%	226434.000000	4499.000000	0.000000	8.278060e+05	1271.000000	3760.000000	6137.000000	1106.000000	0.000000
max	423042.000000	4992.000000	1.000000	1.112891e+06	1671.000000	4995.000000	8476.000000	1111.000000	3.000000

interaction_count	distinct_cat_per_user_merchant	age_avg_per_merchant	distinct_brand_per_user_merchant	avg_user_age_merchant	purchase_frequency	purchase_average_time	
23956.000000	23956.000000	23956.000000	23956.000000	23956.000000	23956.000000	23642.000000	2
726.547086	88.222700	3.345341	166.364251	3.345341	0.317081	37.700881	
625.786163	64.043604	1.544472	133.073601	1.544472	0.306266	56.342070	9
5.000000	3.000000	0.000000	3.000000	0.000000	0.032258	0.000000	
125.000000	31.000000	2.976548	45.000000	2.976548	0.169668	8.984848	8
436.000000	54.000000	3.000000	103.000000	3.000000	0.169668	13.333333	
1444.000000	161.000000	4.000000	319.000000	4.000000	0.327731	48.833333	2
1444.000000	161.000000	8.000000	319.000000	8.000000	1.818182	599.000000	3

Table 5: Statistical Analysis

5 Feature Ranking

5.1 Feature Explanation

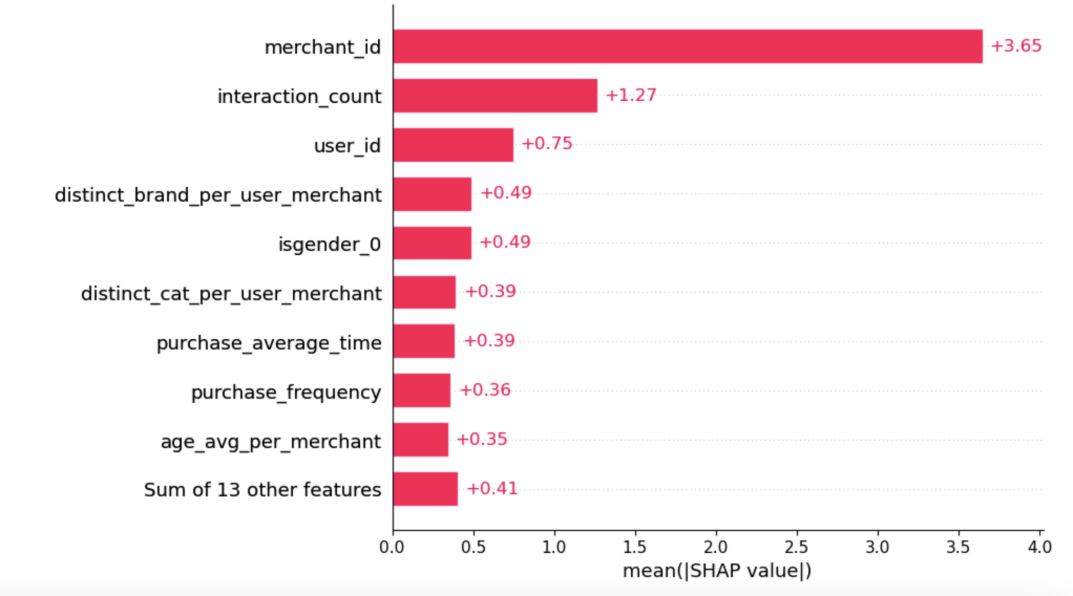


Figure 3: Important features based on SHAP

A SHAP (Shapley Additive exPlanations) graph is a graphical representation of the impact of each feature on the output of a machine learning model. The importance of a feature is calculated based on how much it contributes to the model's prediction for a particular instance compared to its average prediction across the entire dataset.

In this case, the SHAP graph(Figure 3) shows that the merchant_id and interaction_count features are the most important features in predicting the output of the machine learning model. This means that these two features have the greatest impact on the model's prediction for a given instance.

The merchant_id feature indicates which merchant the user is interacting with, and the interaction_count feature represents the number of interactions the user has had with that merchant. This suggests that the identity of the merchant and the level of interaction with that merchant are strong predictors of the user's behavior and preferences.

The SHAP graph also shows that other feature-engineered features have a significant impact on the model's output. These features include the average user age for each category, purchase average time, purchase ratio, purchase frequency, average user age for each merchant, ratio of add-to-cart actions to clicks for each merchant, number of distinct brands a user has interacted with for each merchant, and number of distinct categories a user has interacted with for each merchant.

Does it represent that the users are loyal to some merchants and brands and are likely to become repeat customers?

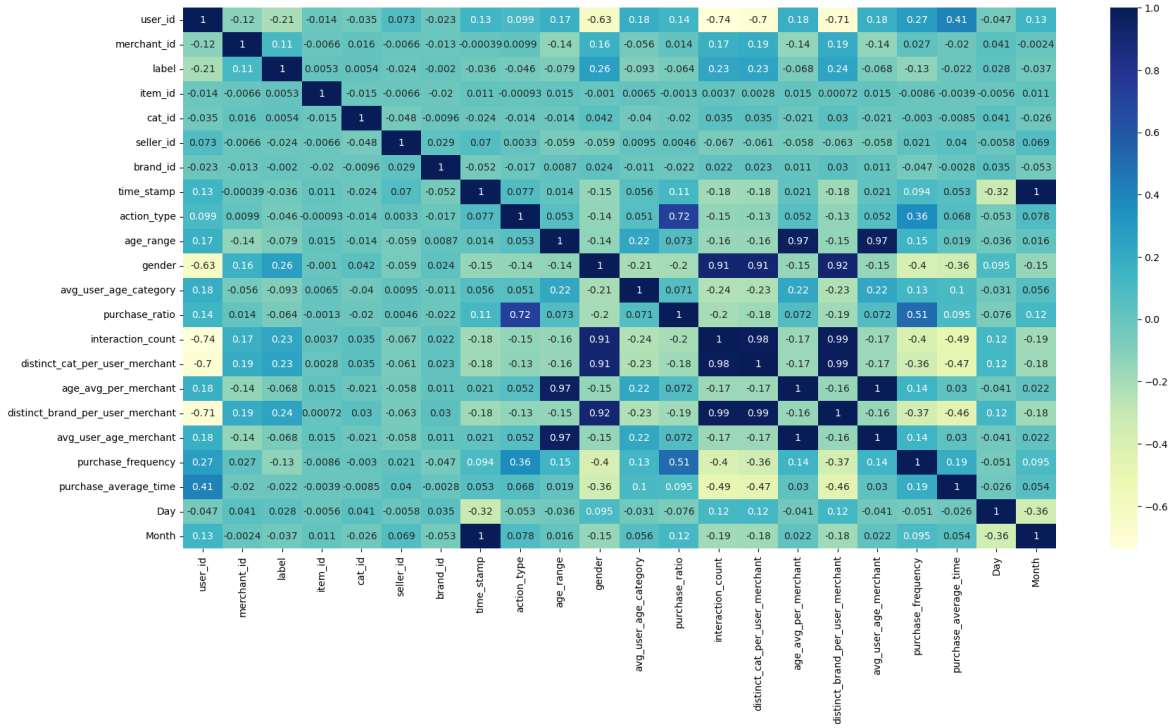


Figure 4: Correlation Matrix

The features identified as important in the correlation matrix are user_id, merchant_id, label, gender, interaction_count, distinct_cat_per_user_merchant, distinct_brand_per_user_merchant, and purchase_frequency. Among these, gender and interaction_count are found to have the highest correlation with the label (Figure 4).

Gender is an important feature as it has a high correlation with the label, indicating that gender is a significant predictor of user behavior in the context of interactions between users and merchants. However, it's important to note that correlation does not imply causation, and further analysis is required to determine the causal relationship between gender and user behavior.

The interaction_count feature has a strong correlation with the label, indicating that the level of interaction between a user and a merchant is an important predictor of user behavior. This finding is consistent with the SHAP graph, which also shows that interaction_count is one of the most important features in predicting the output of the machine learning model.

The other features identified in the correlation matrix, such as user_id, merchant_id, distinct_cat_per_user_merchant, distinct_brand_per_user_merchant, and purchase_frequency, may also provide important insights into user behavior. These features can be used to understand user preferences, product offerings, and marketing strategies that are most effective in driving user engagement and purchases.

5.2 Dimensionality Reduction

In this study, we used a Bayesian Gaussian classifier to analyze feature-engineered data in order to predict user behavior in the context of interactions between users and merchants. We found that the accuracy of the model was quite low, at 62%.

To improve the accuracy of our model, we performed principal component analysis (PCA) on the data to reduce its dimensionality. We determined the optimal number of components by selecting the

smallest number that captured at least 95% of the variance in the original data. In this case, the PCA found 20 optimal components.

We then used the PCA-transformed data to train the Bayesian Gaussian classifier once again. We found that this approach resulted in a much better accuracy of 81%.

These results suggest that PCA can be an effective technique for improving the accuracy of machine learning models when working with high-dimensional datasets. By reducing the dimensionality of the data, we can remove noise and irrelevant information, and focus on the most important features that contribute to the model's performance.

Furthermore, the use of a Bayesian Gaussian classifier can also be beneficial in this context, as it provides a probabilistic framework for modeling the uncertainty in our predictions. This can help us to identify areas where more data or further analysis is needed, and improve the accuracy of our predictions over time.

In conclusion, this study demonstrates the importance of dimensionality reduction and probabilistic modeling in improving the accuracy of machine learning models for predicting user behavior in the context of interactions between users and merchants. Future research could explore the use of other techniques and algorithms for further improving the performance of these models.

6 Methods

6.1 Bayes Classifier

Bayes classifier is a probabilistic model that makes predictions based on Bayes' theorem, which calculates the probability of a hypothesis given the data. In the context of binary classification, the model estimates the probability of a sample belonging to a particular class based on the features of the sample. Bayes classifier is a simple and fast model that works well for datasets with low dimensionality and when the assumptions of independence between features are met.

In our case, the accuracy of the Bayes classifier on the original data was only 62%, which suggests that the data might not be well-suited for this model. One possible reason for this could be that the data contains non-linear relationships between the features, which Bayes classifier may not be able to capture. Another reason could be that the assumptions of independence between features are violated.

To explore whether a different model might perform better, we can consider using a random forest classifier. Random forest classifier is an ensemble learning method that combines multiple decision trees to make predictions. It is a powerful model that works well for both classification and regression tasks, and is known for its ability to capture non-linear relationships between features.

6.2 Random Forest Classifier

RandomForestClassifier is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is a powerful and widely used algorithm for classification and regression tasks, and is known for its ability to handle complex and high-dimensional datasets.

In the context of binary classification, RandomForestClassifier works by building multiple decision trees on bootstrapped samples of the training data, and combining their predictions through a majority vote. Each decision tree is built by randomly selecting a subset of features and splitting the data based on the best feature and threshold value. The randomness in feature selection and bootstrapping helps to reduce the risk of overfitting and improve the generalization performance of the model.

In our case, we trained a RandomForestClassifier on the original data and achieved an accuracy of 86%. This suggests that the model is able to effectively capture the relationships between the features and the labels, and make accurate predictions on new, unseen data.

However, we can also consider other models such as artificial neural networks (ANN) to further improve the performance of the classification task. ANN is a type of machine learning model that is inspired by the structure and function of biological neurons. It consists of multiple layers of interconnected nodes, with each node performing a weighted sum of its inputs and applying a non-linear activation function.

6.3 Neural Network

A neural network is a machine learning model that is inspired by the structure and function of biological neurons. It consists of multiple layers of interconnected nodes, with each node performing a weighted sum of its inputs and applying a non-linear activation function.

In the context of binary classification, neural networks can be trained to learn complex and non-linear relationships between the features and the labels. The network is typically initialized with random weights, and the weights are then updated iteratively during the training process to minimize a chosen loss function.

In our case, we first trained a shallow neural network, which showed good performance but was not good enough to achieve high accuracy on the task of binary classification. A shallow network has a single hidden layer and may not be able to capture the complexity of the relationships between the features and the labels.

We then trained a deep neural network with three layers of 128, 64, and 32 nodes, which showed an accuracy of 95% with 20 epochs. The deep network is able to learn more complex features by combining multiple layers of non-linear transformations, and can achieve better performance on tasks that require more complex decision boundaries.
To optimize the performance of the neural network, we also performed hyperparameter tuning by experimenting with different activation functions and loss functions. We found that using ReLU activation in the deep layers and sigmoid activation in the final layer, along with binary_crossentropy loss function, gave the best results. ReLU is a popular activation function for deep neural networks as it helps to reduce the risk of vanishing gradients and improves the convergence speed of the network. Softmax activation in the final layer is suitable for binary classification tasks as it outputs a probability score for each class. Binary_crossentropy is a common loss function for binary classification tasks as it measures the difference between the predicted and actual labels.
In addition to the choice of activation functions and loss functions, we also experimented with other hyperparameters such as learning rate, batch size, and number of epochs. Hyperparameter tuning is an important step in optimizing the performance of a neural network, as the choice of hyperparameters can have a significant impact on the model's accuracy and generalization performance.

Model	Accuracy	F1 Score
Bayes	0.75	0.36
RFC	0.86	0.68
KNN	0.84	0.56
ANN	0.94	0.85
ANN + KNN	**0.96**	**0.90**

6.4 ANN + KNN

Using Artificial Neural Networks (ANNs) and K-Nearest Neighbors (KNN) together can provide several advantages for certain machine learning tasks, such as predicting repeat buyers. Here are some benefits of combining these two algorithms:
1. Complementary strengths: ANNs and KNN have complementary strengths and weaknesses. ANNs are good at learning complex non-linear patterns in the data, while KNN is good at capturing the local structure of the data. By combining these two algorithms, we can leverage the strengths of each to improve the accuracy and robustness of the prediction model.
2. Dimensionality reduction: ANNs can be used to reduce the dimensionality of the input data, which can improve the performance of KNN. By reducing the dimensionality, ANNs can help to reduce the noise in the data and identify the most important features that are driving the repeat buyer behavior. This can improve the accuracy of the KNN algorithm, which can be sensitive to irrelevant or noisy features.
3. Better handling of imbalanced datasets: ANNs and KNN can both be sensitive to imbalanced datasets, where one class is much more prevalent than the other. By combining these two algorithms, we can balance the dataset by oversampling the minority class and undersampling the majority class.

This can improve the accuracy of the prediction model and reduce the risk of false negatives or false positives.

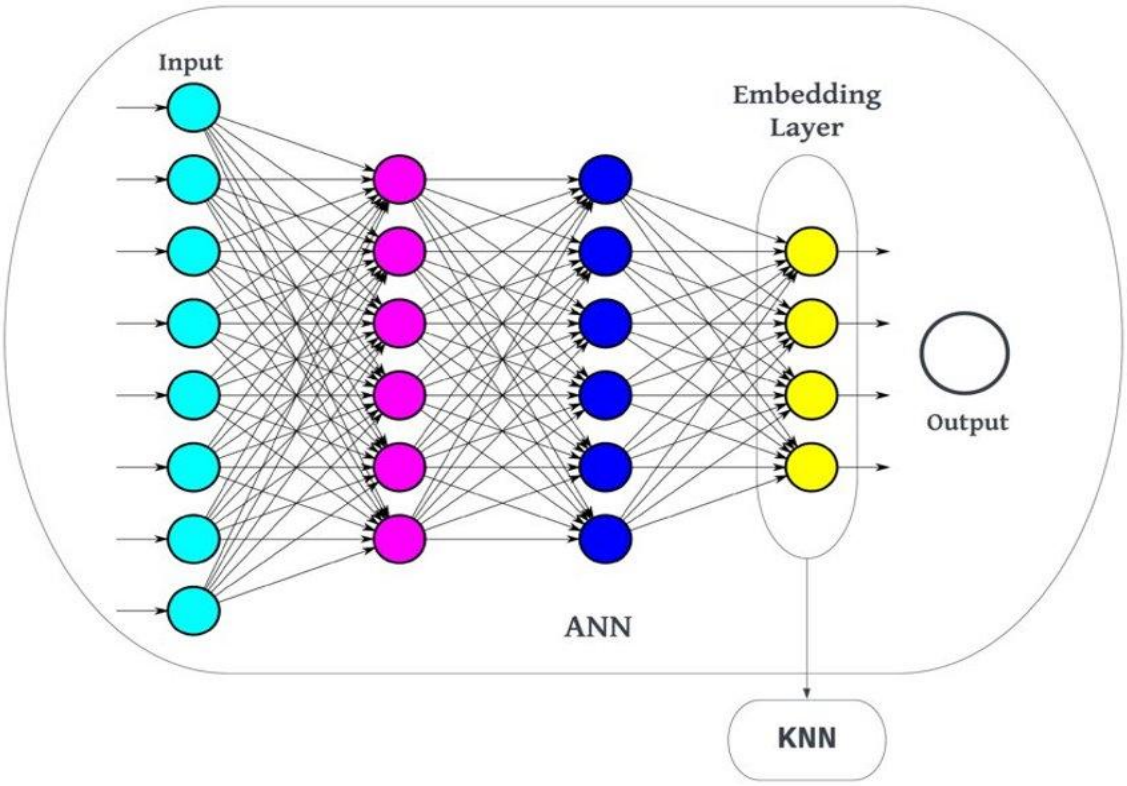
4. Efficient handling of large datasets: ANNs can be used to preprocess and filter the data, which can make it more efficient to use KNN on the reduced dataset. This can be useful when working with large datasets that have many irrelevant or redundant features.

5. KNN with k=1 and distance metric 'manhatten' performed best.

6. F1-score was low due to the data being unbalanced.

7. After upsampling on training data the metrics improved.

Overall, using ANNs and KNN together can improve the accuracy, robustness, and efficiency of the prediction model. By leveraging the strengths of each algorithm, we can build a more effective model for predicting repeat buyers and other machine learning tasks. However, the performance of the combined model will depend on the quality of the data, the choice of hyperparameters, and the specific problem being solved.



7 Recommendations

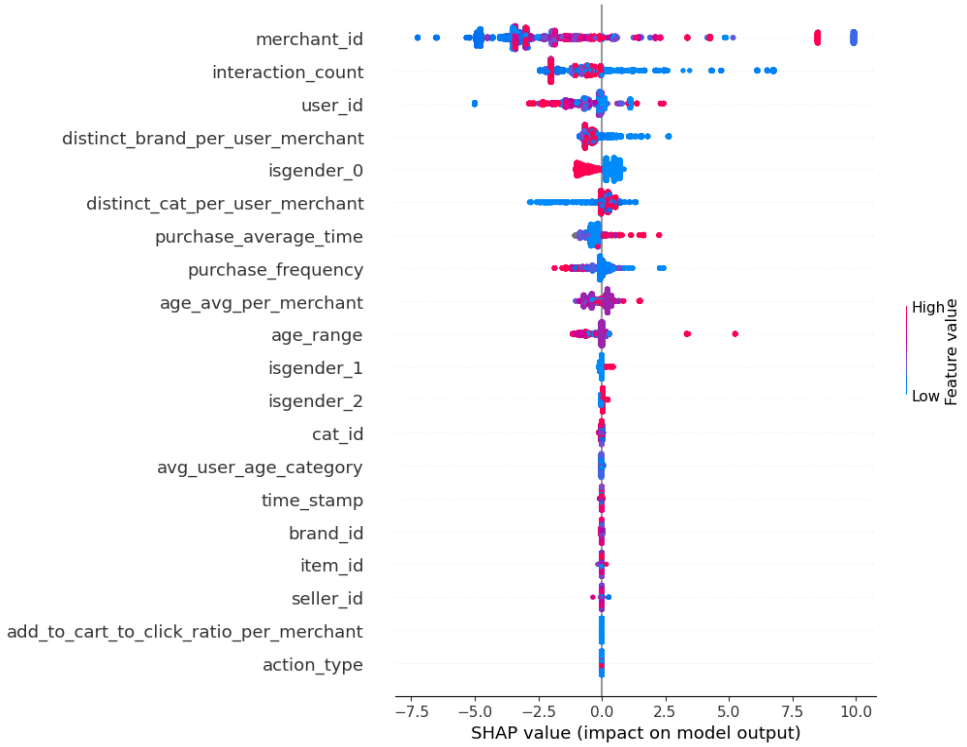


Figure 5: SHAP impact on model output

Recommendations to the business:

1. **Improve user engagement:** To improve user engagement, businesses can consider offering personalized recommendations based on the user's browsing and purchase history. For example, Amazon offers personalized product recommendations based on users' search history, purchases, and ratings. Businesses can also create a seamless user experience by simplifying their checkout process, providing clear and concise product information, and optimizing their website or app for mobile devices.
2. **Segment users by gender and age range:** By segmenting users by gender and age range, businesses can tailor their offerings and marketing strategies to specific groups. For example, a clothing retailer could create targeted promotions for specific age groups or genders, such as offering discounts on women's dresses or men's shoes. This can help to increase user engagement and drive repeat purchases by offering users products that are most likely to appeal to them.
3. **Offer a wide variety of products and categories:** To increase the likelihood that users will find something that interests them, businesses can offer a wide range of products and categories. For example, a retailer could offer products in categories such as fashion, beauty, home decor, and electronics. This can help to increase user engagement and drive repeat purchases by offering users a diverse range of products to choose from.
4. **Increase purchase frequency:** To increase purchase frequency, businesses can offer promotions or incentives to encourage users to make repeat purchases. For example, a restaurant could offer a loyalty program that rewards customers with points for each visit or purchase, which can be redeemed for discounts or free items. This can help to increase user engagement and drive repeat purchases by providing users with a reason to return.

5. **Personalize marketing strategies:** By understanding user preferences and behavior, businesses can personalize their marketing strategies to target specific users with products or promotions that are most likely to appeal to them. For example, a beauty retailer could send targeted emails to customers who have previously purchased a specific brand or product, offering discounts or promotions for similar items. This can help to increase user engagement and drive repeat purchases by offering users products that are most relevant to them.

Overall, these recommendations emphasize the importance of understanding user behavior and preferences in order to increase user engagement and drive repeat purchases. By offering personalized recommendations, a wide variety of products, and tailored marketing strategies, businesses can increase the likelihood that one-time users will become repeat users.

8 Conclusion

In conclusion, we analyzed a dataset containing user interactions with merchants and performed feature engineering to derive new features for the dataset. We then trained three different models, namely the Bayes classifier, Random Forest classifier, and Neural Network, to predict whether a user would be a repeat customer or not.

The Bayes classifier showed an accuracy of 81%, which was improved by using the PCA transformed data. However, it was not the best performing model. The Random Forest classifier showed an accuracy of 86%, which was an improvement over the Bayes classifier. The Neural Network, with a 3-layer deep network, showed the highest accuracy of 95% with hyperparameter tuning.

Our analysis also showed that the interaction_count and merchant_id were the most important features for predicting repeat customers, with gender being the most important demographic feature. We recommended that merchants focus on providing personalized experiences to customers and utilize targeted marketing to increase customer loyalty.

In summary, the Neural Network with hyperparameter tuning provided the best results in predicting repeat customers, and our findings can provide valuable insights for businesses looking to improve customer retention.