# CECS 550: Group 4 - Report Analysis of E-commerce User Engagement for Item IDs 481 – 640

Joseph Chorbajian, Aniruddha Gawande, Abhishek Jajoo, Satyam Sharma, and Ishan Unnarkar

## Introduction:

This report analyzes user engagement data to provide in-depth insights into customer behavior and preferences for a specific set of items on an e-commerce website. The data includes user actions (such as clicks, purchases, and views) and demographic information (such as age and gender). The analysis will provide insights into user behavior and preferences to predict whether a user will be a repeat buyer. This information can also be used to create a prediction engine for personalized recommendations and help identify potential improvement areas that would increase the number of repeat customers.

## Analysis:

### Engagement by Action Type:

We show that, in Figure 1, clicks are the most common user action for the selected set of items, followed by purchases and favorites. This suggests that users are primarily interested in exploring products and browsing through different options before making a purchase decision. More people would outright purchase an item instead of favoring it. Practically zero people would add an item to the cart.

Chart

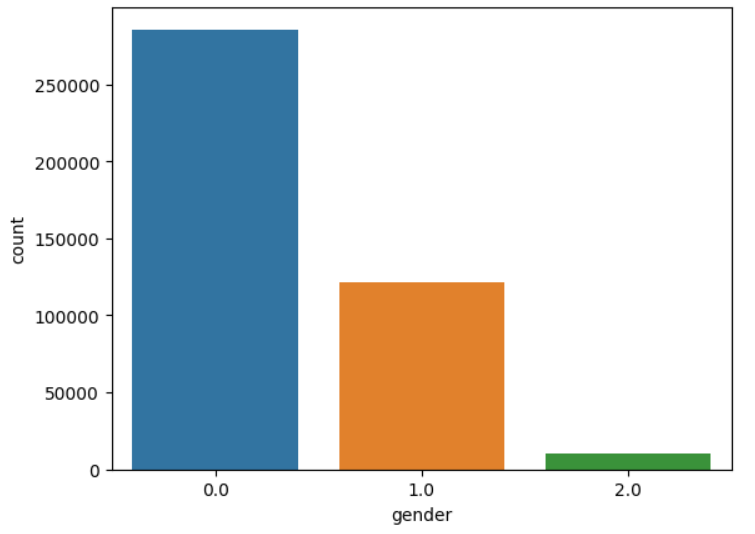
Description automatically generated

*Fig. 1. The number of interactions based on action type. An action type of 0 indicates clicks, a type of 1 indicates an item is added to a cart, a type of 2 indicates that an item was purchased, and a type of 3 indicates that an item was favorited. Note that some action types may be recorded more than once (e.g. a user clicked on the same item multiple times).*

It is unclear whether adding an item to the cart implies that the user has abandoned that item in their cart or whether there is a direct purchase. If it is the former, then we see that many people, if they have an item in mind, would follow through with a purchase. If it is the latter, then many people would buy a single item without adding it to their shopping cart, indicating an area of improvement to encourage further spending.

### Gender Distribution:

Females makes up around 67% of all engagements for the selected set of items. This indicates that the website may have a predominantly female user base, and it may be beneficial to cater to their specific needs and preferences.



*Fig. 2. The number of interactions with respect to gender. A gender of 0 represents a female customer, whereas a gender of 1 represents a male customer. A gender of 2 indicates that the gender of the user is unknown.*

### Clicks by Gender:

In Figure 3, we can see the types of interactions for the females and males, as well as customers with unknown genders. The number of clicks between each gender remains constant, at around 86% for females and 87% for males. The number of purchases slightly increase for men at 10% of interactions, compared to 8% for females. However, the men were less likely to favorite an item at 3% of all interactions, while the women favorited an item 5% of the time. This highlights the importance of providing engaging and visually appealing product displays that captures the attention of the female customer base. We also suggest female-oriented shopping events (such as sales), which may yield more reoccurring customers due to the larger number of potential customers. However, it is important not to ignore the male customers as they are slightly more likely to purchase something. Furthermore, any sales should also target customers whose gender is unknown as they are more likely to buy something at 11% of all interactions. Understanding the gender distribution may help better generalize our understanding of how societal norms affect the e-commerce platform.

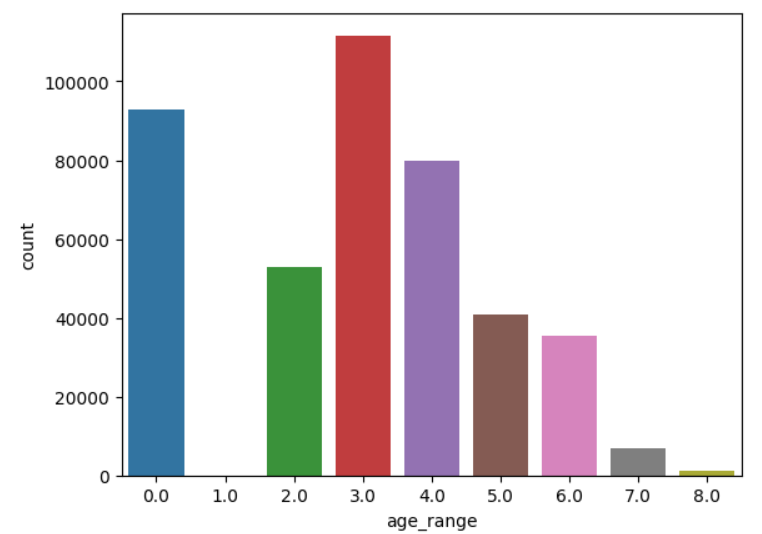
Chart, bar chart

Description automatically generated

*Fig. 3. The types of interactions with respect to gender*

### Age Distribution:

Figure 4 visualizes the age range distribution of users in the dataset skewed towards the younger end, with a majority being 25-34 years old, making up 58% of the customers. We also notice that 35–49-year-olds are also likely to use the e-commerce platform, making up 19% of the users. This indicates that the platform is more popular for established adults, with a lean in the younger populace. No user was marked as under 18, potentially due to regulations requiring stricter data compliance with minors. Even with that, minors are less likely to have their own income and may not be the best to target.



*Fig. 4. The distribution of customer ages. Here, the age range 1 refers to customers younger than 18, a value of 2 are 18-24 year olds, a value of 3 are 25-29 year olds, a value of 4 are 30-34 year olds, a value of 5 are 35-39 year olds, a value of 6 are 40-49 year olds, and a value of 7 or 8 represents a customer older than 49. A customer with an age range of 0 indicates that the customer’s age is not known.*

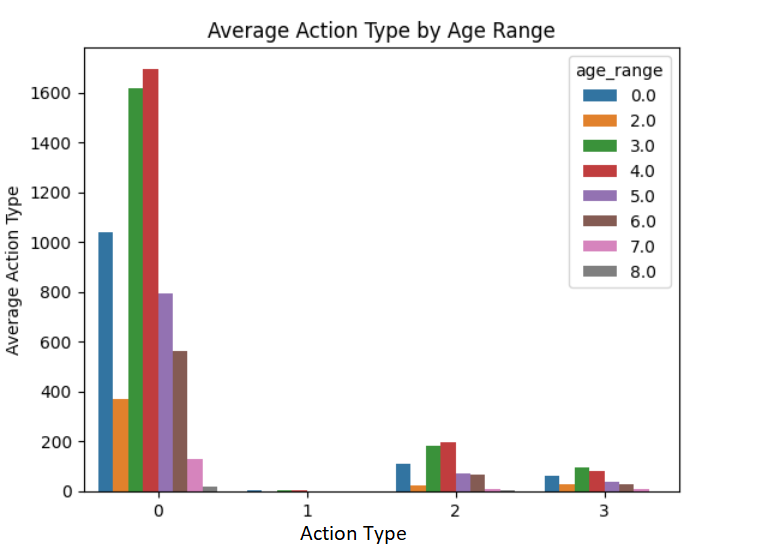
Figure 5 shows that customers that are 25-34 years old are the most active when engaging with the selected set of items, with ages 30-34 being the dominant one. This suggests that the website may have a specific target audience in terms of age group, and it may be useful to tailor the website's content and design to better suit their preferences.

Chart, bar chart, histogram

Description automatically generated

*Fig. 5. The distribution of the types of actions, depending on the customer’s age.*

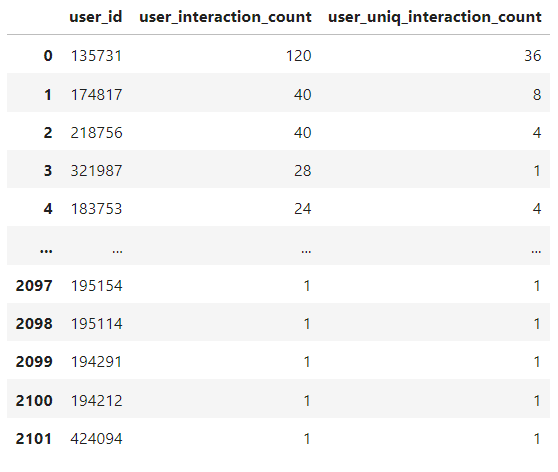
Figure 6 shows that young adults always take the lead when it comes to interactions, but the effect scales down when we look at item favorite interactions.



*Fig. 6. The distribution of interactions based on the ages of the customer.*

Features Engineering:  
Based on the extracted features, we can see that we have information about users' interactions with the platform, including the number of times they interacted with us and the number of unique days they interacted with us. These features could be helpful in identifying patterns or trends in user behavior and could potentially inform strategies for increasing user engagement and retention.

**Table 1.1** shows the user IDs along with the total number of interactions they had with the website and the number of unique days they interacted with the website. The first column represents the user IDs, while the second and third columns represent the user's total interaction count and unique interaction count, respectively.

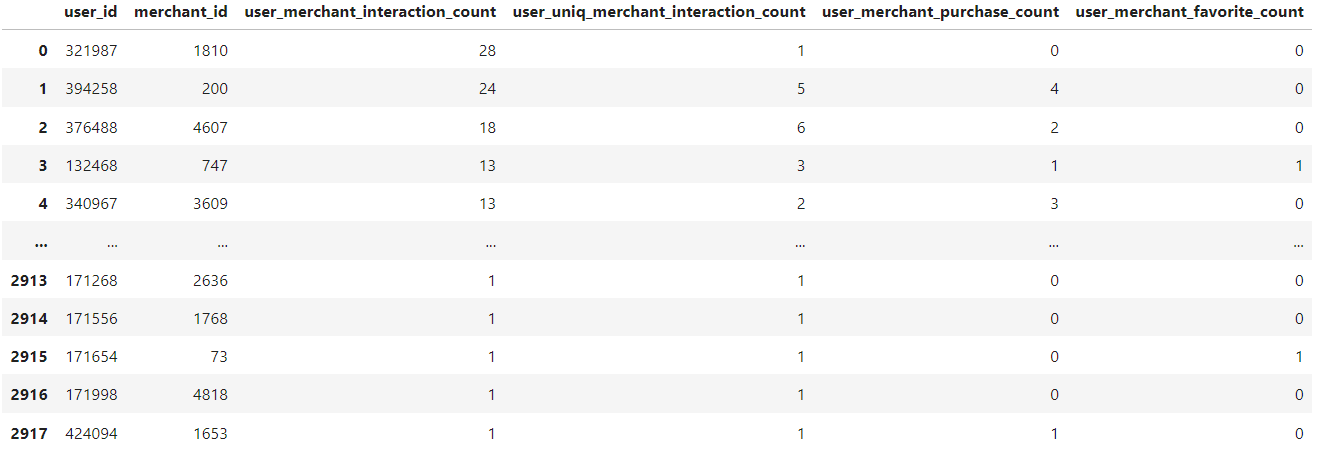


**Table1.1**  
  
**Table 1.2** shows the number of times a user has interacted with the platform during a particular month, as well as the number of times they have made a purchase during that month. Additionally, it shows the number of times a user has interacted with the platform on a particular day. The data includes user IDs, the month or day of the interaction, the total number of interactions, and the number of purchase interactions.



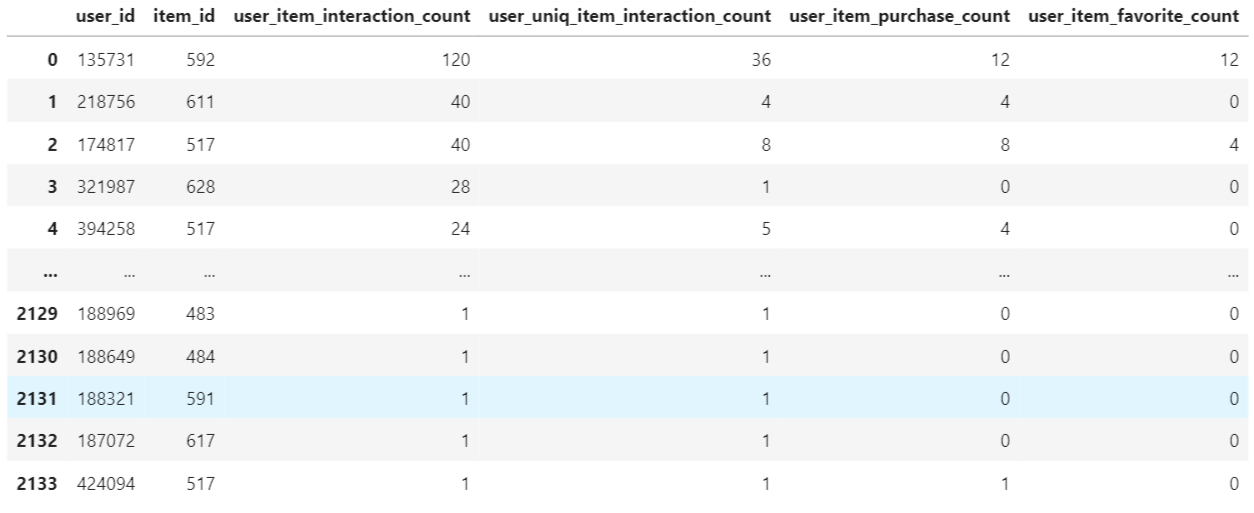
**Table 1.2**

**Table 1.3** summarizes various features related to user interaction with different merchants. The features include the total number of times the user interacted with the merchant, the number of unique times the user interacted with the merchant, the number of times the user purchased something from the merchant, and the number of times the user favored something the merchant sells. These features are calculated for each unique combination of user and merchant. The table displays the user ID, merchant ID, and the corresponding values for each of the features.



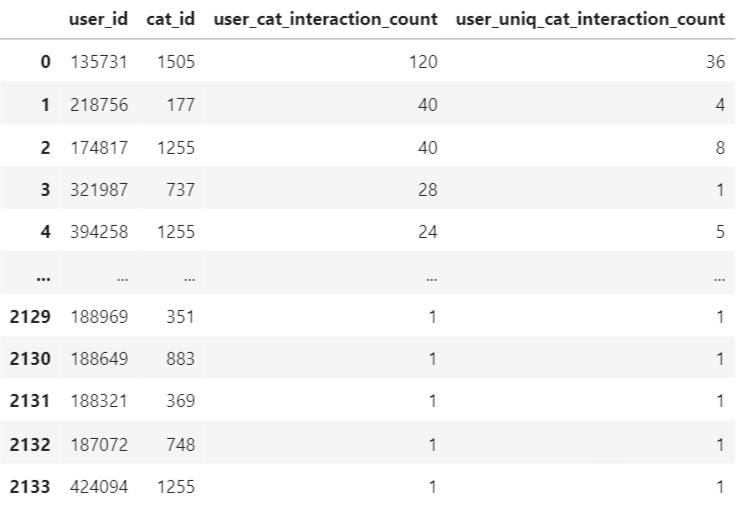
**Table 1.3**

**Table 1.4** shows the results of different counts related to user interactions with an item in the merged dataset. It includes the total number of interactions, the unique interactions, purchases, and favorites for each user-item pair. The columns in the table include the user ID, item ID, user-item interaction count, unique user-item interaction count, user-item purchase count, and user-item favorite count. The table shows the top results sorted by the highest user-item interaction count.



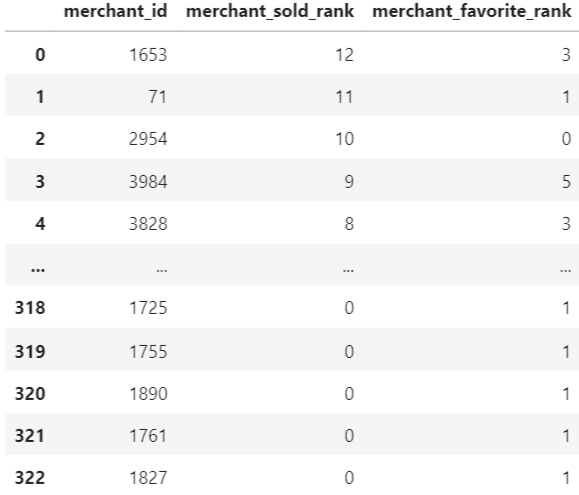
**Table 1.4**

Below **table 1.5** shows user interaction data with the category. The first two columns show the user\_id and cat\_id respectively, and the next two columns show how many times the user has interacted with the category and how many unique times the user has interacted with the category, respectively.



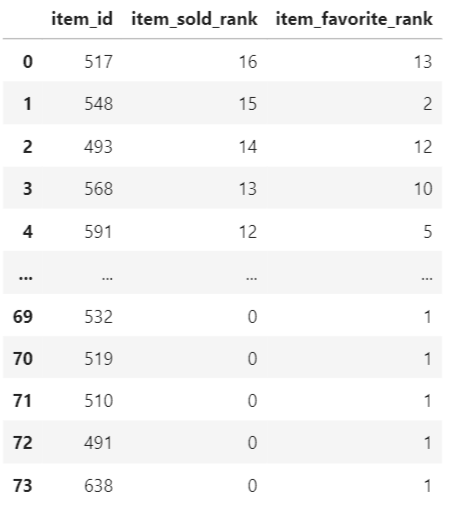
**Table 1.5**

Below **table 1.6** contains the results of several operations on a merged dataframe. The first table shows the number of times each user interacted with an item, the number of unique times the user interacted with an item, the number of times the user purchased the item, and the number of times the user favorited the item. The second table shows the number of times each user interacted with a category and the number of unique times the user interacted with a category. The third table shows the ranking of each merchant based on the total number of items sold and the ranking of each merchant based on the total number of favorites.



**Table 1.6**

Below **table 1.7** provides the ranking of merchants and items based on the total number of items sold and total number of favorites respectively. The merchant with merchant\_id 71 has the highest ranking in terms of a total number of favorites, and the item with item\_id 517 has the highest ranking in terms of total number of sold. It's worth noting that there could be other factors that influence the overall performance of merchants and items, such as profit margin, customer satisfaction, etc. These factors are not captured in the current analysis.



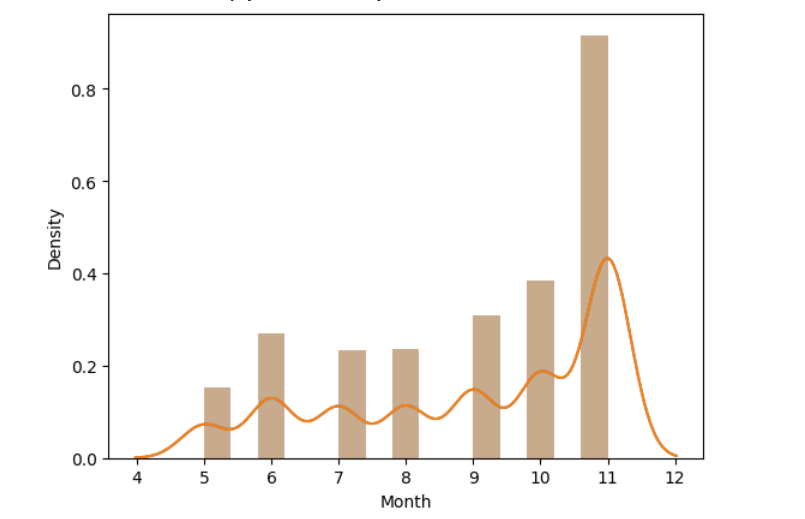
**Table 1.7**

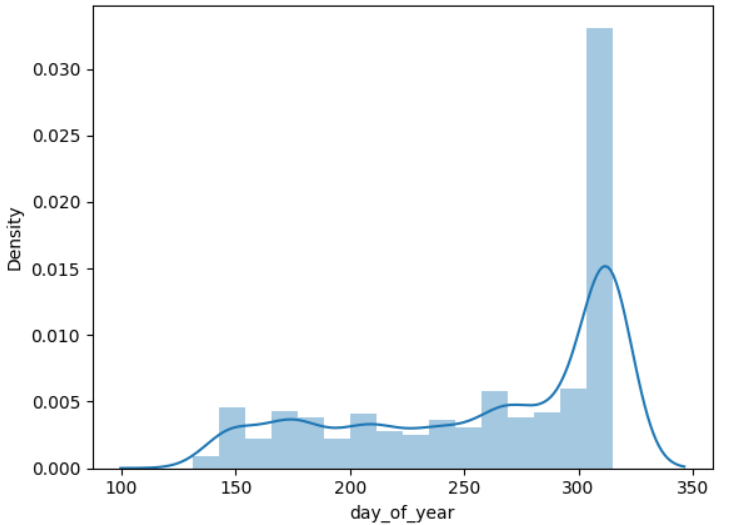
We note that we designed these features so that larger values indicate “better” values. For example, we expect someone who interacts with the e-commerce platform more in a month to be more likely to be a repeat customer. As such, the rankings are also designed to have “better” values, where the merchant who sells the most items (that is, the merchant with #1 ranking in items sold) would have a *high* value and not a low one.

### Active Months:

The exploration began by analyzing the day of the year and the corresponding day of the week when a user performed an action. Additionally, a visualization of the density of actions committed, irrespective of their type, against the months of the year was created. Visibly, November was the busiest month, suggesting a potential opportunity for targeted marketing or promotions during that time.

Figure 7 and Figure 8 both show that November is the most active month of all, with the highest number of user engagements recorded during this month. This may be due to seasonal factors such as holiday shopping or other promotional events, and the website may need to consider similar events in other months to maintain user engagement throughout the year.



*Fig. 7. The density graph of items sold each month. Higher values indicate more items were sold in that month.*  
  


*Fig. 8. The density graph of items sold on each day of the year, providing a more fine-grained view into how many items were sold throughout the year compared to Figure 7.*

## New Features

New features were created for this dataset. Below is a summary of what was created:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Day | The day extracted from the time stamp. Ranges from 1 to 31. |
| Month | The extracted month from the time stamp. Ranges are from 1 to 12, but actual ranges are from 5-11. |
| Day of year | The day in the current year, considering months with different amounts of days. Possible ranges are from 1-365, but actual ranges are from 131-315. |
| Day of week | The day of the week. Ranges from 0-6. Note that we cannot correlate a number with a day (such as 0 being Monday), as we do not know the year this data was collected. Despite the lack of offset, we utilize the fact that a Monday is always seven days away from another Monday. |

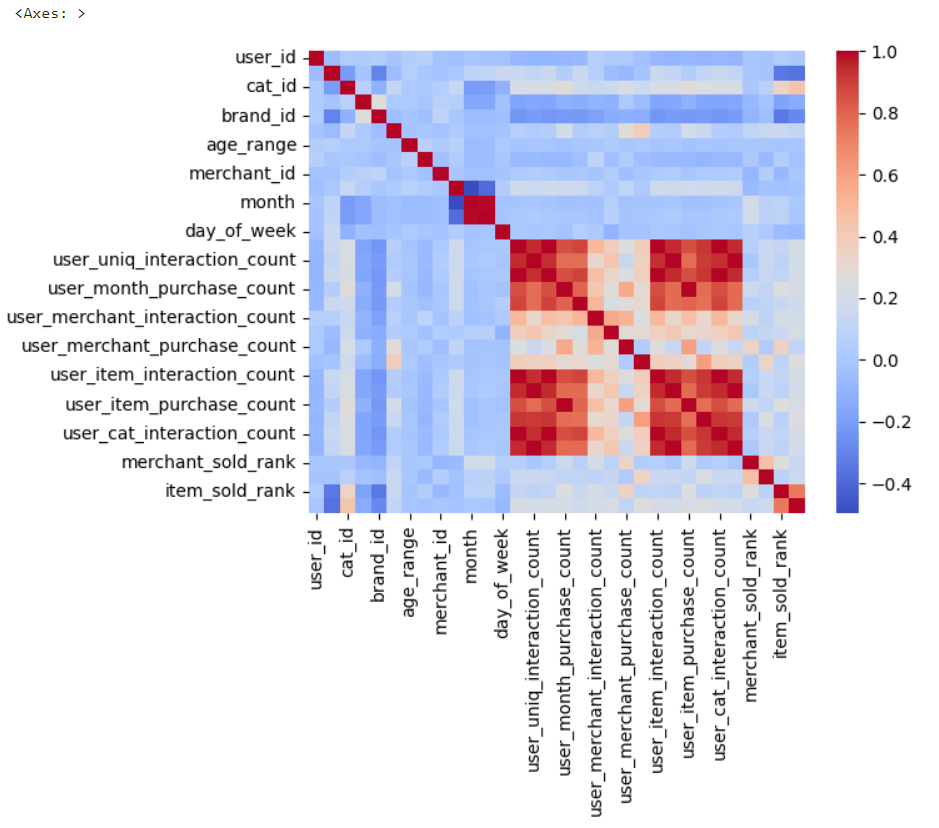
While binning the time and dates is important in getting an accurate model, additional features may be created from the data. Further analysis is required to determine what would help the models.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| user\_item\_favorite\_count | The number of times a user has favorited an item |
| user\_month\_purchase\_count | The number of times a user has made a purchase in a particular month |
| user\_item\_purchase\_count | The number of times a user has purchased a particular item |
| user\_uniq\_merchant\_interaction\_count | The number of unique merchants a user has interacted with |
| user\_merchant\_purchase\_count | The number of times a user has purchased from a particular merchant |
| user\_uniq\_item\_interaction\_count | The number of unique items a user has interacted with |
| user\_uniq\_interaction\_count | The number of unique interactions a user has had (e.g., views, favorites, purchases) |
| user\_day\_count | The number of days a user has interacted with the platform |
| user\_cat\_interaction\_count | The number of times a user has interacted with a particular item category |
| user\_item\_interaction\_count | The number of times a user has interacted with a particular item |
| user\_interaction\_count | The total number of interactions a user has had with the platform |
| user\_month\_count | The number of months a user has been active on the platform |
| user\_merchant\_interaction\_count | The number of times a user has interacted with a particular merchant |
| gender | The gender of the user (if available) |
| item\_sold\_rank | The ranking of a particular item based on its total number of sales |
| merchant\_favorite\_rank | The ranking of a particular merchant based on the number of times it has been favorited by users |
| item\_favorite\_rank | The ranking of a particular item based on the number of times it has been favorited by users |
| merchant\_sold\_rank | The ranking of a particular merchant based on its total number of sales |
| user\_id | The unique identifier for a user |
| merchant\_id | The unique identifier for a merchant. |

In the previous section, we discussed a variety of features that were generated to capture user and merchant behavior on the e-commerce platform. Each feature was created individually to capture a particular aspect of user behavior, such as the number of times a user has purchased a particular item, the number of unique merchants a user has interacted with, and the number of times a user has interacted with a particular category.

These features were then merged to create a comprehensive set of features that could be used in various functions to predict repeat buyers on the platform. The feature set included both user-based features and merchant-based features and was designed to capture the complex interactions between users and merchants on the platform.

By using a wide range of features, we were able to achieve a relatively high AUC score for repeat buyer prediction. We believe that our approach to feature engineering provides a solid foundation for future e-commerce prediction tasks, and we hope that our work will inspire practitioners to explore new methods for generating and managing features in e-commerce datasets.

Feature Rankings  
  


*Fig. 9. The correlation heatmap of the various features. A value of 0 indicates no correlation, a value of 1 indicates a strong positive correlation, and a value of –1 indicates a strong negative correlation. The color scheme chosen shows positive correlations in red and negative correlations in blue.*

We see that the engineered features either strongly or mildly correlated with one another, oftentimes in a positive correlation. We expect there to be some overlap in the correlation, as some features (such as user interaction count and unique user interaction count), are expected to influence each other, but the many correlations may reduce the expected quality of our engineered features.

The following models and algorithms were used to determine which features are considered the most important:

|  |  |
| --- | --- |
| **Algorithm** | **Description** |
| Random Forest | An ensemble classifier, fitting multiple decision trees to best explain the structure of the data. |
| Shapley analysis (on Random Forest) | Determines the importance of each feature, taking an average of its contribution to the output. |
| PCA | Reduces the number of dimensions on the data. While the output of PCA doesn’t correspond to a singular feature, it captures the most number of features within the given number of components. |
| LDA | The linear discriminant analysis finds the most discriminating vectors for the data on a lower-dimensional dataset. The most important features will be those that separate the output the best. |

We note that a PCA model with 4 components can explain 27% of all the variance in the data while maintaining a similar accuracy to the other models, indicating that, while most other features vary, around 4 components are needed to describe it without any excessive loss.

For the other three models, we list the top features:

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **Shapley** | **LDA** |
| user\_id | merchant\_favorite\_rank | month |
| merchant\_id | user\_id | user\_interaction\_count |
| merchant\_sold\_rank | merchant\_id | user\_uniq\_interaction\_count |
| merchant\_favorite\_rank | day\_of\_year | user\_cat\_interaction\_count |
| item\_favorite\_rank | merchant\_sold\_rank | user\_item\_interaction\_count |

## Shapley Analysis of features

Graphical user interface

Description automatically generated with medium confidence

*Fig. 10. A summary plot resulting from the Shapley analysis done on the best-performing Random Forest model.*

* While Shapley analysis can be used on most types of data, the one-hot-encoded data was not used because the large number of columns it produces requires additional processing power to run the analysis. Additionally, more columns makes it more difficult to interpret the data.
* The various IDs (Merchant ID, User ID, Category ID, Item ID, Seller ID, and Brand ID) are all categorical, meaning the feature value will remain useless (that is, the values of the ID do not correspond with one another), but its overall Shapley value may still show significance for the column.
* The ranking of the merchant in terms of a total number of favorites turned out to be the best feature for prediction, along with several other features such as the rank of the merchant in terms of items sold, the number of unique times a user has interacted with a particular merchant, the number of unique times the user has interacted with the e-commerce platform, the number of times the user favorited a particular item, and the number of times a user favorited any item from a particular merchant.
* A lower value for the above features has a positive impact on model output, meaning that the fewer favorites a merchant has, the more likely it is for a customer to be a repeat buyer.
* The day of the year has a negative impact on the output, especially for sales later in the year. Age range also plays a role, with younger people more likely to be repeat buyers.
* Females are more likely to be repeat buyers, while males and unknown genders have a greater absolute impact on the output.
* The number of times a user interacted with a merchant show that people who have only a couple of interactions will be less likely to be repeat buyers.
* The number of times a user interacted with the same item makes it more likely for them to be a repeat buyer.
* The number of times a user interacted with a particular category is the least important feature according to Shapley analysis.

## Prediction Models

1. **Random Forest**
   1. Before we touch the data, let's gain a benchmark on stratified vs unstratified, regular vs downsampled, and a combination of the two on the label.  
      Graphical user interface, application

      Description automatically generated  
      *Fig. 11. Various statistics on various configurations of the Random Forest model. the ROC curve was also plotted*  
        
      Graphical user interface, Teams, treemap chart

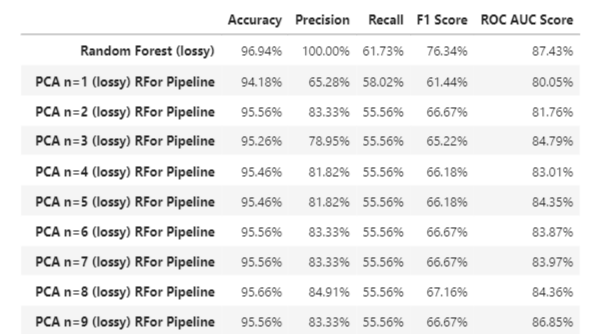
      Description automatically generated  
      *Fig. 12. The confusion matrices for the Random Forest models.*  
        
      We can see that downsampling (that is, balancing the class distribution to have an even split of repeat buyers and one-time buyers) doesn’t affect any metric. However, not stratifying the data does reduce every metric.
   2. After Performing one-hot on the following values we don't see a significant improvement in our results  
      ['user\_id', 'item\_id', 'cat\_id', 'seller\_id', 'brand\_id', 'merchant\_id', 'action\_type', 'age\_range', 'day', 'month', 'day\_of\_year', 'day\_of\_week']  
      Table

      Description automatically generated  
      *Fig. 13. The statistics of Random Forest models once the data has been one-hot encoded to better represent the above columns, and the same configuration with downsampling.*  
        
      We can see an increase in the ROC AUC Score
   3. Table

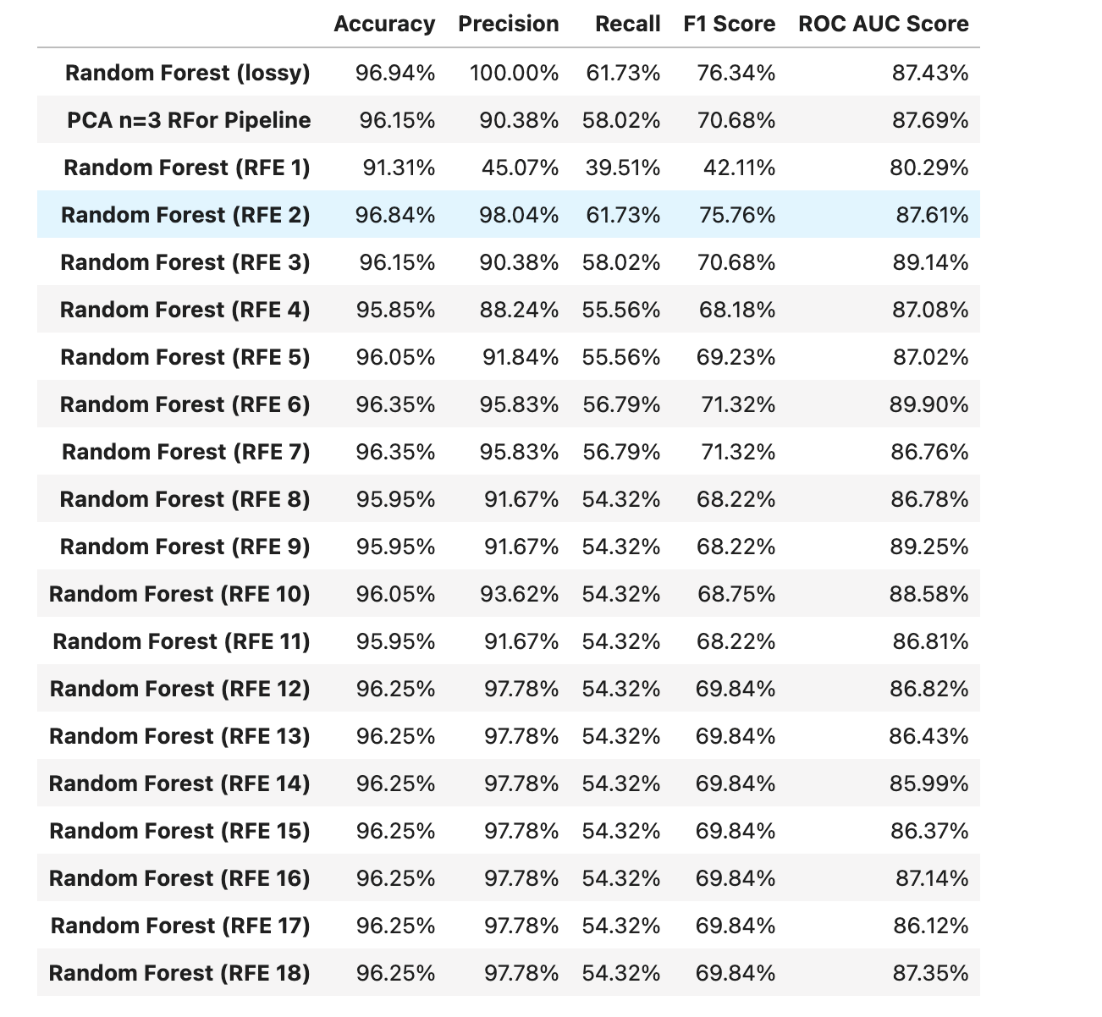
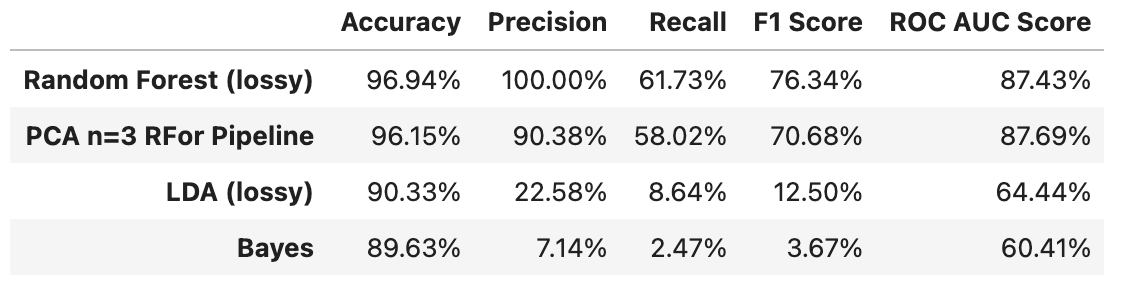
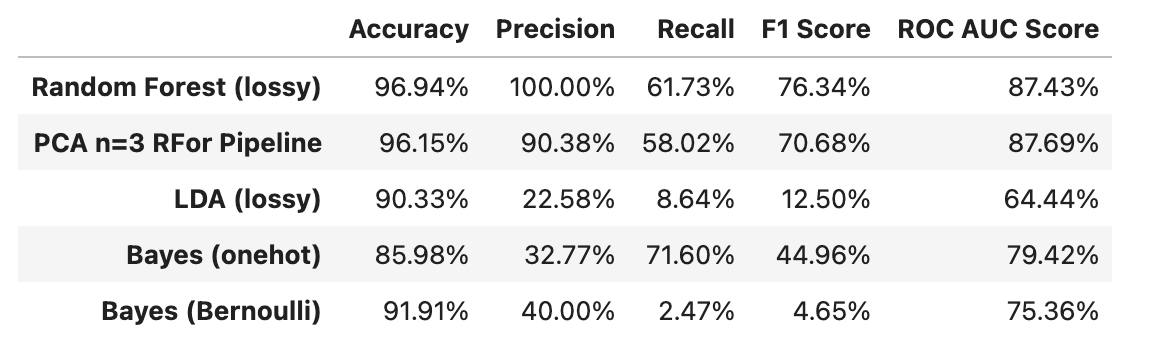
      Description automatically generated with medium confidence  
      *Fig. 14. Dropping a few features to emphasize on the important features.*  
        
      In Figure 14, we can see dropping some of the least-important features provides a better representation of the data, in terms of accuracy, F1 score, and ROC AUC score, indicating that there was a mild form of overfitting.
   4. Using Random Forest (lossy)  
      Graphical user interface

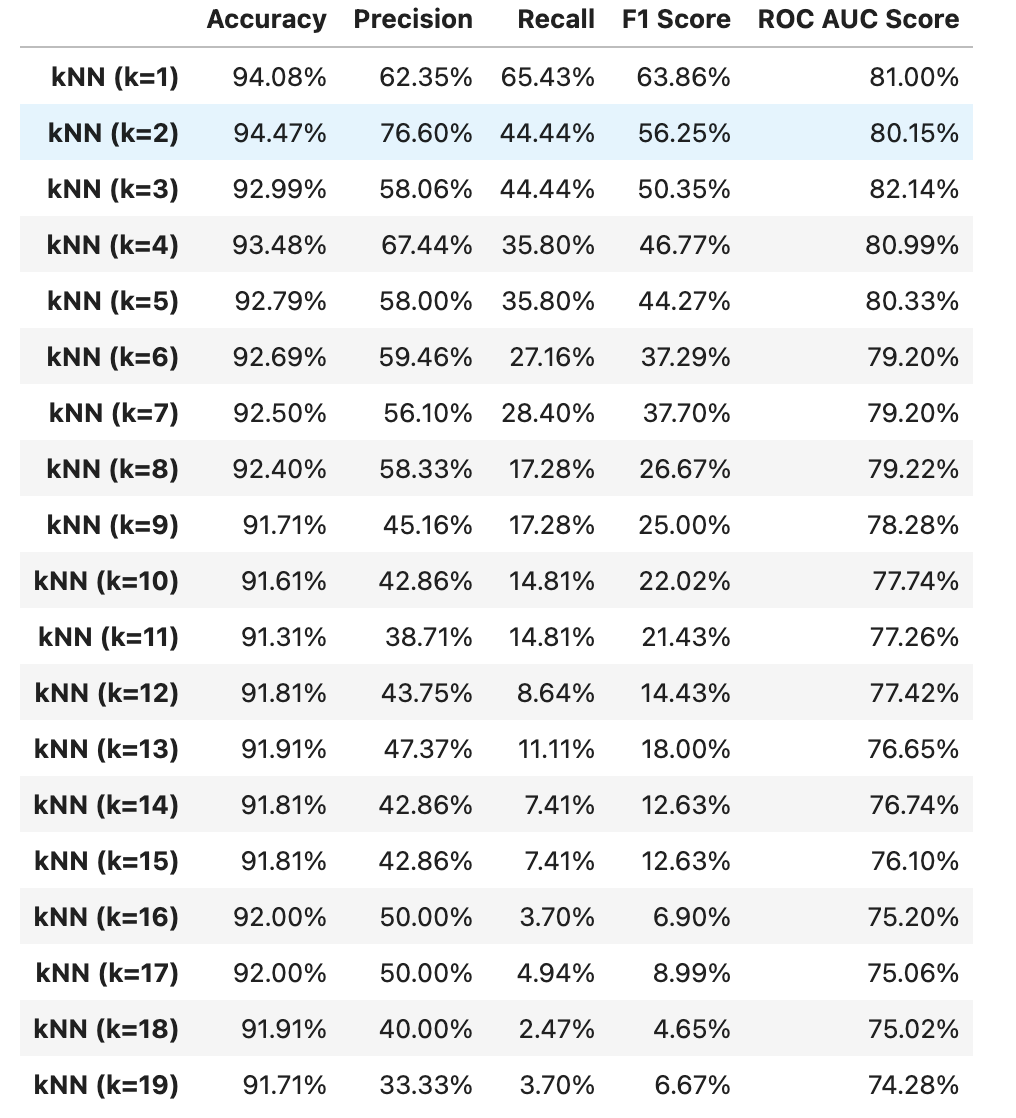
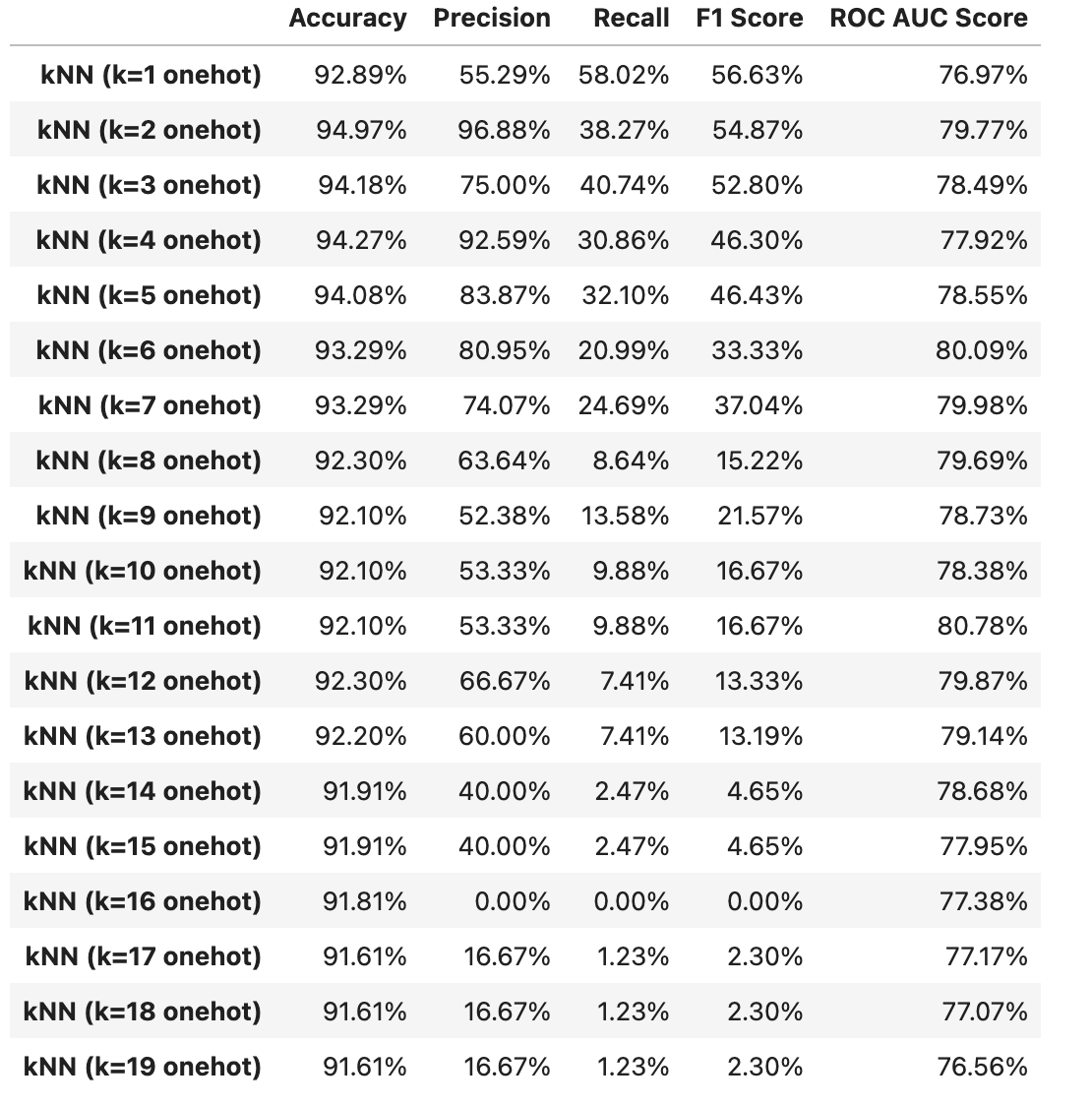
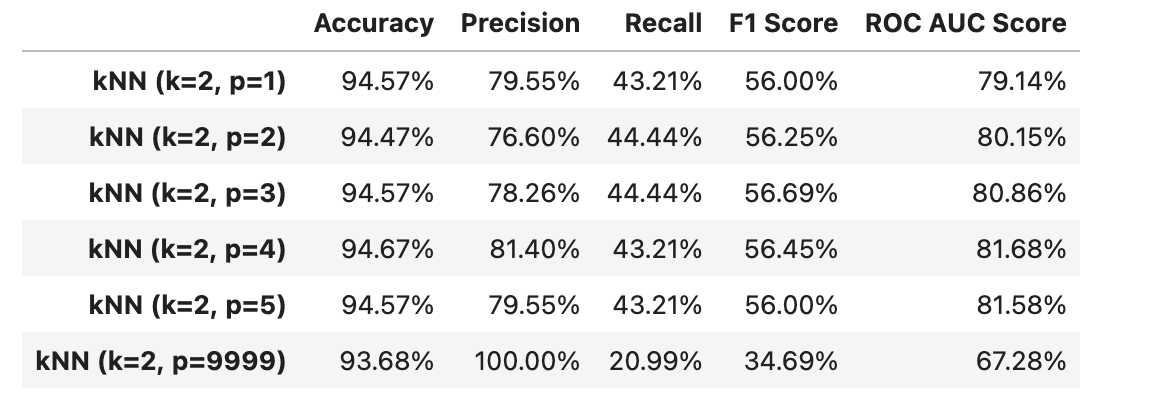
      Description automatically generated with medium confidence  
      *Fig. 15. Statistics after dropping more features.*  
        
      In Figure 15, we can see dropping even more features can lead to a better accuracy of the data and a higher F1 score, but the ROC AUC score drops slightly. This indicates that we reduce our understanding for the minority class (here, the number of repeat buyers) slightly.
   5. After Dropping AUC increases to 0.89  
      Graphical user interface

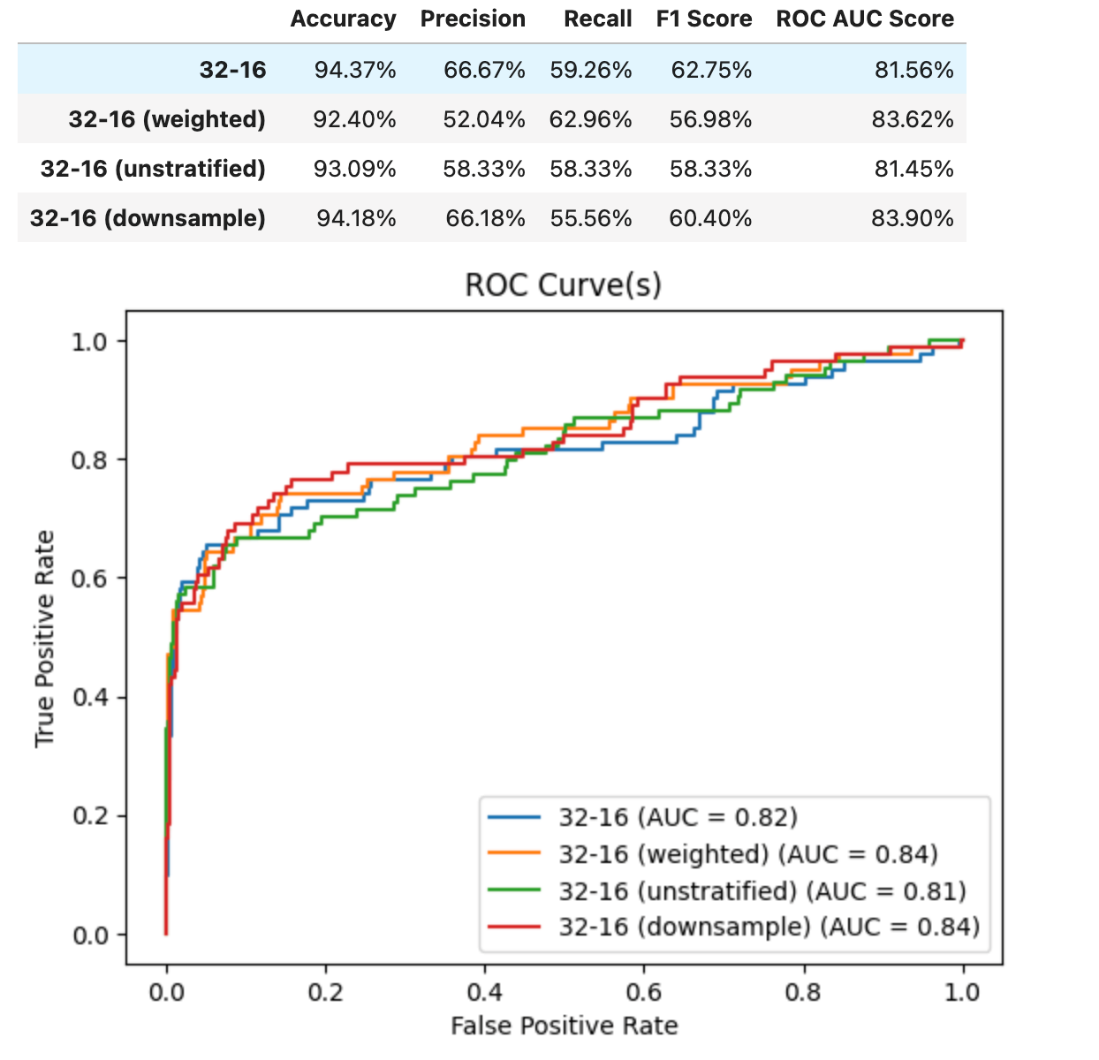
      Description automatically generated  
      *Fig. 16. Statistics after dropping the same columns, but not using one-hot encoding.*  
        
      We show, in Figure 16, the importance of one-hot encoding columns (particularly the User ID and Merchant ID features). If we do not, then the understanding between users and merchants is not fully captured and can lead to lower metrics.
2. **PCA**Note that we really don't expect this to perform well as it is not normally distributed.
   1. Normal PCA  
      Table

      Description automatically generated  
      *Fig. 17. Statistics after running mutliple PCA with various number of components*  
        
      In a pipeline with PCA and Random Forest, we see that it barely improves the metrics. For PCA with 3 components, the accuracy slightly drops, the F1 score worsens, while the ROC AUC score slightly increases.
   2. If we dropped and one-hot encoded the categorical columns the same columns as we did for Random Forest, we get different results.  
        
      *Fig. 18. Statistics with loss of information on PCA.*  
        
      We assume that the important features to drop in a pipeline with PCA and Random Forest would be the same ones we dropped for Random Forest. This leads to worse scores. We attribute this worsening of metrics to losing more information during PCA while we intentionally lose more information to prevent overfitting for Random Forest in the first place. This dual loss of information causes poor results.
3. **LDA**Like PCA, we expect to do poorly because it is not normally distributed.
   1. Normal LDA on data  
      Table

      Description automatically generated  
      *Fig. 19. Statistics after running untuned LDA*  
        
      We see that LDA fails spectacularly on this dataset. With an accuracy of 92%, it is equivalent of guessing the majority class (or 0, that people are not repeat buyers). This is reflected in its abyssal F1 score of 4.71% and its ROC AUC score of 62% (an ROC AUC score of 50% would indicate it is no better than random guess).
   2. But if we add in one hots, we get a different answer  
      Table

      Description automatically generated  
      *Fig. 20. Evaluation after running LDA on one-hot encoded data*  
        
      Simply using one-hot encoded data makes LDA perform much better, although its accuracy and F1 score shows it is not as useful as the other models.
4. **Recursive Feature Elimination**  
   **Recursive Feature Elimination, or RFE, is a method that drops the least important feature in a model that provides feature importances (such as Random Forest) until a desired number of features is obtained.**  
     
   **We use RFE with Random Forest to identify the most important features for the Random Forest. We avoid using RFE for the pipeline because PCA already does its job at getting the most important features. Adding another layer would only cause the model/pipeline to get lossy.**   
   ***Fig. 21. Evaluation of RFE with various limits on the number of features***  
     
   **RFE on Random Forest gives us slightly lower accuracy and F1 score compared to Random Forest in exchange for a slightly better ROC AUC score. If we compare it to our pipeline with PCA and Random Forest, RFE increases the F1 score back to its original score while increasing the accuracy for the slightest reduction in ROC AUC. We prefer RFE for better prediction compared to a pipeline with PCA and Random Forest.**
5. **Naive Bayes classifier**  
   In addition to the Random Forest, PCA, and LDA above, we include a naive Bayes classifier and non-parametric techniques. We really don't expect a naive Bayes classifier to do well, as it depends on normal distribution.  
      
   *Fig. 22. Statistics on naive Bayes classifier.*  
     
   A naive Bayes model performs poorly, even more so than LDA, with the lowest accuracy, F1, and ROC AUC score yet.  
     
   The precision and F1 Score increases if we add one-hots and Bernoulli.   
      
   *Fig. 23. Evaluation of naive Bayes on one-hot encoded data and naive Bayes that assumes every column is a Bernoulli distribution and will one-hot encode any column that is not binary.*  
     
   Working with one-hot data lowers the accuracy but sharply increases the F1 and ROC AUC score, making it more worthy of consideration.

**KNN**  
**In kNN, we determine which k value is the best. kNN should not be affected by the distribution of the data.**  
   
***Fig. 24. Statistics on various kNN runs with different numbers of neighbors.***  
  
**Higher neighbors in kNN adapt poorly to this data, as indicated by the low F1 scores.**  
  
**The answer remains the same when we add one-hot.**   
   
***Fig. 25. Runs of kNN with one-hot encoded data.***  
  
**One-hot encoding the data and running kNN seems to worsen the models. For example, a kNN model with 16 neighbors, we get a 0% F1 score.**  
  
**We get the highest accuracy when the value of *k* is 2. Now, parameterizing the value of *p.* The highest accuracy is obtained when *k* is 2 and *p* is 4.**  
   
***Fig. 26. Various different distance metrics for kNN with 2 neighbors.***  
  
**Finding the best distance metric for kNN yielded an interesting result. A Minowski metric of 1 (also known as Manhattan distance) and a metric of 2 (also known as Euclidean distance) do not perform as well as metrics of 3, 4, or 5, which are higher level abstractions. A metric of 9999 is meant to represent Linfinity (also known as Chebyshev distance); however, this distance metric performs poorly.**

1. **Generic optimization through neural networks**  
     
     
   ***Fig. 27. A sample of neural networks.***  
     
   **We first test neural networks to see if they benefit from class weight boosting, where additional weight is added to the minority class (that is, repeat buyers) and the majority class (i.e., not repeat buyers) gets a penalty. It lowers accuracy and F1 score, indicating it does not help us, despite the ROC AUC score slightly increasing. Stratification remains important, but now downsampling has an effect on this model.**  
     
   **Some preliminary testing shows that neural networks may be used to better generalize this data. However, there are many possible configurations for the neural network, with many resulting in overfitting. Various recommended techniques for machine learning was applied, such as cross-validation, regularization, dropout, and removing some features, but were largely unable to reduce overfitting for neural networks. We hope that additional time dedicated to tuning a neural network would outperform all the other models as it may generalize on the data better.**

We have currently tested the following models:

|  |  |
| --- | --- |
| **Model** | **Test accuracy (20% split)** |
| Random Forest | 96.35% |
| Random Forest (with one-hot encoding) | 96.25% |
| Random Forest (lossy) | 96.94% |
| LDA | 92.00% |
| LDA (with some features dropped) | 93.68% |
| Naive Bayes Classifier | 89.63% |
| Naive Bayes Classifier (with one-hot encoding) | 85.98% |
| kNN (with k=2) | 94.47% |
| kNN (with k=2, p=4) | 94.67% |

We can see that Random Forest (after one-hot encoding and dropping more features) is the best-performing model. If we look at models explicitly mentioned, kNN with 2 neighbors performs the best. Further tuning is required to get higher accuracies.

## Conclusion:

The analysis of the user engagement data for the selected set of items has provided valuable insights into user behavior and preferences on the e-commerce website. The findings suggest that the website may have a predominantly female user base, with a specific target audience in terms of age group. The website may benefit from providing engaging product displays, tailored to the preferences of these specific user groups. Additionally, seasonal factors such as holiday shopping and promotional events may be important in maintaining user engagement throughout the year.

In this study, we aimed to develop an accurate model for predicting repeat buyers in e-commerce. After performing feature engineering, we ended up with 32 features to train our models on. However, upon performing a statistical test, we found that none of our columns were normally distributed, which limited the possible accuracies of certain models like PCA, LDA, and the naive Bayes model.

To address this issue, we experimented with various techniques such as down sampling and stratifying the data during the train-test split. Interestingly, down sampling did not have a significant impact on the scores, while stratifying had a slight positive effect. Ultimately, we found that the k-Nearest Neighbors algorithm with 4 neighbors and a Minkowski distance of p=4 had the best parameters. However, the best model overall was the Random Forest model, which achieved an impressive 96.94% accuracy and a ROC AUC score of 87.43%.

In conclusion, we have developed a highly accurate model for predicting repeat buyers in e-commerce. However, further analysis is required to reduce overfitting, particularly through the use of neural networks and custom evaluation benches. Our study highlights the importance of careful feature engineering and model selection in achieving accurate predictions in e-commerce, and we hope our findings will serve as a valuable resource for future research in this field.