CECS 550- Pattern Recognition

Group 6
Progress Report

Team Members :

Dharaneeswar Reddy Rami Reddy

Madhana Mohit Konduru

Shruthi Venkatachalam

Ashwanth Kumar Alagesan

Sai Abhijyan Tanikella

Under the guidance of Prof. Mahshid Fardadi

Contents

1. Introduction

1.1 Need for Repeat Buyer Prediction

Businesses constantly struggle to understand customer needs, loyalty, and behavior. These businesses have a stockpile of data about the customer's activity. They try to devise a promotions sale or loyalty program using this data. Unfortunately, most customers only seek deals and what is profitable, so they end up as single-time buyers. So the business needs to identify loyal customers who have repeatedly bought items and tailor the deals based on their information to gain profits.

1.2 Problem Statement

In this project, we are provided a dataset with information on promotional shopping events from an e-commerce platform. The task is to design a system that will increase the ROI (in other words, you need to predict the probability that these new buyers would purchase items from the same merchants again within six months), reduce promotional costs, and identify one-time buyers.

2. Data

The data consists of user activity over the e-commerce platform for the last six months and on the "Double 11" promotion day. The label column represents whether the user will be a repeat buyer. The format chosen for this project is data_format2.

The tables below give the general outline of the data structure in Data_format2.

Data Fields	Definition
user_id	A unique id for the shopper.
age_range	User' s 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.
gender	User's gender: 0 for female, 1 for males, 2 and NULL for unknown.
merchant_id	A unique id for the merchant.
label	Value from {0, 1, -1, NULL}. ' 1' denotes ' user_id' as a repeat buyer for ' merchant_id' while ' 0' is the opposite. ' -1' represents that ' user_id' is not a new customer of the

	given merchant, thus out of our prediction. However, such records may provide additional information. ' NULL' occurs only in the testing data, indicating it is a pair to predict.
activity_log	Set of interaction records between {user_id, merchant_id}, where each record is an action represented as ' item_id:category_id:brand_id:time_stamp:action_type' . ' #' is used to separate two neighboring elements. Records are not sorted in any particular order.

As group 6, we are assigned a specific range of item ids to work with from the humungous dataset.

Group	item_id
6	801 - 960

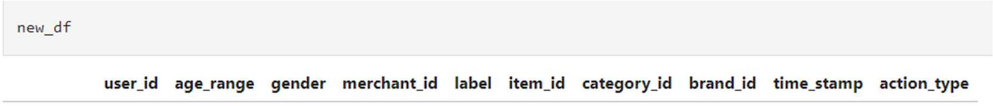
3. Data Pre-processing

3.1 Choosing the Data Format

Initially, Data format_1 was chosen as we felt the feature engineering would be easy. However, not long after we started working with Data_format_1, we understood data cleaning might be difficult. We also encountered problems with merging and a tremendous number of Null values. To tackle these issues, we started working with Data_format_2 as it provided a single unified CSV file and was easier to manipulate and figure out its patterns.

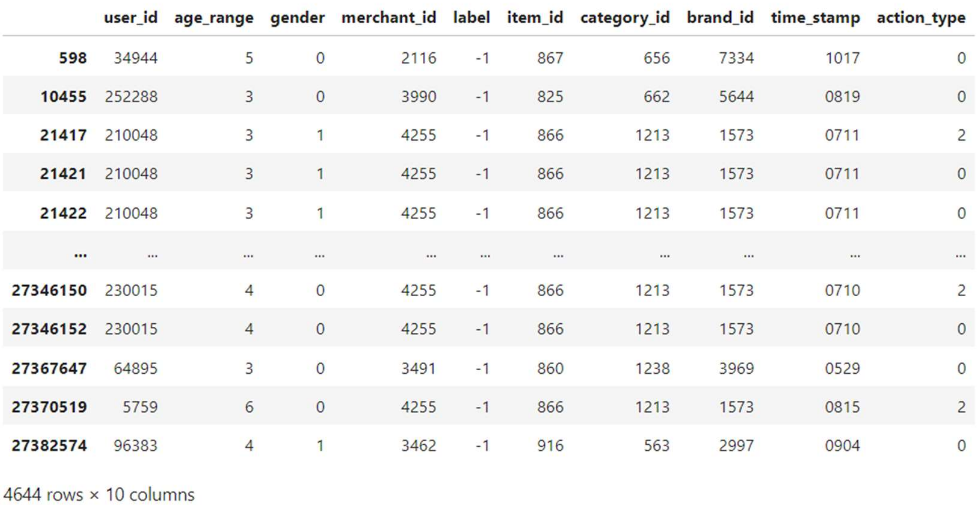
3.2 Pre – Processing steps

In the initial pre-processing step, we replaced null and unknown values with their respective numeric encoding. The second pre-processing step consisted of Splitting the activity_log column at every occurrence of #. The third step involved separating the features encoded in the activity_log by the ":" symbol. Now the new Data sort of looks like this.



```
new_df

        user_id  age_range  gender  merchant_id  label  item_id  category_id  brand_id  time_stamp  action_type
```

Then finally, we filtered out the data based on our group's assigned item_id values. (group 6 : item_id : 801- 960).We also created a copy of our new pre-processed data and transformed it into Data_format_1, which we thought would be helpful during feature Engineering.

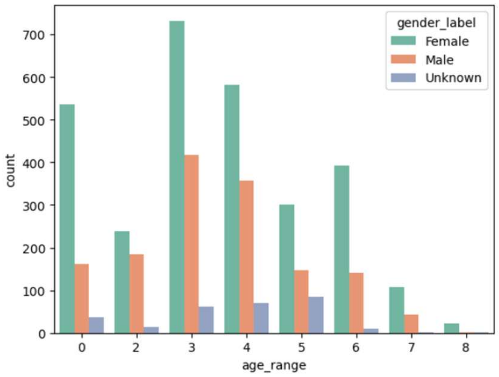


	user_id	age_range	gender	merchant_id	label	item_id	category_id	brand_id	time_stamp	action_type
598	34944	5	0	2116	-1	867	656	7334	1017	0
10455	252288	3	0	3990	-1	825	662	5644	0819	0
21417	210048	3	1	4255	-1	866	1213	1573	0711	2
21421	210048	3	1	4255	-1	866	1213	1573	0711	0
21422	210048	3	1	4255	-1	866	1213	1573	0711	0
...	...	...	...	...	...	...	...	...	...	...
27346150	230015	4	0	4255	-1	866	1213	1573	0710	2
27346152	230015	4	0	4255	-1	866	1213	1573	0710	0
27367647	64895	3	0	3491	-1	860	1238	3969	0529	0
27370519	5759	6	0	4255	-1	866	1213	1573	0815	2
27382574	96383	4	1	3462	-1	916	563	2997	0904	0

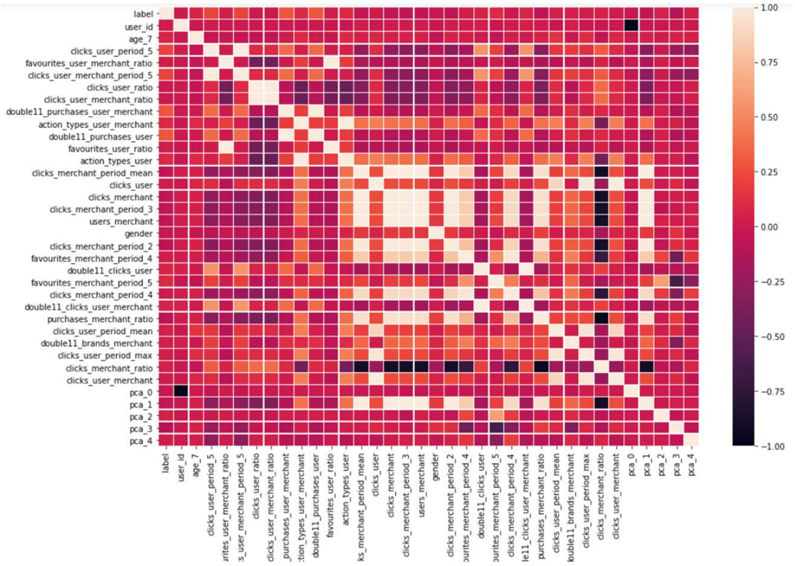
4644 rows × 10 columns

4. EDA

After pre-processing the data, the next part is to figure out the patterns in the data and see what ways can help us in our analysis path. Our first idea was to look at the gender distribution of the data according to their age. We found that most females belonged to age group three, and age groups three, four, and five have almost equal numbers of unknown observations.

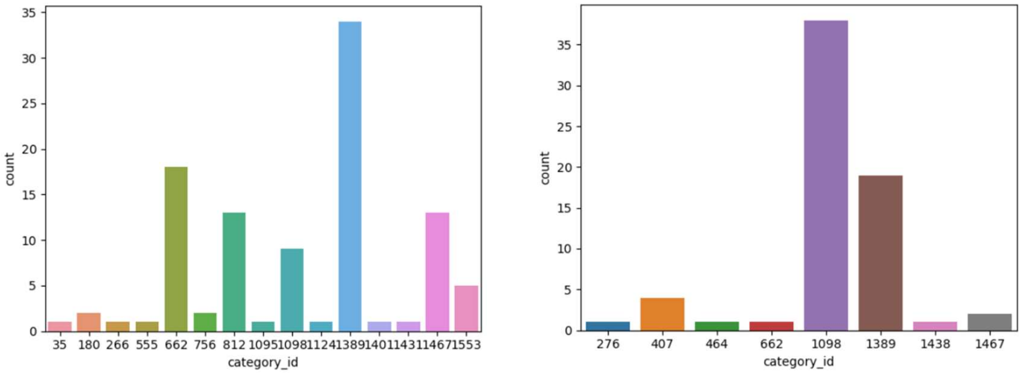


We also performed one of the basic Data statistics like correlation plot.

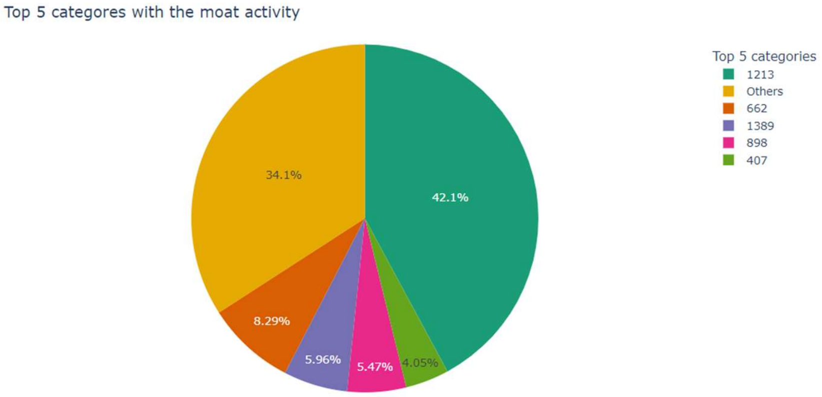


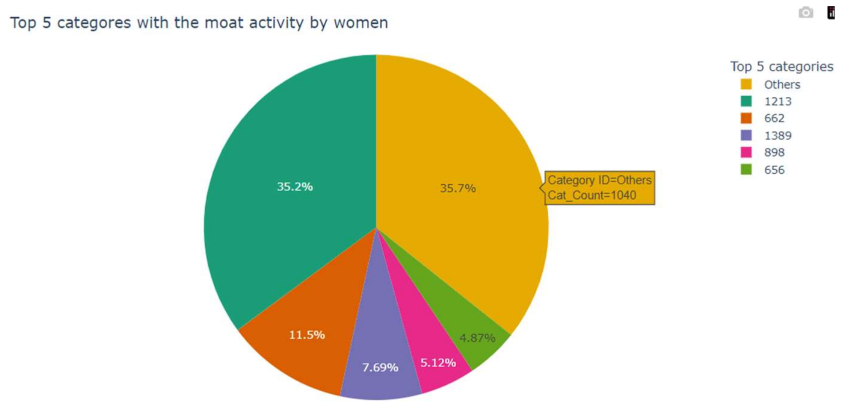
We also saw the most sales happen on the promotion days and visualized how the brands, users, and items worked together.

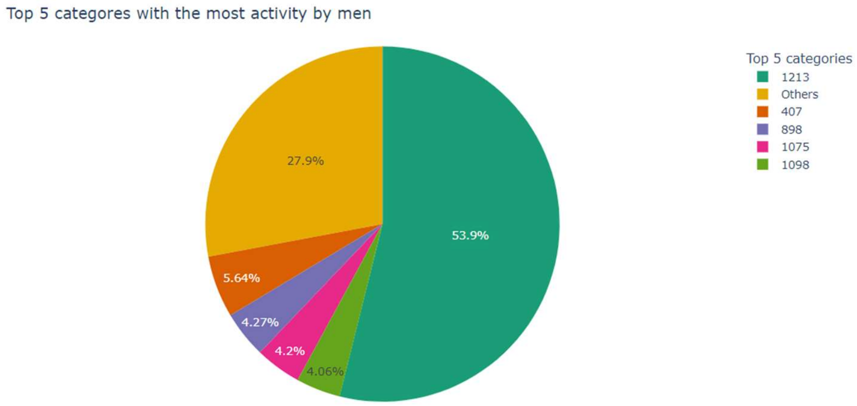
MALES FEMALES

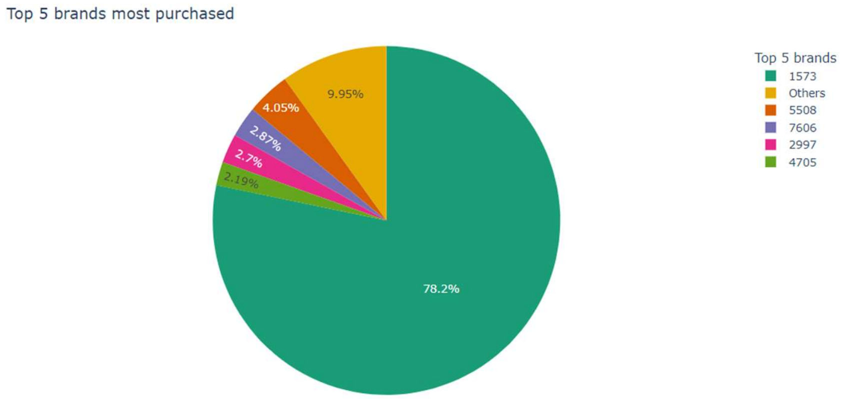


The overall activity by gender was categorized and depicted as pie charts for easy understanding.

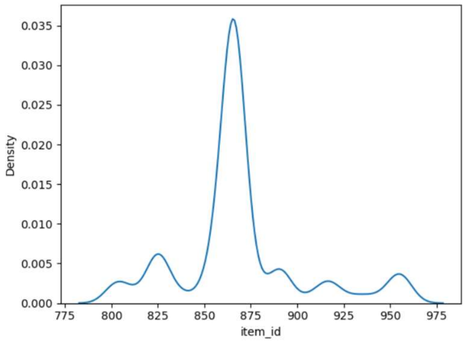








We also wanted to see which item was most popular among the users, so the plot below shows.



And this ended our initial Exploratory data analysis. We realized that all the data is oriented in three perspective user perspective, Merchant perspective, and user-merchant pair perspective. We also understood that all these revolved around the actions performed by the users. Hence using this information, we proceeded to the next step, feature engineering.

5. Feature Engineering And Ranking

5.1 Feature Engineering

We used the transformed dataset obtained from converting the data_format2 into data_format1. Based on this data and the basic user-merchant logic, we were able to engineer a total of 153 features.

The features section is a little long please bear with us 😜 .

time_period:

This feature is obtained by converting the timestamp into numerical days such that 0 indicates the first day any action has occurred in the entire dataset. This number is then divided by 31 to get the time period.

age_0, age_1, age_2, age_3, age_4, age_5, age_6, age_7, age_8:

These features represent if the user is in that particular age group or not. These are binary response variables for the age_range feature.

DAILY FEATURES:

Count features User Perspective

Items_user, categories_user, merchants_user, brands_user:

The number of items/ categories/ merchants/ brands a user has interacted with/on to do any of the actions..

dates_users/ peroids_user:

The number of dates/ periods a user has performed an action.

action_types_user:

The number of action types a user has performed.

Count Features Mechant Perspective

users_merchant:

Total number of users who have interacted with a merchant.

items_merchant/ categories_merchant/ brands_merchant:

The total number of items/ categories/ brands that users have interacted with that belong to a particular merchant.

dates_merchant/ periods_merchant:

The number of dates/ periods a merchant has been interacted with.

Action_types_merchant:

Number of action_types that have been performed on a merchant.

Count Features User-Mechant Perspective

items_user_merchant/ categories_user_merchant/ brands_user_merchant:

The number of items/ categories/ brands a user has interacted with on that particular merchant.

dates_user_merchant/ periods_user_merchant:

The number of dates/ periods a user has interacted with on that particular merchant.

action_types_user_merchant:

The number of action types that have been performed by a user for that merchant.

Action count Features user Perspective

clicks_user/ purchases_user/ favourites_user:

Total number of clicks/ purchases/ add to favorites performed by the user.

Action count Features merchant Perspective

clicks_merchant/ purchases_merchant/ favourites_merchant:

Total number of clicks/ purchases/ add to favorites performed for a merchant.

Action count Features user-merchant Perspective

clicks_user_merchant/ purchases_user_merchant/ favourites_user_merchant:

Total number of clicks/ purchases/ add to favorites a user has performed for that particular merchant.

ratio of actions in user perspective:

clicks_in_merchant_ratio_perspective/ purchases_in_merchant_ratio_perspective/ favourites_in_merchant_ratio_perspective :

The ratio of clicks/ purchases/ add to favorites by a user for a merchant to total number of clicks/ purchases/ add to favorites by user.

ratio of actions in merchant perspective:

clicks_by_user_ratio_perspective/ purchases_by_user_ratio_perspective/ favourites_by_user_ratio_perspective:

Ratio of clicks/ purchases/ add to favorites by a user for a merchant to total number of clicks/ purchases/ add to favorites for a merchant by all users.

ratio of action for given user :

clicks_user_ratio/ purchases_user_ratio/ favourites_user_ratio:

Ratio of total number of clicks/ purchases/ add to favorites performed by a user to the total number of actions performed by the user.

ratio of action for given merchant

clicks_user_merchant_ratio/ purchases_user_merchant_ratio/ favourites_user_merchant_ratio:

Ratio of total number of clicks/ purchases/ add to favorites by a user for that merchant to the total number of actions performed by a user to that merchant.

ratio of action type for user-merchant pair

clicks_merchant_ratio/ purchases_merchant_ratio/ favorites_merchant_ratio:

The ratio of total number of clicks/ purchases/ add to favorites for the merchant to the total number of actions performed for the merchant by all users.

interval:

It is the difference between the first and the last interaction date by a user.

MONTHLY FEATURES:

- Similar count features were generated for a monthly basis.
- Max and mean were considered.
- Monthly features are generated in user perspective, merchant perspective, and user-merchant pair perspective.

User Perspective:

clicks_user_period_max:

It is the max of the total number of clicks performed by a user out of all time periods.

clicks_user_period_mean:

It is the mean of the total number of clicks performed by a user out of all time periods.

Multivariate response variable user perspective:

Clicks_user_period_0, clicks_user_period_0, ……clicks_user_period_0, clicks_user_period_0, ….. Clicks_user_period_5, clicks_user_period_0, ….. Clicks_user_period_5:

These features represent multivariate response variables for various time periods with respect to various action types. For example, **Clicks_user_period_0** represents the total number of clicks by a user in the time period 0.

Merchant Perspective:

Clicks_merchant_period_max:

It is the max of the total number of clicks performed for a user out of all time periods.

Clicks_merchant_period_mean:

It is the mean of the total number of clicks performed for a merchant out of all time periods.

Clicks_merchant_period_0, clicks_mercahnt_period_0, ……clicks_merchant_period_0, clicks_merchant_period_0, …..

Clicks_merchant_period_5, clicks_merchant_period_0, ….. Clicks_merchant_period_5:

These features represent multivariate response variables for various time periods with respect to various action types. For example, **Clicks_merchant_period_0** represents the total number of clicks for a merchant in the time period 0.

DOUBLE 11 DAY FEATURES

The Double 11 features are the same features that have been extracted from the records on the Double 11 day. Double 11 day is a day that occurs on the eleventh day of the eleventh month where promotions happen on a large scale.

Feature extraction with just the D11 days data yields these data.

It is again performed on the user perspective, merchant perspective, and user-merchant pair perspective.

For Example:

Double11_Items_user is the **items_count** feature on the Double 11 day.

Double11_periods_merchant is the **periods_merchant** feature on the Double 11 day.

Double11_brands_user_merchant is the **brands_user_merchant** feature on the Double 11 day.

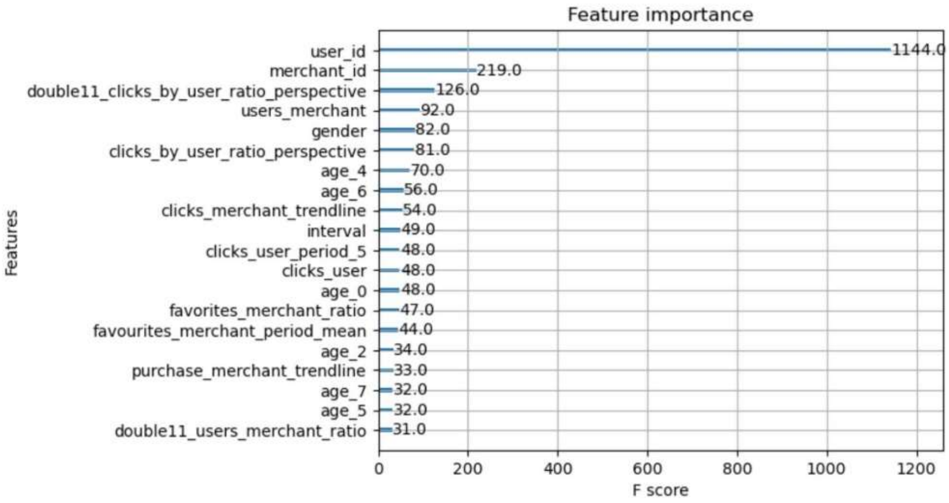
Double11_purchases_user is the **purchases_user** feature on the Double 11 day and so on

NOTE:

- Replace null values of gender and age_rang with their respective numerical variables.
- Make sure that all the columns are in numpy int 64.
- We do not have any add_to_cart in our date set.
- Smoothing has been performed for the ratio features.

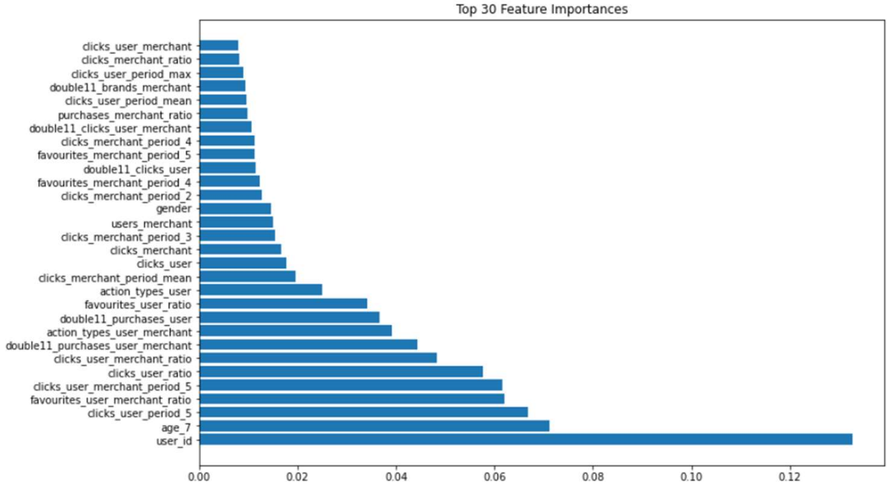
5.2 Feature Ranking

Feature Ranking can be performed in many different ways. In this project, we used three different methods to achieve feature ranking and finally decided to go with the feature that produced the best results. We started our first feature ranking process using a regression model that provided the following importance order.

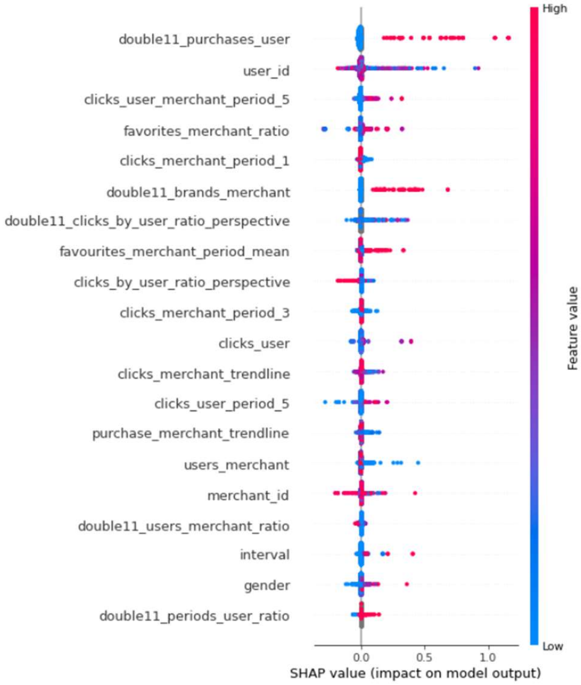


Similarly, we also used SHAP and random forest regressor to achieve two other lists of features. The below picture provides the details of these features. The features provided by the Random forest regressor provided the best results when experimented with the random forest model, KNN and CNN; hence we decided to follow these features for the final prediction task.

Features using Random forest regressor



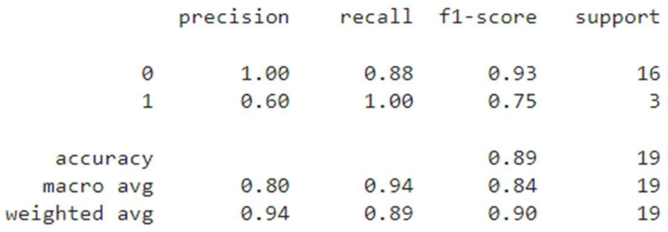
Feature Ranking using SHAP

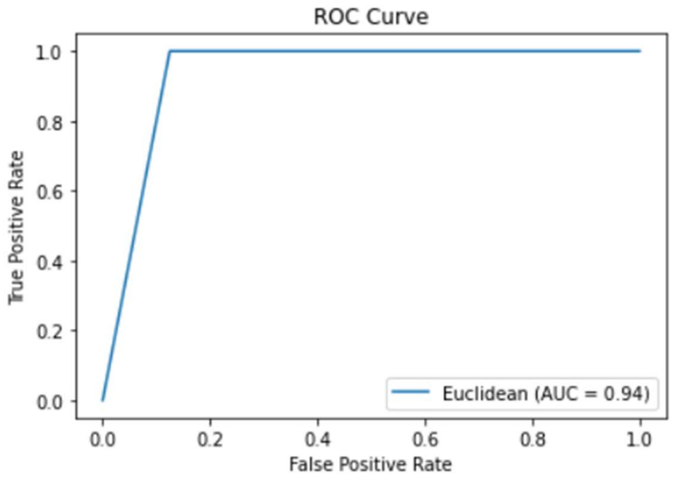


6. Modeling

6.1 Baseline Model

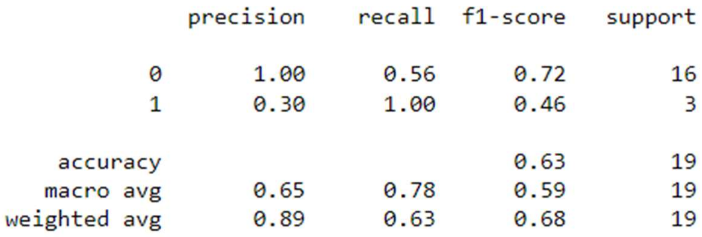
In the feature ranking phase, we used three methods to list the importance of features, and Random forest Regressor was one of the methods used. Hence we decided to start with the random forest model as our baseline model and compare this model with the classic machine learning models. The random forest model was able to achieve an accuracy of 89 percent on the reduced features and PCA data. The picture below provides the performance report and ROC curve of Random Forest.


```
              precision    recall  f1-score   support

           0       1.00      0.88      0.93        16
           1       0.60      1.00      0.75         3

    accuracy                           0.89        19
   macro avg       0.80      0.94      0.84        19
weighted avg       0.94      0.89      0.90        19
```

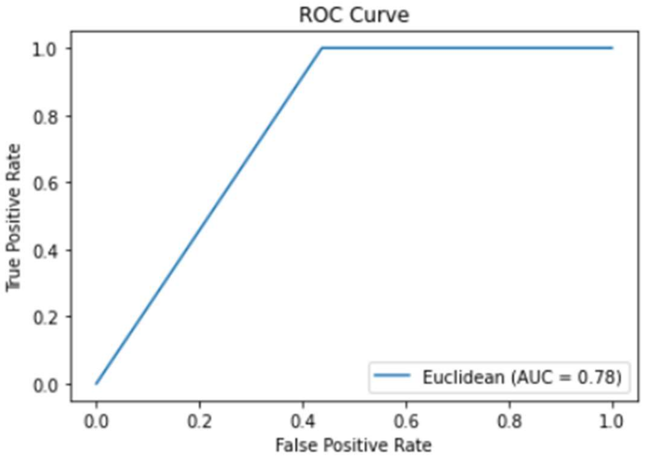


6.2 Bayes Classifier

After looking at the random forest model, we started with a Bayes classifier. A grid search was conducted to test which hyperparameter yielded the best results. Also, five-fold cross-validation was done. After these steps, the modeling and prediction were performed, in which the model gave an accuracy of 0.63. As you can see, it underperformed compared to our baseline model. The picture below provides the performance report and ROC curve of Naïve Bayes.



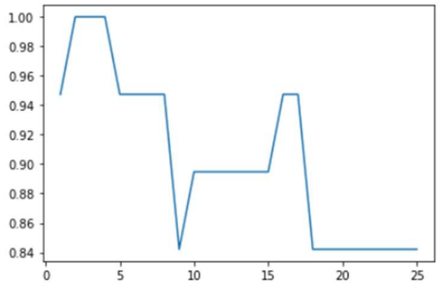
```
              precision    recall  f1-score   support

           0       1.00      0.56      0.72        16
           1       0.30      1.00      0.46         3

    accuracy                           0.63        19
   macro avg       0.65      0.78      0.59        19
weighted avg       0.89      0.63      0.68        19
```



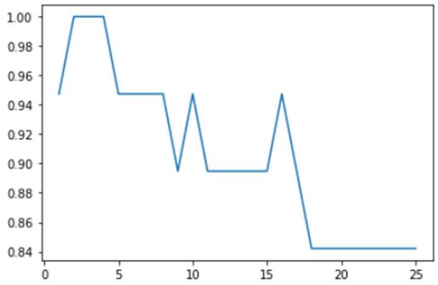
6.3 KNN and Parzen Window

The next step was to use non-parametric models like KNN and parzen windows and analyze the model's performances. So we started with knn. KNN can be performed with four distance measures: Euclidean, Manhattan, Hamming, and cosine. We used grid search on all these to determine an optimal number of neighbors to achieve the best results. The pics below depict the accuracy of the kNN model with respect to the number of neighbors on the x-axis.

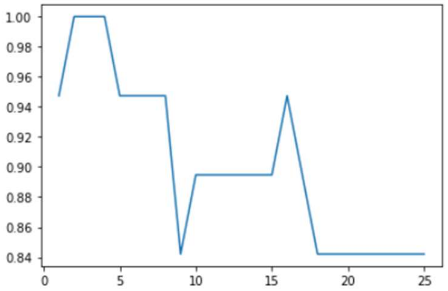
Euclidean



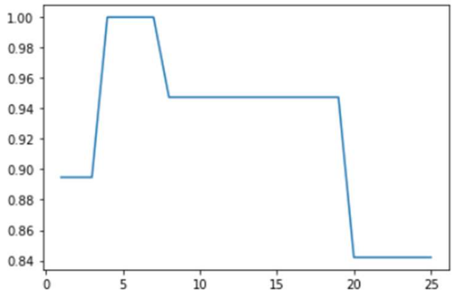
Manhattan



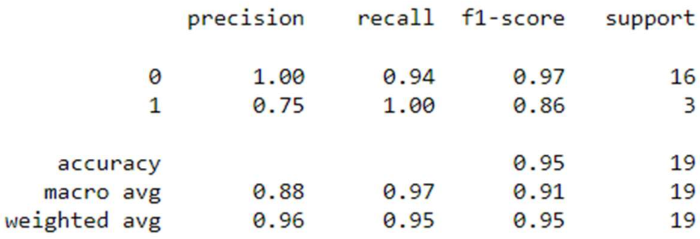
Cosine



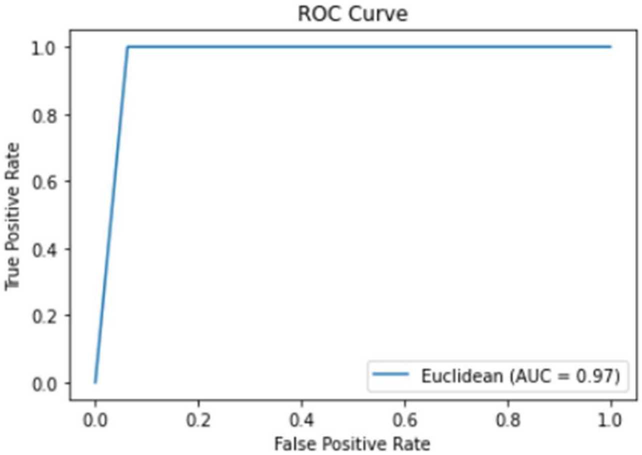
Hamming



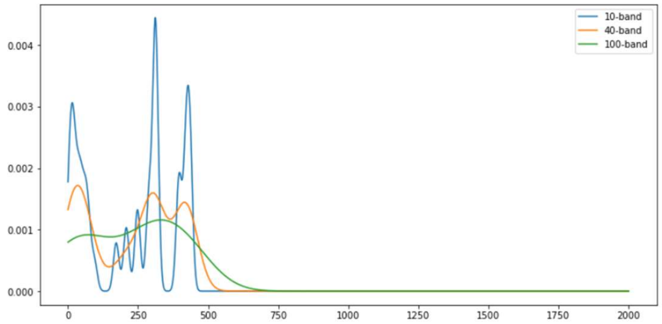
From the graphs, we can see that for 1 to 5 neighbors, almost all the distance measures provide the same results. So let's look at the KNN Model's performance using the Euclidean distance measure.



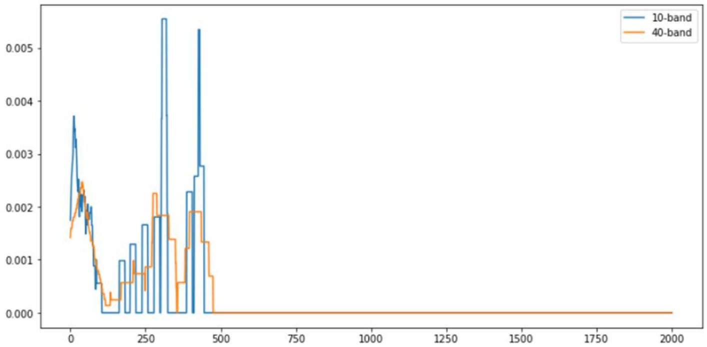
```
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        16
           1       0.75      1.00      0.86         3

    accuracy                           0.95        19
   macro avg       0.88      0.97      0.91        19
weighted avg       0.96      0.95      0.95        19
```



Now let's take a look at the Parzen windows model. We chose "clicks_merchant" as our density feature. Kernel density estimation (KDE) is a proper statistical procedure since it makes no assumptions about the form or distribution of the data. In other words, it is non-parametric, and its probability density estimations depend exclusively on the qualities seen in the data. On the other hand, Parametric approaches make assumptions about the underlying distribution of the data, such as assuming a normal distribution. This graph shows that the user's clicks at a single merchant (clicks_merchant) are primarily between 0-1000 and 2500-3500; however, this is difficult to tell precisely.

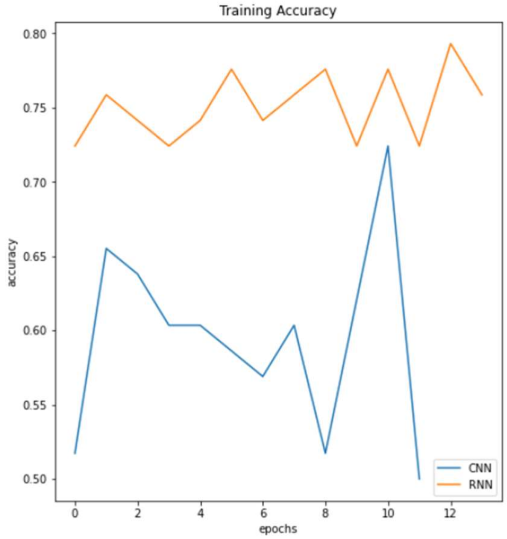


These constraints are undesirable because they make data interpretation more difficult. A KDE smoothes them out by interpolating: generating logical values to "fill in" gaps between data points. Here is an example of a KDE plot using this data (using Seaborn):



6.4 CNN and RNN

The next step was to check if a neural network could somehow increase the performance and prediction of label values. We used two different types of neural networks CNN and RNN. Both of them yielded excellent accuracy of 0.73 and 0.84. Their performance can be compared in the below graph.



7. Conclusion and Insights

To conclude our project, we used customer activity data to predict whether the customer is a loyal/repeat buyer or a one-time buyer. This project explores several machine learning models like the Bayes classifier and non-parametric methods like KNN and Parzen windows. We also explored some neural network approaches like CNN and RNN. After analysis of all the processes, it turns out that KNN has the best accuracy rate of 0.94, and the second is the random forest model with an accuracy of 0.89. Hence non-parametric methods might be the best approach for this data, according to our analysis.

In our analysis of customer behavior on Double 11 Day and the period in which Double 11 Day occurs, we see no significant difference in the click patterns of loyal/repeat customers. This might mean loyalty may not necessarily translate to higher interest in special events or promotions.

However, we found a strong correlation between the click-merchant ratio and the likelihood of a customer becoming loyal. Customers who clicked on a merchant more frequently were likelier to be repeat buyers. We also discovered that customers who browsed more with a single merchant were more likely to become repeat buyers.

Furthermore, we found that merchants who offer a limited number of brands in their stores tended to have more loyal customers. This may indicate that customers prefer a more focused selection of products from a trusted source rather than a more extensive, more varied selection.

8. References :

- https://www.kaggle.com/code/residentmario/kernel-density-estimation-with-ted-talks
- https://machinelearningmastery.com/what-is-imbalanced-classification/
- Repeat Buyer Prediction for E-commerce https://dl.acm.org/doi/10.1145/2939672.2939674
- https://www.researchgate.net/publication/305998405_Repeat_Buyer_Prediction_for_E-Commerce
- https://ieeexplore.ieee.org/document/8955625