# CECS 550 Pattern Recognition

Professor Mahshid Fardadi

|  |  |
| --- | --- |
|  | |
| Submitted By | |
| Akshatha Hebbar  028017780  Akshatha.hebbar01@student.csulb.edu | Pavan Sai Kalyan Badam  029358574  Pavansaikalyan.badam01@student.csulb.edu |
|  |  |

Table of Contents

[CECS 550 Pattern Recognition 1](#_Toc133840792)

[Problem Statement 2](#_Toc133840793)

[Dataset Used 2](#_Toc133840794)

[Data Preprocessing 2](#_Toc133840795)

[Data Filtering activity logs to get information. 2](#_Toc133840796)

[Removing / approximating the null values 2](#_Toc133840797)

[Data Visualization 3](#_Toc133840798)

[Chart 1 Distributions of different action\_types that occurred in the dataset 3](#_Toc133840799)

[Chart 2: Top 10 high demand merchants inferred from user’s purchases. 3](#_Toc133840800)

[Chart 3: Trend in activity 4](#_Toc133840801)

[Chart 4: Preferred categories for female 4](#_Toc133840802)

[Chart 5: Preferred categories for Male 5](#_Toc133840803)

[Chat 6: Top favorite brands among users during the highest activity month (November) 5](#_Toc133840804)

[Chart 7 Top 3% of items that are interested among different age range. 6](#_Toc133840805)

[Chart 8: Activity based on gender. 6](#_Toc133840806)

[Chart 9 : Repeated age range monthly buyers 7](#_Toc133840807)

[Chart 10: top 4% of user who made frequent purchases during the high active months(September, October and November) 7](#_Toc133840808)

[Feature Engineering 8](#_Toc133840809)

[Feature 1 – Mean of unique items purchased. 8](#_Toc133840810)

[Feature 2 – Total actions taken on each merchant page by each user 8](#_Toc133840811)

[Feature 3 - Proportion of purchases made by females within a 30-day period 8](#_Toc133840812)

[Feature 4 – clicks on items during the highest active months 8](#_Toc133840813)

[Feature 6 – Users category interest 9](#_Toc133840814)

[Feature 7 - Users brand interest 9](#_Toc133840815)

[Feature 8 – Purchase Ratio 9](#_Toc133840816)

[Feature 9 – Marking users who purchase top brands. 9](#_Toc133840817)

[Feature 10 - Average days between purchases by a user on a category 9](#_Toc133840818)

[Dataset statistics and feature Ranking 10](#_Toc133840819)

[Statistical Summary 10](#_Toc133840820)

[Prediction Model. 13](#_Toc133840821)

[Non- parameter technique 13](#_Toc133840822)

[Model Evaluation 13](#_Toc133840823)

[Discussion 16](#_Toc133840824)

[k-Nearest Neighbors 16](#_Toc133840825)

[Parzen Window 16](#_Toc133840826)

[Recommendation for e-commerce platform for shopping one-time buyers 17](#_Toc133840827)

[References 17](#_Toc133840828)

## Problem Statement

*Merchants often gain many new customers through promotions, but a significant portion of these customers are only interested in one-time deals. Therefore, the impact of promotions on long-term sales may be limited. To maximize return on investment (ROI) and reduce promotion costs, it is crucial for merchants to distinguish between one-time buyers and potential loyal customers and focus their efforts on converting the latter group.*

*In this project, you are provided a dataset with information on promotional shopping event from e-commerce platform. Your task is to design a system which 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 6 months), reduce promotional cost, and identify one-time buyers.*

## Dataset Used

data\_format2.zip

Down Sampling of data is done to retrieve the items 1121 – 1280

Retrieved data is imbalanced so fetched 20% of rows of label 1.

## Data Preprocessing

### Data Filtering activity logs to get information.

As the data is not readily available,information must be extracted from the activity\_log column. The activity\_log column is used to extract details such as item\_id, category\_id, brand\_id, time\_stamp (converted to %m%d format), and action type.

A screenshot of a computer code

Description automatically generated with low confidence

Fig 1: Data Extraction

### Removing / approximating the null values

Removing unknown or null values is crucial in the data filtering process. In the dataset, we have gender with 2 representing the unknown value. Additionally, we have an age range of 0 corresponding to the unknown. These ambiguous data along with the present 'null' value will hinder getting better results from the model. By replacing these values with mode(calculated using one or more columns) missing information can be handled.

A black background with white text

Description automatically generated with low confidence

Fig 2: Represents data Cleaning.

## Data Visualization

### Chart 1 Distributions of different action\_types that occurred in the dataset

PieChart is used to visualize the data based on different action\_types, 88% of users clicked once, 4.5% added item as their favorite and 7.3% made an actual purchase.

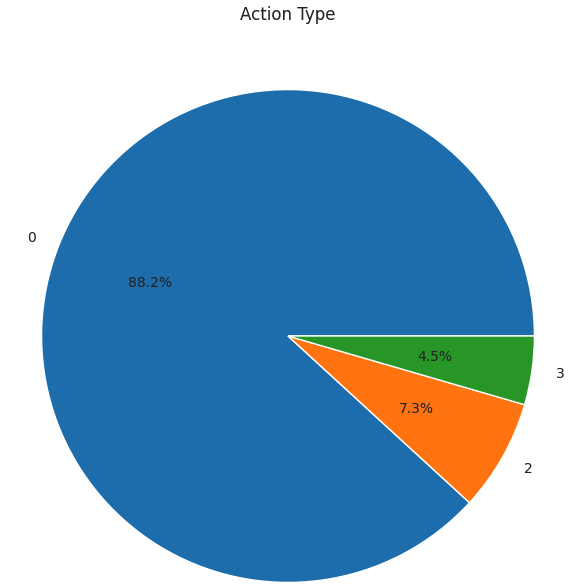


Fig 3: Pie Chart representing action type.

### Chart 2: Top 10 high demand merchants inferred from user’s purchases.

The e-commerce dataset is analyzed to identify the rows indicating purchase (action\_type = 2 ) and to filter out non-repeated users. The resulting data is grouped by merchant\_id to determine the top 10 merchants with the highest number of repeated users (Label = 1). This analysis provided using bar chart with valuable insights into the successful merchants with high customer retention rates.

A picture containing text, screenshot, font, plot

Description automatically generated

Fig 4: Bar Chart representing top 10 merchants.

### Chart 3: Trend in activity

Using the information in the e-commerce dataset trend in activity can be derived. It is primarily derived with a timestamp and label of 1. The information is plotted in a line plot(best represents a trend in activity). The x-axis and y-axis represent the date and number of purchases respectively. For better visualization, the y-axis is scaled logarithmically.

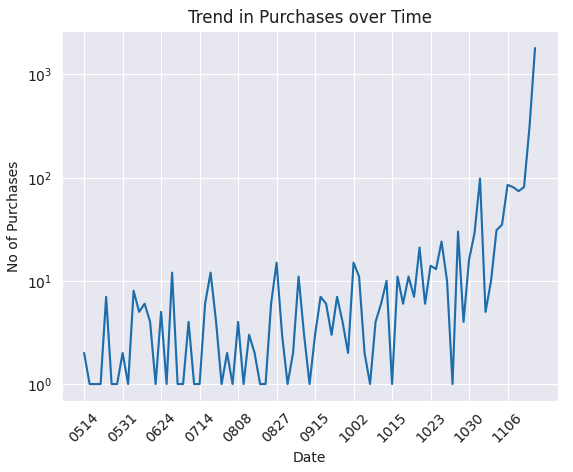


Fig 5: Trend in activity

### Chart 4: Preferred categories for female

A heatmap is generated by using data from the label column to identify purchases and information from the gender(Female) and category\_id columns to determine the preferred categories of female customers. Only categories with more than 15 purchases are considered for this analysis. The resulting heatmap provides an easy-to-read visual representation of purchase patterns across different categories and merchants.

A picture containing text, screenshot, square, number

Description automatically generated

Fig 6: Preferred categories for Female

### Chart 5: Preferred categories for Male

A heatmap is generated by using data from the label column to identify purchases and information from the gender (Male) and category\_id columns to determine the preferred categories of female customers. Only categories with more than 10 purchases are considered for this analysis. The resulting heatmap provides an easy-to-read visual representation of purchase patterns across different categories and merchants.

#### A picture containing text, screenshot, diagram, square Description automatically generated

Fig 7: Preferred categories for Male

### Chat 6: Top favorite brands among users during the highest activity month (November)

A scatter plot displays the brand IDs which were marked as favorites during the month of November, as it is the highest activity month of all. The time stamp in %m%d is used to determine the exact date, each brand was favored by users. From the plot, it can be deducted that brand ID 82 was in the highest demand on November 10th, followed by brand ID 2270 on November 9th.

A picture containing text, screenshot, number

Description automatically generated

Fig 7: Top Brands in November (Highest active month)

### Chart 7 Top 3% of items that are interested among different age range.

The top 3% of interested items based on age range are extracted. We begin by calculating the count of each item and then with a preset threshold of 0.97 we get the top 3% of the data. The report is displayed as a scatter plot, which helps in the identification of the age ranges and their item IDs using which we can get good interpretation of the data.

A picture containing text, screenshot, number, font

Description automatically generated

Fig 8 : Top 3% of items that are interested among different age range

### Chart 8: Activity based on gender.

#### The violin plot presented below illustrates a distinct comparison between the actions of different genders and their associated user age. From the plot, we can deduce that the age group of 25 to 29 years exhibits the highest level of activity and purchases.

A picture containing screenshot, diagram, plot

Description automatically generated

Fig 9: Activity based on gender

### Chart 9 : Repeated age range monthly buyers

A line plot helps in visualizing the purchases made by each age range for the past 6 months. All monthly purchases dominated by age range of 3 [25 to 29] contributing to the most significant purchases. However in the month of May and November, ages between [35 to 39] almost caught up to the purchases made by the users in their 20’s. Additionally, during the month of November all the age ranges spiked up their purchases.

A picture containing line, diagram, plot, slope

Description automatically generated

Fig 10: Repeated age range monthly buyers

### Chart 10: top 4% of user who made frequent purchases during the high active months(September, October and November)

The below code shows a barplot with user\_id as the x axis and count of purchases in the y-axis. he quantile function is applied to a distinct combination of user\_id, timestamp, and gender (known as ‘user\_activity\_each\_day’), and a 4% threshold is set. It is easily noticeable that the top 4% of frequent buyers, female has made purchases than male. This can be one way to help visualize the loyalty of customer based on gender.

A picture containing text, screenshot, line, plot

Description automatically generated

Fig 11: top 4% of user who made frequent purchases during the high active months (september october and november)

## Feature Engineering

### Feature 1 – Mean of unique items purchased.

Profile – User, item

The number of unique items purchased by every user is calculated. It is further grouped with merchant id and user id. Lastly taking the average of the purchase counts for each user, new information is obtained. This new feature stored in ‘avg\_unique\_items\_per\_user’ can provide valuable insight into customer behavior and preferences.

### Feature 2 – Total actions taken on each merchant page by each user

Profile – User, Merchant with timestamp aggregation

From the e-commerce data, ‘total\_actions\_per\_user\_merchant’ column is extracted by taking a count of all actions of a user on a merchant. It helps in forming customer segmentation, developing recommendation system.

### Feature 3 - Proportion of purchases made by females within a 30-day period

Profile – User

The 'proportion of repeat buy days' metric is calculated by comparing the difference between two consecutive purchases per user with a threshold of 30 days. The number of users with at least one repeated buyer day is determined, and this is expressed as a proportion of the total number of users after grouping, but before applying the 30-day threshold. It provides valuable insight into customer retention and loyalty to develop targeted marketing strategy.

### Feature 4 – clicks on items during the highest active months

Associated with Profile – Items

The variable 'item\_clicked\_in\_oct\_nov' provides information on whether an item was clicked during the months of September, October, or November, which are the top three months of high user activity. This variable can be used as an input in machine learning algorithms to forecast user behavior and for further analysis. The existence of this feature indicates that items clicked during these months are more likely to be purchased. It can be useful in predicting user behavior during high activity periods and also give insights into the item’s popularity or in demand during those periods

### Feature 6 – Users category interest

Associated with Profile – Categories and User

This feature is based on, the count of unique categories that a user has made purchases in represented on column ’num\_categories’. It is generated by applying a lambda function to the 'category\_id' column, which counts the number of unique categories in that column. This feature may be useful for predicting user behavior and spending patterns, as users who purchase from a diverse range of categories may have different interests or habits than users who make purchases in a smaller number of categories.

### Feature 7 - Users brand interest

Associated with Profile – brand and User.

The number of clicks on every brand is feature engineered by exploding the brand and action type consecutively. It is then followed by performing necessary grouping, which likely corresponds to clicks. This information is stored in new column in the data frame called ‘brand\_clicks’. If a user has never clicked on a brand, then it will be tagged to 0. This feature is useful in predicting user preferences for particular brands and could provide insights into which brands are most popular or in demand among users.

### Feature 8 – Purchase Ratio

By utilizing the 'action\_type' and 'user' information, the process of Feature Engineering calculates the proportion of total actions taken by a user that resulted in a purchase. The result is captured in purchase\_ ratio column and is used in the model to predict the future trajectory of a category.

### Feature 9 – Marking users who purchase top brands.

Top brands are derived in the data visualization section of preprocessing. It is used the tag a user with a Boolean value in ‘users\_purchased\_top\_brands’ indicating the user's involvement with the top brands. This can be used to deduct, the probability of users' future purchases, loyalty towards top brands, and interest in other high valued brands.

### Feature 10 - Average days between purchases by a user on a category

The unique combination of user id and timestamp is further grouped by category\_id to calculate the average days between purchases by a user on category. It is valuable in making an analysis to get insight on user's purchase behavior and is capable to improving models performance.

## Dataset statistics and feature Ranking

### Statistical Summary

It is a data analysis tool used to summarize, describe the data provided. It gives information about outliers, trends, patterns etc. It helps in drawing conclusion about the relationship between the variables. Some variables include mean, median, mode and dispersion of the data. It is a concise distribution information and variability of the data set. The key characteristics of the data that can be used to better understand its properties and make informed decisions about it.

The below shown is the statistical summary of the data after feature engineering.

A picture containing text, pattern

Description automatically generated

Fig12: Statistical Summary

Feature Ranking using co-relation Matrix.

When using a correlation matrix, the primary focus should be in examining the correlation between the independent and dependent variables. The correlation matrix is used to identify the degree of association between these variables when plotted. The columns that have a high correlation value, irrespective of the sign (positive or negative), are the most important features that are associated with the target variable. It can range between +1 to -1. +1 describe the two variables are directly proportional. Example, in a dataset containing housing process, the number of bedrooms might have a positive co-relation with the price of the house as higher the number of bedroom higher will be the house price. -1 regards to inversely proportional. Example in gas prices dataset, the distance from the downtown, might have a negative co-relation as the gas prices tends to be higher when pump is located closer to the city. The value with 0 indicates there is no relationship between two variables. As discussed, the matrix helps us identify the relationship between variables, understand the underlying data, identify probable problems when we use variables which are highly co-related.

In the correlation matrix below, the target variable is associated with the label, and by focusing on the row and column associated with the dependent variable, we can identify the important features that we can use in feature engineering.

The below shown is the result of the analysis.

A picture containing text, screenshot, number

Description automatically generated

Fig 13: Correlation matrix

A screenshot of a computer

Description automatically generated with low confidence

Fig 14: Feature importance using co-relation.

Understanding of PCA for the data and Feature reduction.

Principle component analysis is a statistical technique specializing in dimensionality reduction. It transforms the large set of variables into a smaller set of un-corelated values called principal components. Principle components helps in summarizing the data. We used all the features created using feature engineering to perform feature reduction. All the values are scaled to be sure that the features are compared on the same scale. With n\_components set to 3, a 3D scatter plot is created with x, y, z representing first second and third principal component respectively. The color in the graph represents the class of the sample. Additionally, the variance ratio gives us the information on the amount of information capture by the components (PC1 is having highest variance). A heatmap is created using the principle components to determine the information captured by each component using which feature reduction can be analyzed. Rows represent components and cell represent each feature. Code with higher positive or higher negative are the feature which has highest weights. Other values not belonging to these criteria has lower weights and hence can be reduced.

A picture containing diagram, design

Description automatically generated

Fig 15: Principle Component Analysis

[0.29340068 0.16108939 0.14183201]

Fig 16: Variance ratio of each component(PC1 has higher variance)

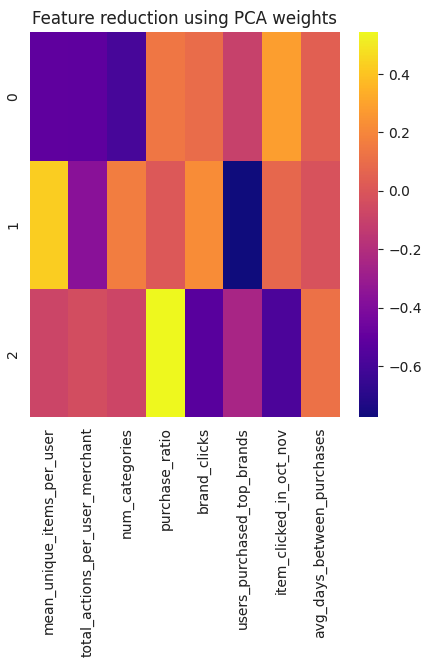


Fig17: Feature reduction analysis using PCA weights

## Prediction Model.

Using the information in the co-relation matrix, we can remove one of the high co-related features. We have considered the co-efficient greater than 0.8 – item\_clicked\_in\_oct\_nov and num\_categories as the protentional features to drop from the data frame. We have consolidated the data frame with the remaining feature and trained the model.

### Non- parameter technique

#### Nearest neighbors

KNN is a supervised learning algorithm from scikit-learn library this is used mainly in classification and regression tasks. The basic idea begins with finding k nearest data points in the training test, to be given to testing data and use their majority vote as the predictable label for test data point. The k and distance measure are hyper parameters that must be tuned to get high performance.

For choosing the value of K, we can use techniques like cross-validation to select the best value of K that maximizes the performance on the validation set. As for the distance measure, we tried different measures such as Euclidean, Manhattan, or Minkowski distance. We have used Euclidean distance measure with k as 5 from KNN classifier to get high performance.

#### Parzen windows

Parzen window is an nonparametric density estimation technique. The idea revolves around estimating the probability function centered at each data point using a window function. The density functions are aggregated to get global probability density function. The parzen windows is usually a kernel function like gaussian kernel, which assign weights to the nearest data points based on distance between the point and central window. The bandwidth (used 1.0) of this function defines the smoothness of the density estimate. The larger the value higher smoothness expected. To use parzen windows for calculation by comparing the density estimation to derive a decision boundary space.

To use non-parametric techniques like nearest neighbors and Parzen windows for classification, we first need to split our consolidated dataset into a training set and a testing set. We can then train our models on the training set and evaluate their performance on the testing set. Let's assume that we have split our dataset into a training set train\_df and a testing set test\_df. We will use the scikit-learn library to implement the nearest neighbors and Parzen windows models using gaussian kernel with density bandwidth bandwidth of 1.0. The trained parzen window model is used calculate log probabilities of the test data using score samples method which are thresholder at 0. Finally, we calculate accuracy. Accuracy is shown below.

*Neural Network*

Neural network is a machine learning algorithm inspired by the structure of neuron. It is deigned to recognize patter and make predictions based on AI. It has interconnectable neuron which received input from previous layers and apply mathematical operations on them to result a next layer. This is call back propagation. The network adjusts weights of neuron to improve accuracy.

### Model Evaluation

Performance Evaluation We evaluated the performance of three classifiers: K-Nearest Neighbors, Parzen Window, and Logistic Regression. The evaluation was based on several performance metrics, including accuracy, precision, recall, F1 score, ROC AUC score, and confusion matrix.

#### K-Nearest Neighbors Classifier

The model evaluation results indicate that it works expectationally. The K-Nearest Neighbors classifier was trained using the training dataset and tested using the testing dataset. The classifier was configured to use 5 nearest neighbors and the Euclidean distance metric. The classifier achieved an accuracy of 0.99 and a ROC AUC score of 0.99. The confusion matrix revealed that the classifier correctly classified all samples of class 1 and class 2, but misclassified one sample of class 0 as class 1.

A screenshot of a computer

Description automatically generated with medium confidence

Fig 18: Result of KNN Accuracy k=5 and Euclidean Distance

Parzen Window Classifier

The Parzen Window classifier was trained using the training dataset and tested using the testing dataset. The classifier was configured to use a Gaussian kernel with a bandwidth of 1.0. The classifier achieved an accuracy of 0.99. The confusion matrix revealed that the classifier correctly classified all samples of class 1. There are some misclassifications in other class but they are less in number.

A screenshot of a computer

Description automatically generated with medium confidence

Fig 19: Result of Parzen window kernel type 🡪 gaussian k=5 and bandwidth 🡪 1.0

*Neural Network*

The implemented neural network model is designed for multi class classification. The input is scaled and convert labels into one hot encoding. Model architecture is defined by two hidden layers of 32 and 16 neurons with “relu” activation function. The output layer with “softmax” activation function gives class probabilities. Initially model is trained on training data using fit() followed by predicting the labels using predict(). Finally convert back the predicted labels from one-hot-encoding to original label. Evaluate the model using ROC (Receiver operating curve) and AUC(Area under curve). The prior is plotted on graph and the latter is computed.

A picture containing text, line, plot, diagram

Description automatically generated

Fig 20: ROC and AUC

#### Logistic Regression Classifier

The Logistic Regression classifier was trained using the training dataset and tested using the testing dataset. The classifier was configured to use L2 regularization with a regularization parameter of 1.0. The classifier achieved an accuracy of 0.98 and a ROC AUC score of 0.99. The confusion matrix revealed that the classifier correctly classified all samples of class 1 and class 2, but misclassified one sample of class 0 as class 1.

#### Comparison Result

Overall, the K-Nearest Neighbors and Parzen Window classifier achieved similar accuracy and ROC AUC score, indicating that they are effective at classifying the samples in the dataset. The K-Nearest Neighbors (8minutes) runs faster than Parzen Window classifier (12 minutes) with easier implementation work and being less expensive.

#### Result

The graphical representation, usually a Receiver Operating Characteristic (ROC) curve, shows the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for different classification thresholds. It plots the TPR on the y-axis and the FPR on the x-axis.

The area under the ROC curve (AUC) is a commonly used metric to evaluate the performance of binary classification models. It quantifies the classifier's ability to distinguish between positive and negative samples. An AUC of 1.0 indicates perfect classification, while an AUC of 0.5 indicates a classifier that performs no better than random guessing.

In general, a good classifier should have an ROC curve that is as close to the top-left corner (TPR=1.0, FPR=0.0) as possible, indicating high TPR and low FPR across all possible classification thresholds. A classifier with an AUC of 0.5 is essentially random, while an AUC of 1.0 represents a perfect classifier.

## Discussion

### k-Nearest Neighbors

Strengths

Simple to understand and interpret Does not make assumptions about the underlying distribution of the data Can handle multi-class classification problems Does not require training time

Weaknesses

Computationally expensive for large datasets Sensitive to the choice of distance metric and number of neighbors May produce poor results if the dataset has a high dimensionality Does not handle missing values well Improvements:

Use distance weighting to give more importance to nearby points Use dimensionality reduction techniques, such as PCA or t-SNE, to reduce the dimensionality of the dataset Use outlier detection techniques to identify and remove noisy data points that may negatively affect the performance of the model Use cross-validation to tune the hyperparameters of the model and optimize the choice of distance metric and number of neighbors.

### Parzen Window

Strengths:

Can handle both continuous and discrete data Does not make assumptions about the underlying distribution of the data Can handle multi-class classification problems Does not require training time

Weaknesses:

Computationally expensive for large datasets Sensitive to the choice of kernel and bandwidth may produce poor results if the dataset has a high dimensionality Does not handle missing values well Improvements:

Use a different kernel, such as a polynomial or exponential kernel, to better capture the underlying distribution of the data Use cross-validation to tune the bandwidth of the kernel and optimize the performance of the model Use dimensionality reduction techniques, such as PCA or t-SNE, to reduce the dimensionality of the dataset and improve the performance of the model Use outlier detection techniques to identify and remove noisy data points that may negatively affect the performance of the model Overall, the best-performing technique will depend on the specific characteristics of the dataset and the problem at hand. It is important to evaluate the performance of each technique using appropriate performance metrics and to select the technique that best balances accuracy, interpretability, and computational efficiency. Additionally, using ensemble techniques or combining multiple techniques may provide further improvements in classification performance.

#### Neural Network

Strengths:

The model is flexible and can handle non-linear data relationships between input and output. It has better capability to learn from large amount of data. They have capability to perform classification, regression, speech, and image recognition. They have capability to automatically extract data from the raw data and can be implemented on both supervised and unsupervised learning tasks.

Weakness:

Computing neural network can become expensive and is much vulnerable to overfitting where model performs poorly on unseen data. It is often tagged as black boxes that they are difficult to analysis the way decision is made. They are very sensitive to hyper parameters which adds difficulty to train them correctly making it more challenging.

### Recommendation for e-commerce platform for shopping one-time buyers

* Use information from brand clicks and users who has purchased top brands to understand popularity and market brand engagement through social media, influencer marketing.
* Use information from mean unique items to sell (complementary samples) of a different product to an existing customer to create brand loyalty thus majorly avoid one-time buyers.
* Using average days between purchases remind customer through emails to restock their product. Having personalized tailored emails will encourage repeated purchase (increase loyalty) and and reduce one-time buyers.

## References

1. <https://www.geeksforgeeks.org/implementing-pca-in-python-with-scikit-learn/>
2. <https://www.udemy.com/course/python-for-data-science-and-machine-learning-bootcamp/learn/lecture/5733554#overview>
3. <https://www.geeksforgeeks.org/create-a-correlation-matrix-using-python/>
4. <https://medium.com/geekculture/feature-selection-in-machine-learning-correlation-matrix-univariate-testing-rfecv-1186168fac12>