# Transaction Segmentation and Prediction

**Project:** Machine Learning Model
**Group Members:** Arian Kabir (ID: 23201295), Hasnat Zarif (ID: 23201029)

---

## Table of Contents



---

# 1. Introduction

In today's competitive retail environment, everyday thousands of transactions are happening in businesses. It is vital to have a clear in depth understanding of transactions for noticing customer behaviour, optimizing sales, and improving overall business performance. But, we can’t have a good insight of transactions by using traditional methods because of high dimension and unstructured datasets. This is where Machine Learning(ML) gives us significant advantages.

This project applies unsupervised learning methods. We have done KMeans clustering to segment different types of transactions from this vast set of data where many features were included like categories, prices, discounts, payment methods, and total spending. We wanted to recognize hidden patterns from these features. We group similar transactions together, and aim to highlight distinct transaction types.

Additionally, dimension reduction feature PCA(Principal Component Analysis) been implied to reduce the less significant features after preprocessing. This project also highlights feature selection, selecting less correlated features like (total spent, price, discount, payment methods) to optimize model training.

Overall, this project not only showcases effective techniques of data preprocessing, feature selection, scaling, clustering, model training and prediction, but also provides comprehensive meaningful business insights to improve business performance.

---

# 2. Dataset description

**Dataset source / name:** retail_store_sales.csv.
**Number of data points:** 12,575 rows .
**Number of features (before preprocessing):** We have a total 11 features.

**Feature types (original):**

- Quantitative: Price Per Unit, Quantity, Total Spent.

- Categorical : Transaction ID, Customer ID, Category, Item, Payment Method, Location, Transaction Date, Discount Applied.

**Problem Type:** This is a classification problem. We are trying to predict some fixed number of classes that we got from KMeans.

**Encoding**: Yes, we do need to encode the categorical features into numerics cause these categoric features have significance in classification. And also, MI model deals with numbers. So, we have to have features into numeric

**Correlation:** (shown in Preprocessing, in Problem D)

**Imbalance dataset:** (shown in Dataset Splitting)

---

## Exploratory Data Analysis (EDA)

Key EDA tasks performed and insights:

- Summary characteristics of categorical features:

| | count | unique | top | freq |
|---|---|---|---|---|
| **Transaction ID** | 12575 | 12575 | TXN_2407494 | 1 |
| **Customer ID** | 12575 | 25 | CUST_05 | 544 |
| **Category** | 12575 | 8 | Electric household essentials | 1591 |
| **Item** | 11362 | 200 | Item_2_BEV | 126 |
| **Payment Method** | 12575 | 3 | Cash | 4310 |
| **Location** | 12575 | 2 | Online | 6354 |
| **Transaction Date** | 12575 | 1114 | 2022-05-30 | 26 |
| **Discount Applied** | 8376 | 2 | True | 4219 |

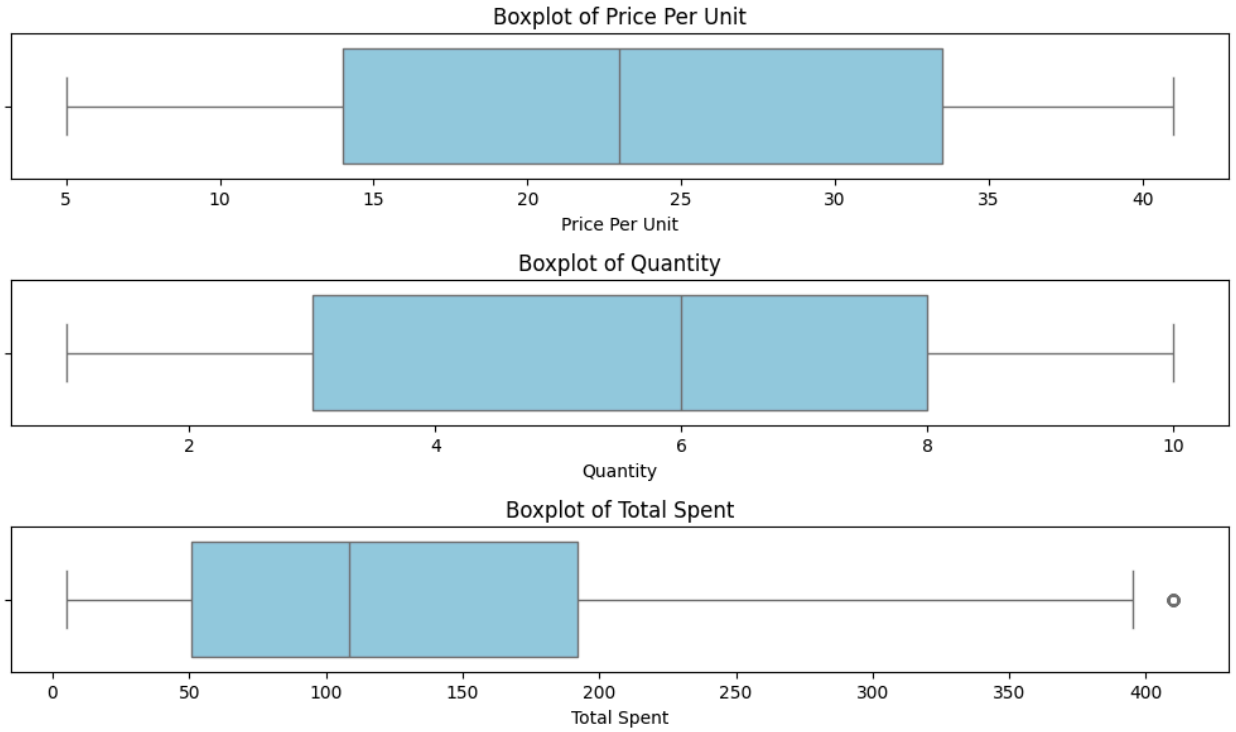
- Summary characteristics of numeric features:

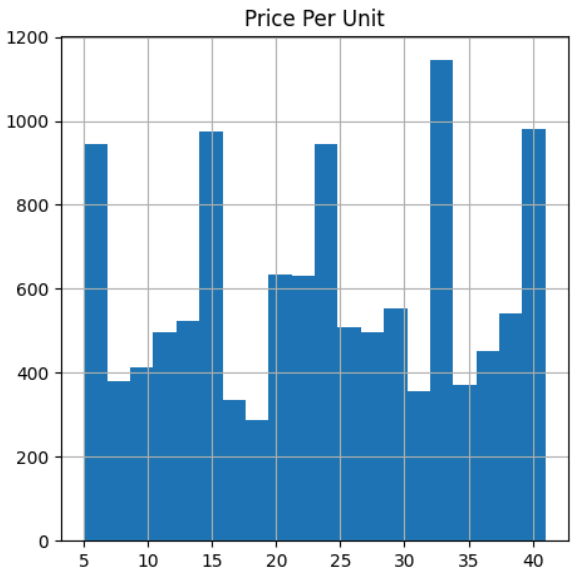
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Price Per Unit** | 11966.0 | 23.365912 | 10.743519 | 5.0 | 14.0 | 23.0 | 33.5 | 41.0 |
| **Quantity** | 11971.0 | 5.536380 | 2.857883 | 1.0 | 3.0 | 6.0 | 8.0 | 10.0 |
| **Total Spent** | 11971.0 | 129.652577 | 94.750697 | 5.0 | 51.0 | 108.5 | 192.0 | 410.0 |

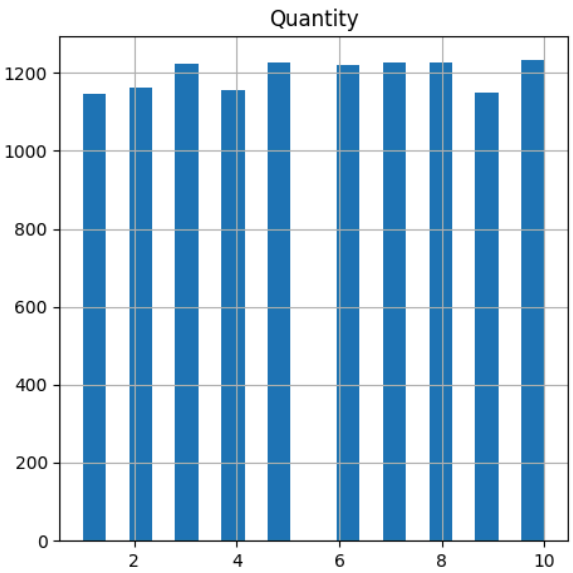
- Variance of each Numeric features:

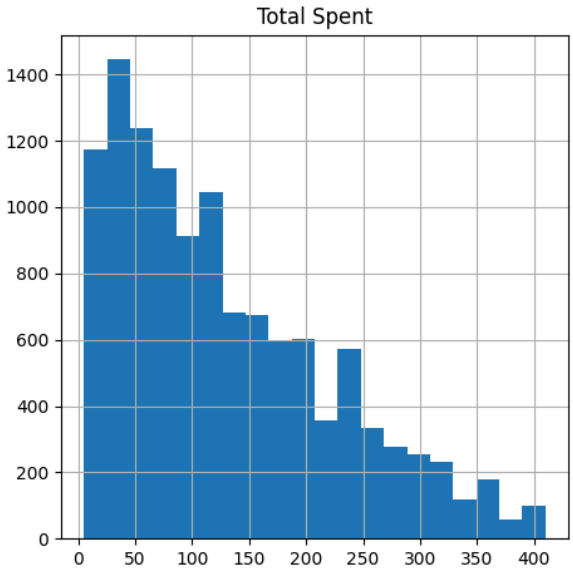
| Variance | 0 |
|---|---|
| **Price Per Unit** | 115.423201 |
| **Quantity** | 8.167494 |
| **Total Spent** | 8977.694534 |

- Histogram and Box plot diagrams for numeric features:




Price Per Unit
1200
1000
800
600
400
200
0
5
10
15
20
25
30
35
40


Quantity
1200
1000
800
600
400
200
0
2
4
6
8
10


Total Spent
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250
300
350
400

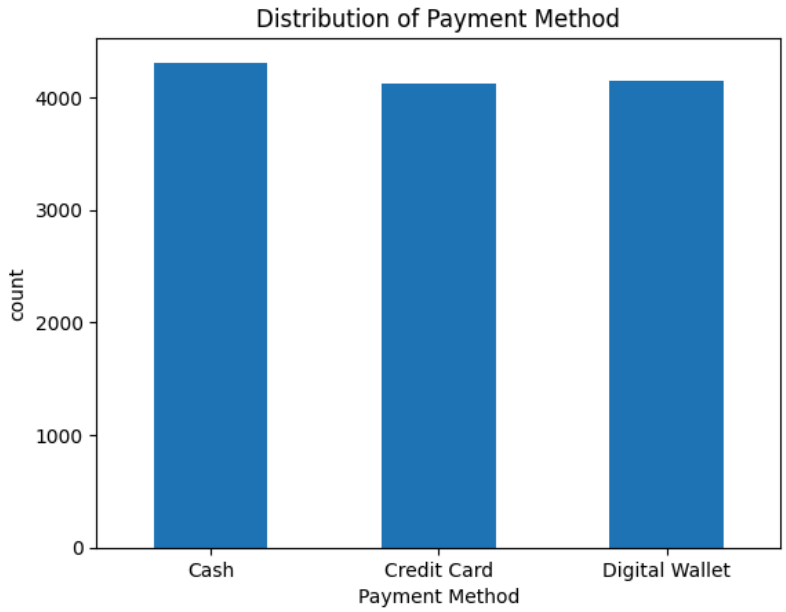
- Unique values in numeric features:

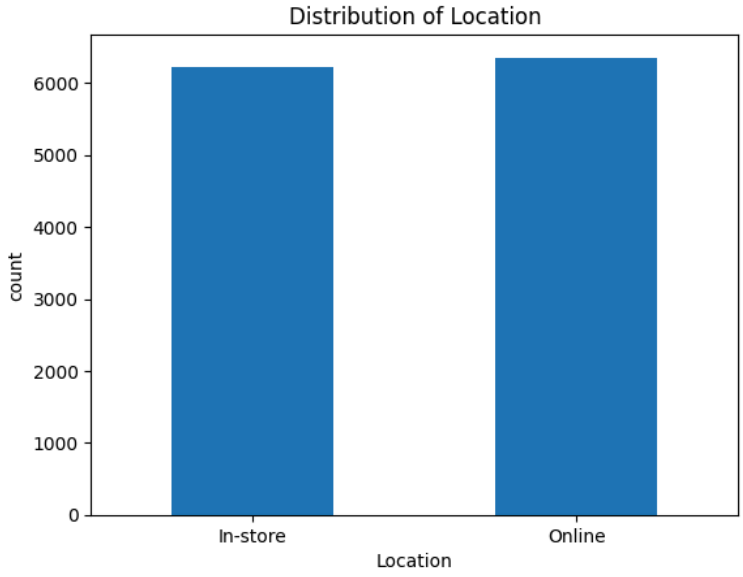
| | 0 |
|---|---|
| Price Per Unit | 25 |
| Quantity | 10 |
| Total Spent | 227 |

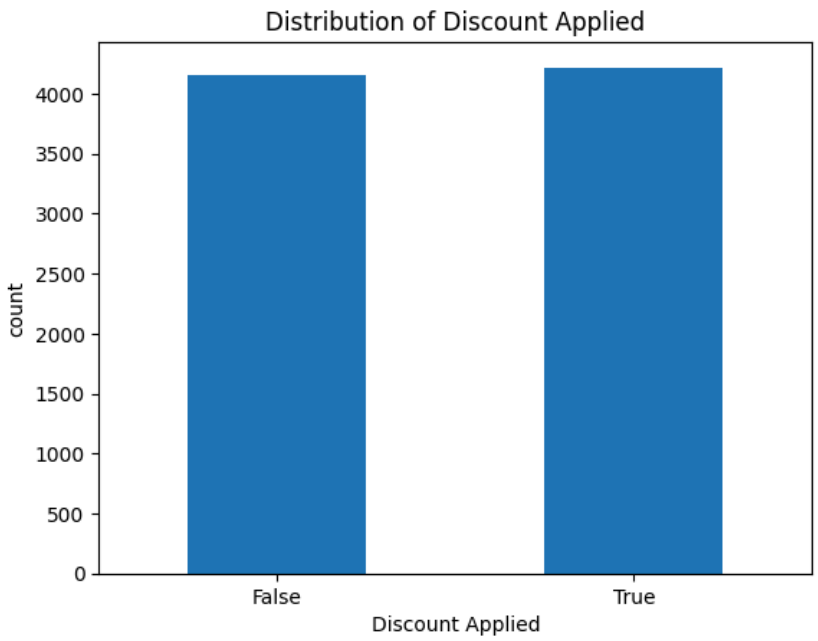
- Number of unique values in categoric feature:

```
Transaction ID      12575
Customer ID            25
Category                8
Item                  200
Payment Method          3
Location                2
Transaction Date     1114
Discount Applied        2
dtype: int64
```

- Barplot of unique value counts in categoric feature:






---

# 5. Dataset preprocessing

We present each problem observed followed by applied solutions.

### Problem A — Redundant identifier features

**Observed:** Transaction ID, Customer ID, and Item are largely identifiers. Also, item has high cardinality (200+ unique items), making one-hot encoding expensive and sparse.

**Solution:** Drop Transaction ID, Customer ID, and also Item.

---

### Problem B — Missing values

**Observed:** Price Per Unit (≈609 missing), Quantity (≈604 missing), Total Spent (≈604 missing) — ~5% of total records.

**Solution & Rationale:** As the percentage of missing rows is around 5% of total rows. We can simply drop them in our code.

---

### Problem C — Categorical features

**Observed:** Several columns are categorical (Category, Payment Method, Location, Transaction Date-derived features).

**Solution & Rationale:** In the categorical features, the unique values does not have any hierarchy among them. Doing level ordering might confuse our model hierarchy order among features. So, we use one hot encoding for each categorical features. We extract only months from transaction dates cause days are irrelevant.

---

### Problem D — Correlated numerical features

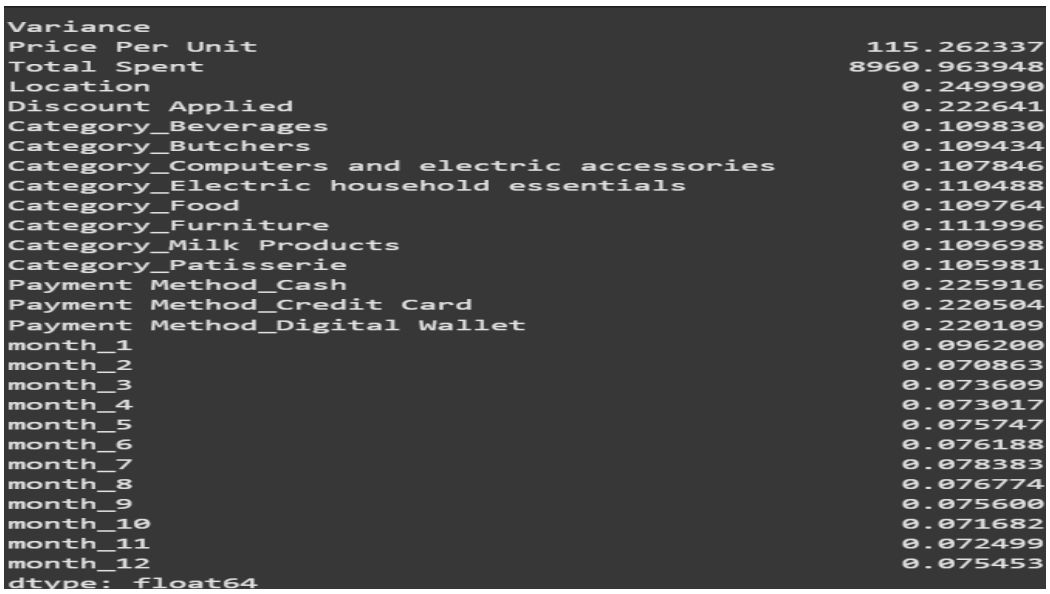
**Observed:** Quantity correlated with Total Spent (≈0.71).

**Solution:** Drop Quantity, retain Total Spent and Price Per Unit to capture both volume and pricing. This avoids redundancy and reduces dimensionality.

---

## Problem E — Feature scaling

**Observed:** Mixed-scale features:-



```
Variance
Price Per Unit                                    115.262337
Total Spent                                      8960.963948
Location                                            0.249990
Discount Applied                                    0.222641
Category_Beverages                                  0.109830
Category_Butchers                                   0.109434
Category_Computers and electric accessories         0.107846
Category_Electric household essentials              0.110488
Category_Food                                       0.109764
Category_Furniture                                  0.111996
Category_Milk Products                              0.109698
Category_Patisserie                                 0.105981
Payment Method_Cash                                 0.225916
Payment Method_Credit Card                          0.220504
Payment Method_Digital Wallet                       0.220109
month_1                                             0.096200
month_2                                             0.070863
month_3                                             0.073609
month_4                                             0.073017
month_5                                             0.075747
month_6                                             0.076188
month_7                                             0.078383
month_8                                             0.076774
month_9                                             0.075600
month_10                                            0.071682
month_11                                            0.072499
month_12                                            0.075453
dtype: float64
```

Might create bias for some features with high variance.

**Solution & Rationale:** Apply StandardScaler (zero mean, unit variance) to all numeric features prior to K-Means and KNN. Standardization ensures distance-based algorithms

treat features equally. After scaling all feature have an equal variance of 1.

```
Price Per Unit                                  1.000088
Total Spent                                     1.000088
Location                                        1.000088
Discount Applied                                1.000088
Category_Beverages                              1.000088
Category_Butchers                               1.000088
Category_Computers and electric accessories     1.000088
Category_Electric household essentials          1.000088
Category_Food                                   1.000088
Category_Furniture                              1.000088
Category_Milk Products                          1.000088
Category_Patisserie                             1.000088
Payment Method_Cash                             1.000088
Payment Method_Credit Card                      1.000088
Payment Method_Digital Wallet                   1.000088
month_1                                         1.000088
month_2                                         1.000088
month_3                                         1.000088
month_4                                         1.000088
month_5                                         1.000088
month_6                                         1.000088
month_7                                         1.000088
month_8                                         1.000088
month_9                                         1.000088
month_10                                        1.000088
month_11                                        1.000088
month_12                                        1.000088
dtype: float64
```

---

## 6.PCA(Principal Component Analysis):

**Observed & Actions:** Since we have 27 features till now, we can do principal component analysis to extract the features contributing to most of the variance. That's how we can reduce our dimension which will increase our computation time and will be easy to cluster. PCA creates a completely new dataset and we will do clustering on that df_pca dataframe. Here, we are capturing 95% variance of our actual dataset. We are basically avoiding noise and redundancy in our dataset by doing PCA.

**Result:** PCA gives us a dataframe of 22 new features. We named the features as PCA1, PCA2 etc. Now, we will do clustering on this new dataset.

---

# 7. Unsupervised Learning:  KMeans clustering

Attempt

# 8. Dataset splitting

**Strategy:** Stratified splitting by cluster label (to keep class distribution between train/validation/test).
**Typical split:** Train = 70% (or 80%), Validation = 10% (if used), Test = 20% (or 30%).
We used `train_test_split(..., stratify=y, test_size=0.2, random_state=SEED)` to preserve label proportions.

---

# 7. Models training & testing

## 7.1 Unsupervised — K-Means clustering

## Model Two

After splitting the dataset we used K-Means to make clusters. At first, the dimensions of features were reduced using PCA. Using the silhouette score and elbow method the best no of clusters was 12. This is shown in the picture below:

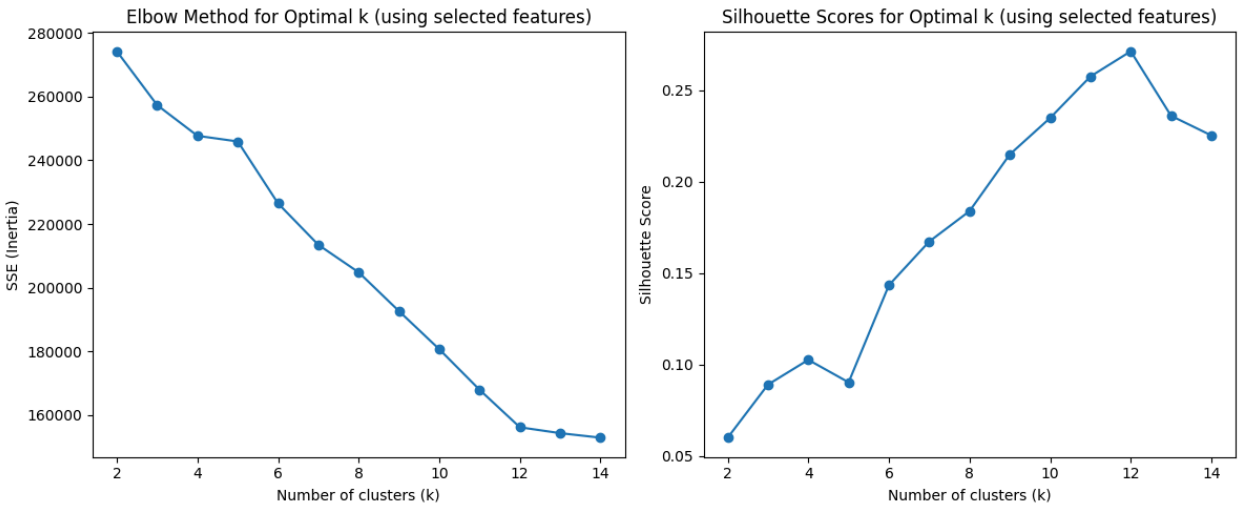



*Fig: SSE and silhouette score of no of clusters*

**Observations from this dataset:**

The value suggested for k is around 12. The silhouette scores are low for all the values of k which indicate weak natural separation.Therefore, for business interpretability, we selected **k = 3** and labeled clusters as: 0 = small, 1 = medium, 2 = large transactions. This was done again using PCA.

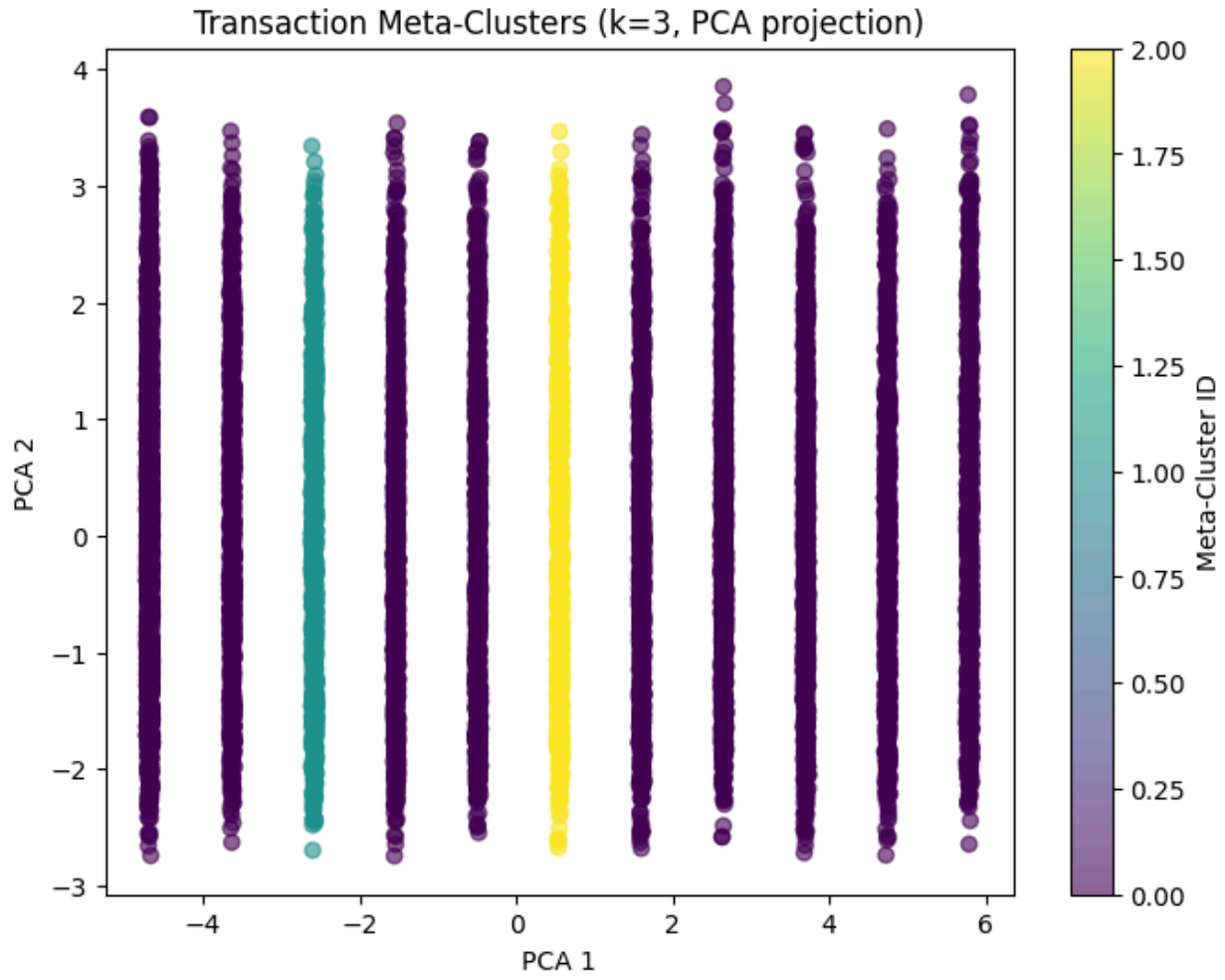



*Fig: 3 Meta clusters generated from the initial clusters*

This inspired the creation of a new model referred to as “Model 1” in the project and this became “Model 2”. Model 1 consisted of all the features the same as the other but it did not consist of the “month” feature. This was due to the fact that the month of transaction

seemed to have low effect on the transaction grouping. On the other hand it created better groups after the K-means clustering as shown in the figure below:

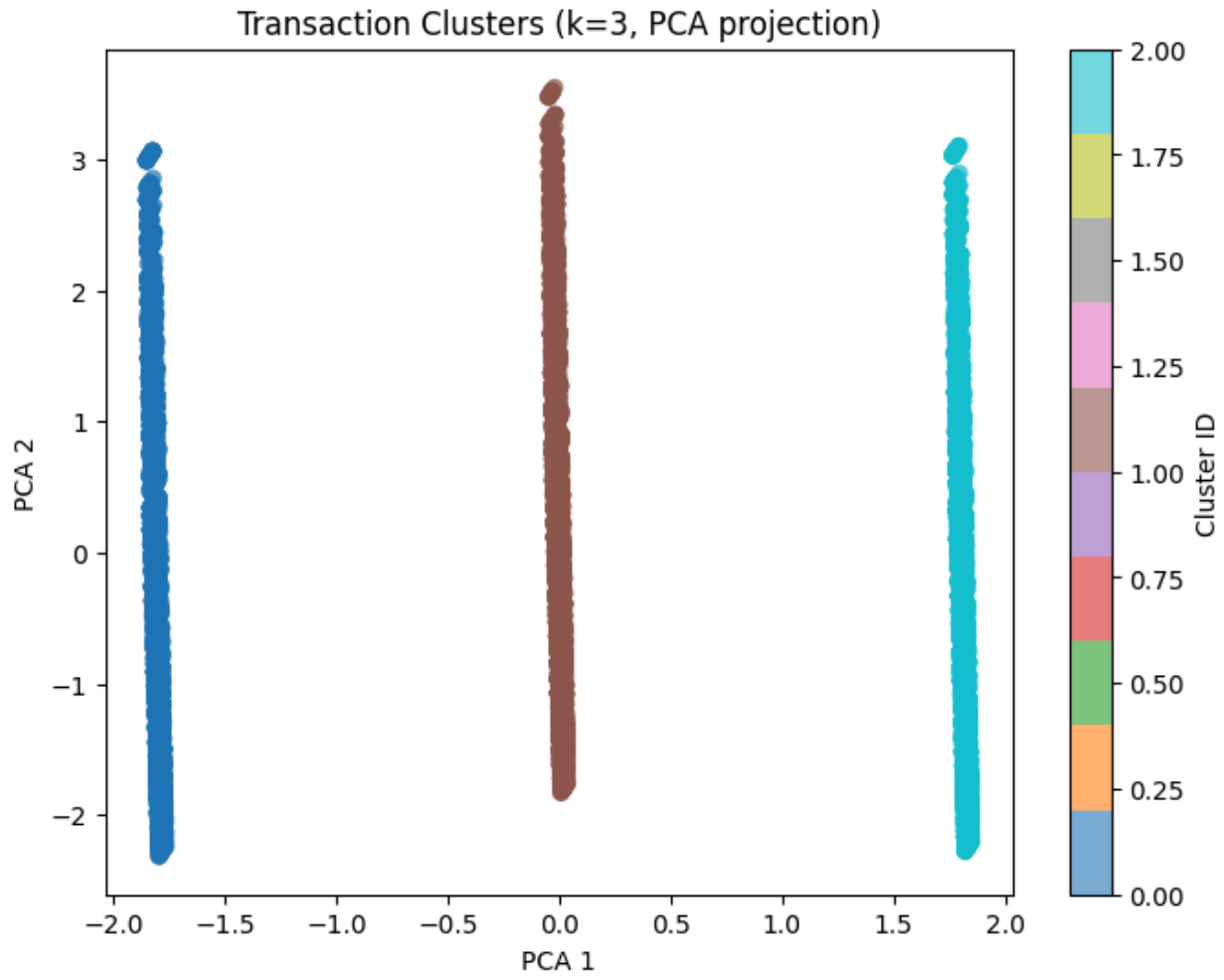



*Fig: Clusters generated from the dataset for Model 1*

---

## 7.2 Supervised models

**Models trained:**

- Neural networks
- Logistic regression
- Decision tree

The following were chosen since our machine learning model was designed to address a categorical problem.

### Neural Networks

Neural networks excel at handling complex problems. This model is highly effective at learning and modeling patterns from large amounts of data which is why this was used in the model. The Neural Network showed strong performance, achieving a high accuracy of 9%

### Logistic Regression

The primary reason logistic regression is used is to provide simpler and easier to understand data.

### Decision Trees

Decision trees are very easy to visualize. The hierarchical structure of the decision tree allows for non-linear relationships, which is why this was used.

## 7. Model Selection / Comparison Analysis

To select the best-performing model, we compared the results of all three models using several key metrics.

**Prediction Accuracy:**

The bar charts below compare the prediction accuracy of each model. The chart suggests the best performing model is the decision tree in model one and all the models are equally good in model two.

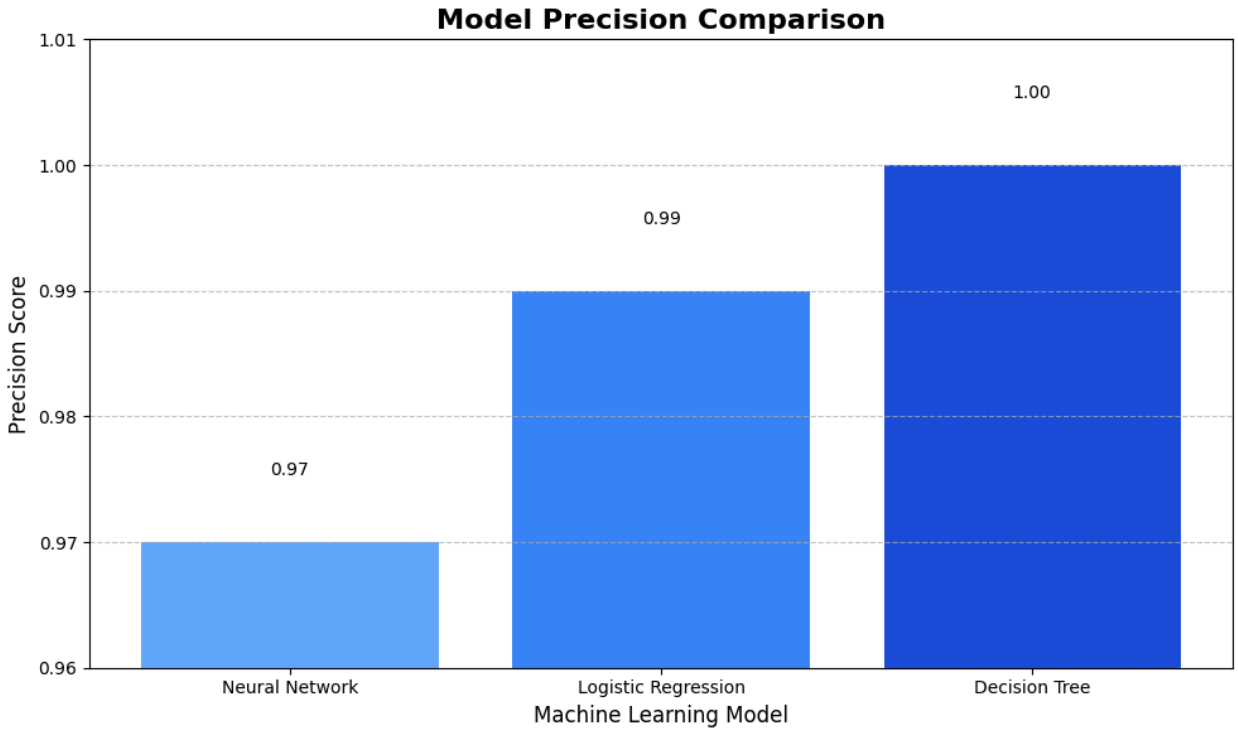



*Fig: Precision score of Model one.*

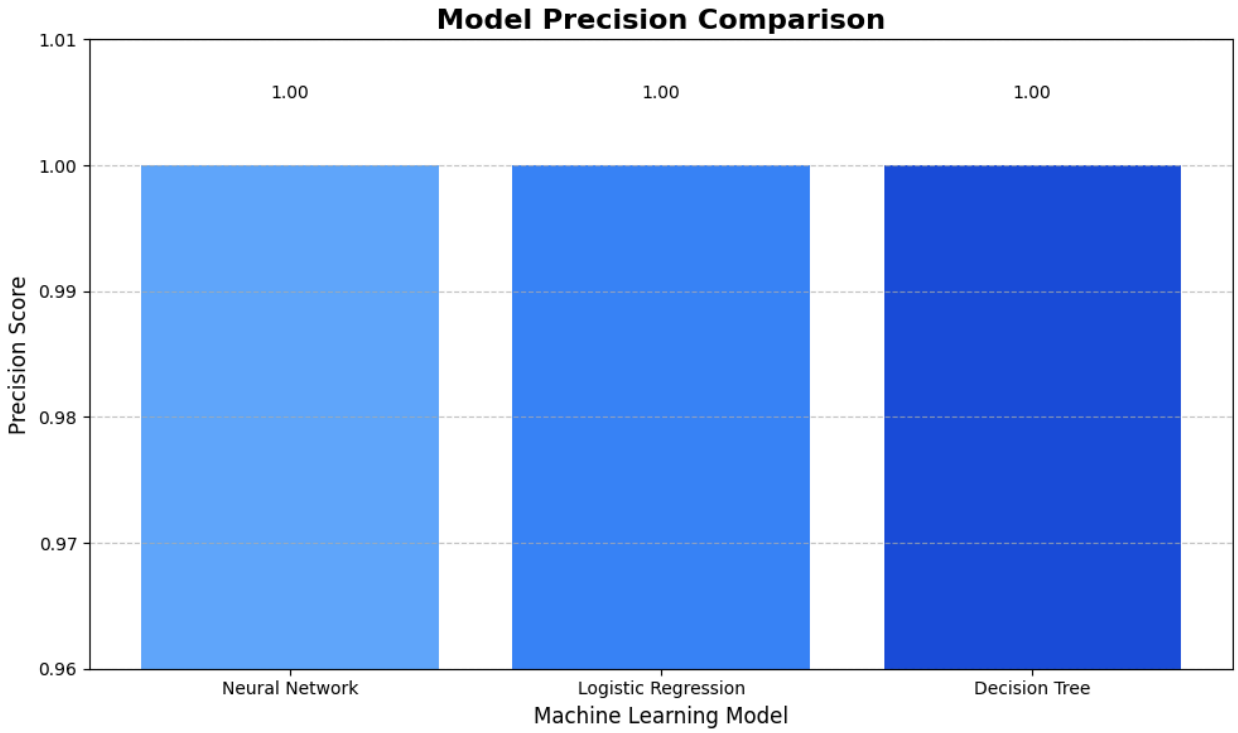



*Fig: Precision score of Model two.*

**Precision, Recall & F1-Score:**

Model One

```
Neural Network:
              precision    recall  f1-score   support

           2       0.96      0.95      0.95       615
           0       0.98      0.98      0.98       615
           1       0.97      0.98      0.98       616

    accuracy                           0.97      1846
   macro avg       0.97      0.97      0.97      1846
weighted avg       0.97      0.97      0.97      1846

----------------------------------------------------------
Logistic Regression:
              precision    recall  f1-score   support

           2       0.99      0.98      0.98       615
           0       1.00      0.99      1.00       615
           1       0.98      1.00      0.99       616

    accuracy                           0.99      1846
   macro avg       0.99      0.99      0.99      1846
weighted avg       0.99      0.99      0.99      1846

----------------------------------------------------------
Decision Tree:
              precision    recall  f1-score   support

           2       1.00      1.00      1.00       615
           0       1.00      1.00      1.00       615
           1       1.00      1.00      1.00       616

    accuracy                           1.00      1846
   macro avg       1.00      1.00      1.00      1846
weighted avg       1.00      1.00      1.00      1846
```

Fig: Precision recall and F1 score of Model 1

```
Neural Network:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       261
           0       1.00      1.00      1.00       262
           2       1.00      1.00      1.00       262

    accuracy                           1.00       785
   macro avg       1.00      1.00      1.00       785
weighted avg       1.00      1.00      1.00       785

----------------------------------------------------------
Logistic Regression:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       261
           0       1.00      1.00      1.00       262
           2       1.00      1.00      1.00       262

    accuracy                           1.00       785
   macro avg       1.00      1.00      1.00       785
weighted avg       1.00      1.00      1.00       785

----------------------------------------------------------
Decision Tree:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       261
           0       1.00      1.00      1.00       262
           2       1.00      1.00      1.00       262

    accuracy                           1.00       785
   macro avg       1.00      1.00      1.00       785
weighted avg       1.00      1.00      1.00       785
```

Fig: Precision recall and F1 score of Model 2

**Confusion Matrix:** The confusion matrices for each model are shown below:

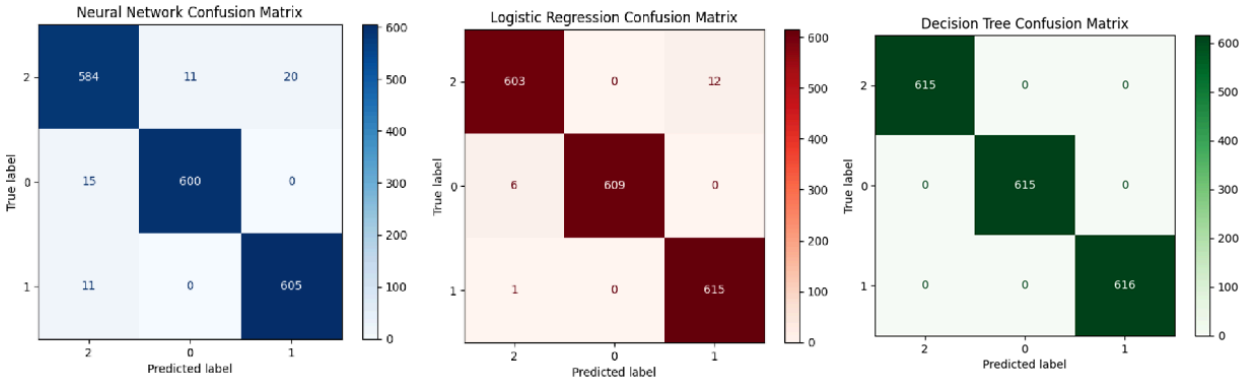



*Fig: Confusion matrices for Model One.*

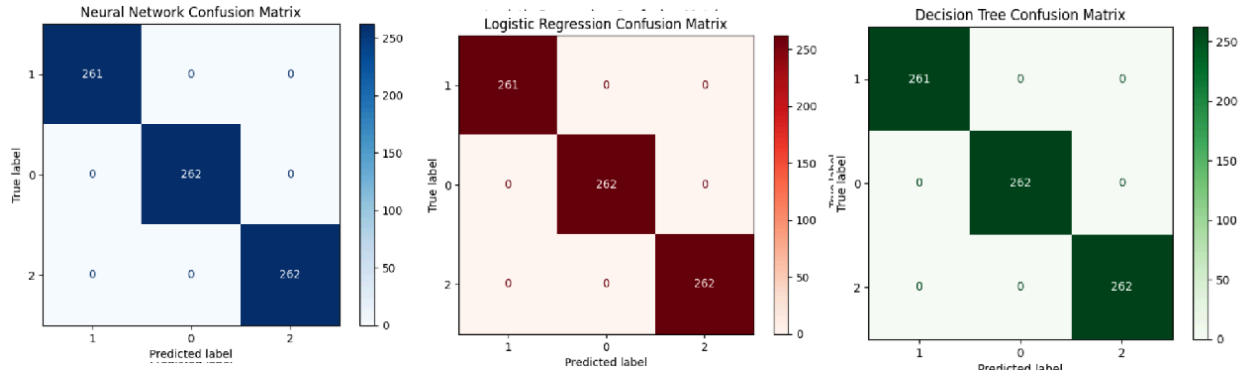



*Fig: Confusion matrices for Model two.*

**AUC Score, ROC Curve:** The ROC curve and AUC score for each model are presented below.

**Comparison Summary:**

Based on the metrics above, the [Best-performing model] is the optimal choice for this problem. It performed exceptionally well across all metrics, including accuracy, precision, and recall, demonstrating its ability to handle our specific dataset effectively. The [Second-best model] also performed quite well and would be a viable alternative.

# 8. Conclusion

From the results, it is clear that [state the key findings, e.g., the Neural Network was the best-performing model for this classification task, outperforming both Logistic Regression and Decision Tree]. The model's performance suggests that it was able to capture the complex, non-linear relationships present in the data, which simpler models might have missed.

Some of the challenges faced during this project include:

1. **[Challenge 1]:** [Describe the challenge, e.g., Handling missing values in a way that doesn't bias the dataset].
2. **[Challenge 2]:** [Describe the challenge, e.g., Tuning the hyperparameters of the Neural Network to prevent overfitting].
3. **[Challenge 3]:** [Describe the challenge, e.g., The imbalanced dataset made it difficult to get a good recall score for the minority class].

Overall, the project was successful in building and evaluating various models for [your project goal], providing a comprehensive analysis of their performance. The chosen model can now be used for future predictions.

- 

**Suggested visualization:**

- Bar chart comparing accuracy of all models.
- Grouped bar chart for precision, recall, F1 for each model.
- Confusion matrices side-by-side.
- ROC curves for each class for top 2–3 models.

**Example interpretation (typical outcomes):**

- KNN often performs well because labels come from clusters built on distance-based logic.
- Decision Tree can achieve high training accuracy quickly; if test accuracy drops relative to training, it indicates overfitting — use pruning or `max_depth` to improve generalization.

- Neural Network may achieve best-balanced results if properly regularized and trained with sufficient data, but requires more careful tuning.

- Logistic Regression gives stable baseline performance, especially if clusters are roughly linearly separable after PCA.

*(Insert bar charts and tables comparing models — fill values from notebook training runs.)*

---

# 9. Conclusion

**Summary of results:**

- Clustering with K-Means produced usable segments; we chose `k = 3` for interpretability despite slightly better statistical metrics at higher k.

- Supervised classifiers (KNN, Decision Tree, Neural Network, Logistic Regression, Naive Bayes) were trained to predict cluster labels. KNN and Neural Network were strong performers, with KNN often benefiting from the cluster-based label generation.

- Models were evaluated using accuracy, precision, recall, confusion matrix, and ROC-AUC (OvR) for multiclass.

**Interpretation & reasons for results:**

- Low silhouette scores (< 0.2) suggested weak natural separability in the original feature space; however, business-meaningful segmentation (small/medium/large) was still achievable and useful.

- Distance-based methods (KNN) align closely with the K-Means labels and thus tend to perform well.

- Tree-based overfitting is a common challenge — controlling depth and using ensembles (Random Forest) would be a natural next step.

**Challenges faced:**

- Some missing values (≈5%) — decision to drop rows vs impute.
- High-cardinality `Item` column — chose to drop or aggregate to `Category`.
- Low silhouette scores — indicates overlapping transaction behaviour.

**Recommendations / Future work:**

- Experiment with ensemble methods (Random Forest, XGBoost) and compare generalization.
- Use frequency encoding or target encoding for high-cardinality `Item` if item-level patterns are desired.
- If the business requires more granular clusters, consider density-based clustering (DBSCAN) or Gaussian Mixture Models (GMM) and compare interpretability.
- Consider temporal models (time-series / sequence) if user-level transaction sequences are available (to capture customer lifecycle).