ENHANCING CUSTOMER SEGMENTATION FOR PERSONALIZED MARKETING

(A CASE STUDY OF KJ MARKETING IN SRI LANKA)

TEAM INSOMNIACS
UNIVERSITY OF PERADENIYA
18/05/2024

Personalized Marketing Strategy Development for KJ Marketing
Customer Segmentation Based on Historical Sales Data

Introduction and Problem Statement

KJ Marketing, a prominent retail supermarket chain in Sri Lanka, seeks to enhance its marketing strategies by adopting a personalized approach tailored to individual customer preferences. The traditional marketing methods have proven ineffective with the current customer base, prompting the need for a more sophisticated strategy. This report focuses on developing an analytical method to identify and classify customer segments using historical sales data, including average monthly sales per customer. By analyzing data from 22 outlets across urban and suburban regions, we aim to classify new customers into one of the six identified segments, optimizing personalized marketing efforts. The implementation of this personalized strategy is expected to increase customer satisfaction, loyalty, and overall sales performance.

Preprocessing and Feature Engineering

The preprocessing and feature engineering steps undertaken in this project are crucial to prepare the dataset for analysis and modeling. Below, we detail each step to ensure clarity and reproducibility.

Methodologies implemented to address missing values, duplicates and outliers within the dataset

1. **Handling Missing Values**

   We drop rows with any missing values to ensure data integrity. This step is vital as missing values can lead to inaccuracies in model training and evaluation. Since the **proportion of data with missing values is very low**, removing these rows has a negligible impact on the overall dataset while significantly improving the reliability of our analysis and model performance.

2. **Converting Textual Representations to Numerical**

   We address **non-numeric entries in sales columns by converting them to numbers**.This function is applied to sales data columns to ensure all entries are numeric.

3. **Handling Invalid Entries/ Outliers**

We correct potential invalid entries in the 'cluster_category' by converting them to a valid range. Since the number of invalid entries is very low, this adjustment ensures data consistency without significantly impacting the overall dataset.

4. **Handling Duplicates**

   After identifying and dropping duplicates, we removed 2940 duplicate entries from the dataset. This step helps to ensure data integrity and prevents redundancy, allowing for more accurate analysis and modeling.

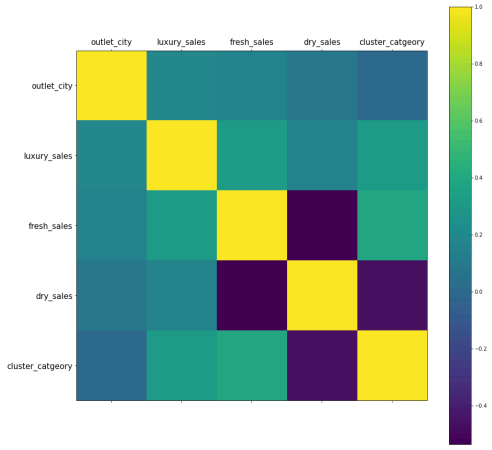
5. **Standardizing Text Data**

   To standardize city names, we convert all city names to uppercase. This ensures uniformity and prevents discrepancies caused by variations in text case. Additionally, **we account for and correct misspelled city names to maintain accuracy**.

Features We Chose for the above Task and Their Relevance to the Problem

Correlation between the Category and other features as well as the correlation between the features was calculated to get a better idea on which features were important.

	outlet_city	luxury_sales	fresh_sales	dry_sales	cluster_catgeory
outlet_city	1.000000	0.183292	0.157711	0.101975	0.013069
luxury_sales	0.183292	1.000000	0.311720	0.154922	0.308915
fresh_sales	0.157711	0.311720	1.000000	-0.535561	0.389056
dry_sales	0.101975	0.154922	-0.535561	1.000000	-0.467755
cluster_catgeory	0.013069	0.308915	0.389056	-0.467755	1.000000

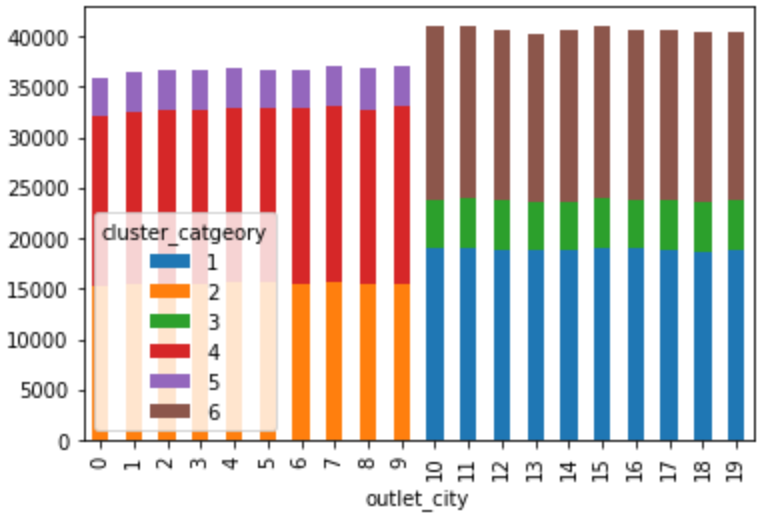


However since the Cities were encoded randomly the correlation values of the cities are not really relevant.

To identify the relationship between the cities and the cluster a bar plot was drawn considering the relative amounts of each cluster in each city.

Here we see that there is a clear relationship between the cluster categories and the outlet city. **The cities from 0-9 mostly have the clusters 5,4 and 2. The cities with the index 10-19 only have clusters 1,3 and 6.**

Since the model already has nice separation given the encoding used for the cities repremutation of the labels was not necessary.



KELANIYA:0
MORATUWA:1
WATTALA:2
HOMAGAMA:3
DEHIWALA-MOUNT
LAVINIA:4
PANADURA:5
KADUWELA:6
PELIYAGODA:7
KOTTE:8
NUWARA ELIYA:9
BATTICALOA:10
COLOMBO:11
JAFFNA:12
GAMPAHA:13
KALMUNAI:14
GALLE:15
KATUNAYAKE:16
NEGOMBO:17
TRINCOMALEE:18
KANDY:19

We drop the 'Customer_ID' column as it is not relevant for modeling, given that **customer IDs are assumed to be completely random**. Removing this column helps streamline the dataset and eliminates any noise that could potentially affect the model's performance.

Feature Scaling or Normalization

Standardization is applied to the sales data and city codes using **StandardScaler,**

$$\hat{x} = \frac{x - \mu}{\sigma}$$

to ensure all features contribute equally to the model. By standardizing the data, we enhance the model's ability to learn effectively from each feature without being biased by their different scales.
This **accelerated the convergence of optimization algorithms but also lead to more accurate and reliable predictions.**

Furthermore, it ensures that the model treats all features with equal importance, allowing for a more balanced and nuanced analysis of the underlying patterns in the data.

Splitting Data for Training and Testing
The dataset is split into training and testing subsets to evaluate the model's performance. Initially, we split the training data into an **8:2 ratio to tune and validate the model**. Once the model is optimized, **we train it on the entire training dataset and then evaluate its performance on the test data. This approach allows us to accurately assess the model's accuracy and generalization capabilities on unseen data.**

Encoding Strategies

We convert city names into numerical values using a dictionary mapping to enhance analytical processes. Upon analysis, we **discovered that the categories are exactly divided into two sets of cities. Therefore, we divided the cities into two sets and applied encoding to differentiate between them effectively**. This approach helps in structuring the data and allows for more efficient analysis and modeling, especially when dealing with categorical variables.

Once the two sets of cities are identified, encoding is applied to differentiate between them effectively. This encoding could involve assigning different numerical ranges or labels to each set.

Set 1: Cities in range 0-9 have clusters 2,4,5

KELANIYA	PANADURA
MORATUWA	KADUWELA
WATTALA	PELIYAGODA
HOMAGAMA	KOTTE
DEHIWALA-MOUNT LAVINIA	NUWARA ELIYE

Set 2: Cities in range 10-19 have clusters 1,3,6

BATTICALOA	GALLE
COLOMBO	KATUNAYAKE
JAFFNA	NEGOMBO
GAMPAHA	TRINCOMALEE
KALMUNAI	KANDY

Impact on Input Requirements

By converting city names into numerical values, the model's input requirements are met, as most machine learning algorithms expect numerical data. This enables seamless integration of city names as features in the model.

Using numerical values ensures consistency and uniformity in the data, which is crucial for accurate modeling and predictions.

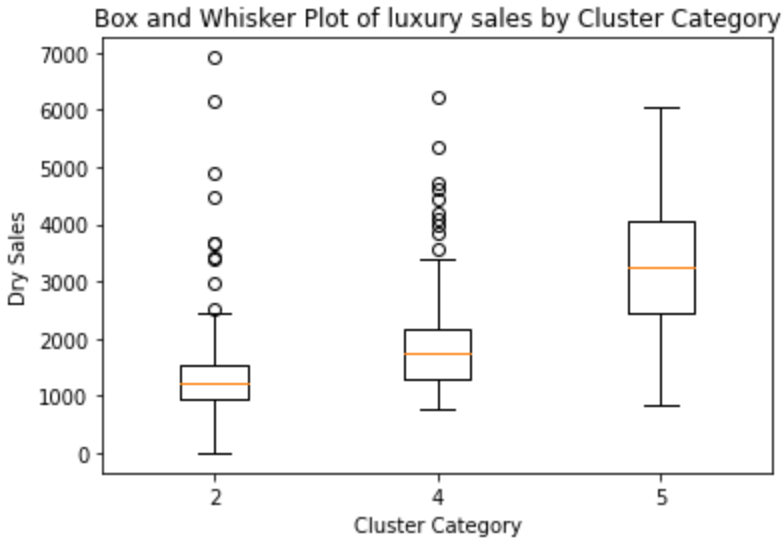
Impact on Performance

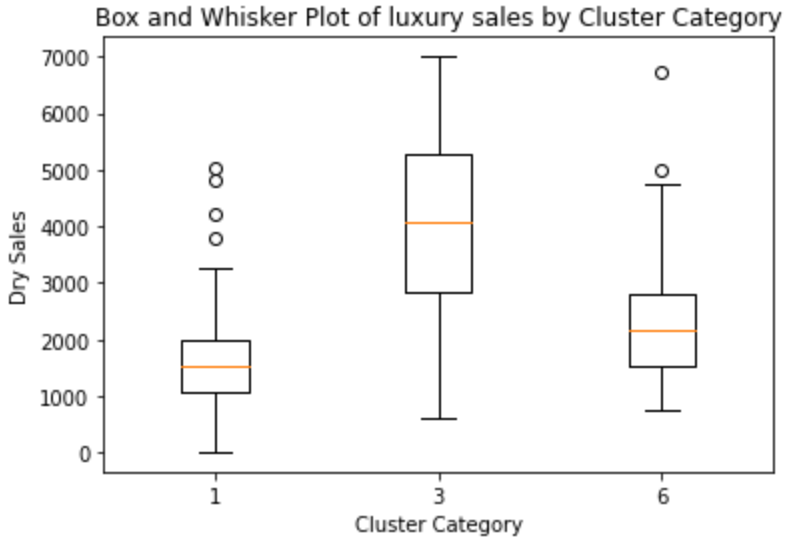
Numeric encoding simplifies data handling and computation, leading to improved efficiency during model training and evaluation.

While numerical encoding makes the data machine-readable, it may reduce the interpretability of the model results, especially if the numerical values are arbitrary or lack meaningful interpretation.

How do the features correlate with the target variable, and notable inter-feature relationships

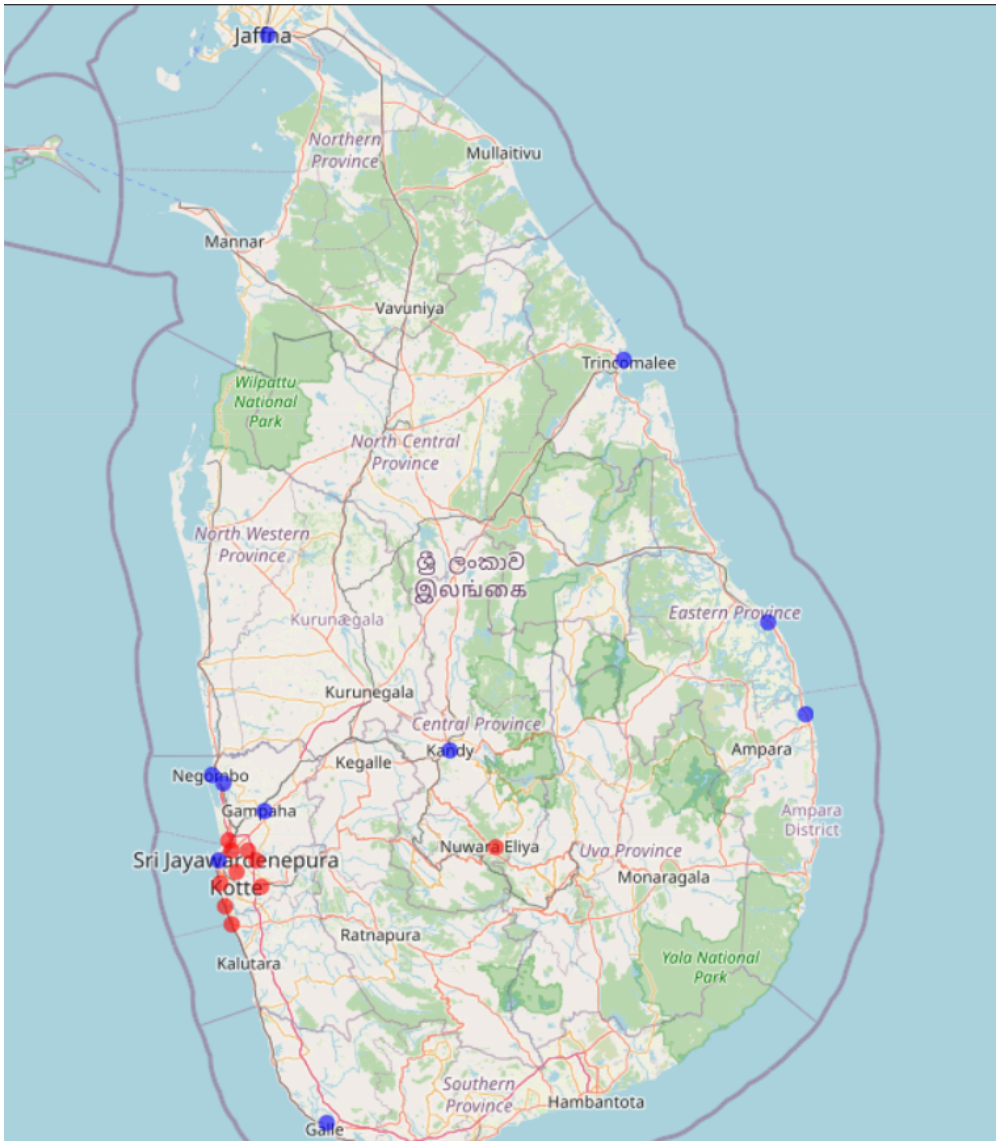
Since a clear separation based on the city was observed between the clusters we separated the clusters to the two sets and analyze the relationship between the sales and the categories we can plot box and whisker plots.



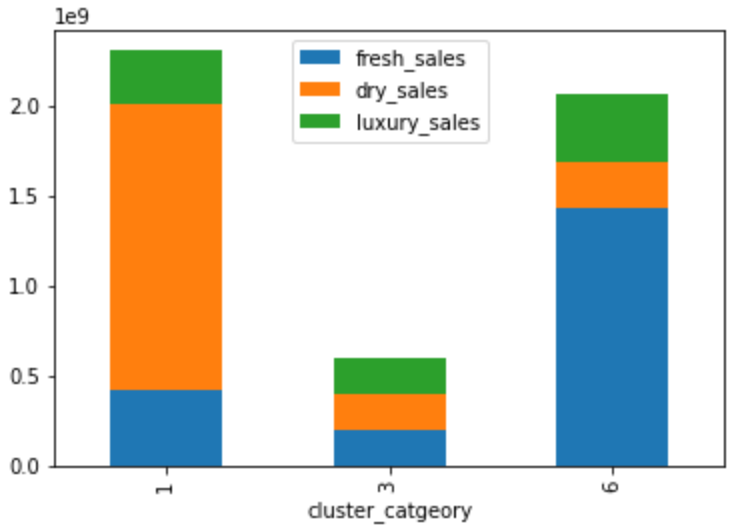


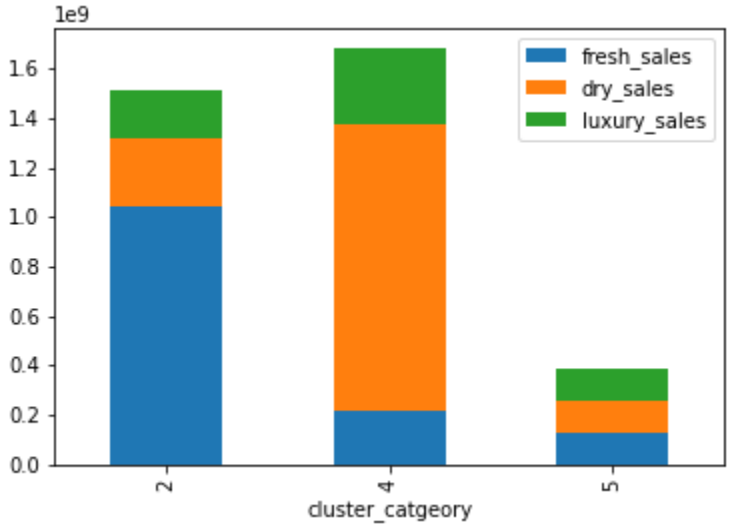
Observing the plots we can see a clear separation for each of the categories

Describe the target variable and interpret each category within it, detailing the characteristics that define the different customer segments.



The above plot represents the two categories of cities identified. We can see a clear pattern among them with most of the cities in the first group near Colombo apart from Colombo itself which is on the second set and Nuwara-Eliya is on the first set. **We have chosen city set 1 as red cities and city set 2 as blue cities.**





From the above two plots it is clear that

1. **Clusters 1 and 4** : Customers Who Buy a Lot of Dry Sales
2. **Clusters 3 and 5** : Customers Who Buy Products in Equal amounts in all three categories (Perhaps, these customers do not frequently buy products from the company)
3. **Clusters 2 and 6** : Customers Who Buy Larger amount of fresh sales when compared to other products.

To summarize

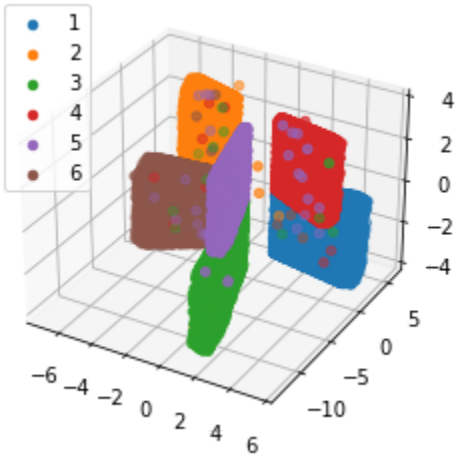
Cluster	City Set	Sale Strategy	Corresponding Cluster in Other City Group
1	1	**Lot of Dry Sales**	4
3	1	**Equal amounts**	5
6	1	**Lot of Fresh Sales**	2
4	2	**Lot of Dry Sales**	1
5	2	**Equal amounts**	3
2	2	**Lot of Fresh Sales**	6

Algorithms we considered for this problem, and why did you choose the final algorithm?

We employ Linear Discriminant Analysis (LDA) to reduce the dimensionality of the feature space for visualization purposes. This transformation allows us to **visualize the data in a 3D scatter plot, making it easier to identify patterns, clusters, and relationships that may not be apparent in higher-dimensional spaces**. By simplifying the data into a three-dimensional view, we can gain valuable insights into the structure and distribution of the data, facilitating better interpretation and communication of the results.

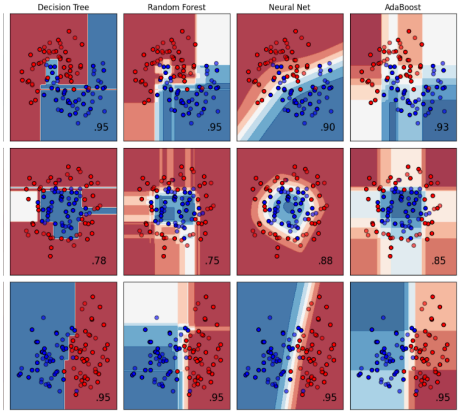
We employ Linear Discriminant Analysis (LDA) to reduce the dimensionality of the feature space for visualization purposes:

This transformation allows us to visualize the data in a 3D scatter plot.



1,2,3,4,5 and 6 represent categories.

We searched for similar classifiers in scikit-learn on synthetic datasets.



Since **Decision Tree, Random Forest, AdaBoost and Neural Networks** show a similar classification, we tried with all these modeling algorithms as well as we tried with **support vector machine and LDA**.

SVM and LDA were chosen for their effectiveness in classification tasks, robustness to overfitting, ability to handle high-dimensional data, and interpretability.

The **Multi-layer Perceptron (MLP)** was selected as one of the modeling algorithms due to its ability to capture complex non-linear relationships in the data. During the experimentation phase, **it was observed that the MLP model didn't overfit the training data and achieved the best accuracy among the classifiers tested.**

We employed a Multi-layer Perceptron (MLP) with 100 neurons in the first hidden layer and 50 neurons in the second hidden layer. The training process is as follows,

If H^1 denotes the output of first hidden layer with 100 neurons, and H^2 denotes the output of second hidden layer with 50 neurons,

The mathematical representation of the MLP is,

Forward Propagation,

$$
\begin{aligned}
Z^{(1)} &= XW^{(1)} + b^{(1)} \\
H^{(1)} &= \text{activation}(Z^{(1)}) \\
Z^{(2)} &= H^{(1)}W^{(2)} + b^{(2)} \\
H^{(2)} &= \text{activation}(Z^{(2)}) \\
Z^{(out)} &= H^{(2)}W^{(out)} + b^{(out)} \\
\hat{y} &= \text{softmax}(Z^{(out)})
\end{aligned}
$$

Loss Calculation,

$$\text{loss} = \text{cross-entropy}(y, \hat{y})$$

Backpropagation,
Updating W and b using **Gradient Descent.**

Repeat
Repeating the process until convergence or for a fixed number of iterations.

Challenges Faced during Model Training

1. **Overfitting**

   The model architecture was too complex relative to the size and complexity of the dataset, it might have captured noise or irrelevant patterns, leading to overfitting**.** This might have happened mainly due to the outliers. Thus, the model might have learned to fit these noise patterns or the outliers.

   Moreover, LDA assumes that the classes have identical covariance matrices and that the features are normally distributed within each class. If the underlying data distribution deviates significantly from these assumptions, LDA might have attempted to fit a decision boundary that is too complex for the true underlying distribution, leading to overfitting.

   MLPs can have a large number of layers and parameters, giving them a high capacity to fit complex patterns in the training data. If the model architecture is too complex relative to the size and diversity of the training data, it may learn to memorize noise or irrelevant patterns, leading to overfitting.

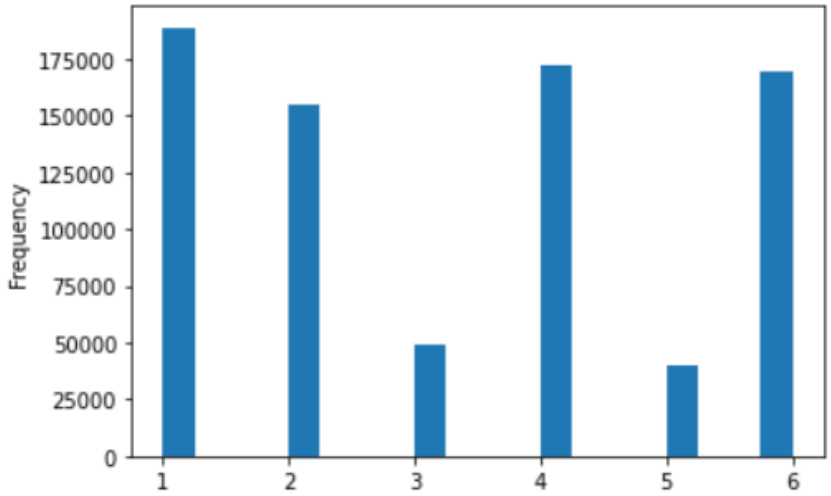
2. **Computational Constraints**

   A significant challenge encountered during the model training phase was the lack of data pertaining to specific geographical areas. For example, the training dataset lacked information regarding Anuradhapura and Medawachchiya. Nonetheless, the testing dataset required predictions for these cities, necessitating the development of strategies to address this disparity in data availability.

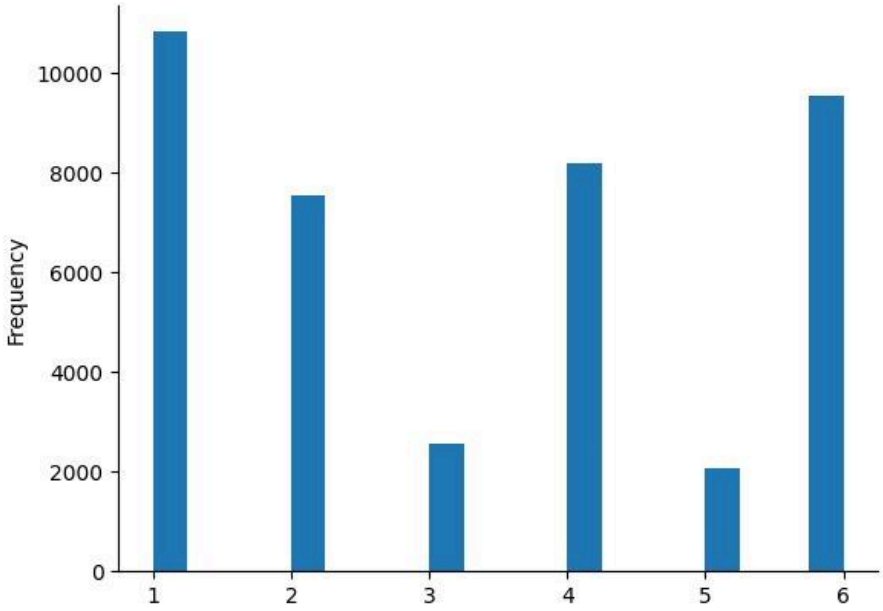
   In addition, LDA involves computing covariance matrices and eigenvectors, which may require significant memory resources, particularly for large datasets with many features. Limited memory capacity may have restricted the size of datasets that can be processed or lead to slower computations due to frequent disk swapping.

   MLP training involves storing model parameters, intermediate activations, and gradients in memory. Large models with many layers or parameters may require substantial memory resources, especially during backpropagation when gradients need to be computed and stored for each layer.

Classified clusters



RESULTED CLUSTER FREQUENCIES FOR TRAIN DATA



RESULTED CLUSTER FREQUENCIES FOR TEST DATA

Based on the results obtained for frequencies of clusters for train data and test data. We can deduce that the model is working properly as well as using above mentioned information we can name the categories as,

- Red City Dry Lovers : 1
- Red City Fresh Freaks : 3
- Red City Equalizers : 6
- Blue City Dry Lovers : 4
- Blue City Fresh Freaks : 5
- Blue City Equalizers : 2

How Does the Solution Enhance the Effectiveness of the Company's Marketing Strategies Based on the Classified Clusters?

Classifying customers into distinct clusters based on purchasing behavior enables KJ Marketing to enhance its marketing strategies effectively.

1. **Personalized Marketing Campaigns:**
   - Tailor promotions and communications to match each cluster's preferences, increasing engagement and conversion rates.

2. **Product Recommendations:**
   - Provide targeted product suggestions and leverage upselling and cross-selling opportunities based on cluster insights.

3. **Inventory Management:**
   - Improve demand forecasting and optimize stock levels by understanding cluster-specific purchasing patterns.

4. **Customer Retention and Loyalty Programs:**
   - Design loyalty incentives for high-value customers and implement retention strategies for clusters showing reduced engagement.

5. **Market Expansion and Outlet Optimization:**
   - Use geographical insights to inform decisions on new outlets and localized marketing campaigns.

6. **Strategic Decision Making:**
   - Guide product development and dynamic pricing strategies to better meet the needs of different customer clusters.

This customer-centric approach enhances satisfaction and loyalty while driving higher sales and profitability.