

- **DataStorm537** -

- Navindu De Silva
- Saeedha Nazar
- Tishan Sathruwan



# STORMING ROUND REPORT



# LastBrainCell

GitHub Link : [click here](#)

# Table of Contents

<b>Table of Contents.....</b>	<b>1</b>
<b>1. Data Preprocessing.....</b>	<b>2</b>
1.1 Missing Data Handling and Data Imputation.....	2
1.2 Data Cleaning (Remove erroneous data entries, outliers and duplicates).....	3
<b>2. Features.....</b>	<b>4</b>
2.1 Feature Engineering.....	4
<b>3. Feature Scaling and Normalisation.....</b>	<b>7</b>
3.1 Min-Max Scaling.....	8
3.2 Log Transformation with Standardization.....	8
<b>4. Feature Encoding Strategies.....</b>	<b>9</b>
4.1 Encoding for 'city_type'.....	9
4.2 Encoding for 'outlet_city'.....	9
4.2.1 Binary Encoding for 'outlet_city'.....	9
4.2.2 Label Encoding for 'outlet_city'.....	10
4.2.2 One-Hot Encoding for 'outlet_city'.....	10
<b>5. Feature Correlation.....</b>	<b>11</b>
5.1 Correlations with Target Cluster Type.....	11
5.2 Inter Feature Correlations.....	11
<b>6. Characteristics of Customer Segments.....</b>	<b>12</b>
<b>7. Model Algorithm Selection.....</b>	<b>17</b>
7.1 XGBoost Algorithm.....	18
7.2 Random Forests.....	18
7.3 K-Nearest Neighbour.....	18
7.4 Logistic Regression.....	19
7.5 Neural Network.....	19
<b>8. Challenges in Model Training.....</b>	<b>20</b>
8.1 Absence of certain outlets from the training dataset.....	20
8.2 High Time Consumption of Training of Small NN model.....	21
<b>9. Description of Classified Customers.....</b>	<b>21</b>
<b>10. Suggested Marketing Strategies.....</b>	<b>21</b>

# 1. Data Preprocessing

Data preprocessing is a crucial step in the machine learning (ML) pipeline. Because in order to make robust and precise predictions, cleaned and formatted data should be extracted from raw data sources. Especially in the business domain, we have to consider this part as one of the major parts in business analysis since these raw data sources have highly impactful insight about the current status of the business environment.

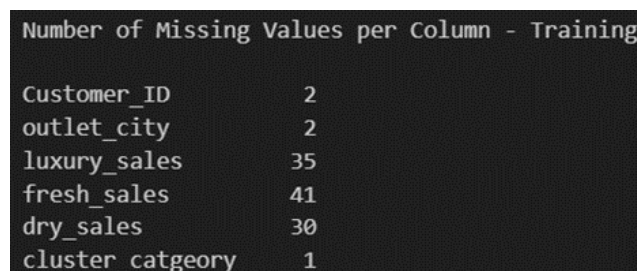
We followed following steps under data preprocessing,

1. Missing Data Handling and Data Imputation
2. Data Cleaning (Remove erroneous data entries, outliers and duplicates)

## 1.1 Missing Data Handling and Data Imputation

In raw data sets, some data entries might be incomplete due to various reasons. As a result of that, some required fields of data might not be available. So, in this kind of situation, we have to impute suitable values for missing sections or exclude the whole entry to prevent erroneous decisions.

So, in this dataset of sale details of customers in different districts, we found out some missing fields in a few columns in csv files.



Number of Missing Values per Column - Training	
Customer_ID	2
outlet_city	2
luxury_sales	35
fresh_sales	41
dry_sales	30
cluster_catgeory	1

Figure 1 : Number of Missing Fields

Therefore, following strategies were used to tackle these missing fields. They have been described under the following section in brief.

### ➤ Customer ID Imputation

We visualised the customer ID values and there was no relationship between those ID values and other features in the dataset. Therefore, the missing data entries were imputed using dummy values and it does not have an impact on the training process.

### ➤ Missing Sales Values

Missing values regarding sales columns such as “luxury\_sales” , “fresh\_sales” and “dry\_sales” could have been handled by taking by assigning as zero to reflect non-purchases. But as there weren’t any other occurrences of zero in the sales columns, it was believed that it would be the model to detect the zeros as outliers. Since the missing sales values were as little as 0.5%, it was safe to drop these features without much impact on the training process.

### ➤ Exclude Partially Filled Rows

“outlet\_city” and “cluster\_category” are essential features in the dataset. But if we had added those values with random values, the precision of predictions would definitely reduce. As well as there were only a few rows which have empty entries in “outlet\_city” and “cluster\_category” .Therefore, we decided to discard those incomplete rows.

These steps were carried out for both training and test dataset to handle the unexpected errors.

## 1.2 Data Cleaning (Remove erroneous data entries, outliers and duplicates)

In raw dataset, erroneous entries might be present due to various scenarios. So we have to remove those data entries to maintain the robustness of the dataset. So, here we found some erroneous entries in “cluster\_category” . Since, there are only 6 clusters in the “cluster\_category”, we removed those erroneous data entries.

Then we checked for outliers and according to the data distribution of “dry\_sales” , “fresh\_sales” and “luxury\_sales” , there were no significantly high or significantly low values. But “luxury\_sales ” showed positively skewed behaviour. So, we did some transformation to prevent the issue and it has been described in detail under the “Feature Scaling” section.

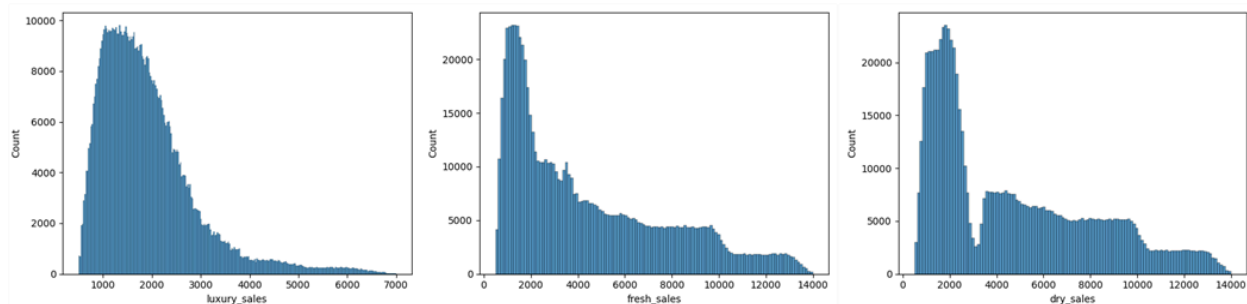


Figure 2 : Histograms of all types of sales.

Duplicates will reduce the performance of the model. Because, due to duplication, the model won’t learn correct distribution of data and this might not generalise the model. Therefore, removing duplicates is a crucial task in data preprocessing. But no duplicates were present in the dataset.



## 2. Features

The original dataset contained very few input features as follows:

- ‘Customer\_ID’: unique identifier of the customers.
- ‘outlet\_city’: identifiers of the cities where the outlets are located.
- ‘luxury\_sales’: average monthly sales per customer for luxury goods.
- ‘fresh\_sales’: average monthly sales per customer for fresh goods.
- ‘dry\_sales’: average monthly sales per customer for dry goods.

As the ‘Customer\_ID’ appeared to be a random number and had no influence on the ‘cluster\_category’ it was disregarded as an input feature.

This resulted with four features and they were analysed to engineer new features to improve the model’s predictions.

### 2.1 Feature Engineering

The boxplot distributions for the ‘outlet\_city’ feature under each sales feature (‘fresh\_sales’, ‘luxury\_sales’, ‘dry\_sales’) were observed and a significant two-class sub-grouping was visible from the way the outlet cities were distributed.

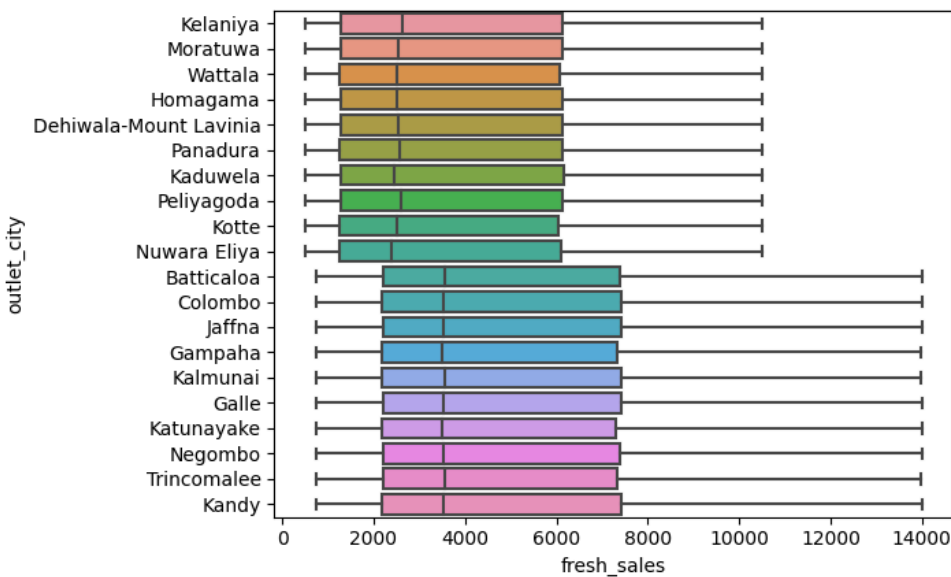


Figure 3: Boxplot of ‘fresh\_sales’ distribution for each ‘outlet\_city’

The fresh sales distribution for each outlet city showcased a two-class sub-grouping as above with Q1 (25th percentile), median and Q3 (75th percentile) falling in the same range as the fresh sales count.

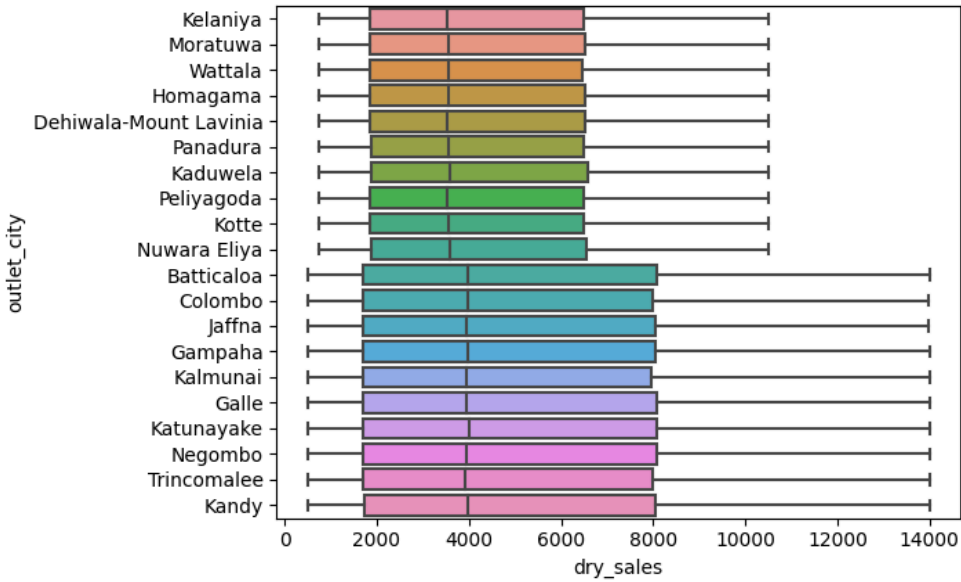


Figure 4: Boxplot of 'dry\_sales' distribution for each 'outlet\_city'

Interestingly, the dry sales distribution for each outlet city also showcased a two-class sub-grouping as above with Q1 (25th percentile), median and Q3 (75th percentile) falling in the same range as the dry sales count.

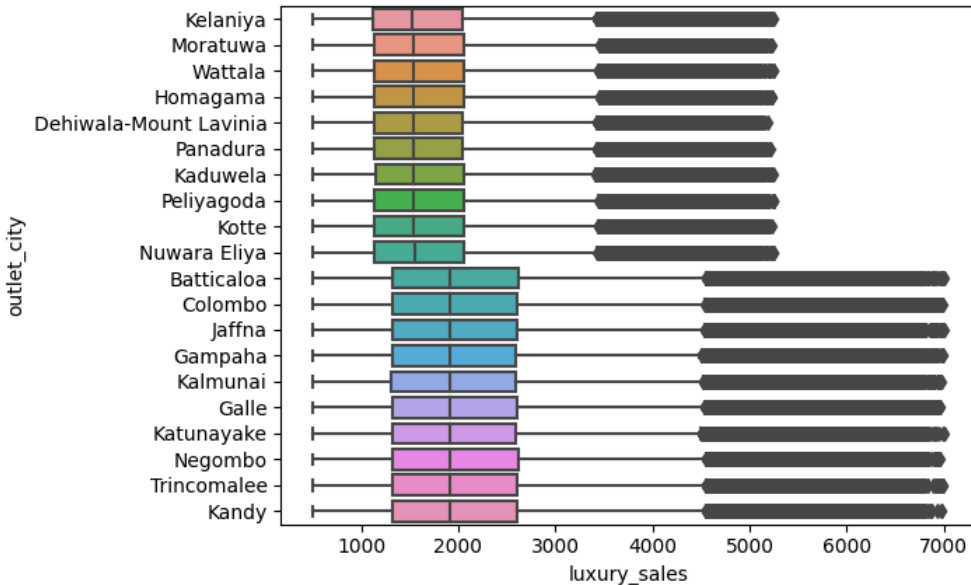


Figure 5: Boxplot of 'luxury\_sales' distribution for each 'outlet\_city'

Additionally, the luxury sales distribution for each outlet city also showcased a two-class sub-grouping as above with Q1 (25th percentile), median and Q3 (75th percentile) falling in the same range as the luxury sales count. The distribution also presented some outliers.

This indicated that there was a sub-grouping in the way all sale types were distributed across the outlet cities.

But the main deciding factor that helped distinguish the sub-grouping of the outlet cities was the number of customer records per outlet.

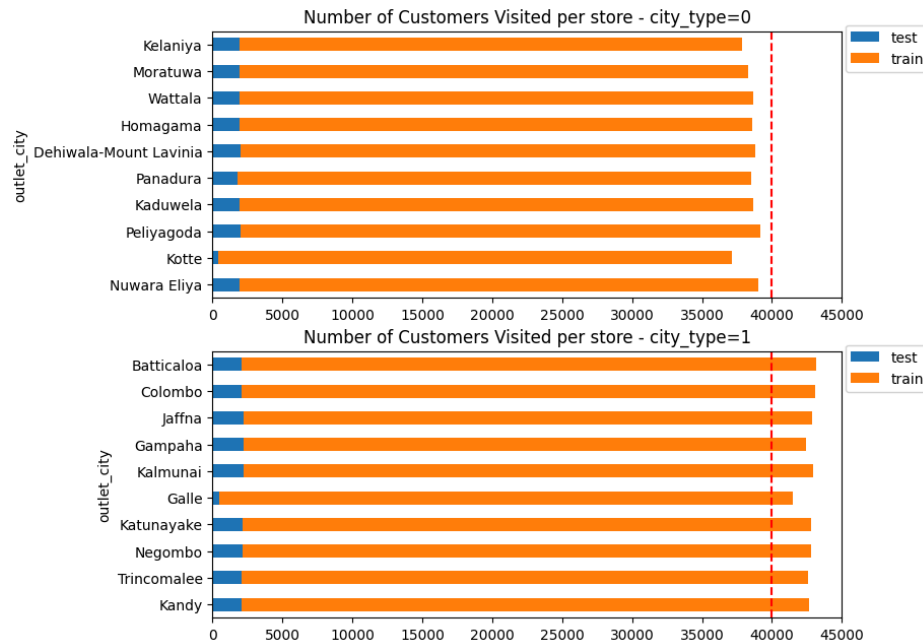


Figure 6: Customer sales count for both city types

It is seen that the first 10 outlet cities, namely:

**Kelaniya, Moratuwa, Wattala, Homagama, Dehiwala-Mount Lavinia, Panadura, Kaduwella, Peliyagoda, Kotte and Nuwara Eliya**

And the next 10 outlet cities, namely:

**Batticaloa, Colombo, Jaffna, Gampaha, Kalmunai, Galle, Katunayake, Negombo, Trincomalee, Kandy** form two informal clusters of cities.

Therefore a new feature was introduced as 'city\_type' to identify which cluster the city belongs to. This is a binary feature with '0' indicating the first cluster where the customer sales are less than 40,000 (low traffic outlets) and '1' indicating the other cluster where customer sales are generally greater than 40,000 (high traffic outlets).

Another type of feature engineering was introduced to provide more insight regarding the customer. They are ratio calculation features such as 'luxury\_fresh\_ratio', 'luxury\_dry\_ratio' and 'fresh\_dry\_ratio'. A mini logistic regression model to infer the 'city\_type' with and without the ratio features was utilised to experiment the need of these features. The results can be seen in the below images.

```

Validation Accuracy: 0.599038994640547
Validation Confusion Matrix:
[[21763 16710]
 [15834 26858]]
Validation Classification Report:

```

	precision	recall	f1-score	support
0	0.58	0.57	0.57	38473
1	0.62	0.63	0.62	42692
accuracy			0.60	81165
macro avg	0.60	0.60	0.60	81165
weighted avg	0.60	0.60	0.60	81165

```

Test Accuracy: 0.597503880935367
Test Confusion Matrix:
[[21887 16838]
 [15831 26610]]
Test Classification Report:

```

	precision	recall	f1-score	support
0	0.58	0.57	0.57	38725
1	0.61	0.63	0.62	42441
accuracy			0.60	81166
macro avg	0.60	0.60	0.60	81166
weighted avg	0.60	0.60	0.60	81166

Figure 7: Results of mini logistic regression model without ratio features

```

Validation Accuracy: 0.8735415511612148
Validation Confusion Matrix:
[[33149 5402]
 [ 4862 37752]]
Validation Classification Report:

```

	precision	recall	f1-score	support
0	0.87	0.86	0.87	38551
1	0.87	0.89	0.88	42614
accuracy			0.87	81165
macro avg	0.87	0.87	0.87	81165
weighted avg	0.87	0.87	0.87	81165

```

Test Accuracy: 0.8722741049207796
Test Confusion Matrix:
[[33040 5353]
 [ 5014 37759]]
Test Classification Report:

```

	precision	recall	f1-score	support
0	0.87	0.86	0.86	38393
1	0.88	0.88	0.88	42773
accuracy			0.87	81166
macro avg	0.87	0.87	0.87	81166
weighted avg	0.87	0.87	0.87	81166

Figure 8: Results of mini logistic regression model with ratio features

The mini logistic regression model without ratio features only contained ‘fresh\_sales’, ‘luxury\_sales’ and ‘dry\_sales’ as input features to infer the ‘city\_type’, this resulted in a lower accuracy as the customer’s information is insufficient to decide the city type. However, the model with ratio features in addition to the sales features inferred the city type more accurately. Hence, it was decided to incorporate the ratio features in the input features as it provided relative information of all sale types for each customer.

### 3. Feature Scaling and Normalisation

Feature Scaling ensures that no single feature dominates others due to its scale, thereby improving the performance of algorithms sensitive to feature magnitude, such as gradient descent and k-nearest neighbours. As well as, this will help the optimization algorithm to converge to an optimal point in a faster manner and that enhances the precision of the model. By bringing all features to a comparable range, feature scaling enables more balanced and efficient learning.

Here we experimented several techniques to improve the performance,

- Min-Max Scaling
- Log Transformation with Standardization



### 3.1 Min-Max Scaling

Min-Max scaling is used to transform data into fixed range within 0 and 1. So, this technique ensures that all features contribute equally to the analysis, preventing features with larger ranges from dominating the results.

$$\text{Scaled Value} = \frac{X - \min(X)}{\max(X) - \min(X)} \text{ where } X \in \mathfrak{R}$$

As well as, this technique preserves the value distribution of the feature and gives a scaled down version of the value distribution. This leads to fast convergence of optimization algorithms.

So, we used this technique to normalize features such as “dry\_sales”, “fresh\_sales”, “luxury\_sales” and “city encodings” (cities were encoded into numbers ). Because sales values are always positive and the range is large. In order to shrink down the range, Min-Max scaler was used.

### 3.2 Log Transformation with Standardization

Log transformation combined with standard scaling is used to stabilize variance and make data more normally distributed. This gives proper representation to continuous data types such as different kinds of sale values. Here, Log transform is used to convert positively skewed dataset into more centralized distribution within the value region.

$$\text{Transformed Value} = \log(X) \text{ where } X \in \mathfrak{R}^+$$

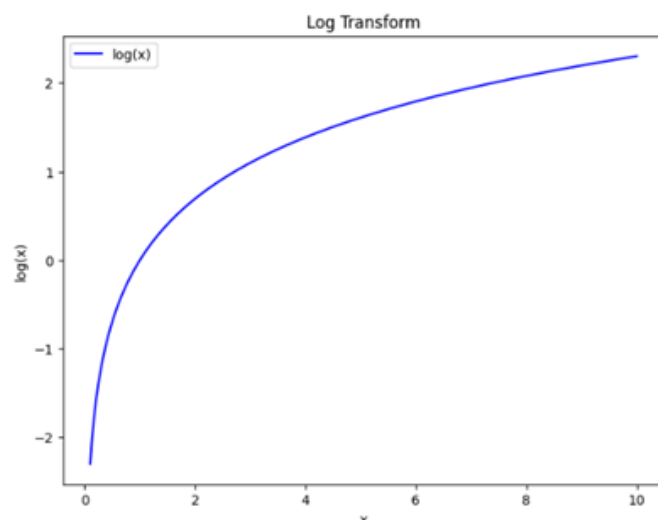


Figure 9 : Log Transformation

This increases the stability of the machine learning algorithms which expect normalised distributions. So, we introduced this normalisation for all kinds of sale features since they are positively skewed continuous distributions.

For an example,

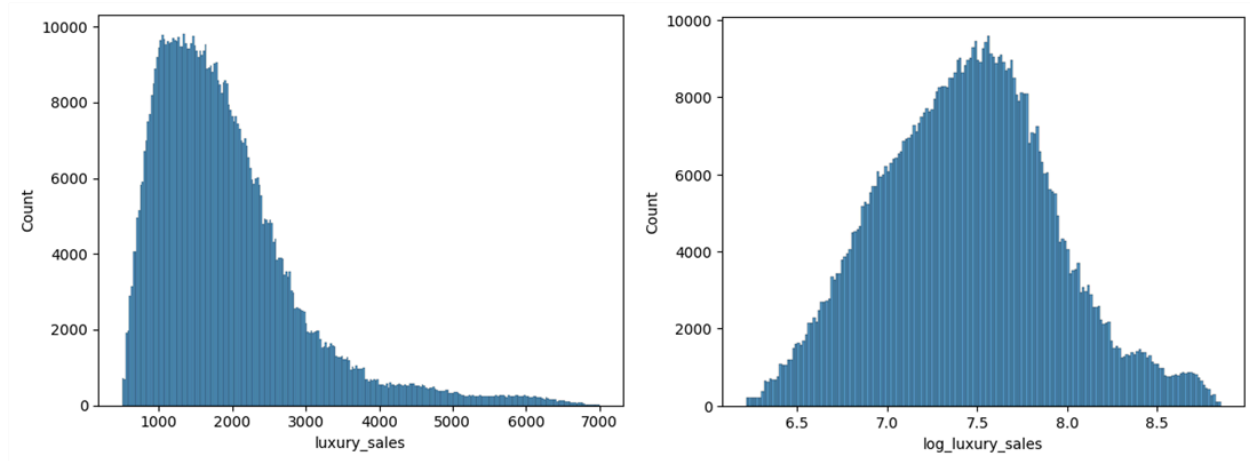


Figure 10: Log Transformation with Standardization for "luxury\_sales" feature

## 4. Feature Encoding Strategies

### 4.1 Encoding for 'city\_type'

The 'city\_type' feature is a binary feature which represents the sub-group of the 'outlet\_city', hence it was encoded as '0' or '1' in binary format as mentioned previously.

### 4.2 Encoding for 'outlet\_city'

This feature was not utilised in the final model, however it was used when experimenting different models and features. It was encoded in three different formats to check this feature's performance.

#### 4.2.1 Binary Encoding for 'outlet\_city'

The different outlet cities were manually encoded in binary format as follows :

Kelaniya: **00000**  
 Moratuwa: **00001**  
 Wattala: **00010**  
 Homagama: **00011**  
 Dehiwala-Mount Lavinia: **00100**  
 Panadura: **00101**  
 Kaduwela: **00110**  
 Peliyagoda: **00111**  
 Kotte: **01000**  
 Nuwara Eliya: **01001**  
 Batticaloa: **01010**  
 Colombo: **01011**

Jaffna: **01100**  
Gampaha: **01101**  
Kalmunai: **01110**  
Galle: **01111**  
Katunayake: **10000**  
Negombo: **10001**  
Trincomalee: **10010**  
Kandy: **10011**

#### 4.2.2 Label Encoding for ‘outlet\_city’

Label Encoding assigns each city an integer value from 0 to 19. This was used when experimenting logistic regression.

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**

#### 4.2.2 One-Hot Encoding for ‘outlet\_city’

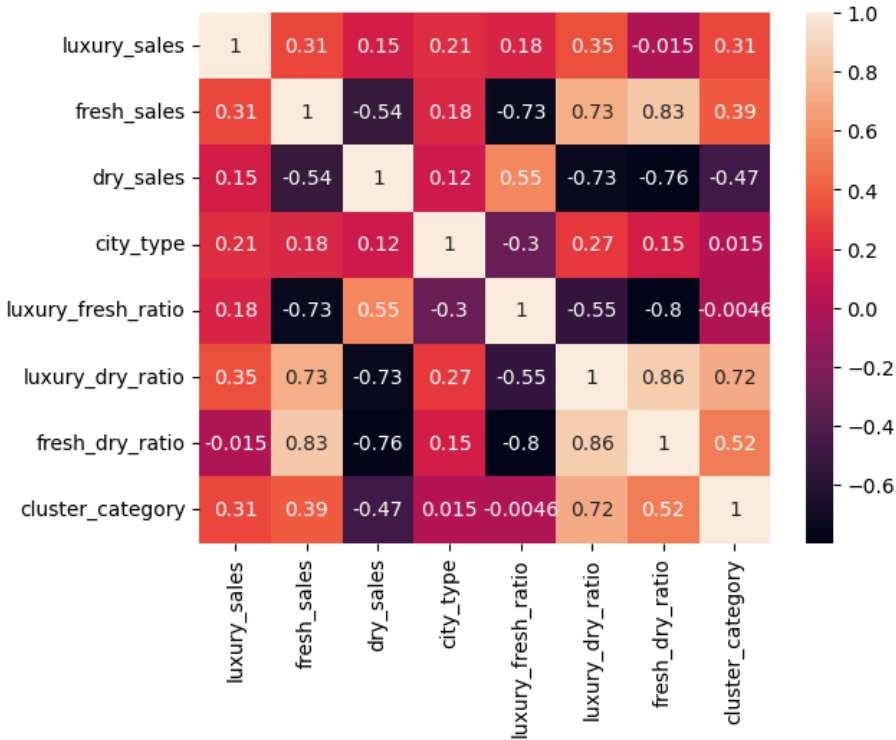
One-Hot Encoding creates a binary column for each category and returns a sparse matrix or dense array. It does not assume any ordinal relationship between categories, which makes it suitable for nominal data. However, the encoded dataset is sparse, meaning that it contains a lot of zeros, which can be inefficient in terms of memory and computational power.

The above cities were converted to one-hot encoding format in the same order.

This encoding was useful when training the neural network model as it is said to perform better with numerical inputs derived from One-Hot Encoding.

## 5. Feature Correlation

The Correlation matrix for the tested features for our solution is presented below.



### 5.1 Correlations with Target Cluster Type

Considering the sales columns for *fresh\_sales*, *dry\_sales* and *luxury\_sales*, there is a reasonable relationship with our target cluster category. While the luxury sales and fresh sales of customers are greater when customer is the later cluster categories, customers in the first few cluster categories have greater purchases of dry items.

Among the sales ratio features designed, the luxury vs fresh sales ratio has very little relation in determining the customer cluster category. But there is an exceptionally high positive correlation with ratios *luxury\_dry\_ratio* and *fresh\_dry\_ratio* with the cluster category. It indicates that customers in the categories 4, 5, 6 are likely to have higher ratios. This vindicates our use of ratio features in this regard.

### 5.2 Inter Feature Correlations

In between features, there are strong relationships between the sales ratio features.

1. Fresh/Dry Ratio & Luxury/Dry Ratio

These 2 features show a very strong positive correlation of 0.86. Since both ratios have Dry Sales as the denominator, it could indicate that when customers prefer Fresh items over Dry items, they tend to show a similar liking towards Luxury items over Dry items.

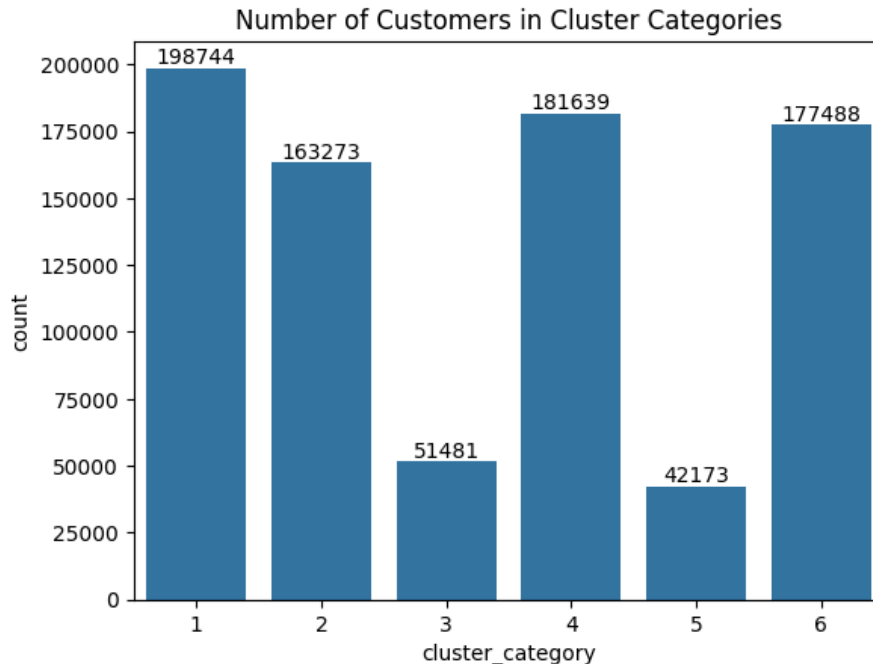
## 2. Luxury/Fresh Ratio & Fresh/Dry Ratio

These 2 features show a strong **negative** correlation of -0.80. It would indicate that when customers much prefer Luxury items over the Fresh items, they tend to have a strong preference to the Dry items when compared to Fresh items.

# 6. Characteristics of Customer Segments

A brief analysis can be viewed [here](#) with plot diagrams.

The following analysis studies the behaviour of the customer cluster categories given for the training dataset and the predicted cluster categories for the test dataset

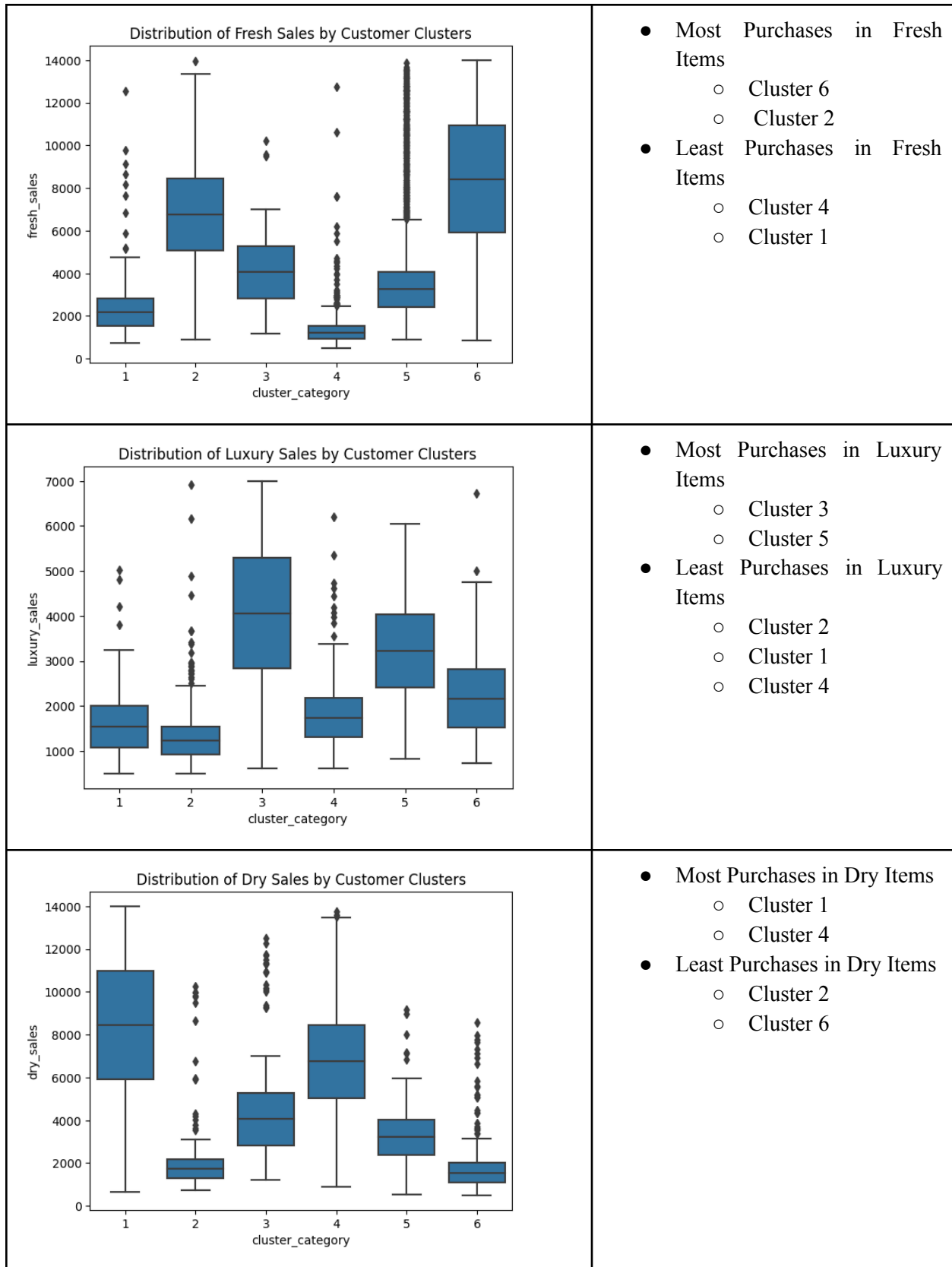


The number of customers allocated to each of the cluster categories are displayed in this graph. **Clusters 3 and 5** have a relatively small number of customers allocated to them compared to other clusters. Hence, it could indicate these clusters have some rare specialized features to cause separate allocation.

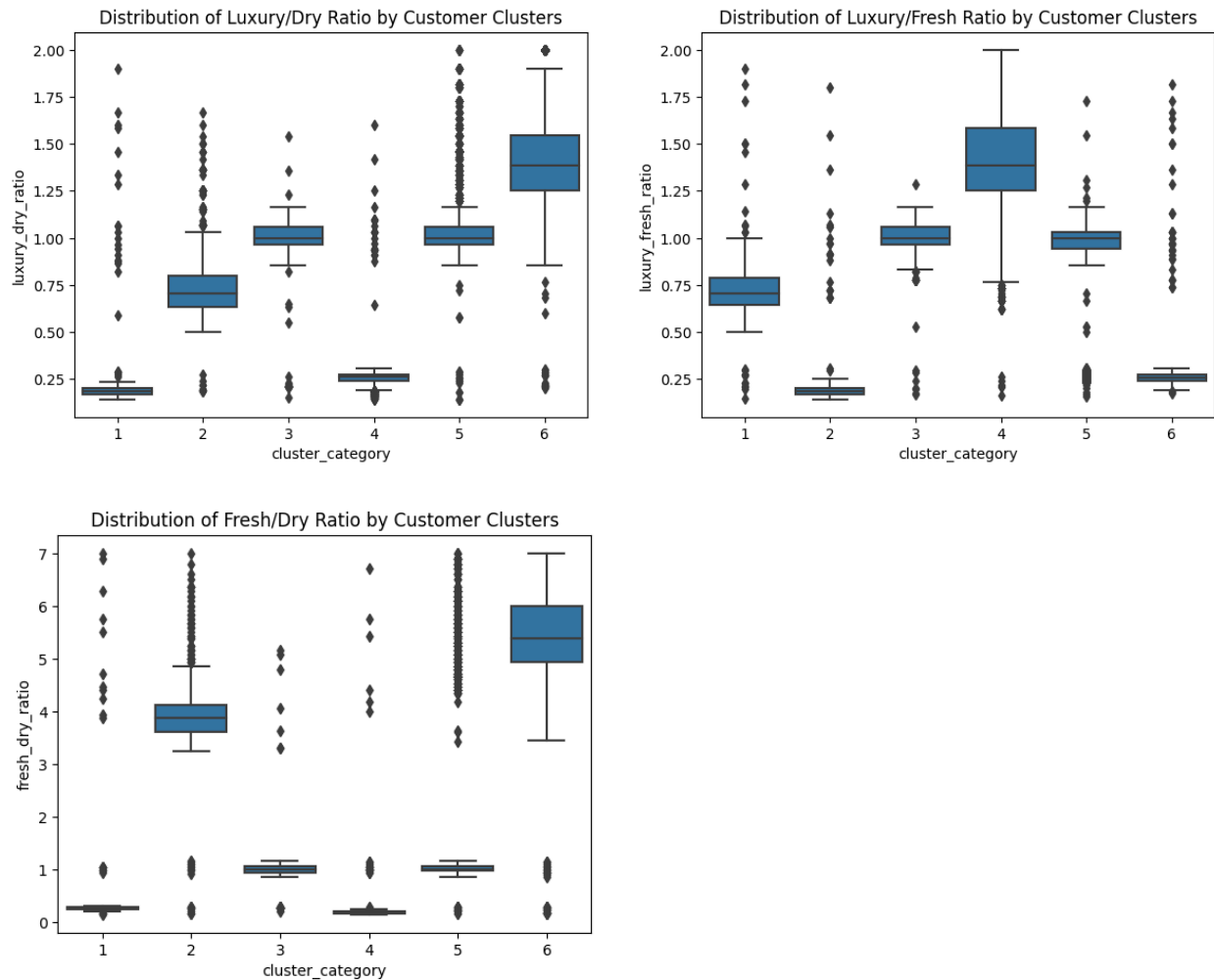
Next we analyze the behaviour of the sales features according to each customer cluster. The box plots help highlight the

cluster categories that have higher sales in each item category and their distributions too.





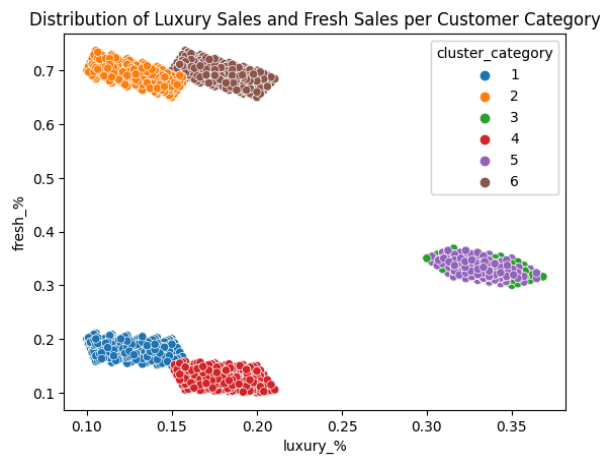
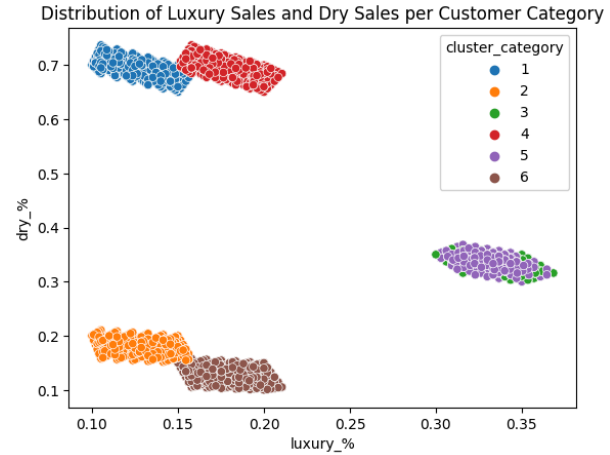
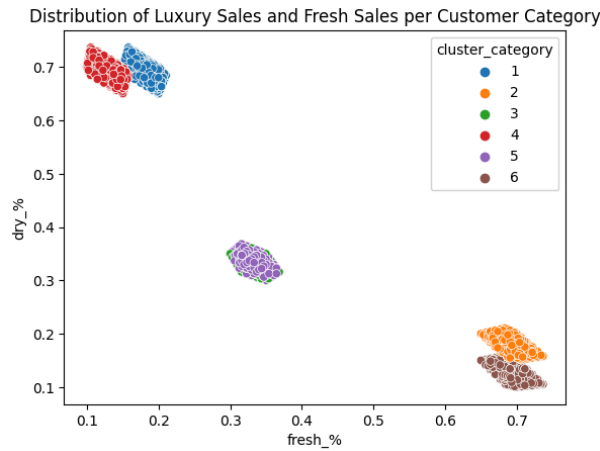
The boxplot diagrams for the distribution of the sales ratio features further solidify the preferences the clusters have over certain type of items.



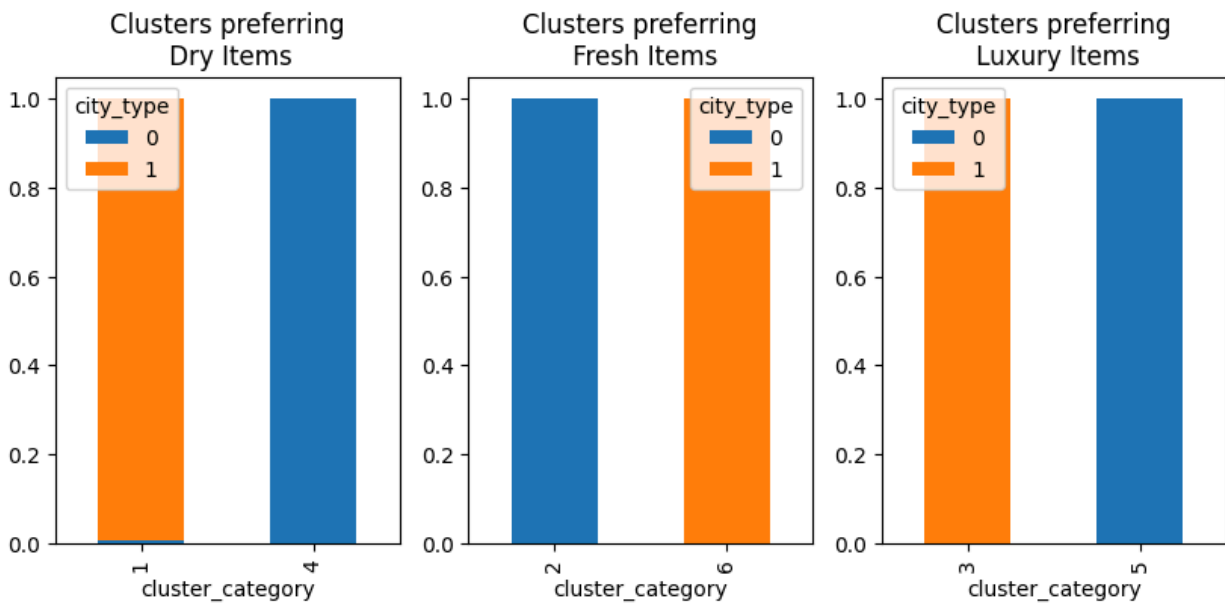
The purchasing pattern based on their total purchasing has indicated clear patterns and behaviours in the customer categories. The following scatterplots are designed using sales features taken as percentage of the customer's total purchases. The clusters can be clearly noted in the scatter plots. On a special note, it is noteworthy to see the clusters are located in the scatterplots as pairs

1. Cluster 1 & Cluster 4
2. Cluster 3 & Cluster 5
3. Cluster 2 & Cluster 6

Moreover, these pairs were the same pairs that were identified in the box plots as well indicating clear behavioural patterns.



The scatterplots lead us to ask us about the difference between each pairing of clusters even though they display similar behaviours in terms of item purchases. Analysing the *city\_type* feature introduced to indicate the grouping each customer's city belongs to, it was found that these pairings can be split based on the city pair they belong to.



According to the above stacked bar charts, it can be seen that each cluster has predominant city type. From the pairings which were identified earlier, we can see that exactly one of the clusters in each pair

belongs to either the city\_type 0 or city\_type 1. Hence we can use the city type to identify the two clusters with similar purchasing behaviours. The characteristics of each cluster can be summarized as follows.

Cluster Type	City Type of Majority Customers	Preference for			Relative Number of Customers
		Dry Items	Fresh Items	Luxury Items	
Cluster 1	Type 1	High	Low	Low	High
Cluster 2	Type 0	Low	High	Low	High
Cluster 3	Type 1	Moderate	Moderate	High	Less
Cluster 4	Type 0	High	Low	Low	High
Cluster 5	Type 0	Moderate	Moderate	High	Less
Cluster 6	Type 1	Low	High	Low	High

*Table of Summarized Characteristics for each Cluster Type*

## 7. Model Algorithm Selection

Model selection is one of the major parts of this competition. So, we have trained various kinds of models to identify the best model among them. We implemented following model for this clustering task,

- ❖ Random Forests
- ❖ XGBoost
- ❖ K-Nearest Neighbour
- ❖ Logistic Regression
- ❖ Small Neural Network

Although we tried out several models, our intuition was to try out tree-based algorithms. Because the available number of features are limited and due to that other algorithms didn't perform better than tree base algorithms such as Random Forests and XGBoost.

The following table represents the accuracy values of clustering tasks.

Model	Validation Accuracy	Test Accuracy
XGBoost	99.979%	98.16%
Random Forests	99.979%	96.06%
K-Nearest Neighbour	99.979%	92.58%
Logistic Regression	99.978%	96.06 %
Small Neural Network	99.979%	96.06%

According to the accuracy values, the XGBoost algorithm showed the best performance among other models. The reason for this is the limitations of input features.



## 7.1 XGBoost Algorithm

XGBoost is a machine learning algorithm which is highly effective for classification and regression tasks. This algorithm builds an ensemble of decision trees in a sequential manner, where each tree aims to correct the errors made by the previous trees. This process increases the performance and convergence speed of the algorithm. As well as due to gradient boosting algorithm, enhance the effectiveness of the algorithm by introducing ensemble techniques for predictions.

In addition to that, in this framework performance enhancing techniques were implemented along with the main algorithm such as “Regularization”, “Tree Pruning” and “Parallel Processing”. These techniques generalize the model and give predictions as fast as possible.

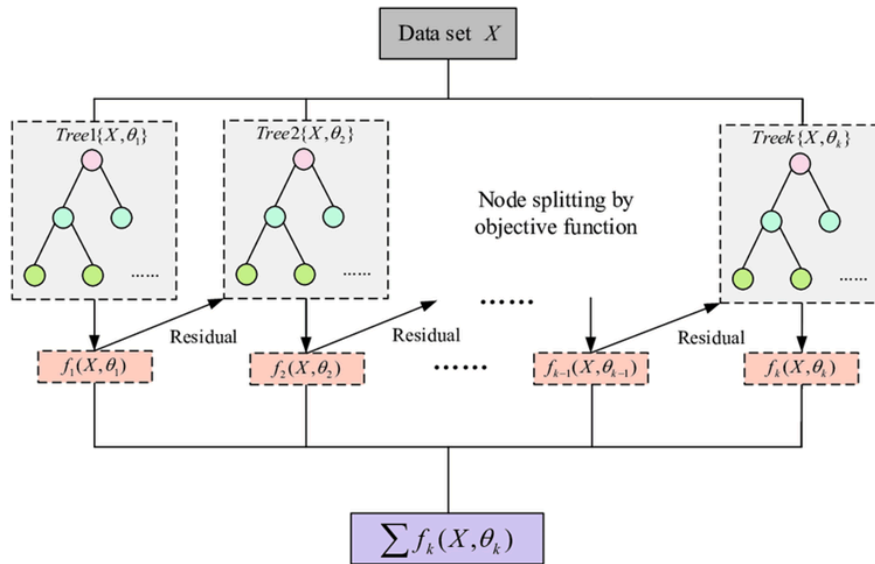


Figure () : XGBoost Algorithm

Due to high precision and fast inference capability, we decided to select XGBoost algorithm as our machine learning algorithm in the data analyzation pipeline.

## 7.2 Random Forests

This is also a tree-based algorithm and this algorithm operates by constructing multiple decision trees during training and outputting the class that has the maximum prediction count from each individual tree. The accuracy of the algorithm also showed considerable performance and other techniques with fast inference capability although this is an ensemble technique.

## 7.3 K-Nearest Neighbour

In this algorithm, we consider a voting technique to find out the relevant cluster by measuring distance from cluster centre to given data points. So, this technique gave good accuracy for both training and validation, but test accuracy value showed decrement with respect to tree-based techniques.



Figure () : Cluster Creation of K-Nearest Neighbor Algorithm

## 7.4 Logistic Regression

Logistic regression is a linear classifier and this creates linear decision boundaries around each cluster category. This is a simple algorithm with respect to other algorithms. Due to the simplicity of the algorithm, it was unable to learn the behaviour of the dataset in a precise manner. Hence, the test accuracy was a bit less than the best model.

## 7.5 Neural Network

Here, we have designed a small neural network with two dense layers of size 64 and 32. “Relu” activation function was used for internal dense layers and “Softmax” activation was used for the last layer with 6 neurons. The input layer was designed according to the number of input features. “Categorical Cross Entropy” loss was used as the loss function and “Adam” optimization algorithm was used.

Due to limitations in the number of features, this model didn’t perform well with respect to other best performing models. But this model also showed high accuracy value for validation dataset.

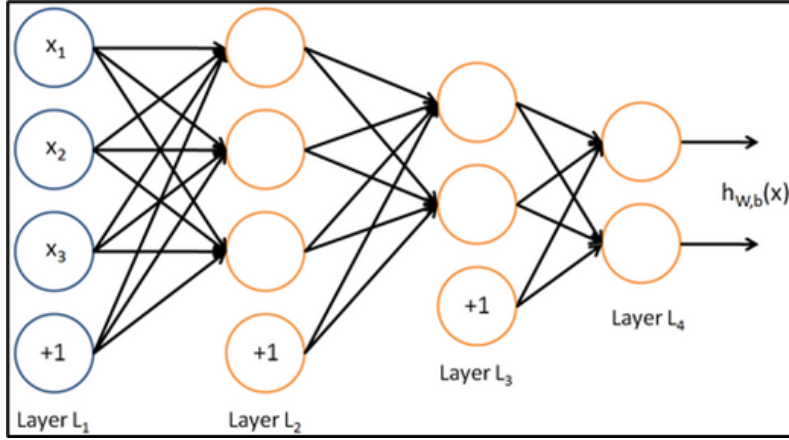
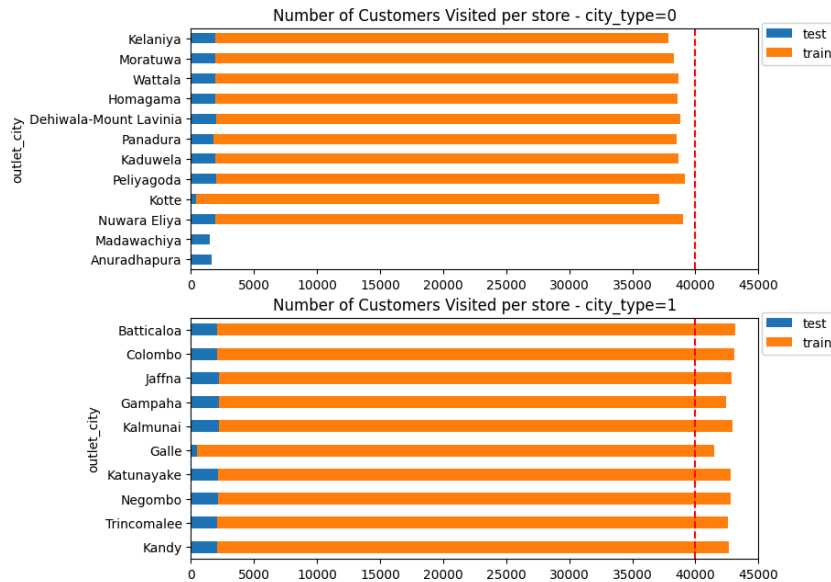


Figure () : Basic Architecture of Small Neural Network with Dense Layers

## 8. Challenges in Model Training

### 8.1 Absence of certain outlets from the training dataset



The outlet cities of Anuradhapura and Madawachiya are absent from the training dataset. Hence it is required to infer the cluster type for these outlets without sufficient training as well. While part of the challenge is mitigated by not using the city encoded features to distinguish each outlet store, the *city\_type* of each outlet in Anuradhapura and Madawachiya could not be determined for certain.

The *city\_type* feature is mainly determined using the number of customers (the size

of traffic to the store), such that, type 0 had less than 40,000 customer records and type 1 had greater than 40,000 customer records in the training dataset. Hence, owing to the lower number of records available for the outlets in Anuradhapura and Madawachiya, the *city\_type* was set to type 0 as a work around.

## 8.2 High Time Consumption of Training of Small NN model

We implemented a small neural network with 4 dense layers. Although the number of parameters is low, due to the high amount of training data, the training process was a highly time-consuming task. As well as, in Kaggle environment, GPUs are not available. Because our account was verified. Due to this issue, we were unable to use higher batch sizes in the training process. In addition to that, the computational capability is a bit low value with respect to GPUs which are specially designed for parallel processing. So, batch processing time was also high. Therefore, the overall training of one epoch task is around 2 minutes even though we had structured data with a small machine learning model.

## 9. Description of Classified Customers

Following the characteristics described in section 6, the clusters can be identified using this table,

Cluster Type	City Type Based on Customer Traffic	Most Preferred Item Category
Cluster 1	Type 1 (High Customer Traffic)	Dry Items
Cluster 2	Type 0 (Low Customer Traffic)	Fresh Items
Cluster 3	Type 1 (High Customer Traffic)	Luxury Items
Cluster 4	Type 0 (Low Customer Traffic)	Dry Items
Cluster 5	Type 0 (Low Customer Traffic)	Luxury Items
Cluster 6	Type 1 (High Customer Traffic)	Fresh Items

Hence the descriptions for each customer cluster would be defined as,

- **Cluster 1 - Customers from high traffic outlets preferring dry items**
- **Cluster 2 - Customers from low traffic outlets preferring fresh items**
- **Cluster 3 - Customers from high traffic outlets preferring luxury items**
- **Cluster 4 - Customers from low traffic outlets preferring dry items**
- **Cluster 5 - Customers from low traffic outlets preferring luxury items**
- **Cluster 6 - Customers from high traffic outlets preferring fresh items**

## 10. Suggested Marketing Strategies

To enhance the effectiveness of KJ Marketing's strategies using the identified customer clusters, the company can adopt several personalised and data-driven approaches. Below, is an outline of the possible strategies that can be carried out based on the aforementioned customer segments:

### 1. Personalized Promotions and Discounts

- Cluster 1 (High Traffic, Dry Items):

High-traffic customers who prefer dry items are likely bulk buyers, and bulk purchase discounts can increase sales volume and customer retention.

- Strategy: Offer bulk purchase discounts and loyalty rewards.
  - Example: Provide discounts on bulk purchases of grains, cereals, and canned goods.
- Cluster 2 (Low Traffic, Fresh Items):  
Low-traffic customers preferring fresh items might visit more frequently for perishable goods; regular promotions can boost their shopping frequency.
  - Strategy: Promote weekly specials on fresh produce.
  - Example: Weekly deals on fruits, vegetables, and dairy products.
- Cluster 3 (High Traffic, Luxury Items):  
High-traffic customers seeking luxury items value exclusivity; special events and offers can enhance their shopping experience and brand loyalty.
  - Strategy: Introduce exclusive member-only sales and limited-time offers.
  - Example: VIP events for premium product launches, early access to new luxury items.
- Cluster 4 (Low Traffic, Dry Items):  
Low-traffic customers can benefit from convenience; subscription services ensure regular purchases and customer retention.
  - Strategy: Implement subscription services for staple items.
  - Example: Monthly subscriptions for essentials like pasta, rice, and snacks.
- Cluster 5 (Low Traffic, Luxury Items):  
Personalised marketing can make low-traffic luxury item buyers feel valued, encouraging repeat purchases.
  - Strategy: Focus on personalised communication and targeted advertising.
  - Example: Personalised emails with recommendations based on past luxury purchases.
- Cluster 6 (High Traffic, Fresh Items):  
High-traffic customers focused on fresh items value quality and experience; guarantees and in-store events can enhance satisfaction and loyalty.
  - Strategy: Introduce freshness guarantees and in-store experiences.
  - Example: Freshness guarantee on all produce and cooking demonstrations using fresh ingredients.

## **2. Optimising Product Placement and Store Layout**



- Cluster 1 & 4 (Dry Items):  
Maximising visibility for frequently purchased items can drive impulse buys and ease of access.
  - Strategy: Position dry goods prominently at the entrance and along high-traffic aisles.
- Cluster 2 & 6 (Fresh Items):  
Fresh items attract frequent visits; a well-organised layout can enhance shopping convenience and satisfaction.
  - Strategy: Ensure fresh produce sections are well-organised, visually appealing, and located at the store's forefront.
- Cluster 3 & 5 (Luxury Items):  
Segregating luxury items enhances their perceived value and provides an exclusive shopping experience.
  - Strategy: Create exclusive sections or aisles dedicated to luxury items.

### **3. Leveraging Data Analytics for Customer Insights**

- Cluster Analysis:  
Dynamic segmentation based on the latest data can help tailor marketing strategies more effectively. Thus, continuously analyse sales data to refine customer segments and predict trends.
- Customer Feedback:  
Direct feedback helps in fine-tuning offerings and improving customer satisfaction. Thus, collect and analyse feedback to understand customer preferences and pain points.

### **4. Enhancing Digital Marketing Campaigns**

- Targeted Advertising:  
Targeted ads increase relevance and effectiveness, driving higher engagement and conversions.
  - Strategy: Use social media and email campaigns tailored to each cluster's preferences.
  - Example: Advertise bulk dry goods on platforms frequented by Cluster 1 customers, and promote fresh item deals in Cluster 2's preferred channels.
- Loyalty Programs:  
Loyalty programs incentivize repeat business and enhance customer lifetime value.
  - Strategy: Develop loyalty programs offering rewards based on purchase patterns.
  - Example: Points systems for repeat purchases of preferred categories.
- E-commerce and Delivery:

Convenience of online shopping caters to both high and low traffic customers, enhancing overall sales.

- Strategy: Strengthen online shopping and delivery options, particularly for fresh and luxury items.
- Mobile Apps:  
Mobile apps provide a direct channel for personalised marketing and customer engagement.
  - Strategy: Develop or enhance a mobile app with personalised offers and easy reordering features.

By leveraging the customer clusters identified from the sales data, KJ Marketing can develop highly targeted and personalised marketing strategies. These strategies should focus on tailored promotions, optimised store layouts, data-driven customer insights and targeted digital marketing. Implementing these strategies will help KJ Marketing better meet customer needs, increase sales, and build stronger customer loyalty.