DataStorm 5.0
Final Report

GPT-5X
(DataStorm502)
University of Colombo School of computing

Introduction.

KJ Marketing, a leading retail supermarket chain in Sri Lanka, operates 22 outlets across urban and suburban regions, offering a wide range of products including dry goods, fresh items, and luxury products. Recently, the company has observed that conventional marketing strategies are no longer effective in engaging their customer base. To address this, KJ Marketing aims to adopt a personalized marketing strategy tailored to individual customer preferences. Utilizing historical sales data, the company seeks to classify new customers into one of six identified segments based on their purchasing behavior. This classification will enable KJ Marketing to enhance their marketing strategies and better cater to the needs of their diverse customer base.

Code Repository Link:

https://github.com/visith1577/DataStorm5.0

1.Elaborate on the methodologies implemented to address missing values, duplicates and outliers within the dataset? Please describe any specific techniques used for imputation or exclusion, and the rationale behind these choices

Handle missing values

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 774155 entries, 0 to 774154
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Customer_ID       774153 non-null  float64
 1   outlet_city       774153 non-null  object
 2   luxury_sales      774120 non-null  object
 3   fresh_sales       774114 non-null  object
 4   dry_sales         774125 non-null  object
 5   cluster_catgeory  774154 non-null  object
dtypes: float64(1), object(5)
memory usage: 35.4+ MB
```

We can see that there are missing values in our dataset.
Missing values in **Customer_ID** refers to the lack of user_id, customer_id is not necessary in the training process, there fore no need to drop them.
Missing values in **outlet_city** may or may not be needed.

We opt not to blindly drop sales missing values, instead we opt to use median value corresponding to each category to fill the missing values.

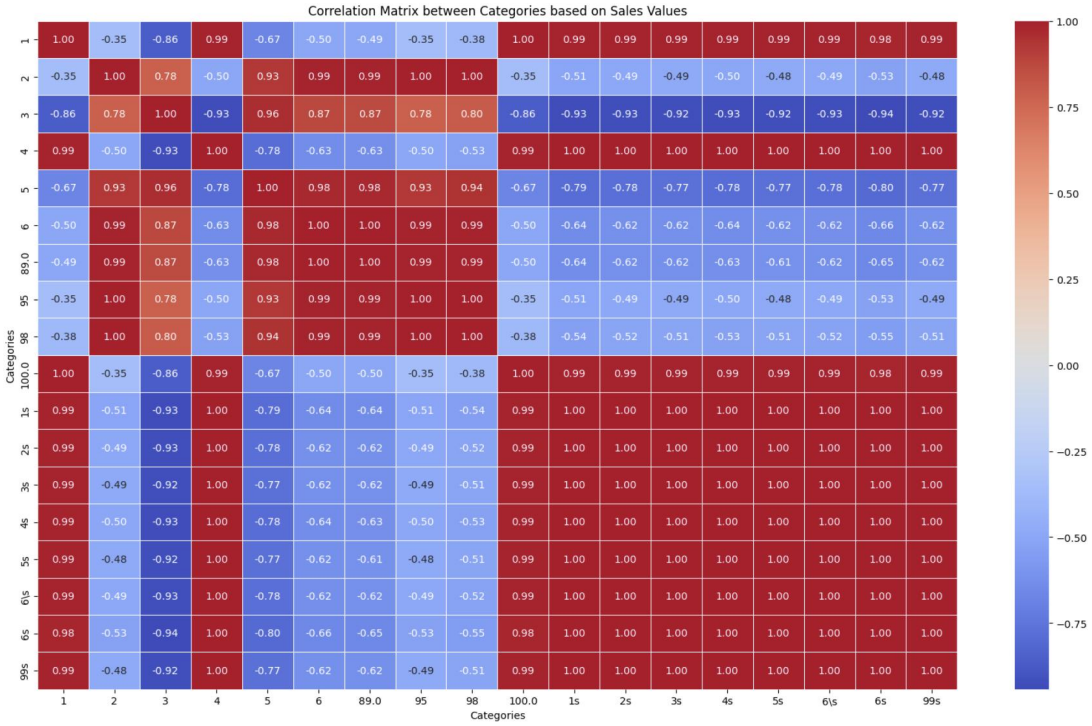
But before that it is necessary to handle corrupt data in all columns.
starting from cluster_category if we peek at our initial dataset.
we can see that there are more than the mentioned 6 labels.

```
train_df['cluster_catgeory'].value_counts()
```

```
cluster_catgeory
1         188975
6         169206
2         155060
4         131039
3          48906
4          41400
5          39531
5              9
1              9
6              8
2              4
3              1
6\             1
95             1
98             1
99             1
100.0          1
89.0           1
Name: count, dtype: int64
```

It is necessary that we either drop the mis-labelled data or find the apropriate label. If we get a correlation matrix between each label

Note: the labels append with s are string labels while the rest are integers



- we can see that the labels [100, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘6\’, ‘99’] are both could be either 1 or 4 mislabelled.
- Since higher correlation with 4 and that ‘4’ has a non negligible amount of values
- assume that all are in category 4.
- 89, 95, 98 are negligible. therefore they will be dropped.
- All the nan cluster_categories will be dropped.

For sales values we can see that some are in string format while some are float.
Then manually convert string value to float values.
We will detect the string values and change them to float for all 3 sales categories.

```python
def convert_to_float_and_find_errors(dataframe, column):
    error_rows = []
    for index, value in dataframe[column].items():
        try:
            float(value)
        except ValueError:
            error_rows.append((index, value))
    return error_rows

# Convert 'luxury_sales' column to float and find rows causing ValueError
error_rows = convert_to_float_and_find_errors(train_df, 'luxury_sales')

# Print rows causing ValueError
print("Rows causing ValueError:")
for index, value in error_rows:
    print(f"Index: {index}, Value: {value}")
```

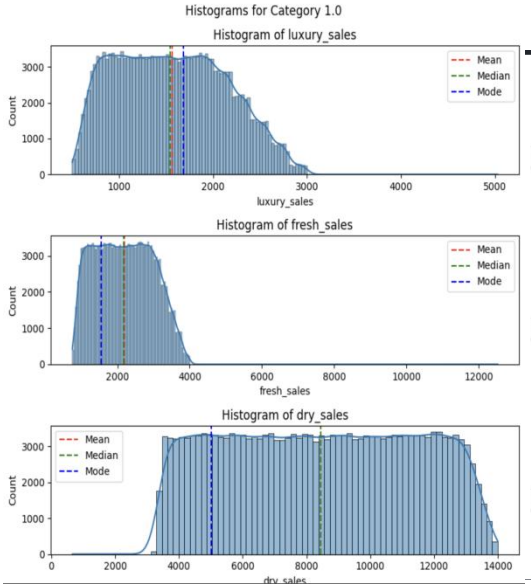
```
Rows causing ValueError:
Index: 80043, Value: One thousand four hundread ruppes
Index: 175278, Value: nul
Index: 296621, Value: nul
Index: 297911, Value: Eight hundread ruppess
Index: 326593, Value: six hundread and hirty
Index: 367935, Value: nul
Index: 497177, Value: Thousand tow hundread
Index: 497245, Value: seven hundread and nine ruppees
Index: 558562, Value: Three thousand two hundread ruppess
Index: 753131, Value: Four thousand one hundread ruppess
```

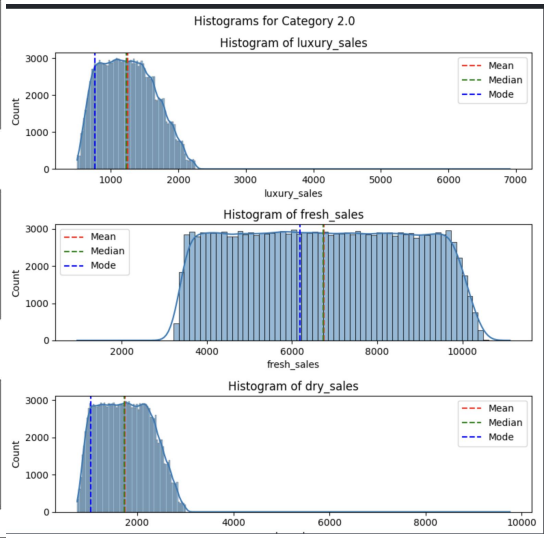
```python
train_df['luxury_sales'][80043] = 1400.0
train_df['luxury_sales'][297911] = 800.0
train_df['luxury_sales'][326593] = 630.0
train_df['luxury_sales'][497177] = 1200.0
```

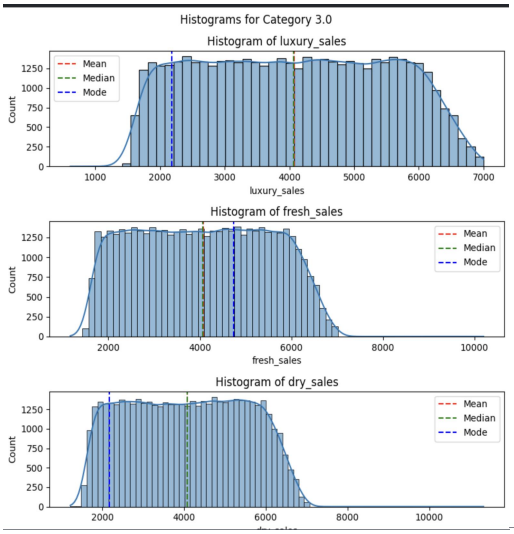
finally for null values take the median under each category corresponding to the column and fill the rows.

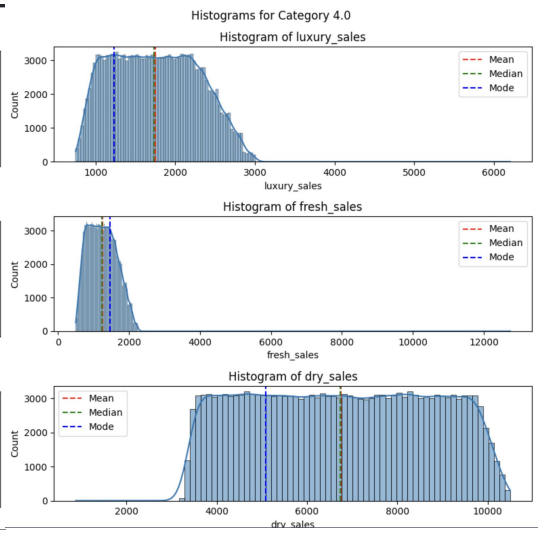
If we plot histograms for sales value distribution for each category, we can see that median == mean in all histograms

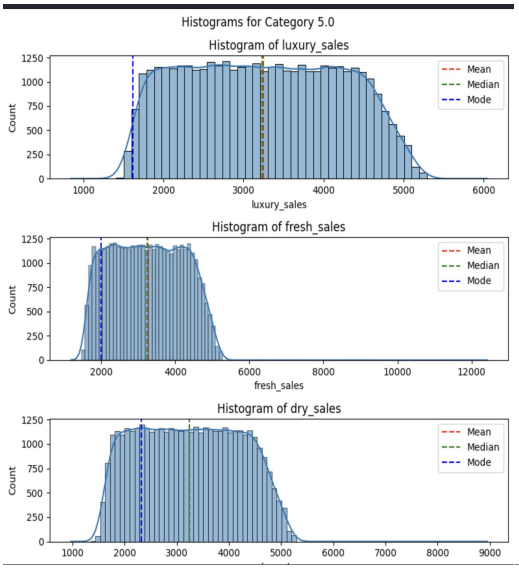
We can also see that most of the hist plots are skewed.
Hence we filled missing values with median.

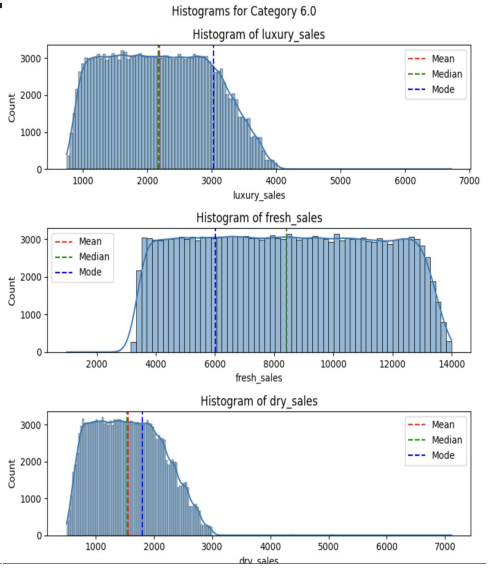

Histograms for Category 1.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales
dry_sales


Histograms for Category 2.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales
dry_sales


Histograms for Category 3.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales

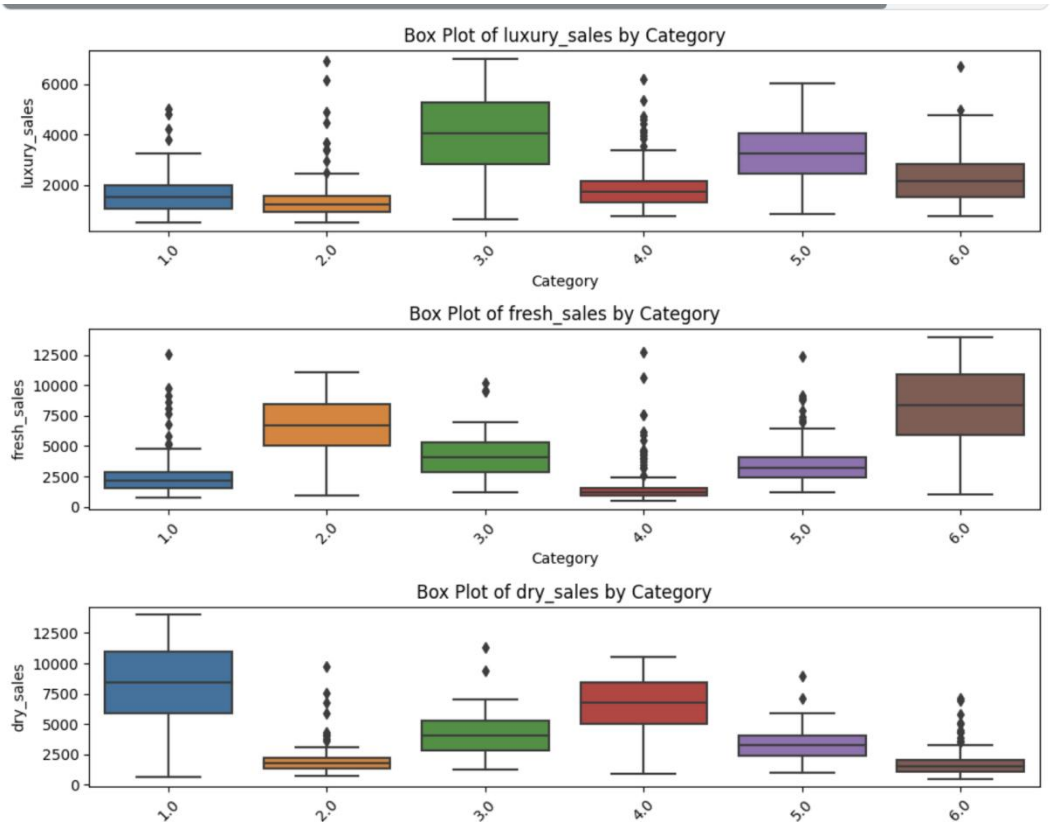

Histograms for Category 4.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales
dry_sales


Histograms for Category 5.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales


Histograms for Category 6.0
Histogram of luxury_sales
Histogram of fresh_sales
Histogram of dry_sales
Mean
Median
Mode
Count
luxury_sales
fresh_sales
dry_sales

For outlet_city, after getting correlation map we could not come to any meaning ful conclusion.

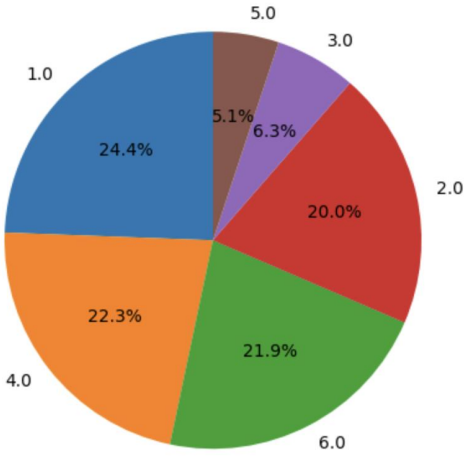
Next we remove outliers, finding outliers by drawing box-plots.



Finally all sales value are changed to float64 (2 precision)

The final dataset after cleaning the dataset has shape (774147, 7) for train set.

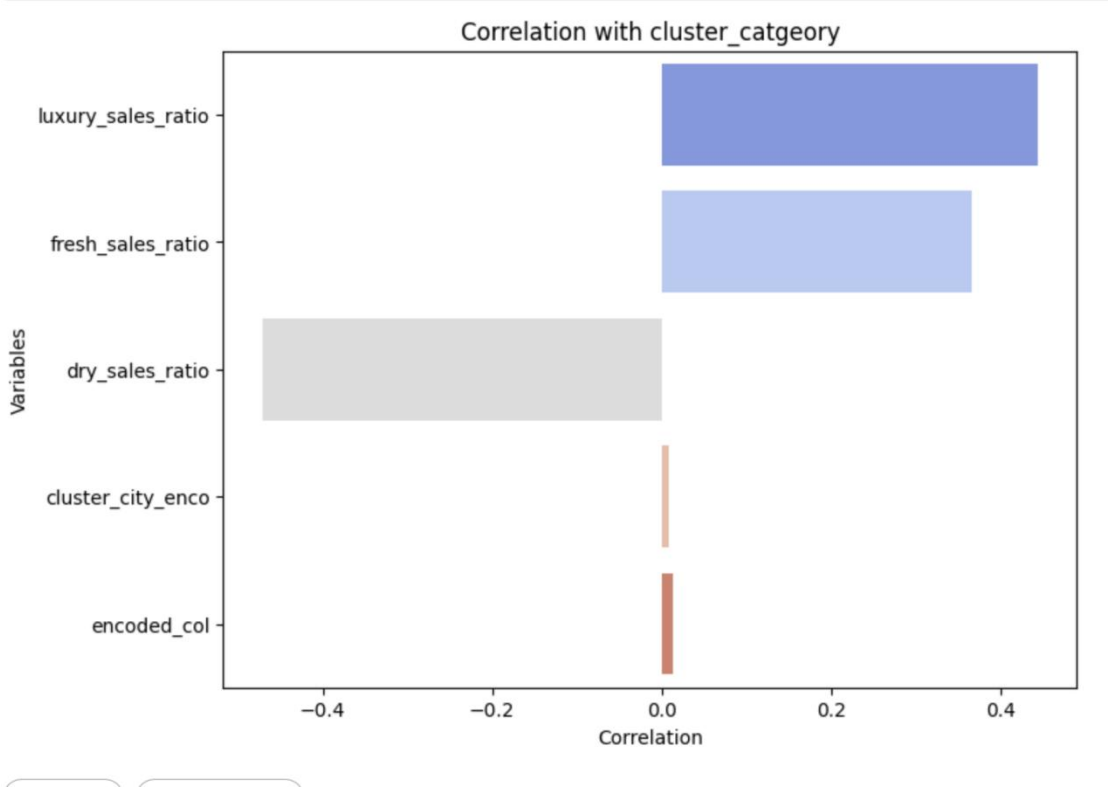
If we peek at the daat distribution for each category. we can see that the dataset is unbalanced. We wil handle that later.



2 .Explain the features you chose for the above task. How did you determine their relevance to the Problem?

In this data set there are six columns.Customer_ID not relevant to solve this problem; it is unique feature.and the problem is to segment the customers so cluster_category is the target and we cannot get it is as a feature.

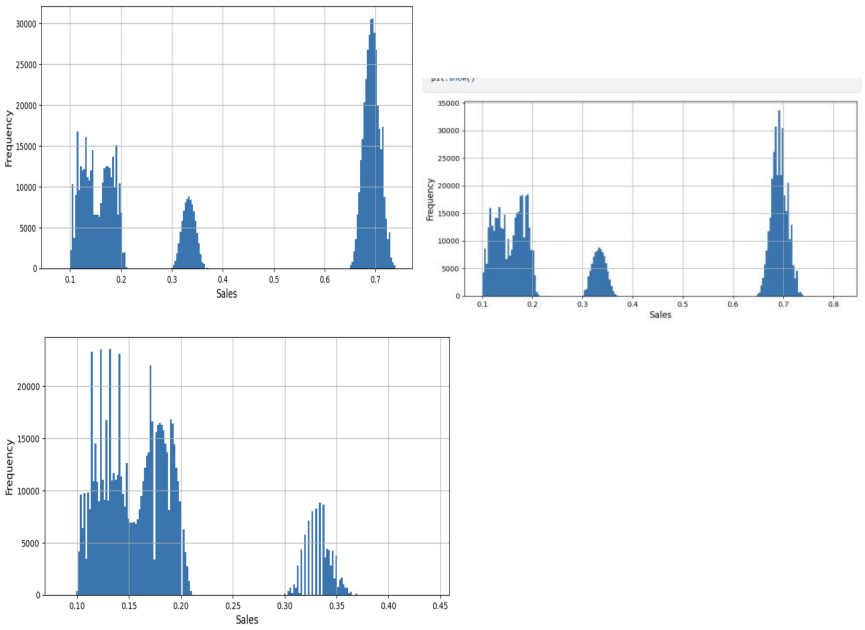
We will take the ratio of <sales_type> / total sales to represent the luxury_sales_rato, dry_sales_ratio, fresh_sales_ratio.



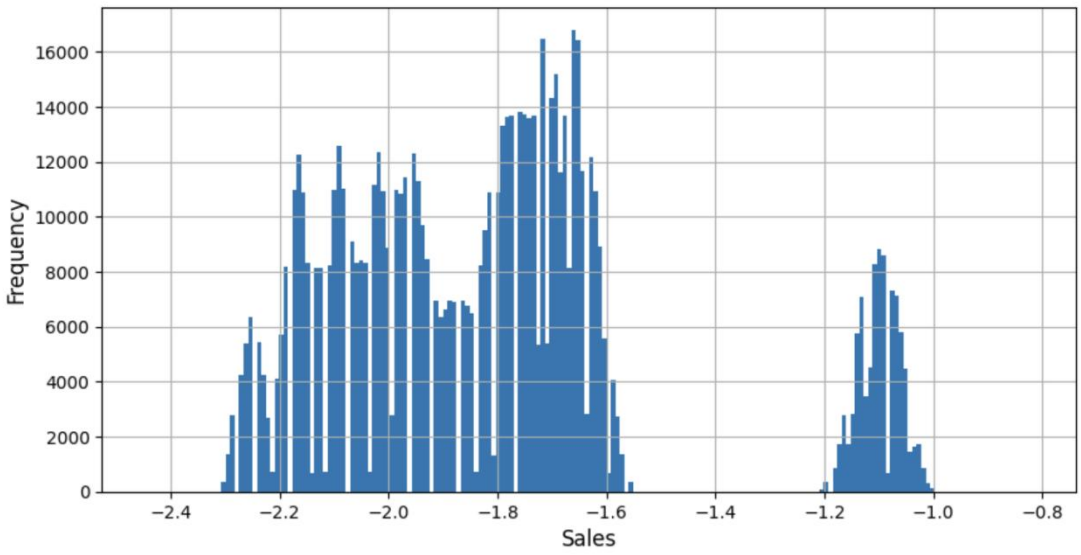
- from this graph we can see that the ratio has high correlation with the cluster_catgeory
- here encoded_col is the outlet_city column encoded to int values, as you can see that column has lee correlation with our categories.
- Hence we can drop outlet_city/ encode_col
- We will use luxury_sales_rato, dry_sales_ratio, fresh_sales_ratio as our main features.

3 .Has feature scaling or normalization been applied to the data? If so, which methods were utilized and explain how these techniques improve the performance of the model?

As stated above we scaled the sales values to 0 - 1 by taking the ratio.
In our data we previosly found how our dataset is unbalanced.

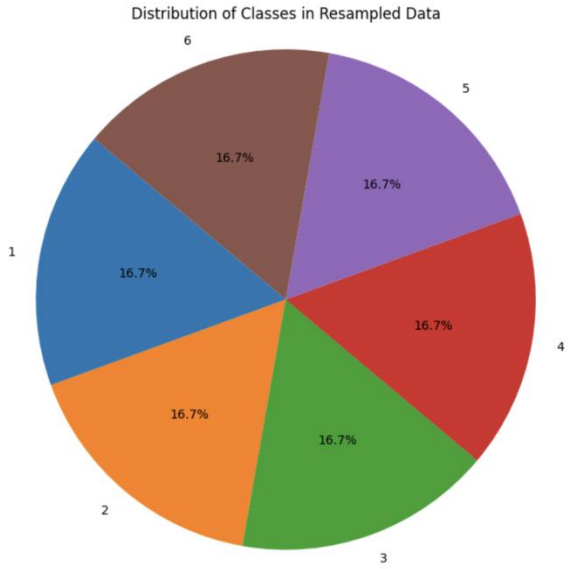


Now our dataset is skewed, therefore we take the log values.



To handle Unbalanced nature of the dataset we use Oversampling with SMOTE paired with undersampling via RandomUnderSampler.

Our dataset is then made into a balanced dataset using imbalance learn pipeline with oversampling and undersampling.



As you can see now our new dataset is balanced.
Finally in the train set use KMeans clustering to find the outliers and remove them.

4 .Have you used any encoding strategies? Provide a comprehensive explanation of the chosen encoding methods and their impact on the model's input requirements and performance.

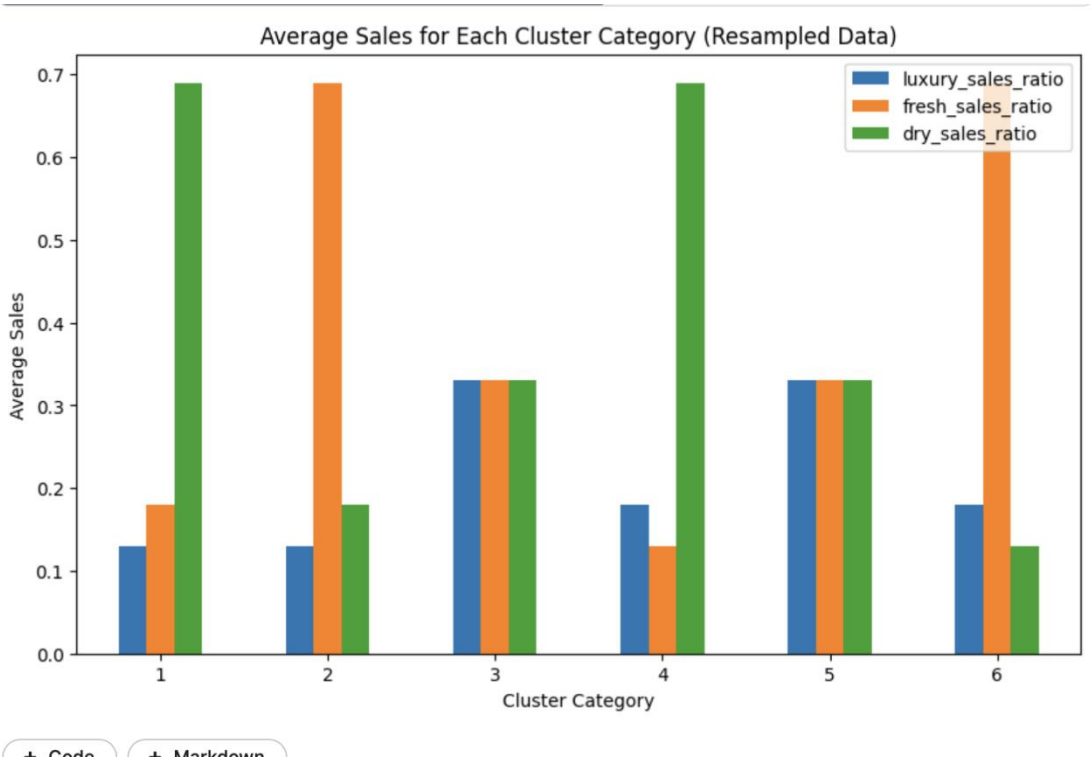
In our initial tests we encoded outlet_city with a custome encoder, how ever the impact of that column on the final result was negligible.
Hence we opt to drop it.

5) **How do the features correlate with the target variable, and are there any notable inter-feature Relationships?**

As per previose mentioned we found several correlations, most notable ones for the final answer.

1. Luxury sales have a strong correlation with total sales.
2. Dry sales and fresh sales have strong negative correlation.
3. Luxury sales and fresh sales have a stronger correlation compared to dry sales and Luxury sales.

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



The target variable in the graph is the average sales for each cluster category, represented as a percentage of the total sales for each cluster category. The x-axis of the graph shows the cluster categories, labeled 1 through 6. The y-axis shows the average sales ratio.

We can see that cat , 41 purchases more dry sales
cat 2, 6 purchases more fresh sales
cat 4 purchases more luxury sales than cat 1
while cat 6 purchases more luxury sales than 2
cat 3 and 5 look exactly the same in the above graph. (model will have difficulty in uniquely identifying these two categories.

7.What are the algorithms you considered for this problem, and why did you choose the final Algorithm

- We attempted models with XGBoost, CatBoost, LightGBM, RandomForest Classifier and 4 layer ANN.
- From the above models all models performed + 0.90 on the accuracy metric for test set. with XGBoost and ANN having accuracy 0.95 - 0.96.
- In our final evaluation with different test set splits XGBoost managed to consistently score 0.95 while catboost and lightGBM struggled.
- XGBoost was much simpler to hyper parameter tuning, Hence final choice was XGBoost.

8. Were there any challenges faced during model training, such as over-fitting or computational Constraints

Training an ANN algorithm with no of layers > 10 was not possible due to GPU constraints.
Running GridSearch with StratifiedKFold to finetune and choose model was not pos sible within the time constraint.

9 .Briefly define and explain all the classified clusters while providing appropriate names.

Category 1: Dry Sales Dominant

Description: This segment exhibits a significantly higher averagesales ratio of dry good sales compared to other categories. The total sales for dry goods in this segment exceed the combined sales of fresh and luxury items.
Appropriate Name: "Dry Goods Shoppers"

Category 2: Fresh Sales Dominant

Description: Customers in this segment have a higher average of fresh item sales, with dry goods sales following closely.
Appropriate Name: "Fresh Goods Shoppers"

Category 3: Balanced Preferences

Description: Segment 3 does not show a clear distinction in preferences based on the provided data. Their purchasing habits for dry goods, fresh items, and luxury items are similar to Segment 5.
Appropriate Name: "Balanced Shopper A"

Category 4: High Dry and Luxury Sales

Description: This segment has higher sales in dry goods and also a notable interest in luxury items. The combination of these preferences indicates a more varied shopping pattern with a lean towards premium products.
Appropriate Name: "Premium Dry Shoppers"

Categories 5: Indistinguishable Preferences

Description: Similar to category 3
Appropriate Name: "Balnced Shopper B"

Categories 6: High Fresh Items and Luxury Sales

Description: Customers in this segment have a significantly higher average of fresh item sales, indicating a strong preference for fresh produce and perishable goods.
Appropriate Name: "Premium Fresh Shoppers"

10 .How can your solution enhance the effectiveness of the company's marketing strategies based on the classified clusters?

Based on the provided customer classifications, here are some potential strategies to enhance the effectiveness of the company's marketing efforts:

- Dry Goods Shoppers:
  Focus marketing campaigns on promoting bulk purchases, value pac ks, and subscription services for dry goods. Highlight the conveniene, cost-savings, and long shelflife of dry goods. Consider offering bundled deals or discounts on popular dry goods items.

- Fresh Goods Shoppers:
  Emphasize the freshness, quality, and health benefits of the fresh produce and perishable items. Promote seasonal offerings, locally sourced products, and recipe ideas incorporating fresh ingredients. Offer loyalty programs or rewards for frequent purchases of fresh items.

- Balanced Shopper A and Balanced Shopper B:
  For these segments with diverse preferences, adopt a well-rounded marketing approach. Highlight the variety and versatility of the product offerings, catering to different dietary needs and preferences. Promote meal planning ideas that incorporate a balance of dry, fresh, and luxury items.

- Premium Dry Shoppers:
  Target this segment with marketing campaigns focused on premium and gourmet dry goods, such as specialty grains, nuts, spices, and imported items. Emphasize the quality, uniqueness, and taste profiles of these products. Consider partnering with influencers or food bloggers to showcase recipes and product reviews.

- Premium Fresh Shoppers:
  Focus on the exclusivity, freshness, and superior quality of the premium fresh items. Highlight the farm-to-table concept, organic offerings,and sustainable sourcing practices. Collaborate with high-end restaurants or chefs to promote recipes and cooking tips using premium fresh ingredients.