

UTM

UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
SEMESTER II SESSION
2025/2026

SECP2753-02 PERLOMBONGAN DATA
(DATA MINING)

PROJECT 2

NAME	MATRIC NUMBER
KAVIVARTHAN A/L MANNIVANAN	A24CS0093
RICHARD YONATHAN JULIO CLAY HENG	X25EC3019
NANDINI VENGAT	X25EC3021
NIVEETHITA A/P PANDIA RAJAN	A24CS0148
KALAITHARAN A/L PALANYVELU	A24CS0091

Table of Content

1.0 Introduction **3**
2.0 Business and Data Understanding **4**
2.1 Problem Statement 4
2.2 Dataset Overview 4
3.0 Data Preparation & Preprocessing **6**
3.1 Overview of the Data Preparation 6
3.2 Preprocessing Step Involved 6
3.3 Sample Preprocessing Inputs and Outputs 7
4.0 Model Development **10**
4.1 Overview of Selected Algorithms 10
4.2 Justification of Algorithms Selection 10
4.3 Selected Algorithm Development 10
4.3 Determination of the Number of Clusters 11
5.0 Model Performance and Evaluation **12**
5.1 Clustering Evaluation Metrics 12
5.2 Analysis of Model Performance & Comparison 12
6.0 Visualization and Discussion of Results **13**
6.1 Model Comparison Scatter Plot Analysis 13
6.2 Cluster Profiling 13
6.3 Algorithm Comparison 14
6.4 Business Insights 15
7.0 Conclusion **16**
8.0 References **17**

1.0 Introduction

Online retail stores receive large amounts of transactions and they have a wide range of customer profiles. In order to expand their business further raw transactional records need to be transformed into accurate marketing strategies. In the modern e-commerce landscape, vast transaction logs are generated daily. This project applies data mining methodologies to group online retail customers based on common purchasing behavior to increase the accuracy of targeted customer relationship management. Clustering analysis is an unsupervised machine learning technique that partitions a heterogeneous dataset into groups of distinct clusters. Unlike supervised learning, clustering operates without predefined class labels by discovering their unique structural patterns through minimizing intra-cluster distances and maximizing inter-cluster distances.

The project aim is to :

1. Create meaningful customer behavior profile using the Recency, Frequency, and Monetary (RFM) model from the online retail transaction dataset
2. Build and compare two unsupervised machine learning models applied which are K-Means Clustering and Agglomerative Hierarchical Clustering.
3. Evaluate the clustering results, analyze the customer groups, and identify insights that help reduce unnecessary marketing costs and improve customer retention.

The dataset captures real-time continuous transaction records from online retail platforms. This dataset is apt for clustering because it has a large volume of various transactions with different expenditure levels such as casual purchases or wholesale ranged purchases. Since these customer behaviors lack predefined categorical labels, this specific dataset helps unsupervised data mining approach to discover the hidden retention and spending patterns.

2.0 Business and Data Understanding

2.1 Problem Statement

Retail has a large number of customer transactions daily and managing these logs to design marketing frameworks is inefficient. Without the knowledge of individual

consumer behaviors there is a risk of budget waste in the aspect of advertisements, discounts, and promotions that is not catered to specific audiences. Moreover, the inability to identify and appreciate valuable loyal shoppers can result in less customer retention. Therefore, online retail businesses require an automated and reliable approach to segment their customer base.

The problem addressed in this project is the difficulty of identifying hidden customer purchasing patterns and grouping online retail customers based on their buying behaviors. By analyzing transaction data, customer groups can be discovered based on factors such as how recently they made a purchase, how often they purchase, and how much they spend. Data mining techniques, especially unsupervised clustering methods, can help automatically identify these customer segments and provide useful information for targeted marketing strategies.

This project aims to analyze customer purchasing behavior and transaction details to develop a three-level customer segmentation model. The findings can help retail businesses reduce unnecessary marketing expenses, improve customer retention, and identify opportunities for cross-selling products. Since customer segmentation focuses on finding patterns without predefined categories, it is suitable for exploratory data analysis and unsupervised learning techniques in data mining.

2.2 Dataset Overview

Attribute	Description
InvoiceNo	Unique 6-digit transaction identification number
StockCode	Unique product identification code
Description	Nominal text name of the product purchased
Quantity	The total units of product per transaction line item
InvoiceDate	The exact date and time the transaction was generated
UnitPrice	The financial cost per single product unit
CustomerID	Unique 5-digit customer account identification number
Country	The location name of the customer's country of residence

Table 2.1 Attributes List

The online retail dataset contains information about past customer transactions, including purchasing, operational, and financial details. The raw dataset consists of

multiple transaction records, where each attribute represents different aspects of customer buying behavior. The variables include categorical attributes such as InvoiceNo, StockCode, Description, CustomerID, and Country, numerical attributes such as Quantity and UnitPrice, and a date attribute, InvoiceDate. Together, these variables help describe customer purchasing patterns, including purchase frequency, buying time intervals, and total spending.

Among the variables, CustomerID is important because it allows individual transactions to be grouped into customer profiles. Quantity and UnitPrice are used to calculate the total amount spent by each customer, while InvoiceDate helps determine how recently a customer made a purchase. These attributes are transformed into Recency, Frequency, and Monetary (RFM) features, which help identify different customer groups, such as inactive customers, regular buyers, and high-value customers.

The dataset contains different types of data, including numerical, date, and text data. Therefore, data preprocessing techniques such as handling missing values, removing duplicate records, applying log transformation to reduce data imbalance, and normalizing features are required before applying clustering algorithms.

Categorical Variables	- InvoiceNo - StockCode - Description - CustomerID - Country
Numerical Variables	- Quantity - UnitPrice
Date Variables	- InvoiceDate

Table 2.2 Datatypes of each Attributes

This dataset is important because it contains detailed transaction records for each customer. The raw transaction-level data allows us to analyze customer purchasing behavior accurately before grouping and summarizing the data. By using this information, we can create customer profiles based on their buying patterns and identify different types of customers.

Expected business insight from the analysis is identifying three customer groups, which are high-spender tier, consistent core tier and inactive tier.

- High-Spender Tier are the customers who contribute a large portion of the company's revenue. These customers can be rewarded through loyalty programs to maintain their relationship with the business.
- Consistent Core Tier are regular customers who make frequent purchases but have potential to spend more. Businesses can use promotions, product recommendations, and bundle offers to increase their spending.

- Inactive Tier are customers who have reduced or stopped purchasing. These customers can be targeted with special offers or reminder campaigns to encourage them to return.

3.0 Data Preparation & Preprocessing

3.1 Overview of the Data Preparation

Raw real-world transactional data is fundamentally messy and poorly suited for direct distance-based machine learning algorithms. The data preprocessing pipeline handles data cleansing, behavioral feature engineering, outlier management, and statistical standardization.

3.2 Preprocessing Step Involved

1. Handling of Missing Values

   Rows lacking a valid identifier in the CustomerID column are completely omitted via dropna() because clustering requires identifying distinct consumer profiles across historical behaviors, rows without an explicit customer token cannot be assigned to a group.

2. Handling of Duplicate Data

   Identical transactional rows represent data logging artifacts or system duplication errors. The pipeline executes drop_duplicates() to retain only unique transactions, preventing synthetic inflation of frequency metrics.

3. Filtering Anomaly Records

   Data integrity checks reveal structural records representing returns, cancellations, or administrative financial adjustments. The system filters these out using Pythondf_clean = df_clean[(df_clean['Quantity'] > 0) & (df_clean['UnitPrice'] > 0)]. This restricts the dataset exclusively to positive, revenue-generating commercial behaviors.

4. Feature Engineering

   To group customers effectively, transactional records are aggregated at the customer level into three essential behavior features. Recency is the number of days elapsed between the customer's most recent purchase date and the absolute maximum date present in the log. Frequency is the grand total count of unique transactions recorded under that specific customer ID. Monetary is the cumulative sum of all financial capital spent by the customer, computed by aggregating the product of Quantity and UnitPrice.

5. Log Transformation

   Retail features like Frequency and Monetary metrics naturally follow heavy right-skewed power-law distributions, where a small handful of customers spend massively compared to the average. Because K-Means is heavily influenced by scale and spatial variance, a log-transformation using $y = ln(x + 1)$ is applied to

compress these wide ranges, stabilizing variance and creating a more normal distribution.

6. Normalization and Scaling

Distance calculations are sensitive to varying scales such as Recency spans hundreds of days whileFrequency might span single digits. Features are uniformly transformed via a standard scaler to a mean of zero and standard deviation of one.

3.3 Sample Preprocessing Inputs and Outputs

```
print("--- Loading and Cleaning Data ---")
df = pd.read_csv('online_retail.csv')

print("--- Sample Raw Input ---")
print(df.head(), "\n")

df_clean = df.dropna(subset=['CustomerID'])

df_clean = df_clean[(df_clean['Quantity'] > 0) & (df_clean['UnitPrice'] > 0)]

df_clean = df_clean.drop_duplicates()
print(f"Original shape: {df.shape} -> Cleaned shape: {df_clean.shape}\n")

print("--- Sample Cleaned Output ---")
print(df_clean.head(), "\n")
```

Diagram 3.1 Code Snippet of

```
--- Loading and Cleaning Data ---
--- Sample Raw Input ---
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1    536365     71053                  WHITE METAL LANTERN         6
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6

           InvoiceDate  UnitPrice  CustomerID         Country
0  2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2  2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4  2010-12-01 08:26:00       3.39     17850.0  United Kingdom

Original shape: (541909, 8) -> Cleaned shape: (392692, 8)

--- Sample Cleaned Output ---
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1    536365     71053                  WHITE METAL LANTERN         6
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6

           InvoiceDate  UnitPrice  CustomerID         Country
0  2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2  2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
```

Diagram 3.2 Code Snippet of Raw and Cleaned Data Output

```
print("--- Extracting RFM Features ---")
df_clean['InvoiceDate'] = pd.to_datetime(df_clean['InvoiceDate'])
max_date = df_clean['InvoiceDate'].max()
df_clean['TotalSum'] = df_clean['Quantity'] * df_clean['UnitPrice']

rfm = df_clean.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (max_date - x.max()).days,
    'InvoiceNo': 'count',
    'TotalSum': 'sum'
}).rename(columns={'InvoiceDate': 'Recency', 'InvoiceNo': 'Frequency', 'TotalSum': 'Monetary'})

print("--- Engineered RFM Sample ---")
print(rfm.head(), "\n")

plt.figure(figsize=(6, 4))
sns.heatmap(rfm.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Feature Selection: RFM Correlation Matrix')
plt.show()
```

Diagram 3.3 Code Snippet of Feature Selection

```
--- Extracting RFM Features ---
--- Engineered RFM Sample ---
            Recency  Frequency  Monetary
CustomerID
12346.0         325          1  77183.60
12347.0           1        182   4310.00
12348.0          74         31   1797.24
12349.0          18         73   1757.55
12350.0         309         17    334.40
```

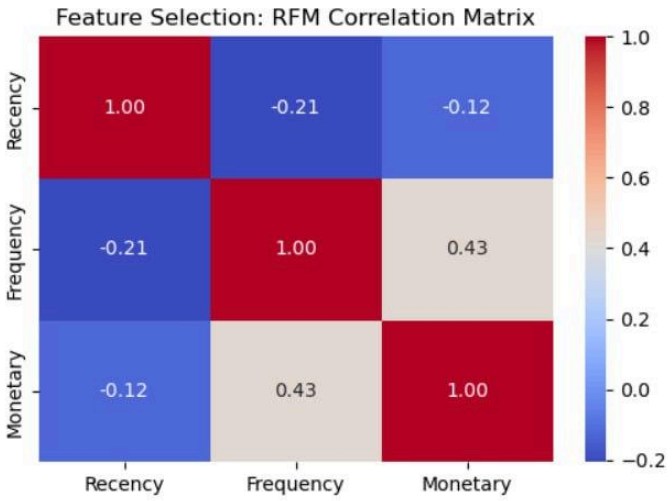


Diagram 3.4 Engineered RFM Sample

```
print("--- Transforming and Scaling Data ---")

rfm_log = np.log1p(rfm)

scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_log)
rfm_scaled_df = pd.DataFrame(rfm_scaled, columns=rfm.columns, index=rfm.index)

print("--- Sample Transformed & Scaled Output ---")
print(rfm_scaled_df.head(), "\n")
```

Diagram 3.5 Code Snippet of Transforming and Scaling

```
--- Transforming and Scaling Data ---
--- Sample Transformed & Scaled Output ---
             Recency  Frequency  Monetary
CustomerID
12346.0     1.409894  -2.438202  3.707716
12347.0    -2.146498   1.188986  1.414903
12348.0     0.383971  -0.211465  0.720024
12349.0    -0.574674   0.461819  0.702287
12350.0     1.374758  -0.673554 -0.614514
```

Diagram 3.6 : Transformed and Scaled Data

4.0 Model Development

4.1 Overview of Selected Algorithms

K-Means is a centroid-based clustering algorithm that groups customers based on the similarity of their numerical data. It works by assigning each customer to the nearest centroid and then continuously updating the centroid positions until the customers within each cluster are as similar as possible. The objective is to minimize the Within-Cluster Sum of Squares. which measures how close the data points are to their assigned cluster center.

On the other hand, Agglomerative Hierarchical Clustering is a bottom-up clustering algorithm. It starts by treating every customer as its own individual cluster. Then, the algorithm repeatedly merges the two closest clusters based on their distance until the required number of clusters is formed. The clustering process can also be visualized using a dendrogram, which shows how the clusters are combined step by step.

4.2 Justification of Algorithms Selection

K-Means was selected because it is simple, fast, and performs well on large datasets. Since the RFM features used in this project are numerical and have been log-transformed and standardized during preprocessing, K-Means is suitable for grouping customers with similar purchasing behaviour. It is also widely used in customer segmentation, making it a good choice for this study.

Agglomerative Hierarchical Clustering was chosen to compare its performance with K-Means. Unlike K-Means, it does not rely on randomly initialized centroids. Instead, it groups customers by gradually merging the closest clusters based on their similarity. Using a second clustering algorithm allows us to validate whether similar customer segments are produced and provides another perspective on the structure of the data.

4.3 Selected Algorithm Development

The K-Means model was created using the KMeans() class with n_clusters=3. The k-means++ initialization method was used to select better starting centroids, while random_state=42 was set to ensure the results are reproducible. The Agglomerative Hierarchical model was built using the AgglomerativeClustering() class with n_clusters=3. The model uses the Euclidean distance metric together with Ward linkage, which merges clusters while minimizing the increase in within-cluster variance. Both clustering algorithms were trained using the preprocessed RFM dataset after log transformation and standardization. Once the models generated the cluster labels 0, 1, and 2, the labels were added back to the original dataset so that each customer could be analysed according to their assigned cluster.

```
# K-Means Clustering
kmeans_model = KMeans(n_clusters=3, random_state=42)
km_labels = kmeans_model.fit_predict(rfm_scaled)
km_sil = silhouette_score(rfm_scaled, km_labels)
km_db = davies_bouldin_score(rfm_scaled, km_labels)
print(f"K-Means (K=3) Silhouette Score: {km_sil:.4f}")
print(f"K-Means (K=3) Davies-Bouldin Index: {km_db:.4f}\n")

# Agglomerative Hierarchical Clustering
agg_model = AgglomerativeClustering(n_clusters=3)
agg_labels = agg_model.fit_predict(rfm_scaled)
agg_sil = silhouette_score(rfm_scaled, agg_labels)
agg_db = davies_bouldin_score(rfm_scaled, agg_labels)
print(f"Agglomerative (K=3) Silhouette Score: {agg_sil:.4f}")
print(f"Agglomerative (K=3) Davies-Bouldin Index: {agg_db:.4f}\n")
```

Diagram 4.1 : Code Snippet of Clustering Algorithms Development

4.4 Determination of the Number of Clusters

The optimal number of clusters was determined programmatically by testing a range of values from 2 to 5 and evaluating the trade-off between Inertia and Silhouette scores.

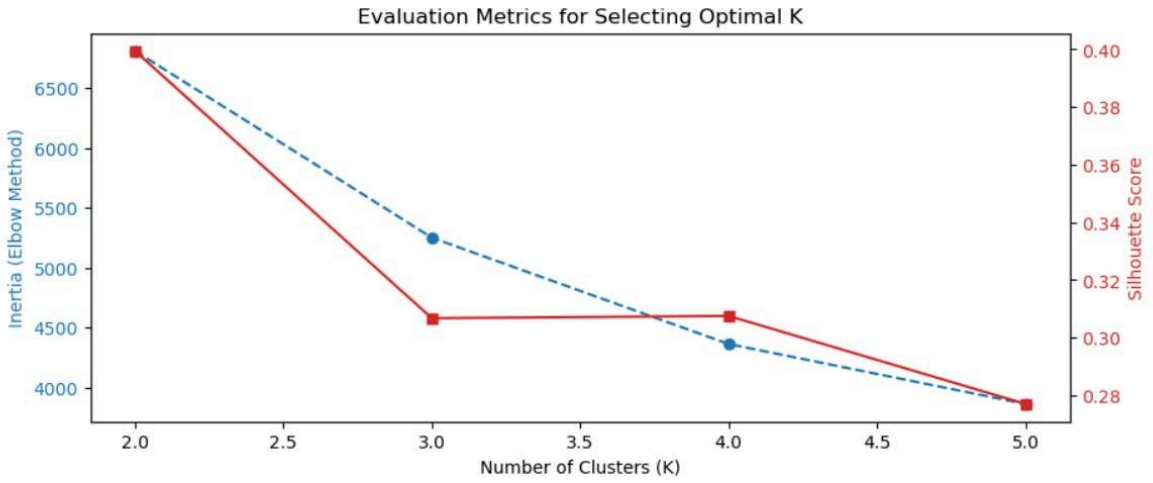


Diagram 4.1 : Graph of Evaluation Metrics for Selecting Optimal K

The Elbow Method shows that the Inertia value decreases sharply when moving from K = 2 to K = 3. After K = 3, the decrease becomes much smaller, forming an elbow in the graph. This indicates that increasing the number of clusters beyond three only provides a small improvement while making the model more complex.

The Silhouette Score is highest when K = 2 which is approximately 0.40, which suggests that the clusters are more clearly separated. However, having only two customer groups is too broad for meaningful business analysis. The Silhouette Score remains relatively stable between K = 3 and K = 4. Therefore, K = 3 was selected because it provides a good balance between clustering quality and practical business interpretation. The three clusters also align well with the objective of segmenting customers into meaningful groups for targeted marketing strategies.

5.0 Model Performance and Evaluation

5.1 Clustering Evaluation Metrics

The Silhouette Coefficient measures how well the data points fit within their assigned clusters. It considers both the similarity of data points within the same cluster and the separation between different clusters. The score ranges from -1 to 1, where a value closer to 1 indicates that the clusters are well separated and the data points are grouped appropriately. On the other hand, Davies-Bouldin Index measures the quality of clustering by comparing the compactness of each cluster with the separation between clusters. A lower DB Index indicates better clustering performance, as it means the clusters are more compact and have less overlap with one another.

5.2 Analysis of Model Performance & Comparison

```
K-Means (K=3) Silhouette Score: 0.3068
K-Means (K=3) Davies-Bouldin Index: 1.0946

Agglomerative (K=3) Silhouette Score: 0.2616
Agglomerative (K=3) Davies-Bouldin Index: 1.1297
```

Diagram 5.1 : Result of K-Means and Agglomerative Hierarchical Model Evaluation Results

Based on the evaluation results, the K-Means model performed better than the Agglomerative Hierarchical model for customer segmentation.

The K-Means model achieved a Silhouette Coefficient of 0.3068, which is higher than the 0.2616 obtained by the Agglomerative model. This indicates that the customer groups formed by K-Means are more distinct and that customers within the same cluster have more similar purchasing behaviour.

In addition, the Davies-Bouldin Index for K-Means is 1.0946, which is lower than the 1.1297 achieved by the Agglomerative model. Since a lower DB Index represents better clustering quality, this shows that the clusters produced by K-Means are more compact and have less overlap.

The better performance of K-Means is likely because it is well suited for numerical data such as the standardized RFM features used in this project. By repeatedly updating the cluster centroids, K-Means is able to group customers with similar purchasing patterns more effectively. On the other hand, Agglomerative Hierarchical Clustering builds clusters by merging the closest groups step by step, which may not produce clusters that are as well separated for this dataset.

6.0 Visualization and Discussion of Results

6.1 Model Comparison Scatter Plot Analysis

To compare the clustering results, a side-by-side scatter plot was created using Frequency and Monetary Value on a log-log scale. These two features provide a clear view of customers' purchasing behaviour. Applying a log transformation helps reduce the effect of extreme values, making the customer distribution easier to visualize and compare.

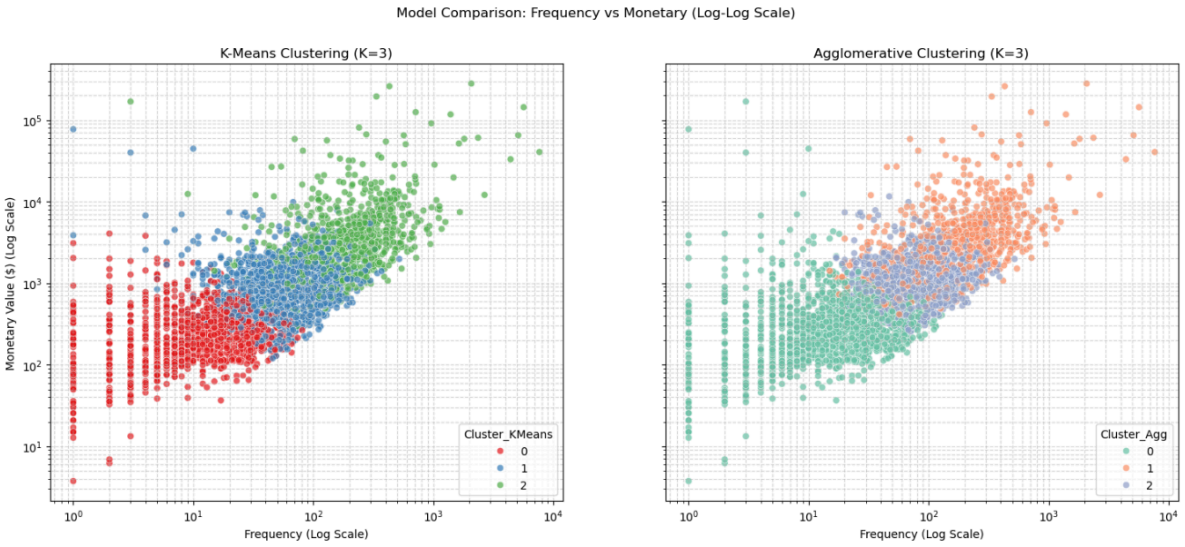


Diagram 6.1 : Scatter Plot Comparison of Frequency vs. Monetary Value for K-Means and Agglomerative Clustering

K-Means Clustering

The K-Means scatter plot shows three clearly separated customer groups. The clusters have more distinct boundaries, indicating that customers with similar purchasing behaviour are grouped together effectively.

Agglomerative Hierarchical Clustering

The Agglomerative Hierarchical scatter plot also forms three customer groups. However, the boundaries between the middle and lower-value customer groups are less clear, with some overlap between the clusters. This suggests that the model is slightly less effective at separating customers with similar purchasing behaviour.

6.2 Cluster Profiling

Cluster 0: At-Risk Customers

Recency = 167.67 days

Frequency = 15.23 orders

Monetary = $296.56

Customers in this cluster purchase infrequently, spend the least, and have not made a purchase for a long time. Their average recency of around 168 days indicates that many of them are inactive or at risk of leaving the platform. From a business perspective, these customers contribute less revenue and should not be the main focus of expensive marketing campaigns. Instead, businesses can target them with promotional offers or re-engagement campaigns to encourage them to return.

Cluster 1: Standard Customers

Recency = 66.85 days

Frequency = 66.11 orders

Monetary = $1,197.35

This cluster represents regular customers who make purchases consistently. They return to the platform approximately every two months, place a moderate number of orders, and contribute a steady amount of revenue. These customers form the main customer base of the business. Maintaining their satisfaction through personalised recommendations, promotions, or loyalty programmes can help increase repeat purchases and overall sales.

Cluster 2: Premium Customers

Recency = 12.18 days

Frequency = 265.13 orders

Monetary = $6,669.54

Customers in this cluster are the most valuable. They purchase very frequently, spend the most, and return to the platform regularly, with an average recency of only about 12 days. Although this group is smaller than the other clusters, they contribute a significant portion of the business revenue. Therefore, they should be prioritised through exclusive rewards, personalised services, and customer retention strategies.

6.3 Algorithm Comparison

Both K-Means and Agglomerative Hierarchical Clustering successfully identified the high-value customer group. For example, the Premium customer segment identified by K-Means has an average spending of $6,669.54, while the corresponding Agglomerative cluster has an average spending of $6,480.38. This shows that both algorithms are able to recognise customers with high purchasing value.

However, there are noticeable differences in the middle and lower-value customer groups. Agglomerative Hierarchical Clustering places more customers into its middle cluster, resulting in a higher average frequency of 87.77 orders and monetary value of $1,396.84 compared to K-Means which is 66.11 orders and $1,197.35.

Overall, the evaluation metrics and scatter plot show that K-Means produces clearer and more balanced customer segments, making it more suitable for this retail dataset.

6.4 Business Insights

Insight 1: Premium customers generate the highest revenue

The Premium customer segment is the most valuable group. These customers purchase frequently, return to the platform every few weeks, and spend significantly more than other customers. Businesses should focus on retaining these customers through loyalty programmes, exclusive promotions, and personalised services to maintain long-term revenue.

Insight 2: Core customers support business growth

The Standard customer segment represents loyal and active customers who contribute consistent revenue. Encouraging these customers to purchase more often through product recommendations, cross-selling, or special promotions can further increase overall sales.

Insight 3: At-risk customers need re-engagement

Customers in the Budget or At-Risk segment have not purchased for several months and contribute the lowest spending. Instead of investing heavily in expensive marketing

campaigns, businesses should use cost-effective strategies such as discount vouchers, reminder emails, or win-back campaigns to encourage them to make another purchase.

7.0 Conclusion

This project successfully applied two unsupervised machine learning algorithms, K-Means and Agglomerative Hierarchical Clustering, to analyse customer purchasing behaviour using the Online Retail dataset. By using the Recency, Frequency, and Monetary (RFM) model, the customer transaction data was grouped into three meaningful customer segments. Moreover, from the evaluation results, K-Means performed better than Agglomerative Hierarchical Clustering which produced clearer customer groups with a higher Silhouette Coefficient of 0.3068 and a lower Davies-Bouldin Index of 1.0946. This indicates that **K-Means is more suitable for clustering customers** in this dataset.

The three customer segments identified are Premium, Standard, and At-Risk customers. The Premium customers are the most valuable group as they purchase frequently, spend the most, and contribute a large portion of the business revenue. The Standard customers represent the main customer base by making regular purchases and providing consistent revenue. Meanwhile, the At-Risk customers purchase less frequently, spend the least, and have not returned for a long time, making them suitable targets for re-engagement campaigns.

Although the clustering models have produced meaningful customer segments, this study has several limitations. The dataset only contains transaction information and does not include demographic details such as age, gender, occupation, or income, which could provide a better understanding of customer behaviour. In addition, the analysis was performed using historical transaction data, meaning it does not capture changes in customer behaviour over time due to seasonal trends.

In future work, the study can be improved by including product information to better understand customers' purchasing preferences and buying patterns. Other clustering algorithms, such as DBSCAN or Gaussian Mixture Models (GMM) can also be explored to compare their performance with K-Means. Furthermore, the customer segments generated from this study could be used as labels for supervised machine learning models to predict customer behaviour who are likely to become inactive in the future.

8.0 References

Data Mining Project IPYNB files [Google Drive folder]. Google Drive. Retrieved June 8, 2026, from https://drive.google.com/drive/folders/1OvFg2jF6H7WmYD0oawSk3PHKnESley0Z?usp=sharing

Pedersen, U. T. (n.d.). *Online retail dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/ulrikthygepedersen/online-retail-dataset

IBM. (2024). *Hierarchical clustering* https://www.ibm.com/topics/hierarchical-clustering

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents, and usage data* (2nd ed.). Springer.