Journal of Student Research Exploration

https://shmpublisher.com/index.php/josre

p-ISSN 2964-1691 | e-ISSN 2964-8246

Clustering Analysis of Customers Based on Purchasing Patterns with K-Means Clustering

Wayne Joel Marcelino Lubis [1]

[1] Information Systems, Department of Computer Science, Universitas Negeri Semarang

Article Info

Article history:

Received December 17, 2024
Revised April 2, 2026
Accepted April 14, 2026

Keywords:

Data mining
K-Means
Elbow method
clustering

ABSTRACT

There are various techniques to classify data, one of which is clustering. What distinguishes clustering techniques from classification techniques is that they do not rely on the labels in the dataset. The main purpose of clustering is to divide data into several clusters based on similar characteristics, while Classification Technique is a technique of grouping data based on the similarity of the labels of the data under study. In this study, the dataset was created using secondary data from Kaggle. The analysis process begins with data pre-processing to normalize the variables used, followed by the application of the K-Means Clustering method to group customers into several clusters based on the similarity of their purchasing patterns. This research demonstrates the potential of using clustering analysis to improve understanding of customer behaviour and develop more effective business strategies.

1. Introduction

In the era of digital transformation, the global economy is shifting towards a data-driven model, often referred to as the data-driven economy. In this model, data becomes one of the most valuable assets for organizations and businesses. Data is used to support decision-making, improve operational efficiency, and design more effective marketing strategies. One important element in the data-driven economy is the ability to understand customer behaviour through the analysis of data generated from their transactions.

In the world of data analysis, data mining plays a very important role. Data mining is an area of intersection between computer science and statistics that is used to find patterns in information. The main goal of the data mining process is to extract useful information from

[1] ***Corresponding Author:***
Wayne Joel Marcelino,
Department of Computer Science,
State University of Semarang,
Email: devastator0209@students.unnes.ac.id
DOI: https://doi.org/10.52465/josre.v4i1.512

data files and shape it into an understandable structure for future use [1]. Some of the main methods in data mining include predictive analysis, classification, regression, association, and clustering. One of the most frequently used methods for customer segmentation purposes is clustering, Clustering is one of the various techniques in data grouping. Clustering is a process for grouping data into several clusters or groups so that data within a cluster has the maximum level of similarity and data between clusters has minimum similarity [2]. With the Clustering Technique, researchers are facilitated in the process of grouping datas that has similarities. Clustering can also be interpreted as a data segmentation method that is implemented in several fields, including marketing, financial analysis [3].

One of the most popular clustering algorithms is K-Means Clustering. This algorithm works by dividing data into a number of clusters [4], based on the similarity of predefined features or attributes. K-Means was chosen as the method in this study due to its simplicity, processing speed, and ability to identify patterns in large datasets. With this algorithm, customers can be grouped based on their purchasing behaviour, so that companies can identify customer segments more clearly. This research uses a dataset that includes attributes such as Customer_ID, Price_per_Unit, and Units Sold to analyse customer purchasing patterns.

2. Method

This research is divided into several stages which can be seen in Figure 1.

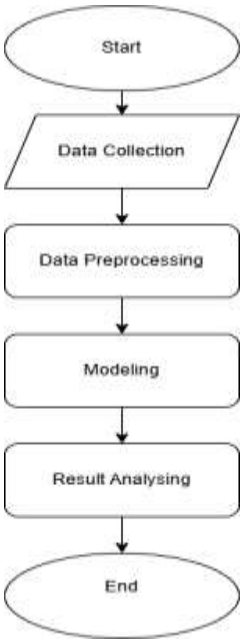


Figure 1 Research flowchart

To make this research more systematic, the sections contain; data collection by searching data from the Kaggle website. The second is data preprocessing which aims to normalize data on Google Collab using MinMaxScaler included in preprocessing because it aims to prepare data so that the K-Means algorithm can work optimally. The third is the data analysis stage in Google Collab using the K-Means algorithm. And the last is the analysis of each cluster.

Data Collection

The data taken is secondary data. Secondary data is data taken from data archives, usually taken from the internet [5]. On this article I'm taking data from the Kaggle website named "Houseware_Retail_Sales_Data_Updated.csv" which is household appliance sales data.

Data Preprocessing

Data preprocessing is an important stage in the data mining process. There are two objectives in using data preprocessing, which are to solve problems in the data, and to learn more about the nature of the data [6]. Data preprocessing involves data preparation, data cleansing, normalization, and data transformation. The result is expected to be correct data and can be used for data mining algorithms which in this research I use the K-Means algorithm [7].

Result Analysis

At this stage, the data from the existing clusters will be analysed. This stage will also identify the characteristics of each cluster that can later be used to create a marketing strategy for each market segmentation.

3. Results and Discussion

In this section, it is explained the results of research and at the same time is given the comprehensive discussion.

3.1. Data Collection

The data obtained from Kaggle is secondary data. The data retrieved contains the attributes Customer_ID, Product_ID, Transaction_Date, Price_per_Unit, Units_Sold with a total of 51425 data from home appliance sales.

3.2. Data Preprocessing

This research will use google Collab as a data processing tool to help researchers to clustering the data. The first step we do is to see if there are missing values contained in the data we want to process. We can see the process using google Collab as shown in Figure 2.

```
import pandas as pd

df = pd.read_csv('Houseware_Retail_Sales_Data_Updated.csv')

missing_values_count = df.isnull().sum()

# Print the result.
missing_values_count
```

	0
Customer_ID	0
Product_ID	0
Transaction_Date	0
Price_per_Unit	0
Units_Sold	0

dtype: int64

Figure 2. Identify missing value

In the data there are no missing values, so the next step is selecting data that is relevant to the research which can be seen in Figure 3.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

df = pd.read_csv('Houseware_Retail_Sales_Data_Updated.csv')
df.head()
```

	Customer_ID	Product_ID	Transaction_Date	Price_per_Unit	Units_Sold
0	384	1286	2020-01-01	432.83	6
1	103	422	2020-01-01	203.12	10
2	664	832	2020-01-01	703.74	9
3	456	1692	2020-01-01	60.02	9
4	527	41	2020-01-01	658.94	9

Figure 3. Import library and showing data

After inputting all the libraries, the next step we're going to do is choosing the relevant data for clustering as you can see in Figure 4.

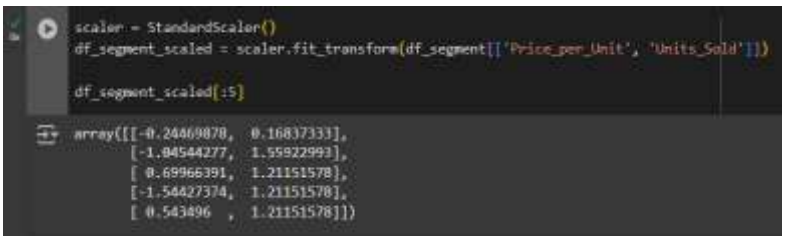
```
df_segment = df[['Customer_ID', 'Price_per_Unit', 'Units_Sold']]

df_segment = df_segment.dropna()
df_segment.head()
```

	Customer_ID	Price_per_Unit	Units_Sold
0	384	432.83	6
1	103	203.12	10
2	664	703.74	9
3	456	60.02	9
4	527	658.94	9

Figure 4. Selecting relevant data for clustering

Next is to normalize using StandardScaler so that each data is on the same scale which can be seen in Figure 5.



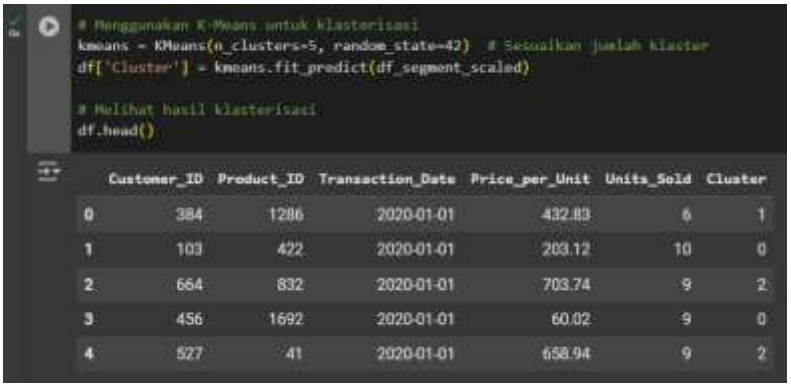
```
scaler = StandardScaler()
df_segment_scaled = scaler.fit_transform(df_segment[['Price_per_Unit', 'Units_Sold']])

df_segment_scaled[:5]
```

```
array([[-0.24469878,  0.16837333],
       [-1.04544277,  1.55922993],
       [ 0.69966391,  1.21151578],
       [-1.54427374,  1.21151578],
       [ 0.543496  ,  1.21151578]])
```

Figure 5. Data normalize

The data normalization process starts here. StandardScaler is used to normalize the data to be on the same scale. StandardScaler expects your information to be normally adjusted within each component and will scale it such that its current distribution hovers around 0, with a standard deviation of 1 [8]. fit_transform() is used to learn and apply the transformation to the same dataset in a one-time manner [9] so that the data has a mean of 0 and a standard deviation of 1. After normalizing the data, we can continue clustering the data using K-Means Algorithm, showed on Figure 6.



```
# Menggunakan K-Means untuk klasterisasi
kmeans = KMeans(n_clusters=5, random_state=42)  # Sesuaikan jumlah klaster
df['Cluster'] = kmeans.fit_predict(df_segment_scaled)

# Melihat hasil klasterisasi
df.head()
```

	Customer_ID	Product_ID	Transaction_Date	Price_per_Unit	Units_Sold	Cluster
0	384	1286	2020-01-01	432.83	6	1
1	103	422	2020-01-01	203.12	10	0
2	664	832	2020-01-01	703.74	9	2
3	456	1692	2020-01-01	60.02	9	0
4	527	41	2020-01-01	658.94	9	2

Figure 6. Clustering the data using K-Means

And the next step is to identify the silhouette score. silhouette score method, which is a measure of the quality of a cluster, was used to find the mean silhouette co-efficient of all the samples for different number of clusters.

```
from sklearn.metrics import silhouette_score
for k in range(2, 10):
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(df_segment_scaled)
    sil_score = silhouette_score(df_segment_scaled, labels)
    print(f'For n_clusters = {k}, Silhouette Score = {sil_score}')
```

```
For n_clusters = 2, Silhouette Score = 0.3981610913012895
For n_clusters = 3, Silhouette Score = 0.3965448963332856
For n_clusters = 4, Silhouette Score = 0.41140830210032614
For n_clusters = 5, Silhouette Score = 0.38861946778915457
For n_clusters = 6, Silhouette Score = 0.3705006106032533
For n_clusters = 7, Silhouette Score = 0.39930948027717367
For n_clusters = 8, Silhouette Score = 0.3864243536357085
For n_clusters = 9, Silhouette Score = 0.36484074856537474
```

Figure 7. Silhouette score

The highest silhouette score indicates the optimal number of clusters [10]. Silhouette score values lie between -1 to +1. The value of +1 indicates correct clustering of objects while the value of -1 show that objects are not properly clustered [11]. We can see on Figure 7, that 4 number of clusters is having the highest silhouette score than the other, so we can conclude that 4 clusters is the most ideal cluster. To help validate the most optimal number of clusters, we will use the elbow method. The elbow method is an approach to provides data for determining the most optimal amount of clusters through observing the percentage of the comparison between the number of clusters that would create an elbow at one location [12]. As you can see on Figure 8.

```
# Elbow Method untuk menentukan jumlah kluster optimal
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(df_segment_scaled)
    inertia.append(kmeans.inertia_)

# Plot Elbow Method
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()
```

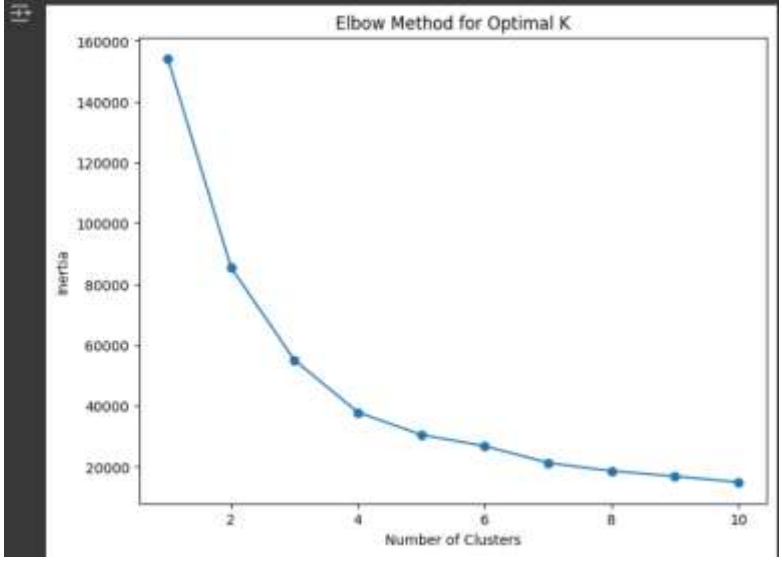
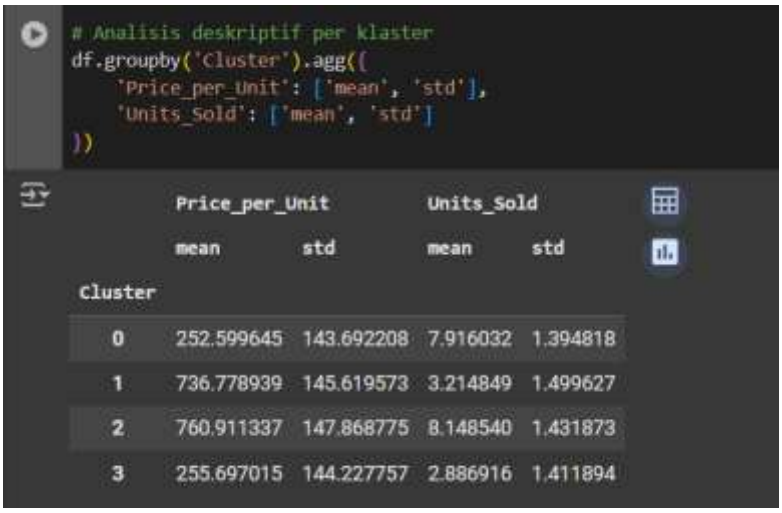


Figure 8. Elbow method

As you can see in Figure 8, the number of clusters closest to the 90-degree angle is in the number of clusters 4, therefore it can be stated that the number of clusters 4 is the most optimal. The last step is to analyse each cluster as shown in Figure 9.



```
# Analisis deskriptif per klaster
df.groupby('Cluster').agg({
    'Price_per_Unit': ['mean', 'std'],
    'Units_Sold': ['mean', 'std']
})
```

	Price_per_Unit		Units_Sold	
	mean	std	mean	std
Cluster				
0	252.599645	143.692208	7.916032	1.394818
1	736.778939	145.619573	3.214849	1.499627
2	760.911337	147.868775	8.148540	1.431873
3	255.697015	144.227757	2.886916	1.411894

Figure 9. Cluster analysis

Group by ('Cluster') groups the data by cluster. The agg function calculates descriptive statistics for each cluster, such as the average product price and the number of units sold.

Table 1. Descriptive statistics of cluster analysis

Cluster	Price_per_Unit (Mean)	Price_per_Unit(Std)	Units_Sold(Mean)	Units_Sold (Std)
0	252.60	143.69	7.92	1.39
1	736.78	145.62	3.21	1.50
2	760.91	147.87	8.15	1.43
3	255.70	144.23	2.89	1.41

3.3. Result Analysing

The descriptive statistics for each cluster are summarized in Table 1.

1. Cluster 0: Customers who purchase products at lower prices and buy more units per transaction. (More affordable prices and more purchases)
2. Cluster 1: Customers who buy higher priced products but purchase fewer units. (Premium products and fewer purchases)
3. Cluster 2: Customers who purchase products at a high price and purchase more units per transaction. (Premium products with bulk purchases)
4. Cluster 3: Customers who purchase products at lower prices but purchase fewer units per transaction. (Affordable products with fewer purchases)

4. Conclusion

In this analysis, we used K-Means clustering to segment customers based on their purchasing behaviour. The dataset included data on Price_per_Unit and Units_Sold, and after preprocessing (handling missing values and normalizing the data), we applied K-Means to create clusters. We determined the optimal number of clusters using the Elbow Method and

chose 4 clusters, which gave a moderate Silhouette Score of 0.42. This score suggests that the clusters are somewhat well-separated but could be improved. The four clusters identified were:

1. Cluster 0: Customers who buy cheaper products in larger quantities.
2. Cluster 1: Customers who buy expensive products in smaller quantities.
3. Cluster 2: Customers who buy expensive products in larger quantities.
4. Cluster 3: Customers who buy cheaper products in smaller quantities.

These insights can help businesses create targeted strategies, such as promotions and product recommendations, based on customer buying behaviour. Although the clustering results are good, further adjustments might help improve the separation between the clusters.

REFERENCES

[1] S. Agarwal, "Data mining: Data mining concepts and techniques," *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013,* pp. 203-207, 2014, doi: 10.1109/ICMIRA.2013.45.

[2] P. Tan, M. Steinbach, V. Kumar, T. Pang-Ning, M. Steinbach, and V. Kumar, "Introduction to data mining: Instructur's," in *Library of Congress*, 2006, vol. 769.

[3] M. Cui, "Introduction to the k-means clustering algorithm based on the elbow method," *Accounting, Auditing and Finance,* vol. 1, no. 1, pp. 5-8, 2020.

[4] R. T. Ng and H. Jiawei, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, no. 5, pp. 1003-1016, 2002, doi: 10.1109/TKDE.2002.1033770.

[5] J. J. Hox and H. R. Boeije, "Data collection, primary vs. secondary."

[6] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent data analysis,* vol. 1, no. 1, pp. 3-23, 1997.

[7] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer.

[8] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020,* no. Icssit, pp. 729-735, 2020, doi: 10.1109/ICSSIT48917.2020.9214160.

[9] E. Bisong, "Introduction to Scikit-learn," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, E. Bisong Ed. Berkeley, CA: Apress, 2019, pp. 215-229.

[10] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," *Proceedings - 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA 2020,* pp. 747-748, 2020, doi: 10.1109/DSAA49011.2020.00096.

[11] G. Ogbuabor and U. F. N, "Clustering Algorithm for a Healthcare Dataset Using Silhouette Score Value," *International Journal of Computer Science and Information Technology,* vol. 10, no. 2, pp. 27-37, 2018, doi: 10.5121/ijcsit.2018.10203.

[12] R. Nainggolan, R. Perangin-Angin, E. Simarmata, and A. F. Tarigan, "Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method," *Journal of Physics: Conference Series,* vol. 1361, no. 1, 2019, doi: 10.1088/1742-6596/1361/1/012015.