Journal Of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Exploring Customer Trends with K Means Clustering Analysis based on Machine Learning

Neha Gupta
Associate Professor
School of Computer Science & Information Technology, Symbiosis University of Applied Sciences, Indore
Email: neha.gupta@suas.ac.in
Devendra Chouhan
Assistant Professor
School of Computer Science & Information Technology, Symbiosis University of Applied Sciences, Indore
Email: devendra.chouhan@suas.ac.in
Ankit Upadhyay
Assistant Professor
Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore.
Email: Ankitu009@gmail.com

Abstract-The zeitgeist of the modern era is innovation, where everyone is embroiled into a competition to be better than others. Today's business run on the basis of such innovation having the ability to enthral the customers with the products, but with such a large raft of products leave the customers confounded, what to buy and what to not and also the companies are nonplussed about what section of customers to target to sell their products. This is where machine learning comes into play, various algorithms are applied for unravelling the hidden patterns in the data for better decision making for the future. This elude concept of which segment to target is made unequivocal by applying segmentation. The process of segmenting the customers with similar behaviours into the same segment and with different patterns into different segments is called customer segmentation. In this paper clustering algorithms k-Means have been implemented to segment the customers and finally compare the results of clusters obtained from the algorithms. A Python program has been developed and the program is been trained by applying standard scales onto a dataset having various features of training sample data collected by various means. The features are the mean of the amount of shopping by customers and the average of the customer's visit to the shop annually.

1. INTRODUCTION

Over the years, business competition has intensified, and the huge amount of historical data available has led in the widespread usage of data mining techniques in extracting relevant and strategic information from an organization's database. Data mining is the process of extracting data patterns and presenting them in a human-readable way for decision assistance.

Data tuples are treated as objects in clustering algorithms [1, 2]. The data objects are divided up into groups or clusters so that they may be compared to one another and distinguished from those in other clusters. Client segmentation is the process of dividing a customer base into various groups known as customer segments, each of which contains consumers with comparable characteristics.

The segmentation is based on similarities in several marketing-relevant factors, including gender, age, hobbies, and various purchasing behaviours.

Customer segmentation is important because it contains the ability to change market programmes so that they are appropriate for each customer segment, support in business decisions, identification of items connected with each customer segment and managing the demand and supply of that product, identifying and targeting the potential customer base, forecasting customer defection, and providing directions in finding solutions.

There is a lot of competition among businesses to attract new customers and hold on to existing ones as a result of the creation of numerous competitors and entrepreneurs. Because of the aforementioned, outstanding customer service is required, regardless of the size of the company. Any company will also benefit more from targeted customer services and the creation of unique customer care strategies if it can comprehend the needs of each of its clients. Structured customer service makes it feasible [3 – 5].

Each brand has a distinct group (or groups) of customers and a wealth of data that informs various methods of customer segmentation. Both large and small firms can benefit from machine learning's ability to sift through data and derive insightful conclusions.

By using machine learning, consumer segmentation separates a client base into distinct groups that have common traits. Numerous methods exist for segmenting customers. Sorting can be done either demographically or psychologically.

- Age
- Sex
- Highest level of education achieved
- Stage of life
- Income
- Religion
- Attitudes towards products/services

A customer segmentation model enables businesses to target particular client groups, allowing for the efficient allocation of marketing resources and the optimization of cross- and up-selling opportunities. Customer segmentation enhances customer service and encourages repeat business.

Compared to impersonal branding that ignores all forms of customer relationships, this technique further helps to increase brand value and client loyalty. The customer generally values and appreciates these split

2. LITERATURE REVIEW

The practise of storing consumer information on paper and in computer software (digital data) is becoming more widespread. At the end of the day, personnel will analyse their statistics, including the number of items sold, the real number of customers, etc. They can identify customers that will benefit their firm and boost sales by analysing the data that has been gathered. Both time and paperwork are increased. Finding the required consumer data using this method is also not very efficient [6 – 8].

The commercial world has grown more competitive over time as a result of companies like these having to better their enterprises by attracting new clients and satisfying the requirements and desires of their existing clientele. Identifying and addressing the wants and expectations of each client in the organisation is a demanding undertaking. This is because clients differ in terms of their demands, desires, demographics, shapes, flavour and taste, characteristics, and so on. Currently, it is not a good business strategy to serve every consumer equally. This difficulty has resulted in the introduction of the notion of customer segmentation or market segmentation, in which customers are separated into subcategories or segments, with members of each subcategory exhibiting comparable market behaviours or traits. Customer segmentation is hence the process of separating the market into local populations.

Marketing strategies are largely centred on consumer-retailer interactions. One strategy to boost revenues is to discover client needs through consumer dialogue. Communicating with consumers on a personal level is almost impossible, yet without creating communication, marketing catastrophes are unavoidable. Retailers may connect with one another using customer data to solve this issue. Retailers can categorise their customers based on their habits and afterwards create business plans around it. Customer segmentation is a way to better connect with customers and to understand their interests so that the right connections can be made. At the moment,

consumer segmentation is done by processing customer datasets, such as demographic information or purchase history.

Research Scope

At its most basic, customer segmentation is the division of potential customers in a given market into discrete groups. That division is based on variables and descriptors of those customers having similar enough: 1. Needs, i.e., so that a single whole product can satisfy them. 2. Buying characteristics, i.e., responses to messaging, marketing channels, and sales channels, that a single go-to-market approach can be used to sell to them competitively and economically.

The premise of market segmentation is that to maximize sales to a large population of customers, it is best to divide it into logical subgroups. The assumption is that by dividing one large, amorphous mass into subgroups, you can fine-tune your product, messaging, support, or distribution channels to meet the specific needs of unique customer groups. Thus, the goal is to use a market segmentation model to improve marketing success and optimize marketing ROI [9, 10].

3. DATA ACQUISITION AND ANALYSIS

Understanding the Data

The dataset used in the project was collected from the UCI Machine Learning Repository. This is a set of geographic data containing all transactions occurring between 1/1/2/10 and 9/12/2011 in an unregistered and unregistered UK broker. The company mainly sells unique gifts all together. Many of the company's customers are shopkeepers. The database contains 8 attributes. These attributes include:

- Invoice No: This is a 6-digit number assigned separately for each transaction.
- Stock Code: This is a 5-digit number assigned only to each unique product.
- Description: This is the description of the product.
- Quantity: This indicates the number of products sold in the transaction.
- Invoice Date: The dates and time of each transaction.
- Unit Price: This depicts the price of a single product or price per unit of measurement
- Customer ID:. This is a 5-digit number assigned to each customer.
- Country: Name of the country where each customer lives.

```
[84] data.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom

Table 1. List of Attributes

Statistical Analysis and Variable Identification

The very first step after getting familiarized with the data dictionary was to get some basic data insights about the contents of the data frame. This included knowing the types of various variables, null values and descriptive statistical information such as mean, standard deviation, number of rows and columns (i.e., shape of the dataframe) and quartile values of the data to know its distribution.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
None

********************************

            Quantity      UnitPrice     CustomerID
count  541909.000000  541909.000000  406829.000000
mean        9.552250       4.611114   15287.690570
std       218.081158      96.759853    1713.600303
min    -80995.000000  -11062.060000   12346.000000
25%         1.000000       1.250000   13953.000000
50%         3.000000       2.080000   15152.000000
75%        10.000000       4.130000   16791.000000
max     80995.000000   38970.000000   18287.000000
```

Table 2. Statistical Values

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
None
```

Table 3. Data Types of Attributes

4. IMPLEMENTATION

Missing Value Treatment

The first step in an EDA process always begins with identifying the missing/null values in the dataset. These could either be a single value or an entire tuple of data. Typically, is dealt with on a case-by-case basis depending on the size and complexity of the data. This missing data would further lead to biased approximations, or even incorrect conclusions in extreme cases.

So, the first step would involve treating these missing values. We'd essentially try to find the percentage of missing values for each of the characteristics [11 – 14]. Percentages are a better estimate as they're easier to understand and will give a quick and concise idea of what we need to work with.

```
[85] missing_percentage = data.isnull().sum() / data.shape[0] * 100
     missing_percentage

InvoiceNo      0.000000
StockCode      0.000000
Description    0.268311
Quantity       0.000000
InvoiceDate    0.000000
UnitPrice      0.000000
CustomerID    24.926694
Country        0.000000
dtype: float64
```

Table 4. Percentage Estimation

Running the query gives a result displaying that we have 0.2% data missing from descriptions and a significant 24.8 ~ 25% of CustomerIDs are unknown. This makes for a messy data set. Upon further exploration, we could also find that descriptions with missing values also have their CustomerId missing and unit prices as 0 which leads to the question of how and why these entries were recorded. Moving on to the missing CustomerIDs, other characteristics connected to these key values, like quantities and price, can further express outliers [15]. And a significant number of outliers will imbalance the data as we know. We might need to create features that may depend on these quantities and prices.

Simply running a null query would not be enough though. A further need for inspection is necessary as the previous query has given us an idea of the strangeness of the data collected. A command was run to find hidden missing values, i.e., 'nan' strings and even "" empty cells. These hidden, additional null values were then transformed to the same as null query values - 'NaN'. As we lack an understanding into why these descriptions or IDs are missing, as well as quantity/price outliers and 0 prices we would decide to drop these values to be a little precautious [16].

Finally, there is a need for verification if there are any missing values left before we proceed any further.

Univariate Analysis

The individual attributes were analysed, and information was collected from the columns. The attribute Invoice No. has some transactions that start with "c". These transactions simply reflect the transactions that have been cancelled. Therefore, a new feature, Is Cancelled was created to indicate the cancelled transactions.

```
data["IsCancelled"]=np.where(data.InvoiceNo.apply(lambda l: l[0]=="C"), True, False)
data.IsCancelled.value_counts() / data.shape[0] * 100
```

```
False    97.81007
True      2.18993
Name: IsCancelled, dtype: float64
```

Table 5. Cancelled Transaction Summary

Approximately, 2.2% of the transactions in the datasets have been cancelled. Furthermore, it was found that all cancellations had negative quantities, but positive and non-zero unit prices. Moving further to the StockCode attribute, most common stock codes were identified as the percentage of the entire data. It was inferred that most stock codes highly common, indicating that the retailer sells many different products and that there is no strong specialization of a specific stock code. Moreover, the length of stock code was also analysed for their alpha-numeric characteristics [17, 18].

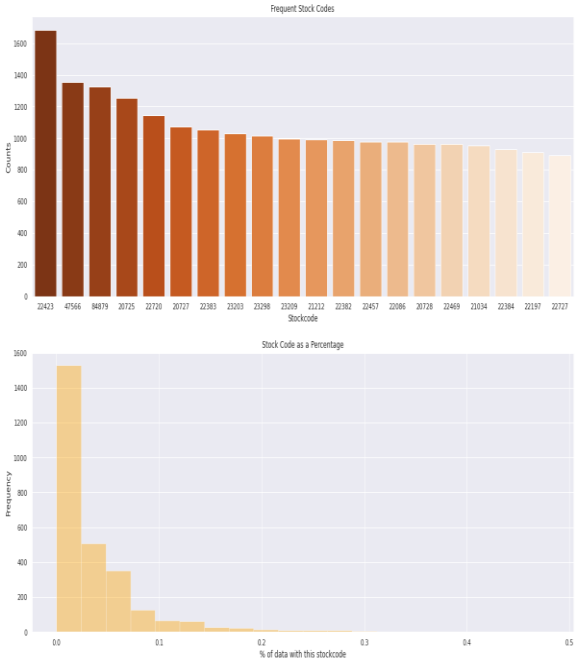

Figure 1. Stock Code Analysis

After analysing the CustomerID, it was found that four customers accounted for the majority of the transactions.

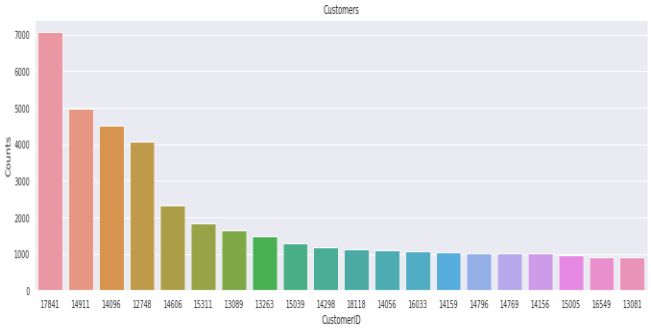

Figure 2. Customer ID Analysis

Furthermore, the retailer sold a large chunk of the products in the UK, which was followed by some European countries.

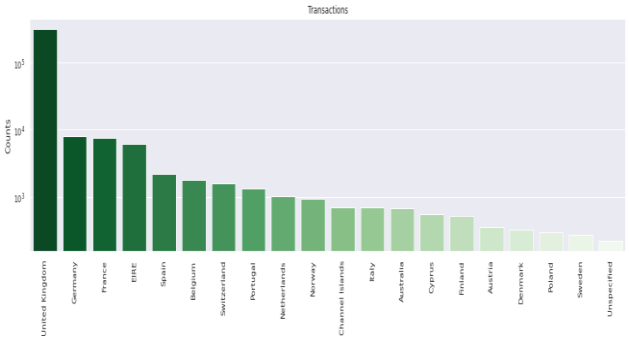

Figure 3. Transaction Analysis

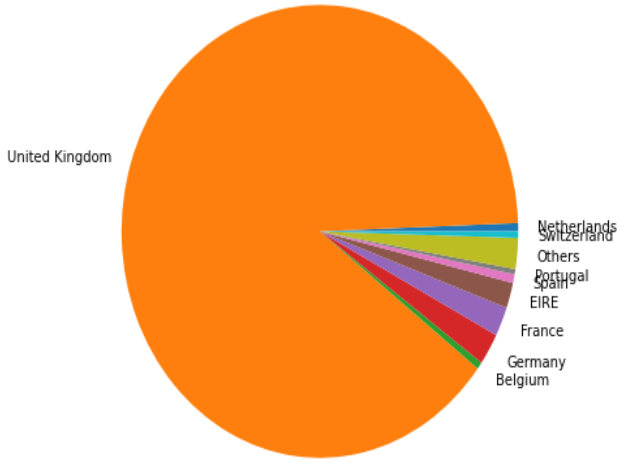

Figure 4. Country wise Analysis

```
[128] data.loc[data.Country=="United Kingdom"].shape[0] / data.shape[0] * 100
0s

89.10192031784572
```

Here, we could see that almost 89% of the transactions were inside the UK. Furthermore, it was also found that the retailer delivered the products to 37 different countries. The gross amount of the total quantity purchased was also segregated according to their respective countries and the top seven countries were found out.

Since the given website is based out of the United Kingdom, the entire dataframe consists the variables (like no. of Customers and the Gross total sales) that are dominated by the United Kingdom. Furthermore, the remaining data is occupied by the neighbouring European countries.

The Kernel Density Estimation graph was also plotted for Quantity and UnitPrice to estimate the probability density of these variables.

Applying K-Means

K-Means clustering is a machine learning algorithm that separates the data into a given number of clusters [19]. In this algorithm, K is the given number of predefined clusters. Each cluster has a centroid assigned to it because the algorithm is centroid-based. The primary goal is to close the gap between each data point and its corresponding cluster centroid. The algorithm separates the dataset into clusters using the raw, unlabelled data as its input before repeating the procedure until the optimal clusters are identified.
The key to this approach is determining the ideal number of clusters. Elbow Method is a widely used technique for determining the ideal K value. In the Elbow approach, the number of clusters (K) is truly variable and ranges from 1 to 10. We are calculating WCSS for each value of K. (Within-Cluster Sum of Square) [20 – 21]. The sum of the squared distances between each point and the cluster's centroid is known as WCSS. The plot of the WCSS with the K value resembles an elbow. The WCSS value will begin to drop as the number of clusters rises. The highest WCSS value is at K = 1. When we examine the graph, we can observe that it abruptly changes at one point, forming an elbow. The graph then begins to travel nearly parallel to the X-axis from this point on. The best K value, or the most clusters, is the one that corresponds to this location.

	num_clusters	cluster_errors
0	1	8678.000000
1	2	3167.406530
2	3	2006.354230
3	4	1574.253185
4	5	1182.757571
5	6	1001.370727
6	7	858.942915
7	8	750.744686
8	9	666.428164
9	10	582.988507

Table 6. Cluster Errors

```
[ ] plt.figure(figsize=(15,6))
    plt.plot(clusters_df["num_clusters"],clusters_df["cluster_errors"],marker = 'o')
    plt.xlabel('count of clusters')
    plt.ylabel('error')

Text(0, 0.5, 'error')
```

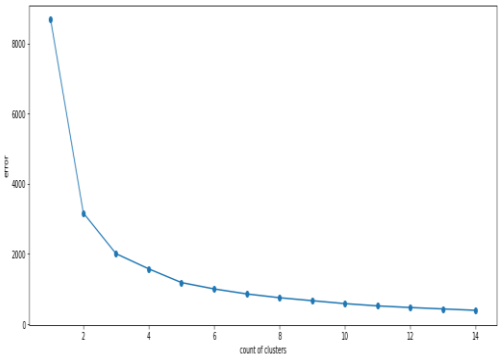

Figure 5. Count of Cluster

From the Elbow Plot, we observed that there is a sharp bend after the second cluster. Furthermore, after the number of clusters crosses 6, the plot starts to flatten up. Therefore, to choose the optimum number of clusters we would also analyse the Silhouette Score.

```
[ ] for num in range(2,16):
        clusters = KMeans(n_clusters=num,random_state=0)
        labels = clusters.fit_predict(df_pca)

        sil_avg = silhouette_score(df_pca, labels)
        print('For',num,'The Silhouette Score is =',sil_avg)
```

```
For 2 The Silhouette Score is = 0.5316173539289467
For 3 The Silhouette Score is = 0.46938033910712523
For 4 The Silhouette Score is = 0.448983308636758
For 5 The Silhouette Score is = 0.43620408327157767
For 6 The Silhouette Score is = 0.4283012994329389
For 7 The Silhouette Score is = 0.4355180378885031
For 8 The Silhouette Score is = 0.4393227054799325
For 9 The Silhouette Score is = 0.4448248070230853
For 10 The Silhouette Score is = 0.43730324152304423
For 11 The Silhouette Score is = 0.44392301049015087
For 12 The Silhouette Score is = 0.441380530537108
For 13 The Silhouette Score is = 0.43678789078974434
For 14 The Silhouette Score is = 0.4439714220360542
For 15 The Silhouette Score is = 0.44365694590196053
```

After calculating the Silhouette Score, we can see that the score is highest for 2 clusters. However, when the number of clusters increases from 2 to 4, there is also a large decrease in cluster error, and after 4, there is little reduction. In order to correctly categorise our customers, we will select n clusters = 4.

```
[ ] kmeans = KMeans(n_clusters = 4)
    kmeans = kmeans.fit(df_pca)
    labels = kmeans.predict(df_pca)
    centroids = kmeans.cluster_centers_

    print(labels)
    print()
    print('Cluster Centers')
    print(centroids)
```

```
[2 2 2 ... 1 2 3]

Cluster Centers
[[ 1.06020799 -0.57758139]
 [ 1.72062887  0.27234113]
 [-1.70745845  0.04812617]
 [-0.13227033  0.11820651]]
```

```
[ ] df_pca['Clusters'].value_counts()
```

```
3    1555
2    1184
1     802
0     798
Name: Clusters, dtype: int64
```

```
[ ]
    sns.pairplot(df_pca,diag_kind='hist',hue='Clusters')
```

```
<seaborn.axisgrid.PairGrid at 0x7f0e7169d250>
```

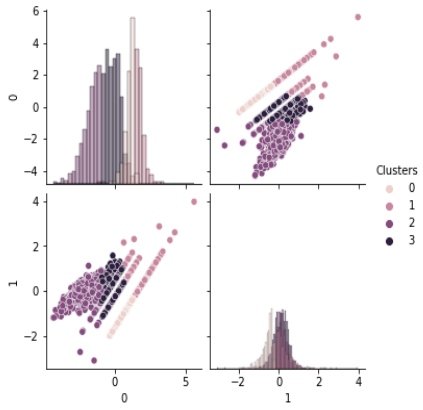

Figure 6. Pairgrid Clustering

```
[ ] X = df_pca[[0,1]]
    Y = df_pca['Clusters']

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42, stratify=Y)
    lr = LogisticRegression(max_iter=1000,random_state=0)
    lr.fit(X_train, y_train)
    y_pred = lr.predict(X_test)
    print('Test accuracy = ', accuracy_score(y_test, y_pred))
```

```
Test accuracy =  0.9946236559139785
```

```
knn = KNeighborsClassifier(n_neighbors = 4)

knn.fit(X_train, y_train)
pred = knn.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, pred))

print(classification_report(y_test, pred))
```

```
[[239   0   0   0]
 [  0 240   0   1]
 [  0   0 355   0]
 [  0   0   4 463]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       239
           1       1.00      1.00      1.00       241
           2       0.99      1.00      0.99       355
           3       1.00      0.99      0.99       467

    accuracy                           1.00      1302
   macro avg       1.00      1.00      1.00      1302
weighted avg       1.00      1.00      1.00      1302
```

```
[ ] from sklearn.tree import DecisionTreeRegressor
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42, stratify=Y)
    regressor = DecisionTreeRegressor(random_state = 213)
    regressor.fit(X_train,y_train)

    pred = regressor.predict(X_test)
    print(confusion_matrix(y_test, pred))

    print(classification_report(y_test, pred))
```

```
[[238   0   0   1]
 [  0 240   0   1]
 [  0   0 355   0]
 [  0   0   0 467]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       239
           1       1.00      1.00      1.00       241
           2       1.00      1.00      1.00       355
           3       1.00      1.00      1.00       467

    accuracy                           1.00      1302
   macro avg       1.00      1.00      1.00      1302
weighted avg       1.00      1.00      1.00      1302
```

5. CONCLUSION

Based on a database that contains information on purchases made on an e-commerce platform over the course of a year, the work is discussed in this notebook. Each entry in the dataset describes a product purchase made by a certain consumer on a specific date. The database contains about 4000 clients in total. Based on the information available, it was determined to create a classifier that can predict the type of purchase a client would make, as well as the number of visits he will make throughout the year, even before the user enters the e-commerce site.

Basic data visualisation was the focus of the analysis's next section. This was done to gain knowledge on the nation that used the e-commerce website the most. The analysis's findings were displayed using simple graphs. Other significant considerations include a country's Gross Purchase as well as which of the following descriptions was utilised the most.

Customers' clusters from the RFM dataset were used as independent variables in classification models such Logistic Regression, KNeighborsClassifier, and DecisionTree, with Cluster serving as the target variable. The categorization models' predicted clusters match K-Means clustering exactly. Therefore, we may say that our clusters are accurate.

The customer segmentation phase made up the last portion of the analysis. Using the RFM (Recency, Frequency, Money) table to group the customers is the primary workaround for this process. Following the creation of the RFM table, 4 clusters into which the consumers should be divided were created using K-Means clustering (Elbow curve and Silhouette scores). After each consumer was divided into their appropriate categories, models like Logistic Regression, K-Nearest Neighbours Classifier, and Decision Tree were used to measure the accuracy of the clustering, yielding a score of 0.98.

REFERENCE

1. V. Dutta, N. Gupta, “Sentimental analysis using Deep Learning techniques”, International journal of scientific and research publications, volume 12, issue 2, 2022, pp. 550-575.

6. N. Gupta, “Embedding Color Watermark in a Digital image by Adjusting DCT Coefficients through Back Propagation Neural Network using RGB Gray Scale watermarking and subsequent Union of RGB planes”, Test Engineeing and Management, 2020, pp.375- 380.

7. C. L. Liu, D. P. Mohapatra, “Discrete mathematics”, 2008, New Delhi, India: Tata McGraw Hill

8. A. B. Diogo Fernandes, “Security issues in cloud environments: a survey”, International. Journal of Information Security, Springer, Vol. 13, 2014, pp. 113–170.

9. D. Forte, "Security audits in mixed environments", Network Security,March, Vol.3, No. 3, 2009, pp. 17-19.

10. R. S. Chandel, A. D. Singh, D. Chouhan, "Solution of Linear and Non – Linear Fractional Order Differential Equation Using Legendre Wavelet Operational Matrix", Journal of Mathematical Analysis, Kosove, 6 (5), (2015), 32 – 42.

11. R. S. Chandel, A. D. Singh, D. Chouhan, "A Wavelet Operational Matrix Method for Solving Initial – Boundary Value Problems for Fractional Partial Differential Equations", Journal of Mathematical and Computational Science, 6 (4), (2016), 527 – 539.

12. A. K. Gupta, "Management Information System", 2012, New Delhi, India: S Chand Publishing.

13. R. S. Chandel, A. D. Singh, D. Chouhan, "Numerical Solution of Fractional Relaxation – Oscillation Equation Using Cubic B – Spline Wavelet Collocation Method", Italian Journal of Pure and Applied Mathematics, 36 (2016), 399 – 414.

14. R. S. Chandel, A. D. Singh, D. Chouhan, "Numerical Solution of Fractional Order Differential Equations Using Haar Wavelet Operational Matirx", Palestine Journal of Mathematics, 6 (2), (2017), 515 – 523.

15. Y. Chen, Y. Wu, Y. Cui, Z. Wang, D. Jin, "Wavelet method for a class of fractional convection – diffusion equation with variable coefficients", J. Comput. Sci. 1 (2010), 146 – 149.

16. D. Chouhan, R. S. Chandel, "Numerical Solution of the Convection Diffusion Equation by the Legendre Wavelet Method", Jnanabha, Vijnana Parishad of India. 49 (1), (2019), 26 – 39.

17. D. Chouhan, V. K. Mishra, H. M. Srivastava, "Bernoulli wavelet method for numerical solution of anomalous infiltration and diffusion modeling by nonlinear fractional differential equations of variable order", Results in Applied Mathematics, 10 (2021).

18. A. Waller, "Special issue on Identity Protection and Management", Journal of Information Security and Applications, 2014, Vol. 19.

19. D. Chouhan, R. S. Chandel, U. Dolas, "Solving fractional differential equations characterizing the dynamics of a current collection system for an electric locomotive using Shannon Wavelet", Jnanabha, Vijnana Parishad of India, 51 (1), (2021), 79 – 87.

20. A. D. Singh, R. S. Chandel, D. Chouhan, "A Numerical approach for Solving boundary value problems for fractional differential equations using Shannon Wavelet", J. Math. Comput. Sci., 6(6), (2016), 1085 – 1099.

21. K. Bernard, "Discrete mathematical structures", 2007, New Delhi, India: Person Education India (PHI).

22. A. D. Singh, R. S. Chandel, D. Chouhan, "An Introduction to Wavelets and Their Applications, Bulletin of the Calcutta Mathematical Society", 107 (3), (2015), 241 – 262.

23. A. D. Singh, R. S. Chandel, D. Chouhan, "Solving Multi – Order Linear and Non – Linear Fractional Differential Equations Using Chebyshev Wavelets", JNANABHA, Vijnana Parishad of India, 44 (2014), 69 – 80.

24. E. B. Heinlein, "Principles of Information Systems Security", Computers & Security, Vol. 14, 1995, pp. 197-l98.

25. A. D. Singh, R. S. Chandel, D. Chouhan, "Solution of Higher Order Volterra Integro – Differential Equations by Legendre Wavelets, International Journal of Applied Mathematics", Bulgaria, 28 (4), (2015), 377 – 390.