Plag

by Sss Sss

Submission date: 25-Nov-2022 04:49AM (UTC-0800)
Submission ID: 1963073419
File name: F.docx (229.66K)
Word count: 1403
Character count: 7686

Customer Segmentation Using K-Means and DBSCAN Clustering: A Comparative Study

Methodology

Tools and techniques

This paper consists of the customer segmentation process. For this purpose different kinds of tools have been used. In the modern scenario, different advanced machine learning and deep learning-based tools have occupied the market for customer segmentation. Python is used for the visualization of this segmentation process. But in this project customer segmentation is done by a comparative study using K-Means and DBSCAN clustering techniques. A very useful python library, "pandas", is used for analyzing data and machine learning purposes. Numpy helps to handle multi-dimensional data. Seaborn is another library that helps to visualize graphs by providing some attractive themes. Matplatlib creates plotted graphs by analyzing the dataset. The DateTime module helps to manipulate dates and times to implement customer segmentation. For clustering purposes, different algorithms are used. For analyzing and categorizing raw data it is used. The dendrogram is used to visualize the relationship among different clusters. Linkage, dendrogram, etc are imported for retrieving different relationships from the database. In this project a web-based code editor, Jupyter Notebook is used which provides a very simplified visualized interface (Chindyana *et al*. 2021).

Data Collection

Data collection is the most important part of differentiation from the customers. In this process, the transactional dataset used belongs to online retail companies. The data of customers is verified as all the details are taken from the ideal clients of the companies. It helps to understand the actual necessity of the customer and better understand them. In this data collection process, it is very important to take all the necessary information of the clients which includes invoice no, stock code, name, and details of customer's purchased products, the quantity of the products, the actual date of making the invoice, customers country and the most important the customer id.

Data Analysis

More than one hundred clustering algorithms are used for segmentation purposes. But few of them are popular till now. But the most widely used methods for making segments are K-means and Density-based spatial clustering of applications with noise (DBSCAN) Clustering. When the data is unsupervised and iterative we use the K-means data mining approach. The

DBSCAN is a comparative data approach process. The k-means clustering process aims to cluster customers by observing the data. The main advantage of using the K-means data mining approach is that it can cluster a large amount of data very efficiently. The iterative algorithm of K-means can make clusters in different segments by analyzing the attributes of the customers. That categorized the data into different groups in a convenient way, unlike machine learning which needs training over time for accuracy. K-means has a centroid-based algorithm that makes each cluster by attaching a centroid. The main purpose for associating centroids is to minimize the total distances between each cluster and their corresponding dataset.

Though the same approaches are used for all clustering, the DBSCAN clustering method is used for creating arbitrarily shaped clusters. K-means fails to create this. DBSCAN clustering is capable of forming clusters that are based on varying densities. In this clustering method, the clusters are made in dense regions in space that are separated by lower-density regions. The most interesting feature of this clustering is that the number of clusters is not required beforehand unlike K-means. DBSCAN consists of two parameters "epsilon" and "minPoints". "Epsilon" refers to the radius of the circle of each data point which is used to check density and "minPoints" refer to the lowest number of data points that are required inside the circle (Shirole *et al*. 2021).

Results

```
1
2
3  import numpy as np
4  import pandas as pd
5  import matplotlib.pyplot as plt
6  import seaborn as sns
7  import datetime as dt
8
9
10 from sklearn.preprocessing import StandardScaler
11 from sklearn.cluster import KMeans
12 from sklearn.metrics import silhouette_score
13 from scipy.cluster.hierarchy import linkage
14 from scipy.cluster.hierarchy import dendrogram
15 from scipy.cluster.hierarchy import cut_tree
16
17
18 import warnings
19 warnings.filterwarnings("ignore")
```

Figure 1: Importing library functions

(Source: Created in Jupyter Notebook)

To complete the data analysis first the library functions are imported into the Jupyter notebook. The following figure describes the imported functions such as the NumPy function, it is the library function of python that gives the ability to work with the arrays. The library

function is beneficial for linear algebra, Fourier transforms, and matrix work (Marisa *et al*. 2019). In 2005 Travis Oliphant created it and this is an open-source project; here numpy stands for numerical python. In this work, numpy is defined as np for making the work easier. Another important function is pandas which are defined in this work as PD, this library function is important in analyzing the data (Koul *et al*. 2021). This library function is very useful for creating the data frame, importing CSV files, and preparing datasets. To visualize the data another important function that is used is matplotlib as this is an important function for such a purpose; this is defined in this work as "plt".

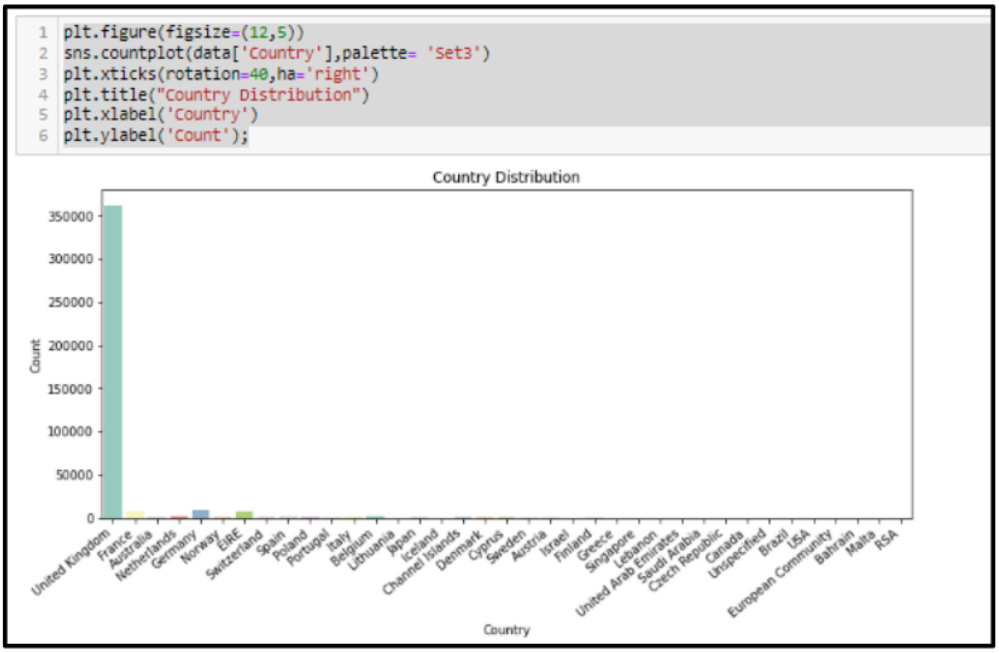

Figure 2: Country distribution analysis

(Source: Created in Jupyter Notebook)

```
plt.figure(figsize=(8,5))
data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'], errors='coerce')
sns.countplot(data['InvoiceDate'].dt.year,palette= 'Set1')
plt.xticks(rotation=40,ha='right')
plt.title("Year Distribution")
plt.xlabel('Year')
plt.ylabel('Count');
```

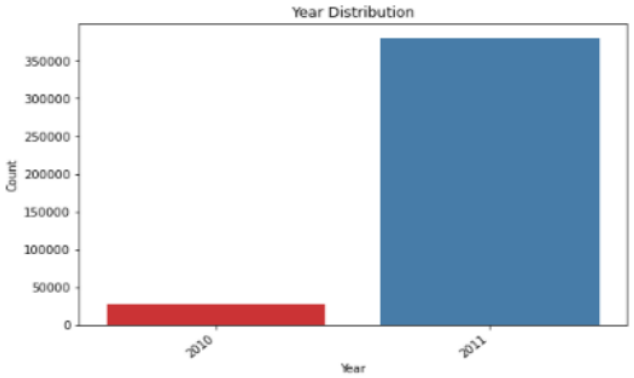

Figure 3: year distribution analysis

(Source: Created in Jupyter Notebook)

```
plt.figure(figsize=(8,5))
plt.xticks(rotation=40,ha='right')
sns.countplot(data['InvoiceDate'].dt.month_name(),palette= 'Spectral')
plt.title("Month Distribution")
plt.ylabel('Count')
plt.xlabel('Month')
```

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

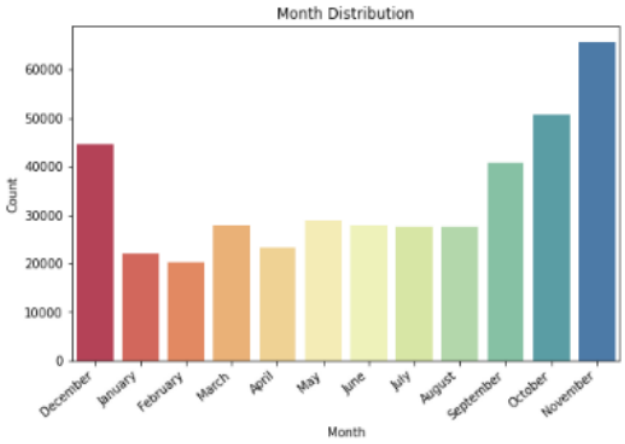

Figure 4: Monthly distribution analysis

(Source: Created in Jupyter Notebook)

```
In [18]:
plt.figure(figsize=(8,5))
sns.countplot(data['InvoiceDate'].dt.day_name(),palette= 'Set1')
plt.xticks(rotation=40)
plt.title("Week Distribution")
plt.xlabel('Week')
plt.ylabel('Count')

Out[18]: Text(0, 0.5, 'Count')
```

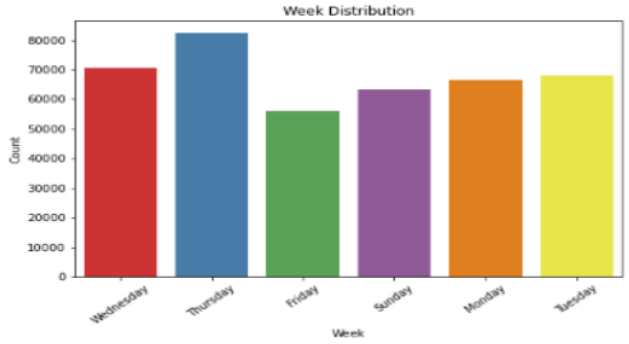

Figure 5: Weekly distribution analysis

(Source: Created in Jupyter Notebook)

In this research work, the analysis for the following datasets is represented using several diagrams and all this analysis was completed in a Jupyter notebook. The analysis for the weekly, monthly, year-wise, and country-wise is done. The weekly analysis says that the weekly distribution is the highest on Thursday and it is represented by the blue colour. The second highest weekly distribution is Wednesday, which is represented by red colour. The monthly distribution says that the distribution is highest in November and it is represented by the deep blue colour. The second highest is October month which is represented by the specified colour. Here is the respective monthly distribution for the other months also represented. The yearly distribution analysis tells the analysis report is highest in 2011 and the graph is generated for only two years 2010 and 2011. The country distribution analysis report says that the United Kingdom has the highest data amazon the respective countries.

```
attributes = ['Amount','Frequency','Recency']
plt.rcParams['figure.figsize'] = [10,8]
sns.boxplot(data = rfm[attributes], orient="v", palette="Set2" ,whis=1.5,saturation=1, width=0.7)
plt.title("Outliers Variable Distribution", fontsize = 14, fontweight = 'bold')
plt.ylabel("Range", fontweight = 'bold')
plt.xlabel("Attributes", fontweight = 'bold')
```

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

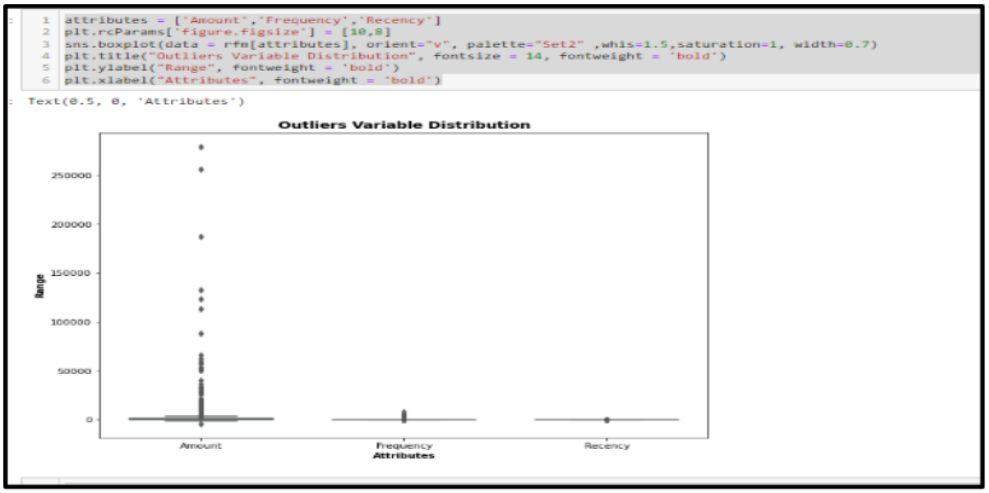

Figure 6: Outliers variable distribution

(Source: Created in Jupyter Notebook)

This is the figure that tells about the outlier variable distribution of different parameters such as amount frequency attributes and Rency. Outliers distribution is an observation that indicates the abnormalities of the data by its spacing over the sample space. In this analysis, three parameters abnormalities of their data are checked and the result is amount has the highest amount of abnormalities in the data (Brahmana *et al*. 2020).

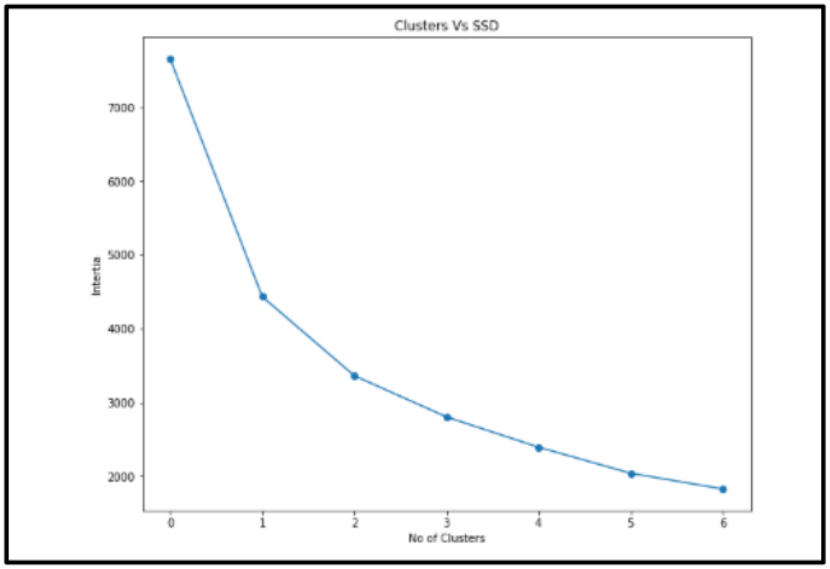

Figure 7: Clusters vs SSD diagram

(Source: Created in Jupyter Notebook)

This is the cluster versus SSD diagram and in this diagram number of clusters, values are zero when the inertia value is 7000 and when the inertia is 2000 the number of clusters is 6.

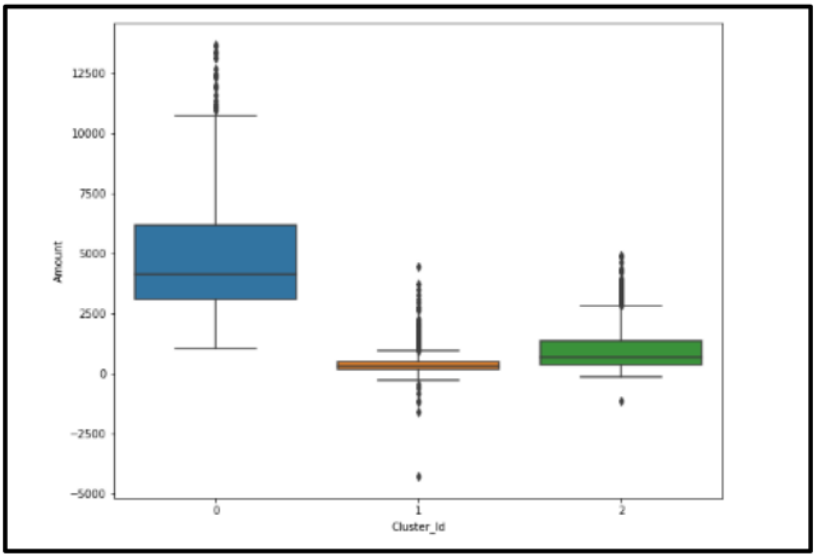

Figure 8: Cluster_ID distribution analysis

(Source: Created in Jupyter Notebook)

This figure depicts the cluster id versus amount diagram where the amount is highest when the cluster id is zero and the amount is near zero when cluster id is 1 and 2.

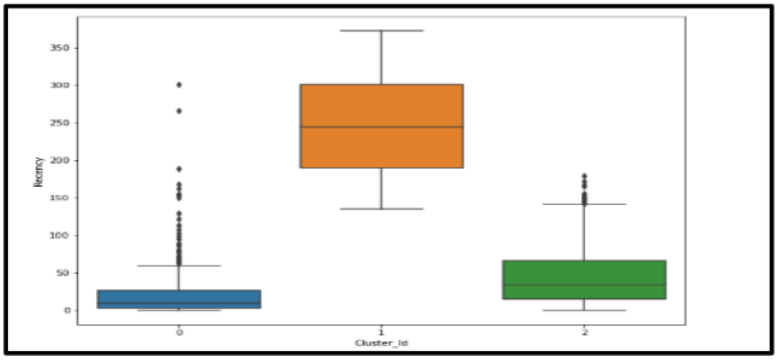

Figure 9: Cluster_ID distribution analysis

(Source: Created in Jupyter Notebook)

This figure depicts the cluster id versus amount diagram for different parameters and different representation styles. The amount is the highest when the cluster id is one and the amount is between zero to fifty when the cluster id is zero and two (Monalisa *et al*. 2019).

```
]:  1 rfm_data_scaled = pd.DataFrame(rfm_data_scaled)
    2 rfm_data_scaled.columns = ['Amount', 'Frequency', 'Recency']
    3 rfm_data_scaled.head()
```

]:

	Amount	Frequency	Recency
0	-0.723738	-0.752888	2.301611
1	1.731617	1.042467	-0.906466
2	0.300128	-0.463636	-0.183658
3	0.277517	-0.044720	-0.738141
4	-0.533235	-0.603275	2.143188

Figure 10: Cluster_ID distribution analysis

(Source: Created in Jupyter Notebook)

This figure describes the cluster id distribution analysis of the different parameters such as amount, frequency, and Rency. In general, this type of analysis is done to identify the potential customers for the marketing campaign, here this analysis is done to identify the strong customer ids who have done the highest amount of the products.

Conclusion:

In this competitive market, companies are required to make new products based on customers' age, occupation, gender, taste, culture, geography, and preferences which will also help to retain the customer's attention. In this case study, customer segmentation has been made so that companies can make products according to the customers' desires. K-means and DBSCAN clustering algorithms are used to differentiate the customer by analyzing the database that contains the information about clients. In this work, the segmentation of customers is calculated from the raw dataset. It will help the companies create more attractive advertisements, optimize the prices of the product, and improve the distribution channel. Different tools and techniques are used for this process. Jupyter Notebook is used as a code editor. Matpltlib, Numpy, seaborn, etc libraries are imported to python to make plot diagrams, better visualization, and link with the database.

Plag

ORIGINALITY REPORT

5%	3%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Lancaster Student Paper	2%
2	Rajani Shetty, Indiramma. "Clustering Analysis of Traffic Accident Dataset using Canopy K Means", 2021 IEEE Mysore Sub Section International Conference (MysuruCon), 2021 Publication	1%
3	www.analyticsvidhya.com Internet Source	1%
4	www.mdpi.com Internet Source	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		