Customer Segmentation using RFM Rankings

Report Submitted to SASTRA Deemed to be University

as the requirement for the course

CSE300 - MINI PROJECT

Submitted by

Mourya Manohar Reddy I (123018037,CSBS)

Harshini A (123018036, CSBS)

Nithin C (123003048,CSE)

June-2022

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA-613401

Bonafide Certificate

This is to certify that the report titled "Customer segmentation using RFM Rankings" submitted as a requirement for the course, CSE300: MINI PROJECT for B.Tech is a bonafide record of the work done by, Mr.Mourya Manohar Reddy I (Reg.No.123018037, B.Tech-CSBS), Mr. Nithin C (Reg.No.123003048, B.Tech-CSE), Ms. Harshini A (Reg.No.123018036, B.Tech-CSBS) during the academic year 2021-22, in the School of Computing, under my supervision.

Signature of the Project Supervisor:

Name with affiliation : **Mr. Bharathy C,** Faculty(SoC)

Date : **June 2022**

Mini Project Viva voce held on _____

Examiner 1

Dr. M. RAJA

Examiner 2

Acknowledgement

We have taken efforts in this project as a team. However, it would have not been possible without the kind of support and help of many individuals and the organization. I would like to express my sincere thanks to all of them.

We are grateful to our Vice Chancellor **Dr. S. Vaidhyasubramaniam** and our Registrar **Dr. R. Chandramouli** for giving me the opportunity to work on this project and for providing their continuous support.

We are also thankful to our Dean **Dr. A. Umamakeswari** for her guidance and support in completing the project.

We are highly indebted to **Mr. Bharathy C** for his guidance and constant supervision as well as providing necessary information regarding the project continuously and also for her support in completing the project on time. Working on this project was an immense source of knowledge to us. We also express our sincere thanks to our guide for extending valuable guidance, support and critical review of the project and for the moral support provided at all points of time.

I would also like to express my gratitude towards my parents and members of SASTRA University for their kind cooperation and encouragement which helped me in completion of this project.

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1.1	Syntax of pandas *groupby* function	3
3.1.2	Syntax of numpy *percentile* function	4
3.1.3	Formula to recalculate the centroid of the cluster in FCM	4
3.2.1	Snapshot of first few rows of the dataset	6
3.2.2a	Count of null values in the dataset	7
3.2.2b	Count of null values after removing them	7
3.2.2c	Number of duplicate values	7
3.2.3a	Pair plots for Quantity and Unitprice attributes	8
3.2.3b	Heatmap for the pairplots	8
3.2.3c	Histogram of Quantity attribute	8
3.2.3d	Histogram of Unitprice attribute	8
3.2.4a	Dataset after finding RFM values and eliminating duplicates	11
3.2.4b	RFM values for each customer	11
3.2.5	RFM values and corresponding scores for each customer	13
3.2.6	K-Means plot	14
3.2.7	Fuzzy C-Means plot	15
3.2.8	RM K-Means plot	16
3.2.9	Silhouette plot of K-Means clustering with 10 clusters	17
3.2.10	Silhouette plot of K-Means clustering with 5 clusters	19
4.1	Home page of developed GUI	21
4.2	Uploading the dataset csv file	21
4.3	Submitting the dataset csv file	22
4.4	Sample data of the uploaded csv file	22
4.5	Selection of attributes	23
4.6	Enter the scores(0 to 5) to predict the cluster	23

4.7	Selection of the algorithm to be applied	24
4.8	Display of the cluster to which the given scores belong to	24

List of Tables

TABLE NO.	TITLE	PAGE NO.
3.1	Comparative analysis of algorithms	**20**

Abbreviations

RFM- Recency, Frequency, Monetary
RM K-Means- Repetitive Median K-Means
FCM- Fuzzy C-Means
GUI- Graphical User Interface

Notations

No specific notations were used.

ABSTRACT

The efficient segmentation of customers of an enterprise is categorized into groups of similar behavior based on the RFM (Recency, Frequency and Monetary) values of the customers. The transactional data of a company is analyzed over a specific period. Segmentation gives a good understanding of the needs of the customers and helps in identifying the potential customers of the company. Dividing the customers into segments also increases the revenue of the company.

It is believed that retaining the customers is more important than finding new customers. For instance, the company can deploy marketing strategies that are specific to an individual segment to retain the customers.

This study initially performs an RFM analysis on the transactional data and then extends to cluster the same using traditional K-means and Fuzzy C- Means algorithms. In this paper, a novel idea for choosing the initial centroids in K- Means is proposed.

The results obtained from the methodologies are compared with one another by their iterations, cluster compactness and execution time.

KEY WORDS: *RFM Values, K-Means, Fuzzy C-Means, RM K-Means*.

Table of Contents

Title	Page No.
Bonafide Certificate	ii
Acknowledgements	iii
List of Figures	iv
List of Tables	v
Abbreviations	vi
Notations	vi
Abstract	vii
1. Summary of the base paper	1
2. Merits and Demerits of the base paper	2
3. Project Phases & Source Code	3
4. Snapshots	21
5. Conclusion and Future Works	25
6. References	26
7. Appendix -Base Paper	27

CHAPTER 1
SUMMARY OF THE BASE PAPER

Title: RFM Ranking - An effective approach to customer segmentation

Journal: King Saud University-Computer and Information Sciences

Publisher: Elsevier, Radarweg 29a, Amsterdam, Netherlands, 1043 NX

Web of science Core Collection: Science Citation Index Expanded

Published: 10 December 2020

Recent years, especially in the pandemic have seen a sharp rise in business competition to stay alive in their industry. To make the current clients stay is more important than getting the new customers. Because the retended customers alone contribute more to the profit of the company than the rest of the customers. Therefore, in this paper, a customer segmentation model was suggested to help the businesses by informing them of the pickup times of the clients.

For this segmentation RFM analysis was used – Recency, Frequency, Monetary.

Due to its availability, continual change over time, and history of purchases, behavioral data is frequently utilized for segmentation. The data is initially preprocessed to separate outliers and filter important situations. The number of days a customer waits between two purchases is referred to as Recency in RFM analysis. F stands for Frequency, or how many purchases a customer makes in a certain time frame. M stands for money, or the sum that the customer spent.

Scores are supplied on a scale of 0 to 5 as RFM scores following analysis and observation of the rfm values acquired for a specific set of data. Clustering is done after acquiring the pertinent scores. Three different methods have been used to cluster the customers based on RFM analysis. Customers are segmented using the clustering methods K-Means, Fuzzy C-Means, and Repetitive Median based K-Means (RM K-Means). The efficiency of the clustering techniques is then assessed in terms of the execution time, cluster compactness, and iteration count.
As segmentation is concluded based on the rfm scores the respective enterprise can alter their marketing approaches to the customers based on their purchasing ways.

CHAPTER 2

Merits and Demerits of the Base Paper

2.1 Merits:

Recency, frequency, and monetary (RFM) analysis is a potent and well-known database marketing tool. Based on their previous purchasing histories, the clients are ranked using this strategy. The analysis is useful in many different contexts when there are plenty of clients, like internet shopping, retailing, etc. Customer segmentation will strengthen the engagement with customers.

In this study, segmentation is carried out using RFM analysis, and it is then expanded to other algorithms, including K-Means clustering, Fuzzy C-Means, and a new method called RM K-Means, by making slight adjustments to the already-existing K-Means clustering.

Each algorithm's execution time is studied, and the algorithm that is observed to take the least amount of time and require the fewest iterations is the result.

2.2 Demerits:

We defined and tallied the benefits after giving a brief introduction to RFM analysis. In addition, I want to point out that RFM analysis might not be enough to achieve ambitious targets and boost ROAS. Sadly, RFM segmentation is insufficient for ad optimization. So why?

RFM analysis measures consumer behavior, which is one of the initial causes. Although it is employed to track visitor activity, it cannot be stated to be a useful technique for enhancing visitor data. Many businesses, nevertheless, aim to increase both existing and future customers.

Second, RFM analysis simply uses three metrics to make calculations: recency, frequency, and monetary. The findings of measurements produced using these metrics might, however, not always be complete. To better understand visitor and consumer behavior, there ought to be more and more varied indicators. Even data analytics platforms like Enhancer have capabilities that are more sophisticated than this.

Thirdly, e-commerce businesses are transforming and growing every day. In the world of data analytics and advertising, instant campaigns and tactics are practically required. As a result, the machine learning algorithm's self-learning and autonomous segment analysis are also more valuable and practical.

CHAPTER 3

Project Phases and Code

3.1 Project Phases:

3.1.1 Phase-1:

In this project, in order to cluster the customers efficiently we have to find the RFM values and assign the scores on a scale of 0-5. Apply various clustering techniques and compare the efficiency and working of each technique. First dataset was collected from the UCI Machine Learning Repository.

The dataset used is mentioned below

- Online Retail Dataset.

3.1.2 Phase-2:

Then the collected dataset was pre-processed using various preprocessing techniques. The operations carried out on the dataset are listed below:

1. Data Cleaning
    1. Removal of Null Values
    2. Removal of Redundant Values
    3. Removal of Outliers
2. Missing Values

3.1.3 Phase-3:

RFM values were calculated from the given dataset by doing various operations on specific attributes and corresponding scores were given to them on the scale of 0 to 5.
Functions used to calculate the scores are:

1. **Pandas groupby:** A groupby operation involves some combination of splitting the object, applying a function, and combining the results. This can be used to group large amounts of data and compute operations on these groups.

```
DataFrame.groupby(by=None, axis=0, level=None, as_index=True, sort=True,
group_keys=True, squeeze=NoDefault.no_default, observed=False,
dropna=True)                                                    [source]
```

Fig 3.1.1. Syntax of a Pandas groupby function

2. **Numpy percentile:** Computes the q-th percentile of the data along the specified axis. Returns the q-th percentile(s) of the array elements.

```
numpy.percentile(a, q, axis=None, out=None, overwrite_input=False,
method='linear', keepdims=False, *, interpolation=None) #          [source]
```

Fig 3.1.2. Syntax of Numpy percentile function

3.1.4 Phase-4:

Here after attaining RFM scores, we use various clustering algorithms to fit the data and plot them. We use K-Means Algorithm, Fuzzy C-Means Algorithm and RM K-Means Algorithm. Corresponding Silhouette Plots were drawn.

1. **K-Means Algorithm:** It segregates unlabeled data into various groups called clusters. It is an unsupervised learning algorithm because it performs on an unsupervised dataset. It is an Iterative algorithm and makes sure that items belonging to the same group has similar characteristics.

2. **Fuzzy C-Means Algorithm:** This is an unsupervised learning algorithm which allows an item to be present in more than one cluster. This is way more powerful than the standard K-Means algorithm because membership is assigned based on the distance between the center of the cluster and the point, which ultimately leads to the summation of memberships of a datapoint to 1.

$$\mu_k(n+1) = \frac{\sum_{x_i \epsilon k} x_i * P(\mu_k|x_i)^b}{\sum_{x_i \epsilon k} P(\mu_k|x_i)^b}$$

Fig 3.1.3. Recalculate the centroid of the cluster as the weighted centroid given the probabilities of membership of all data points.

3. **RM K-Means Algorithm:** Traditional k-means choose initial centroids randomly. The problem is that there is a chance that clusters become less meaningful due to random centroids. So, we find the median of RFM Scores after ordering them in ascending order to pass as arguments to the k-means. Choosing the initial centroids as median values reduces the number of iterations and increases the cluster's meaningness.

4. **Silhouette Plot:** The silhouette plot offers a visual approach to evaluate factors like the number of clusters by displaying a measure of how close each point in one cluster is to points in the neighboring clusters. The range of this metric is [-1, 1].

The Silhouette Score is a measure of how similar an object is to its own cluster in comparison to other clusters.

3.1.5 Phase-5:

Performance Improvement: In the Silhouette plot, some clusters don't meet the Average Silhouette Score, so it was decided to reduce the number of clusters to 5 instead of 10. After, the Average Silhouette Score is calculated and found out that all clusters have a score at least that of the new Silhouette Score.

3.1.6 Phase-6:

Web Application: In addition to what was proposed in the base paper, we went on and created an aesthetic GUI and included functionality which enables the user to select the attributes and then select the clustering algorithm to fit the data. Then user can give input of RFM scores of their own and the application predicts to which cluster the entered values belong.
Technologies used is listed below:

- HTML
- CSS
- Python
- Flask (for backend integration)

3.2 Source Code:

Mounting Google Drive for loading Dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

Importing Necessary Packages to Implement

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
!pip3 install fuzzy-c-means
from yellowbrick.cluster import SilhouetteVisualizer
from sklearn.metrics import silhouette_score
```

Loading DataSet and Preparation

```
data = pd.read_excel("/content/drive/MyDrive/Mini Project DataSet/Online Retail.xlsx")
data.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Fig 3.2.1. Snapshot of first few rows of Dataset

Data Preprocessing

```
#Removing Null Values

data.isnull().sum()                    #Counting no.of null values initially
data=data.dropna()
data.isnull().sum()                    #Counting no.of null values after dropping them


#Removing Duplicate Values
```

```
data.duplicated().sum()                    #Counting no.of duplicate values initially
data_new= data.drop_duplicates()



#Removing Outliers
   #Finding Q1,Q2,Q3(Quartiles), Max and Min for given attributes using five-number summary

cols = ['Quantity', 'UnitPrice']

Q1 = data[cols].quantile(0.25)
Q3 = data[cols].quantile(0.75)
IQR = Q3 - Q1

data_cleaned = data[~((data[cols] < (Q1 - 1.5 * IQR)) |(data[cols] > (Q3 + 1.5 * IQR))).any(axis=1)]
```

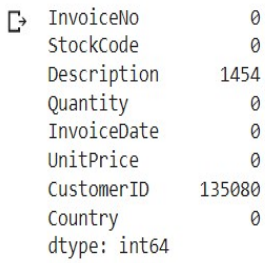


Fig 3.2.2a Count of null values in the dataset.

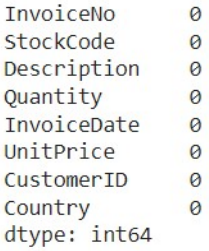


Fig 3.2.2b. Count of null values after removing them.

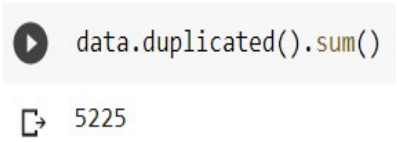


Fig 3.2.2c. Number of duplicate values

Plottings

```
#PairPlots

import seaborn as sns
sns.pairplot(data_cleaned[['Quantity', 'UnitPrice']].loc[:10000])

#Corresponding HeatMap

sns.heatmap(data_cleaned[['Quantity', 'UnitPrice']].corr(method='pearson'), annot=True)


#Histograms
```

```
plt.hist(data_cleaned['Quantity'])
plt.hist(data_cleaned['UnitPrice'])
```

```
<seaborn.axisgrid.PairGrid at 0x7fbe15287050>
```

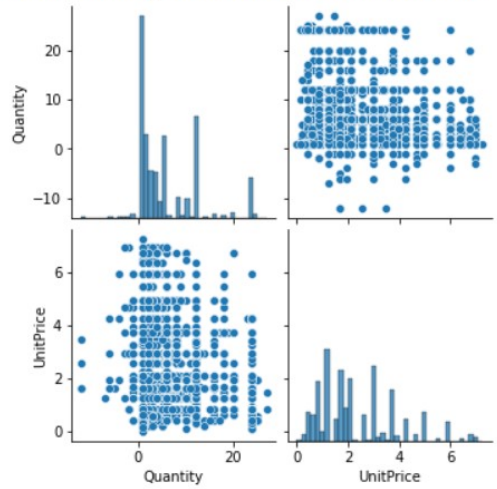


Fig 3.2.3a Pair Plots for Quantity and UnitPrice attributes

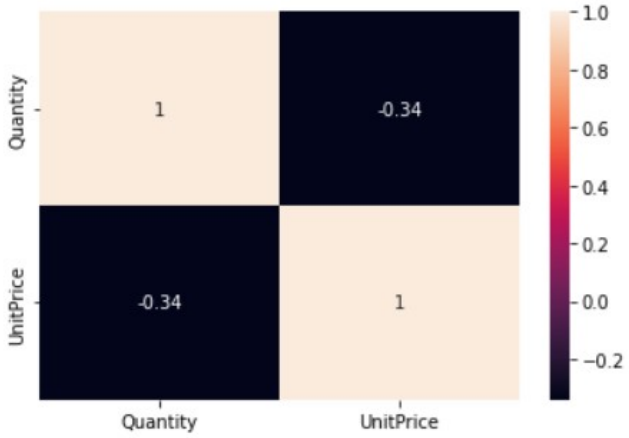


Fig 3.2.3b Corresponding HeatMap

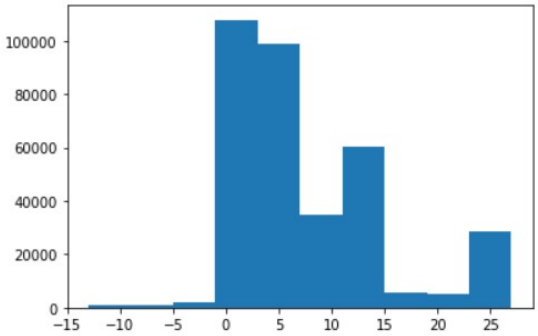


Fig 3.2.3c Histogram of Quantity

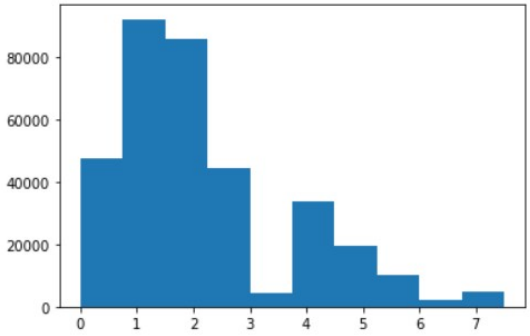


Fig 3.2.3d Histogram of UnitPrice

Calculating RFM Values

Finding Recency Value:

```
#Step-1: Split the InvoiceDate to Date and Time separately

data_cleaned['Date'] = pd.to_datetime(data_cleaned['InvoiceDate']).dt.date

#Step-2: Copy data to a dataframe and find the most recent purchase date of a customer
```

```
x = pd.DataFrame()
x['Recency'] = data_cleaned.groupby('CustomerID')['Date'].max()
x = x.reset_index()
print(x)

#Step-3: Insert the Max. date generated into the original dataset

data_cleaned['Recency1'] = 0
for j in x['CustomerID']:
  data_cleaned['Recency1'][data_cleaned['CustomerID'] == j] =
x['Recency'][x['CustomerID']==j].values[0]

#Step-4: In another dataframe find the difference between the Max. date and the actual date
of every purchase of every customer

data_r = data_cleaned.copy()
data_r['r'] = abs(data_r['Date'] - data_r['Recency1']).dt.days

#Step-5: Remove the null values

data_r = data_r[data_r['r']!=0]

#Step-6: Find the minimum of the differences calculated above for each customer which
ultimately is the Recency Value

x_r = pd.DataFrame()
x_r['Recency'] = data_r.groupby('CustomerID')['r'].min()
x_r = x_r.reset_index()
print(x_r)


#Step-7: Load these values into original dataset as RECENCY

data_cleaned['Recency'] = 0
for j in x_r['CustomerID']:
  data_cleaned['Recency'][data_cleaned['CustomerID'] == j] =
x_r['Recency'][x_r['CustomerID']==j].values[0]
```

Finding Frequency Value:

```
#Load the data into a new dataframe and find the no.of unique Invoice Numbers for each
```

```
customer which is our Frequency Value

y = pd.DataFrame()
y['count'] = data_cleaned['InvoiceNo'].groupby(data_cleaned['CustomerID']).nunique()
y = y.reset_index()
y

#Load these values into original dataset as FREQUENCY

data_cleaned['Frequency'] = 0
for j in y['CustomerID']:
  data_cleaned['Frequency'][data_cleaned['CustomerID'] == j] =
y['count'][y['CustomerID']==j].values[0]

#Checking for Null Values

data_cleaned['Frequency'].isna().sum()
```

Finding Monetary Value:

```
#Calculate the Amount for each row, group it by customers and adding them gives the
Money spent by the each customer

data_cleaned['bill'] = data_cleaned['Quantity']*data_cleaned['UnitPrice']
z = pd.DataFrame()
z['total'] = data_cleaned['bill'].groupby(data_cleaned['CustomerID']).sum()
z = z.reset_index()
z


#Load the values into original dataset

data_cleaned['Monetary'] = 0
for i in z['CustomerID']:
    data_cleaned['Monetary'][data_cleaned['CustomerID'] == i] =
z['total'][z['CustomerID']==i].values[0]
```

#Checking for Null Values in all fields

```
data_cleaned.isnull().sum()


#Load required attributes into a new dataframe

data_rfm = data_cleaned[['CustomerID', 'Recency', 'Frequency', 'Monetary']]
data_rfm


#Drop the duplicates

data_rfm.duplicated().sum()
data_rfm= data_rfm.drop_duplicates()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Date	Recency1	Frequency	bill	Monetary	Recency
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010-12-01	2011-02-10	35	15.30	4359.58	70
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12-01	2011-02-10	35	20.34	4359.58	70
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010-12-01	2011-02-10	35	22.00	4359.58	70
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12-01	2011-02-10	35	20.34	4359.58	70
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12-01	2011-02-10	35	20.34	4359.58	70

Fig 3.2.4a Dataset after finding RFM values and eliminating duplicates.

	CustomerID	Recency	Frequency	Monetary
0	17850.0	70	35	4359.58
8	13047.0	15	17	2537.97
21	12583.0	16	16	5252.69
72	14688.0	32	25	3561.06
88	17809.0	113	5	728.39
...	...	...	...	...
341781	13436.0	0	1	113.54
341851	15520.0	0	1	320.70
342507	13298.0	0	1	90.00
343189	14569.0	0	1	80.09
344264	12713.0	0	1	722.85

4222 rows × 4 columns

Fig 3.2.4b RFM values for each customer

Assigning Scores on the scale 0-5 for RFM values

```
#Using percentiles, Recency Scores were awarded

data_rfm['RecencyScore'] = pd.cut(data_rfm["Recency"],
                                  bins=[-1,
                                      np.percentile(data_rfm["Recency"], 20),
                                      np.percentile(data_rfm["Recency"], 40),
                                      np.percentile(data_rfm["Recency"], 60),
                                      np.percentile(data_rfm["Recency"], 80),
                                      data_rfm["Recency"].max()],
                                  labels=[5, 4, 3, 2, 1]).astype("int")
data_rfm["RecencyScore"].value_counts()


#Frequency Scores are awarded below

data_rfm['FrequencyScore'] = pd.cut(data_rfm["Frequency"],
                                  bins=[-1,
                                      np.percentile(data_rfm["Frequency"], 20),
                                      np.percentile(data_rfm["Frequency"], 40),
                                      np.percentile(data_rfm["Frequency"], 60),
                                      np.percentile(data_rfm["Frequency"], 80),
                                      data_rfm["Frequency"].max()],
                                  labels=[1, 2, 3, 4, 5]).astype("int")
data_rfm["FrequencyScore"].value_counts()


#Monetary Scores are awarded below

data_rfm['MonetaryScore'] = pd.cut(data_rfm["Monetary"],
                                  bins=[-1,
                                      np.percentile(data_rfm["Monetary"], 20),
                                      np.percentile(data_rfm["Monetary"], 40),
                                      np.percentile(data_rfm["Monetary"], 60),
                                      np.percentile(data_rfm["Monetary"], 80),
                                      data_rfm["Monetary"].max()],
                                  labels=[1, 2, 3, 4, 5]).astype("float")
data_rfm["MonetaryScore"].value_counts()
```

Finalizing the dataset

```
#Checking for Null Values

data_rfm.isna().sum()

#Drop them if found any
data_rfm = data_rfm.dropna()
```

	CustomerID	Recency	Frequency	Monetary	RecencyScore	FrequencyScore	MonetaryScore
0	17850.0	70	35	4359.58	2	5	5.0
8	13047.0	15	17	2537.97	3	5	5.0
21	12583.0	16	16	5252.69	3	5	5.0
72	14688.0	32	25	3561.06	2	5	5.0
88	17809.0	113	5	728.39	1	4	4.0

Fig 3.2.5. RFM values and corresponding scores for each customer.

Applying K-Means Clustering Algorithm

```
#Fit the data before plotting

from sklearn.cluster import KMeans
from datetime import datetime
start_time = datetime.now()
kmeans = KMeans(n_clusters = 10, random_state = 42)
kmeans.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
data_rfm['cluster'] = kmeans.predict(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']])
end_time = datetime.now()


#Time taken to fit the data

timetaken = end_time - start_time
kmeanstime = timetaken
print(kmeanstime)


#Plotting K-Means Plot

import matplotlib.pyplot as plt
```

```
x = data_rfm['RecencyScore']
y = data_rfm['FrequencyScore']
z = data_rfm['MonetaryScore']
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection ="3d")
ax.scatter3D(x, y, z, c = data_rfm['cluster'])
plt.title("RFM cluster plot")
plt.show()
```

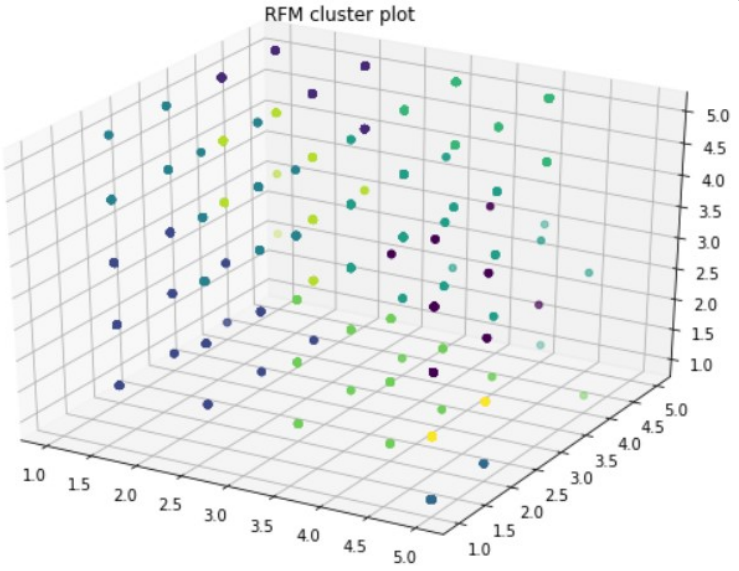


Fig 3.2.6. K-Means plot

Applying Fuzzy C-Means Algorithm

```
#Fuzzy C-Means only supports input file type of CSV, so generate a csv file of required attributes and give as input

data_rfm.to_csv("/content/drive/MyDrive/Mini Project DataSet/rfm_with_clusters.csv", index = False)

#Fit the data

from fcmeans import FCM
import numpy as np
from datetime import datetime
data = np.array(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
start_time = datetime.now()
fcm = FCM(n_clusters = 10)
fcm.fit(data)
```

```
data_rfm['fcmcluster'] = fcm.predict(data)
end_time =  datetime.now()

#Time taken to fit the data

timetaken = end_time-start_time
fcmtime = timetaken
fcmtime


#Plotting Fuzzy C-Means plot

import matplotlib.pyplot as plt
x = data_rfm['RecencyScore']
y = data_rfm['FrequencyScore']
z = data_rfm['MonetaryScore']
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection ="3d")
ax.scatter3D(x, y, z, c = data_rfm['fcmcluster'])
plt.title("RFM fcm cluster plot")
plt.show()
```

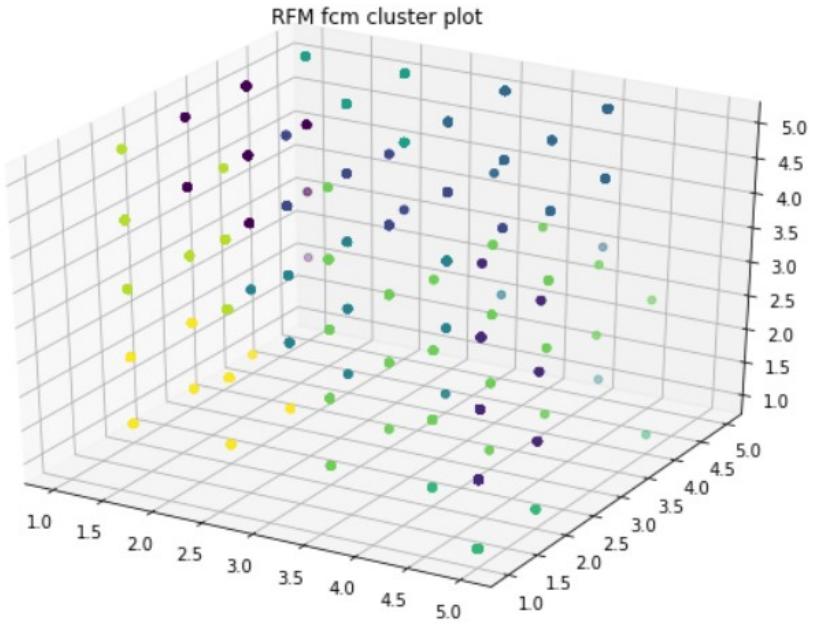


Fig 3.2.7. Fuzzy C-Means plot

Applying RM K-Means Clustering Algorithm

```
#Finding the Centers of clusters

Kmeans.cluster_centers_

#Fitting the data and Plotting RM K-Means plot along with time calculation

start_time=datetime.now()
kmeans_r = KMeans(n_clusters=10, init='random')
kmeans_r.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
data_rfm['cluster_r'] = kmeans_r.predict(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']])
end_time=datetime.now()
time taken=end_time-start_time
rmtime=timetaken
x = data_rfm['RecencyScore']
y = data_rfm['FrequencyScore']
z = data_rfm['MonetaryScore']
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection ="3d")
ax.scatter3D(x, y, z, c = data_rfm['cluster_r'])
plt.title("RFM rkmeans cluster plot")
plt.show()


#Time taken to fit the data

rmtime
```

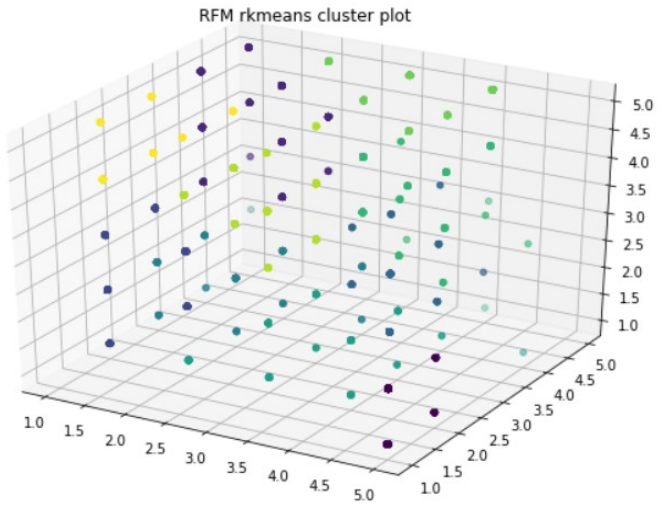


Fig 3.2.8. RM K-Means plot

Silhouette Plot

```
#Plot

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 10, random_state = 42)
visualizer = SilhouetteVisualizer(kmeans, colors='yellowbrick')
visualizer.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
visualizer.show()

#Silhouette Score

silhouette_score(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']],data_rfm['cluster'])
```

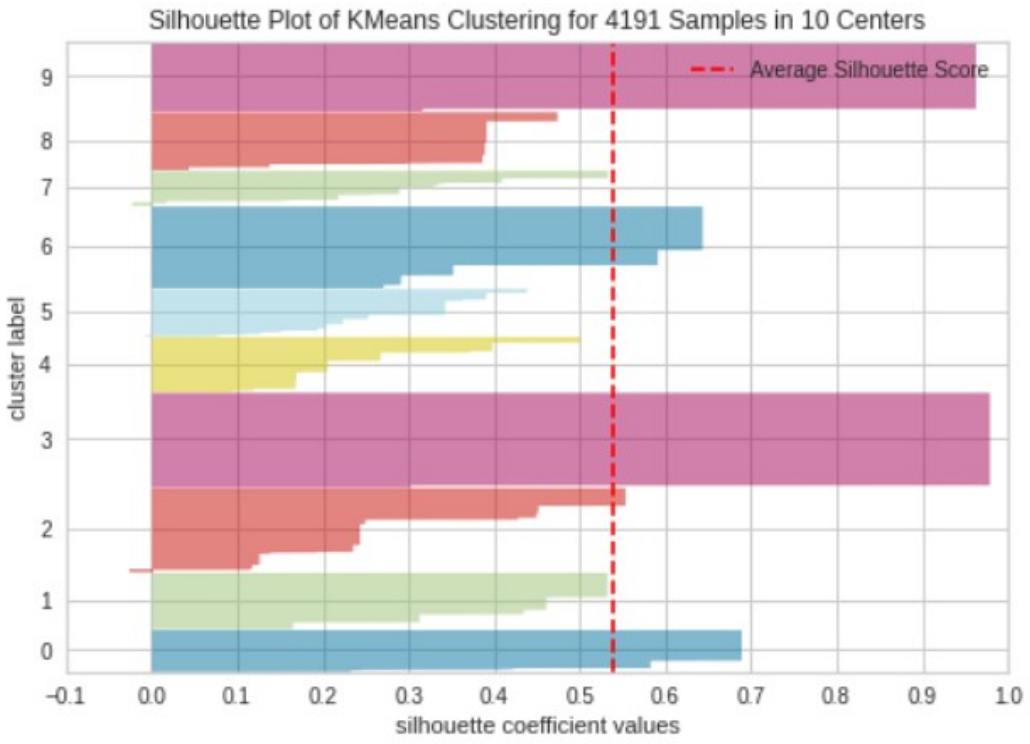


Fig 3.2.9 Silhouette plot of K-Means clustering with 10 clusters

Improving the plots

#Reduce the no.of clusters to 5 and fit the data for all the clustering algorithms and find the time taken to fit the data

K-Means Algorithm:

```
#Fit the data

from sklearn.cluster import KMeans
from datetime import datetime
start_time = datetime.now()
kmeans = KMeans(n_clusters = 5, random_state = 42)
kmeans.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
data_rfm['cluster'] = kmeans.predict(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']])
end_time = datetime.now()

#Time taken

timetaken= end_time-start_time
kmeans5clusters=timetaken
kmeans5clusters
```

Fuzzy C-Means Algorithm:

RM K-Means Algorithm:

```
#Fit the data

start_time=datetime.now()
kmeans_r = KMeans(n_clusters=5, init='random')
kmeans_r.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
data_rfm['cluster_r'] = kmeans_r.predict(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']])
end_time=datetime.now()

#Time taken
```

```
timetaken=end_time-start_time
rm5time=timetaken
rm5time
```

Silhouette Plot

```
#Plot

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 5, random_state = 42)
visualizer = SilhouetteVisualizer(kmeans, colors='yellowbrick')
visualizer.fit(data_rfm[['RecencyScore', 'FrequencyScore', 'MonetaryScore']])
visualizer.show()

#Silhouette Score

silhouette_score(data_rfm[['RecencyScore', 'FrequencyScore',
'MonetaryScore']],data_rfm['cluster_5'])
```

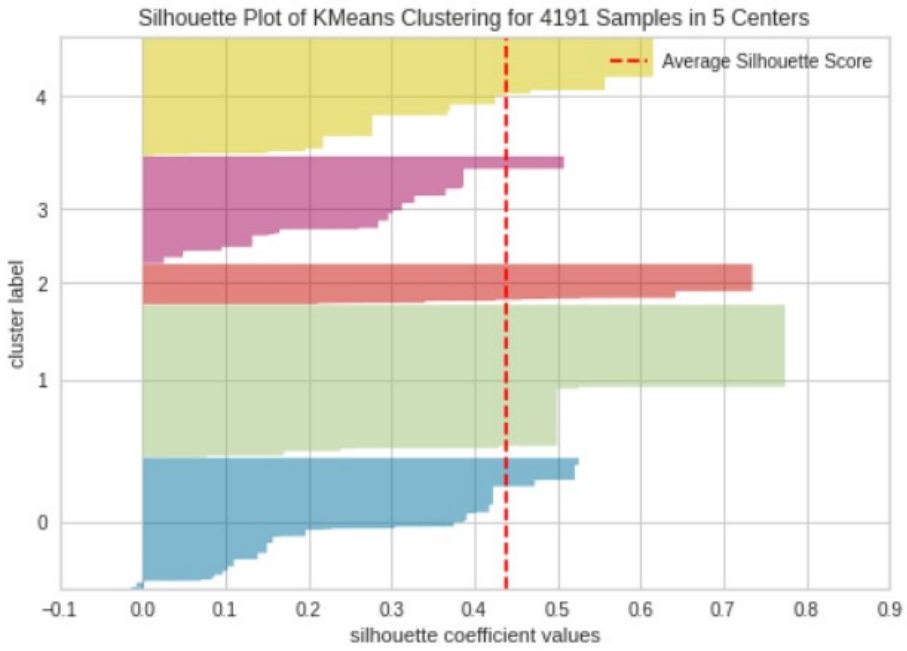


Fig 3.2.10. Silhouette plot of K-Means clustering with 5 clusters

Results of Different Models

	K-Means	Fuzzy C-Means	RM K-Means
No.of Clusters	10	10	10
Time Taken (in sec)	0.375	0.81	0.069
Time Taken for 5 Clusters (in sec)	0.25	0.58	0.05

Table 3.1Comparative analysis of algorithms.

Source Code for Prediction in the Web Application:

```
def algores(algo):
    alg = None
    if algo[0] == "KMeans()":
        alg = KMeans(n_clusters=5)
    elif algo[0] == "FCM()":
        alg = FCM(n_clusters=5)
    data = pd.read_csv("D:\\Mini project\\file.csv", index_col=0)
    x = X[-1]

    alg.fit(data[x])
    x_2 = [int(r[-1]), int(f[-1]), int(m[-1])]
    x_22 = np.reshape(x_2, (1, -1))
    preds = alg.predict(x_22)
    return preds[0]
```

```
def result():
    result = algores(algo)
    if result == 5:
        res = "Very Good Customer"
    elif result == 4:
        res= "Good Customer"
    elif result==3:
        res="Average Customer"
    elif result==2:
        res="Below Average Customer"
    elif result==1:
        res="Poor Customer"
    elif result==0:
        res="Very Poor Customer"
    return render_template("result.html", result=result, res = res)
```

CHAPTER 4

SNAPSHOTS

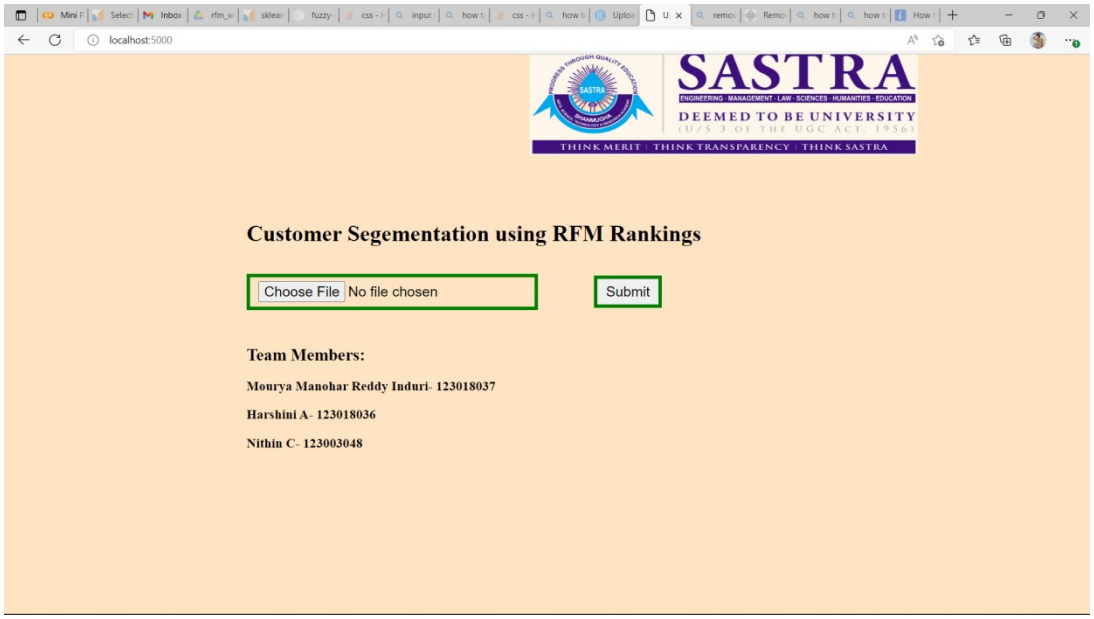


Fig. 4.1 Home page of the developed user interface.

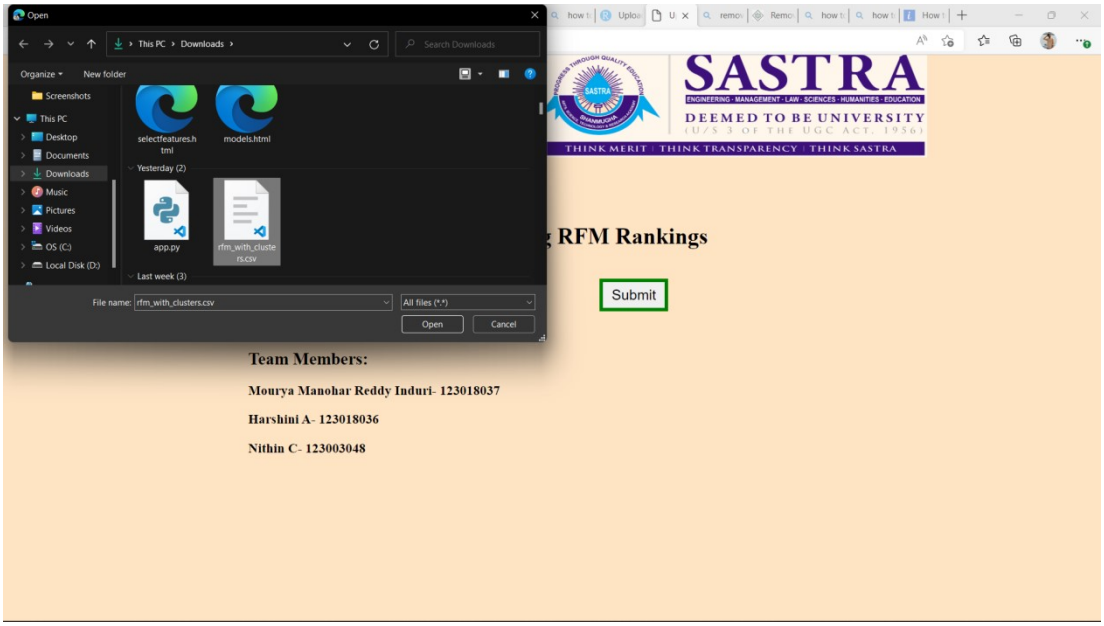


Fig. 4.2 Uploading the dataset csv file.

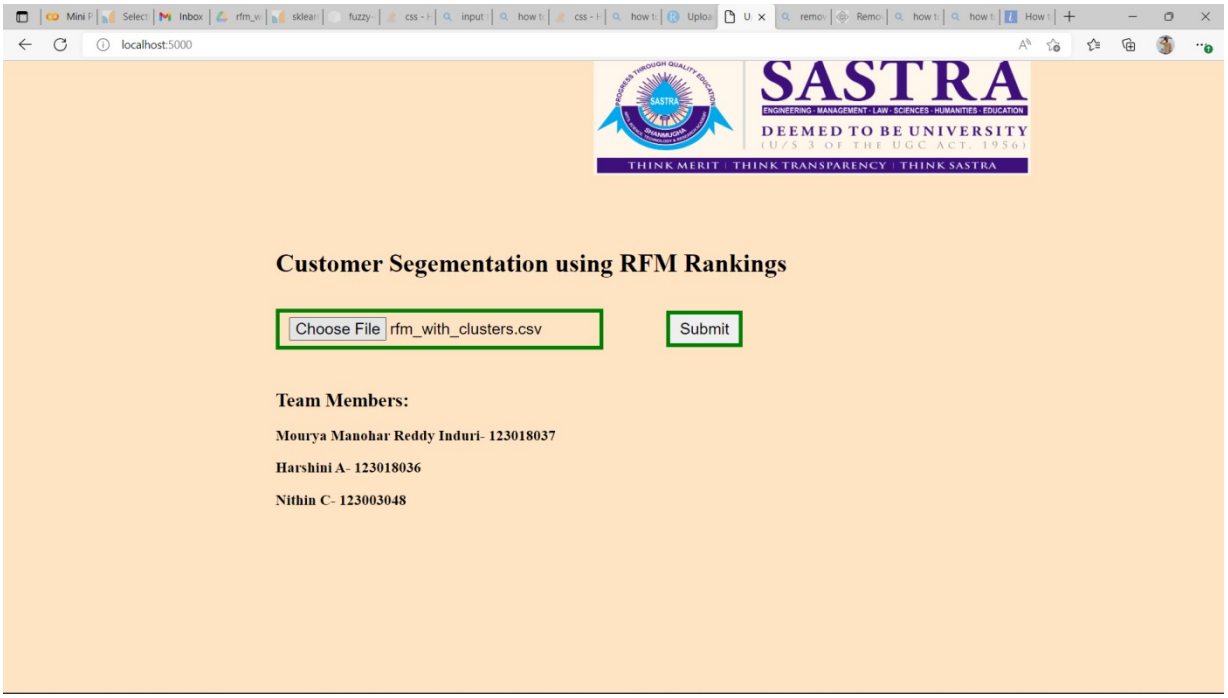
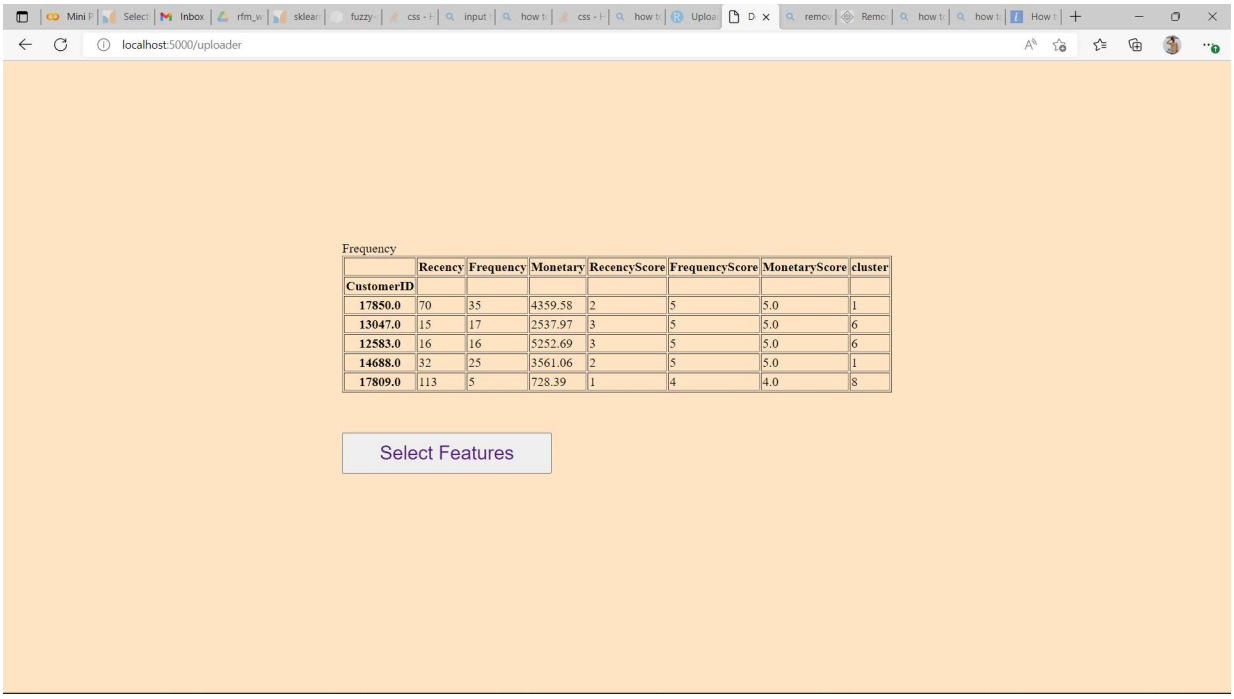


Fig. 4.3 Submitting the dataset csv file.



	Recency	Frequency	Monetary	RecencyScore	FrequencyScore	MonetaryScore	cluster
CustomerID							
17850.0	70	35	4359.58	2	5	5.0	1
13047.0	15	17	2537.97	3	5	5.0	6
12583.0	16	16	5252.69	3	5	5.0	6
14688.0	32	25	3561.06	2	5	5.0	1
17809.0	113	5	728.39	1	4	4.0	8

Fig. 4.4. Sample data of the uploaded csv file.

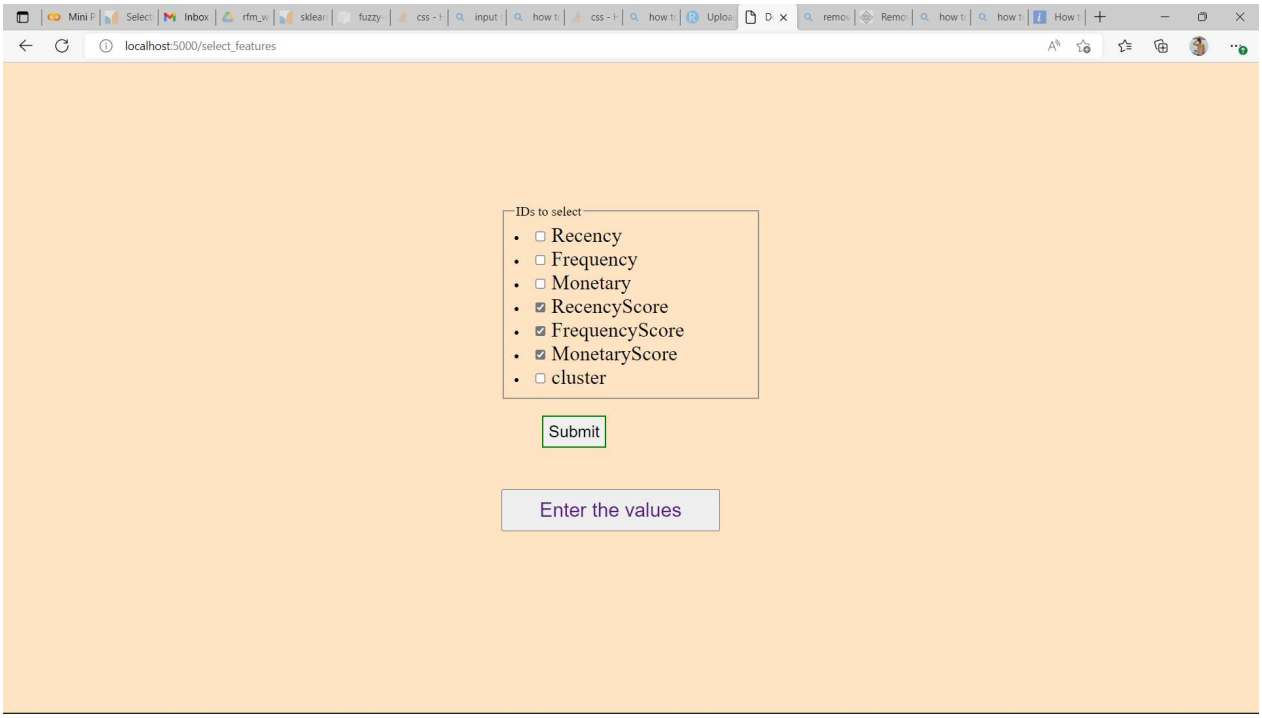


Fig. 4.5 Selection of attributes.

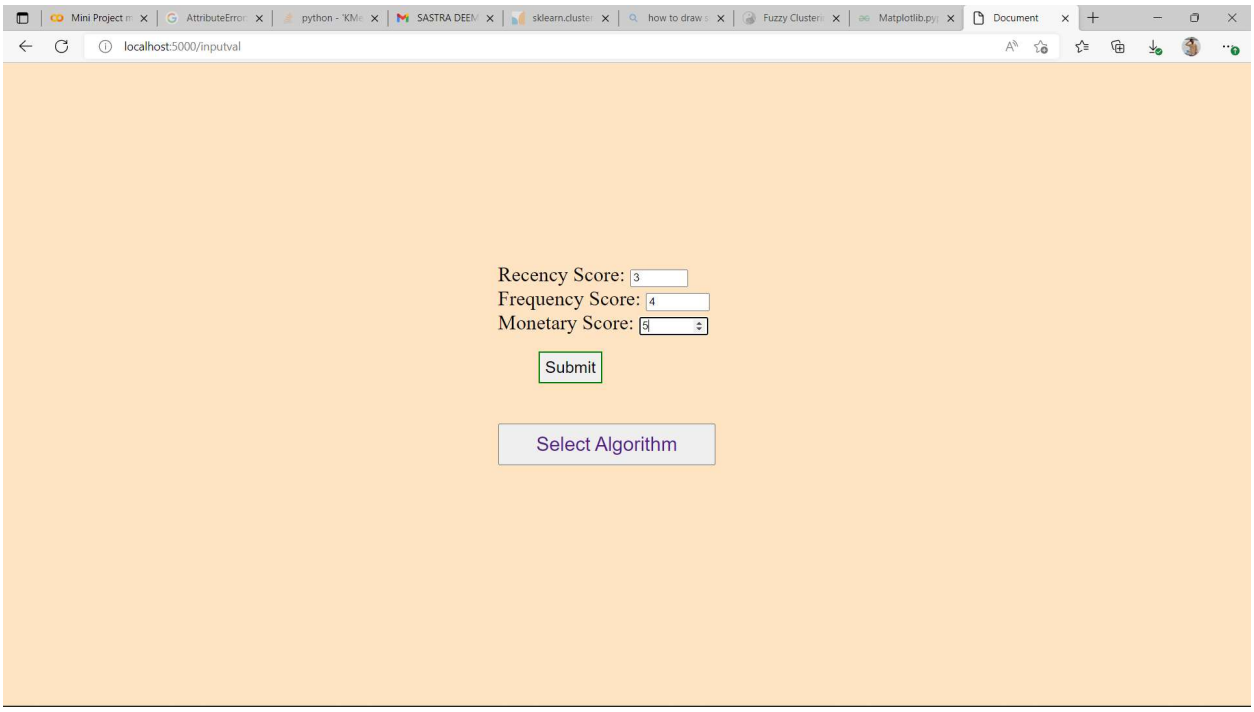


Fig. 4.6 Enter the scores(0 to5) to predict the cluster.

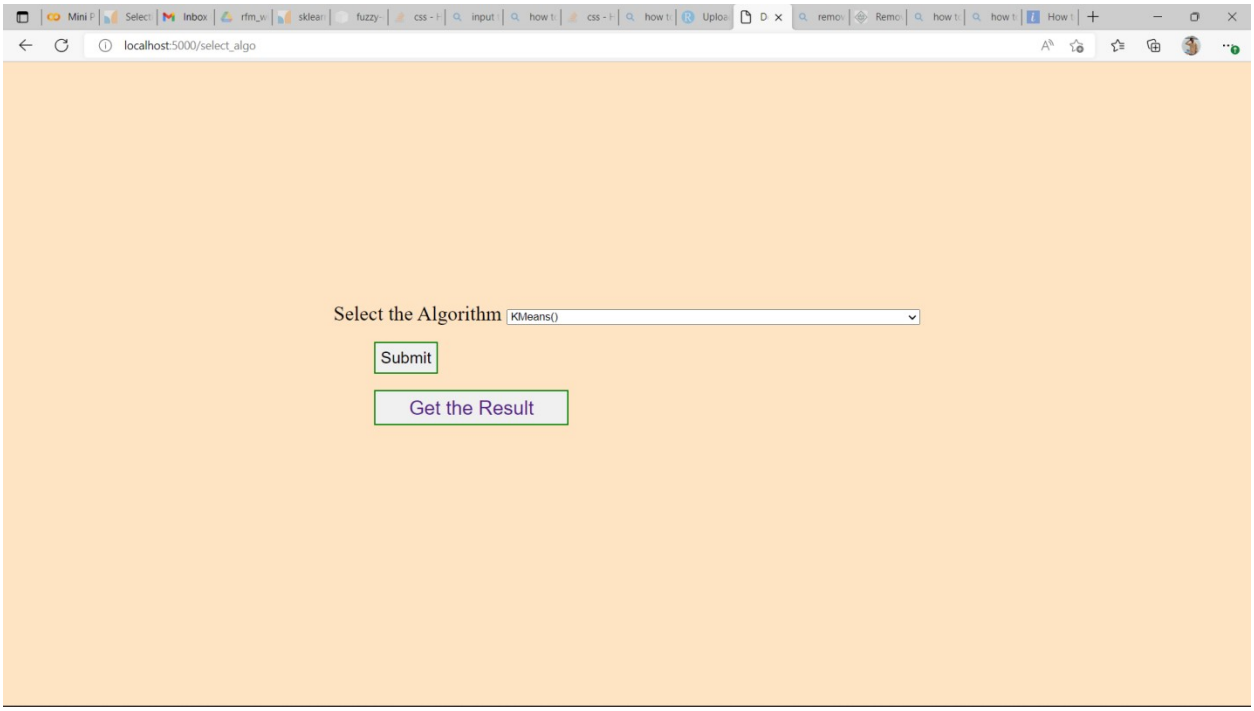


Fig. 4.7 Selection of the algorithm to be applied.

Fig. 4.8 Display of the cluster to which the given scores belong.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion:

Customer segmentation will strengthen customer relationships.In this study, segmentation is carried out using RFM analysis and then expanded to other methods, such as K-Means clustering and Fuzzy C, by making a small adjustment to the already-existing K-Means clustering. These methods' operation is examined.The execution times of each algorithm are studied, and it is shown that the suggested K-Means strategy uses less time and requires fewer iterations.The centroids are more significant and are determined at the outset based on the efficient medians of the data distribution, making the proposed technique more efficient.Based on their purchasing habits, the corporation can tailor its marketing strategy to each individual customer.

5.2 Future Work:

Future research will examine customer behavior in each segment, such as the products that consumers in each segment purchase regularly.This would make it easier to offer specific products better promotional incentives.

CHAPTER 6

REFERENCE

[1] He X., Li, C., *The research and application of customer segmentation on e-commerce websites*. In: 2016 6th International Conference on Digital Home (ICDH), Guangzhou, pp. 203–208.

[2] Haiying, M., Yu, G., *Customer Segmentation Study of College Students Based on the RFM*. In: 2010 International Conference on E-Business and E-Government, Guangzhou, pp. 3860-3863

[3] R. Srivastava, *Identification of customer clusters using RFM model: a case of diverse purchaser classification,* Int. J. Bus. Anal. Intell., 4 (2) (2016), pp. 45-50.

[4] Memon, K.H., Lee, D.H., 2017. *Generalized fuzzy c-means clustering algorithm with local information.* In: IET Image Processing, vol. 11, no. 1, pp. 1-12, 1.

[5] Shah, S., Singh, M., 2012. *Comparison of a Time Efficient Modified K-mean Algorithm with K-Mean and K-Medoid Algorithm.* In: 2012 International Conference on Communication Systems and Network Technologies, Rajkot, pp. 435–437.

CHAPTER 7

APPENDIX-BASE PAPER

Literature Survey

Source Content	Using various clustering algorithms to compare the efficiency of variuos algorithms while performing customer segmentation using RFM rankings.
Source Authors	A. Joy Christy , A. Umamakeswari , L. Priyatharsini , A. Neyaa.
Author's Work	Proposed a novel technique of choosing the initial centroids as the median values of the required attributes for the K-Means clustering algorithm.
Overall Comment	We would like to thank the author's work on proposing the novel technique. We have used their work to understand this project.