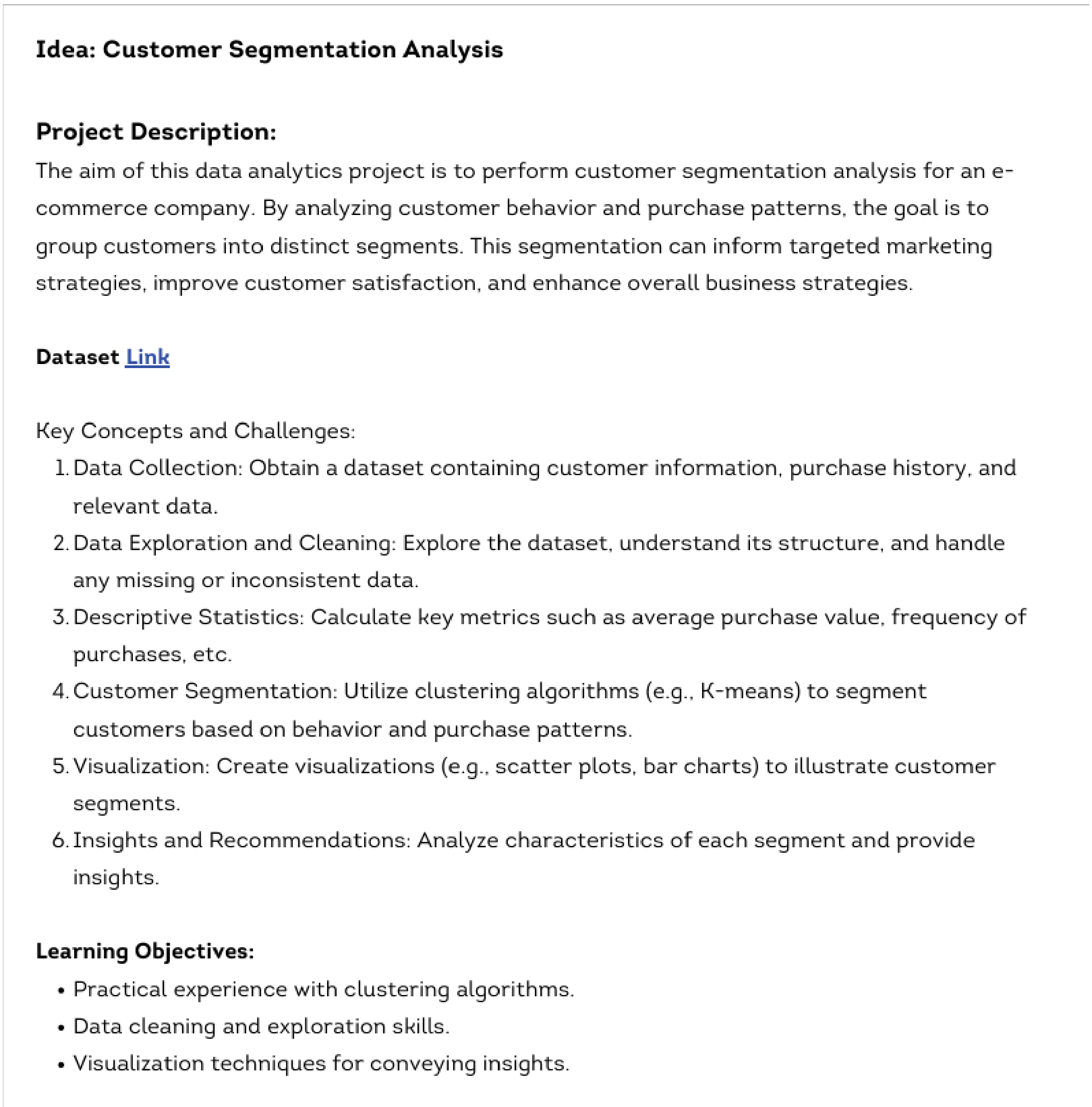

Idea: Customer Segmentation Analysis

Project Description:

The aim of this data analytics project is to perform customer segmentation analysis for an e-commerce company. By analyzing customer behavior and purchase patterns, the goal is to group customers into distinct segments. This segmentation can inform targeted marketing strategies, improve customer satisfaction, and enhance overall business strategies.

Dataset Link

Key Concepts and Challenges:

1. Data Collection: Obtain a dataset containing customer information, purchase history, and relevant data.
2. Data Exploration and Cleaning: Explore the dataset, understand its structure, and handle any missing or inconsistent data.
3. Descriptive Statistics: Calculate key metrics such as average purchase value, frequency of purchases, etc.
4. Customer Segmentation: Utilize clustering algorithms (e.g., K-means) to segment customers based on behavior and purchase patterns.
5. Visualization: Create visualizations (e.g., scatter plots, bar charts) to illustrate customer segments.
6. Insights and Recommendations: Analyze characteristics of each segment and provide insights.

Learning Objectives:

- Practical experience with clustering algorithms.
- Data cleaning and exploration skills.
- Visualization techniques for conveying insights.

Dataset: Link

Python code:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans
```

```
#Data collection
#loading the data from csv file to a Pandas DataFrame
customer_data = pd.read_csv('C:/Users/tanya/OneDrive/Desktop/Oasis Infobytes Customer Clustering/ifood_df.csv')
#info about our data
print(customer_data.head())
print(customer_data.shape)
print(customer_data.info())
print(customer_data.isnull().sum())

#Main features that we want to select: Income (0th column), MntTotal (36th column)
X = customer_data.iloc[:,[0,36]].values
print(X)

#choosing the number of clusters using WCSS - Within Cluster Sum of Squares
#finding wcss value for different no of clusters.
# we need less wcss valued clusters
wcss=[]

for i in range(1,11):
   kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state=42)
   kmeans.fit(X)
   wcss.append(kmeans.inertia_)

#plottimg elbow graph to choose kth value(optimum no of clusters)
sns.set()
plt.plot(range(1,11),wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

#optimum no of clusters = 4
#training the k-means Clustering model
kmeans = KMeans(
   n_clusters = 4, init = 'k-means++', random_state=42
)

#return a label for each data point
Y = kmeans.fit_predict(X)
print(Y)

#visualizing all clusters
#plotting all the clusters and their centroids
plt.figure(figsize=(8,8))
plt.scatter(X[Y==0,0],X[Y==0,1],s=50,c='green', label='Cluster 1')
plt.scatter(X[Y==1,0],X[Y==1,1],s=50,c='red', label='Cluster 2')
plt.scatter(X[Y==2,0],X[Y==2,1],s=50,c='yellow', label='Cluster 3')
plt.scatter(X[Y==3,0],X[Y==3,1],s=50,c='blue', label='Cluster 4')
#centroids
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=100,c='black',label='Centroids')

plt.title('Customer Groups')
plt.xlabel('Income')
plt.ylabel('Total amount spent')
plt.show()
```

Output:

```
PS C:\Users\tanya\OneDrive\Desktop\Oasis Infobytes Customer Clustering> python
clustering.py

   Income  Kidhome  Teenhome  Recency  MntWines  ... education_Master
education_PhD  MntTotal  MntRegularProds  AcceptedCmpOverall

0  58138.0       0        0       58       635  ...              0           0     1529           1441
0

1  46344.0       1        1       38        11  ...              0           0       21             15
0

2  71613.0       0        0       26       426  ...              0           0      734            692
0

3  26646.0       1        0       26        11  ...              0           0       48             43
0

4  58293.0       1        0       94       173  ...              0           1      407            392
0


[5 rows x 39 columns]

(2205, 39)

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2205 entries, 0 to 2204

Data columns (total 39 columns):

 #   Column              Non-Null Count  Dtype

---  ------              --------------  -----

 0   Income              2205 non-null   float64

 1   Kidhome             2205 non-null   int64

 2   Teenhome            2205 non-null   int64

 3   Recency             2205 non-null   int64

 4   MntWines            2205 non-null   int64

 5   MntFruits           2205 non-null   int64

 6   MntMeatProducts     2205 non-null   int64

 7   MntFishProducts     2205 non-null   int64
```

```
8   MntSweetProducts      2205 non-null   int64
9   MntGoldProds          2205 non-null   int64
10  NumDealsPurchases     2205 non-null   int64
11  NumWebPurchases       2205 non-null   int64
12  NumCatalogPurchases   2205 non-null   int64
13  NumStorePurchases     2205 non-null   int64
14  NumWebVisitsMonth     2205 non-null   int64
15  AcceptedCmp3          2205 non-null   int64
16  AcceptedCmp4          2205 non-null   int64
17  AcceptedCmp5          2205 non-null   int64
18  AcceptedCmp1          2205 non-null   int64
19  AcceptedCmp2          2205 non-null   int64
20  Complain              2205 non-null   int64
21  Z_CostContact         2205 non-null   int64
22  Z_Revenue             2205 non-null   int64
23  Response              2205 non-null   int64
24  Age                   2205 non-null   int64
25  Customer_Days         2205 non-null   int64
26  marital_Divorced      2205 non-null   int64
27  marital_Married       2205 non-null   int64
28  marital_Single        2205 non-null   int64
29  marital_Together      2205 non-null   int64
30  marital_Widow         2205 non-null   int64
31  education_2n Cycle    2205 non-null   int64
32  education_Basic       2205 non-null   int64
33  education_Graduation  2205 non-null   int64
34  education_Master      2205 non-null   int64
35  education_PhD         2205 non-null   int64
36  MntTotal              2205 non-null   int64
```

```
 37  MntRegularProds       2205 non-null   int64
 38  AcceptedCmpOverall    2205 non-null   int64
dtypes: float64(1), int64(38)
memory usage: 672.0 KB
None
Income                 0
Kidhome                0
Teenhome               0
Recency                0
MntWines               0
MntFruits              0
MntMeatProducts        0
MntFishProducts        0
MntSweetProducts       0
MntGoldProds           0
NumDealsPurchases      0
NumWebPurchases        0
NumCatalogPurchases    0
NumStorePurchases      0
NumWebVisitsMonth      0
AcceptedCmp3           0
AcceptedCmp4           0
AcceptedCmp5           0
AcceptedCmp1           0
AcceptedCmp2           0
Complain               0
Z_CostContact          0
Z_Revenue              0
Response               0
```

```
Age                    0
Customer_Days          0
marital_Divorced       0
marital_Married        0
marital_Single         0
marital_Together       0
marital_Widow          0
education_2n Cycle     0
education_Basic        0
education_Graduation   0
education_Master       0
education_PhD          0
MntTotal               0
MntRegularProds        0
AcceptedCmpOverall     0
dtype: int64
[[5.8138e+04 1.5290e+03]
 [4.6344e+04 2.1000e+01]
 [7.1613e+04 7.3400e+02]
 ...
 [5.6981e+04 1.2170e+03]
 [6.9245e+04 7.8200e+02]
 [5.2869e+04 1.5100e+02]]
```

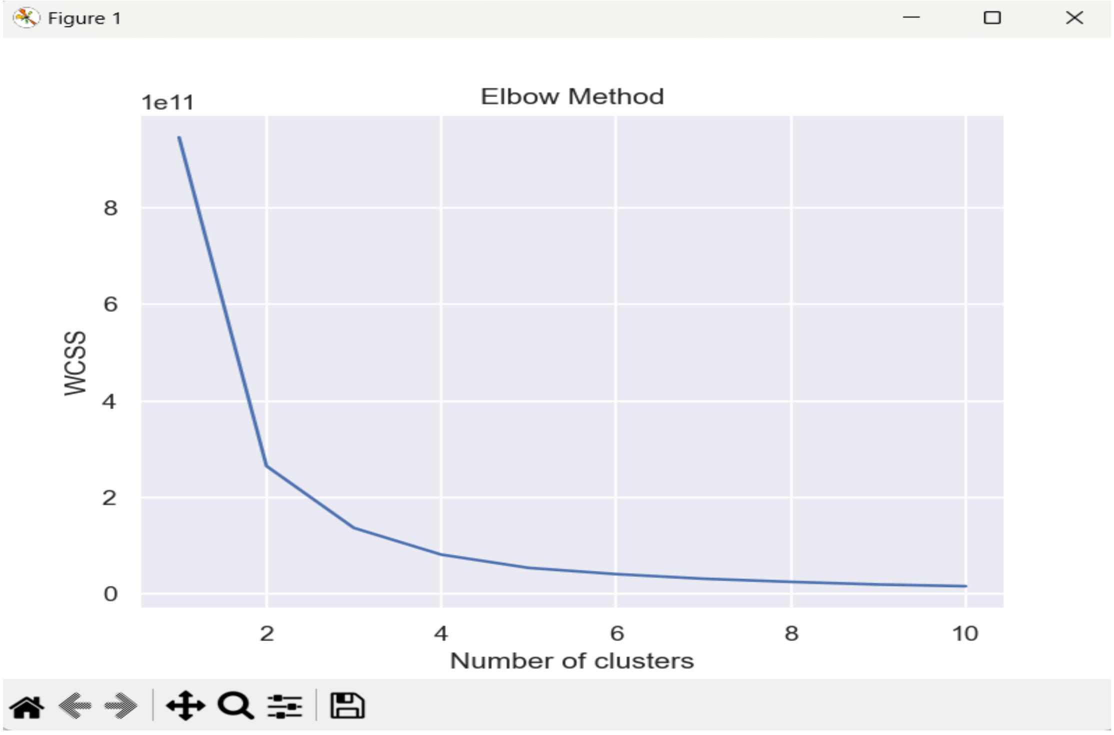
Figure 1
Elbow Method
1e11
WCSS
0
2
4
6
8
2
4
6
8
10
Number of clusters

```
[2 3 0 ... 2 2 2]
```

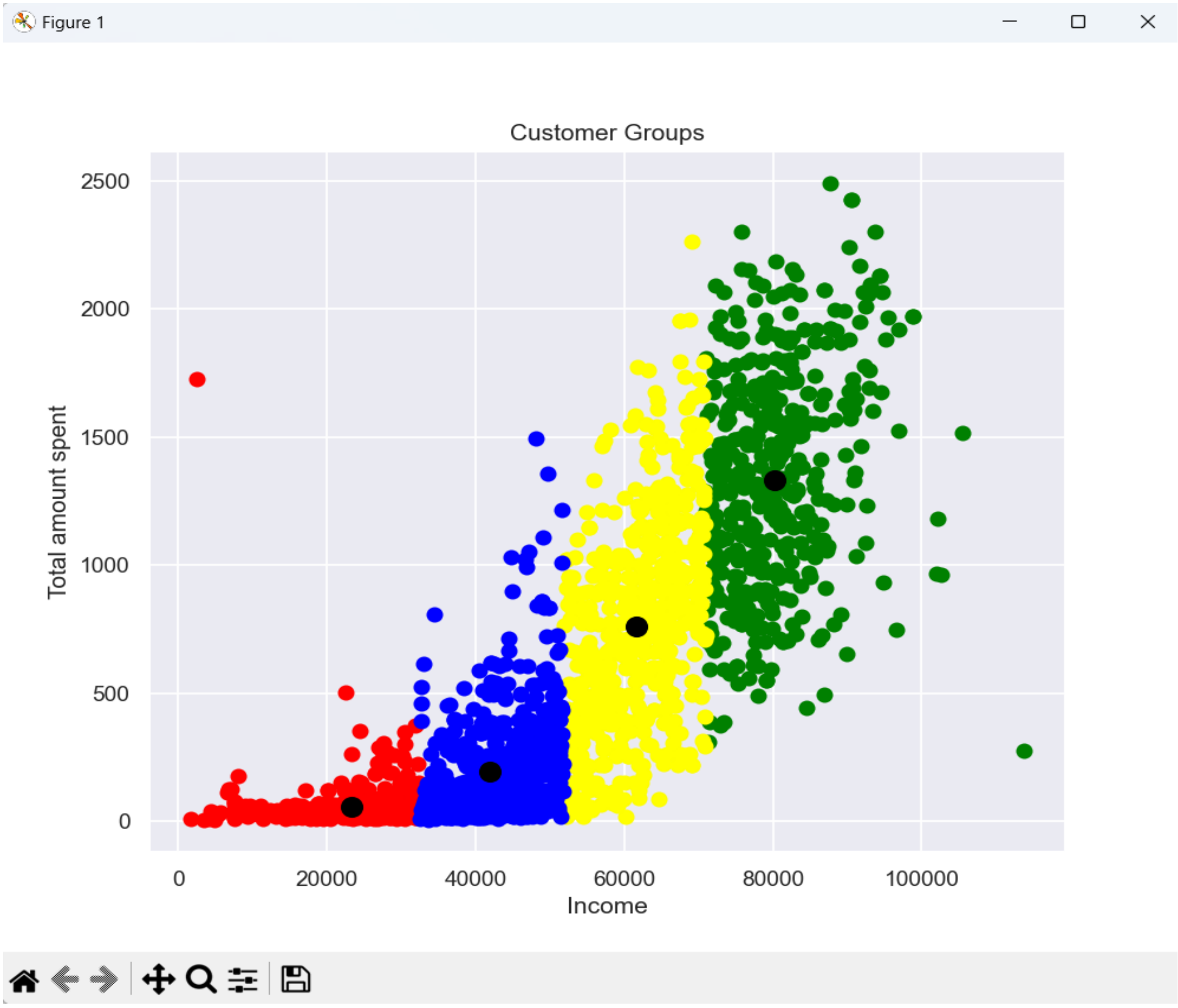
Figure 1
Customer Groups
Total amount spent
0
500
1000
1500
2000
2500
0
20000
40000
60000
80000
100000
Income