Customer Segmentation Using Data Science

Mall customer

* This is a mall’s dataset from Kaggle, and it has some basic data about the customers such as Customer ID, age, gender, annual income, and spending score.

Data Import:

Import the customer data into your chosen data analysis tool or programming environment. Common tools include Python with libraries like Pandas, R, or spreadsheet software like Excel.

#Import Packages

import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

import seaborn as sns

sns.set(context="notebook", palette="Spectral", style = 'darkgrid' ,font\_scale = 1.5, color\_codes=True)

Data Collection:

First, gather the customer data from your chosen sources. Ensure you have access to the data and that it is in a format that can be easily imported for analysis.

Gather customer data from various sources, such as your CRM (Customer Relationship Management) system, transaction records, customer surveys, and any other relevant sources. Ensure you have all the necessary permissions and adhere to data privacy regulations.

# Importing the dataset

dataset = pd.read\_csv('../Mall\_Customers.csv',index\_col='CustomerID')

Data Cleaning and Preprocessing:

This is a critical step to ensure the data is accurate and consistent. Here's what you can do:

* + Remove duplicate records.
  + Handle missing data (either by imputation or removal).

Standardize data formats (e.g., converting date formats to a consistent format).

* + Remove outliers that might skew the analysis.
  + Check for data integrity and correctness.

This step is crucial to ensure data quality and consistency. Here are some common tasks:

* + Remove duplicates: Check for and eliminate any duplicate records.
  + Handle missing data: Decide whether to impute missing values, remove rows with missing values, or use other methods.
  + Data type conversion: Ensure that data types are appropriate for analysis (e.g., numeric, categorical).
  + Outlier detection and handling: Identify and address outliers if they exist.
  + Standardize or normalize data if necessary.

dataset.head()

|  | **Genre** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| --- | --- | --- | --- | --- |
| **Customer ID** |  |  |  |  |
| **1** | Male | 19 | 15 | 39 |
| **2** | Male | 21 | 15 | 81 |
| **3** | Female | 20 | 16 | 6 |
| **4** | Female | 23 | 16 | 77 |
| **5** | Female | 31 | 17 | 40 |

Exploratory Data Analysis (EDA):

Perform EDA to get insights into your customer data. This might include:

* + Descriptive statistics to summarize data.
  + Data visualization to identify patterns, trends, and relationships.
  + Identifying customer segments based on preliminary findings.

Create new features or modify existing ones if it can help improve the quality of your segmentation. For example, you might calculate the recency, frequency, and monetary (RFM) values for each customer based on their transaction history.

dataset.info()

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

Int64Index: 200 entries, 1 to 200

Data columns (total 4 columns):

# Column Non-Null Count Dtype

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

0 Genre 200 non-null object

1 Age 200 non-null int64

2 Annual Income (k$) 200 non-null int64

3 Spending Score (1-100) 200 non-null int64

dtypes: int64(3), object(1)

memory usage: 7.8+ KB

Data Scaling and Transformation:

* Depending on the segmentation technique you plan to use, you may need to scale or transform your data. For example, if you're using K-Means clustering, it's essential to standardize the data to have zero mean and unit variance.

Assess the quality and validity of the segments. You can use metrics like Silhouette Score, Davies-Bouldin Index, or domain-specific evaluation criteria.

dataset.isnull().sum()

Genre 0 Age 0

Annual Income (k$) 0

Spending Score (1-100) 0

dtype: int64

# Using the elbow method to find the optimal number of clusters

from sklearn.cluster import KMeans

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state = 42)

kmeans.fit(X)

# inertia method returns wcss for that model

wcss.append(kmeans.inertia\_)

he optimal number of clusters

Decide on the segmentation criteria and prepare the data accordingly. For instance,

if you're segmenting based on purchase behavior, make sure your dataset includes relevant columns like purchase history.

Keep detailed records of your data collection, preprocessing, analysis, and decision-making process for future reference and collaboration with colleagues.

# Using the elbow method to find the optimal number of clusters

from sklearn.cluster import KMeans

wcss = []

for i in range(1, 11):

    kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state = 42)

    kmeans.fit(X)

    # inertia method returns wcss for that model

    wcss.append(kmeans.inertia\_)

plt.figure(figsize=(10,5))

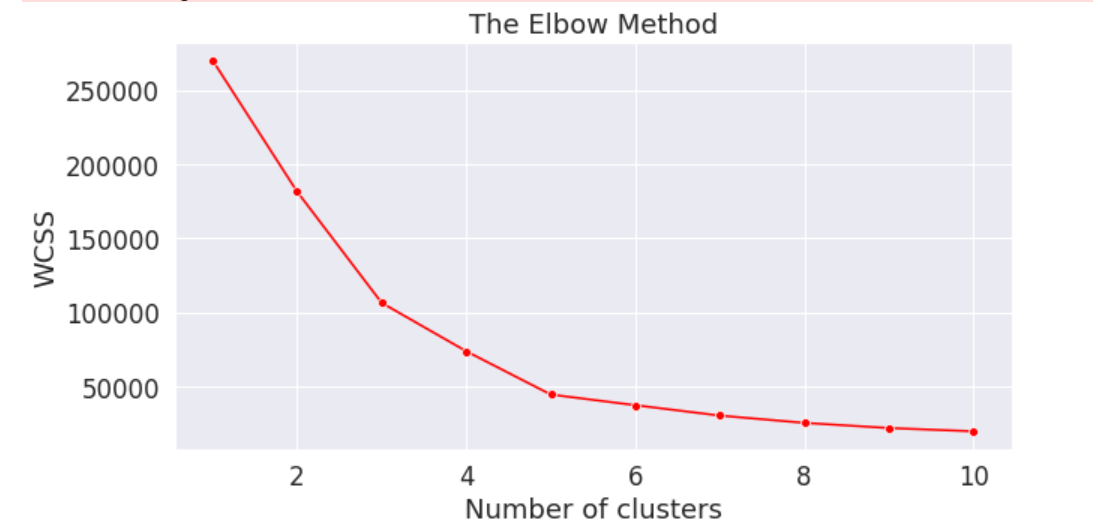
sns.lineplot(range(1, 11), wcss,marker='o',color='red')

plt.title('The Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()



Data Visualization:

* Use data visualization techniques to gain insights into the data and understand potential patterns and relationships. This can help in making informed decisions about the segmentation criteria.

# using only Spending\_Score and income variable for easy visualisation

X = dataset.iloc[:, [2, 3]].values

# Visualising the clusters

plt.figure(figsize=(15,7))

sns.scatterplot(X[y\_kmeans == 0, 0], X[y\_kmeans == 0, 1], color = 'yellow', label = 'Cluster 1',s=50)

sns.scatterplot(X[y\_kmeans == 1, 0], X[y\_kmeans == 1, 1], color = 'blue', label = 'Cluster 2',s=50)

sns.scatterplot(X[y\_kmeans == 2, 0], X[y\_kmeans == 2, 1], color = 'green', label = 'Cluster 3',s=50)

sns.scatterplot(X[y\_kmeans == 3, 0], X[y\_kmeans == 3, 1], color = 'grey', label = 'Cluster 4',s=50)

sns.scatterplot(X[y\_kmeans == 4, 0], X[y\_kmeans == 4, 1], color = 'orange', label = 'Cluster 5',s=50)

sns.scatterplot(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], color = 'red',

label = 'Centroids',s=300,marker=',')

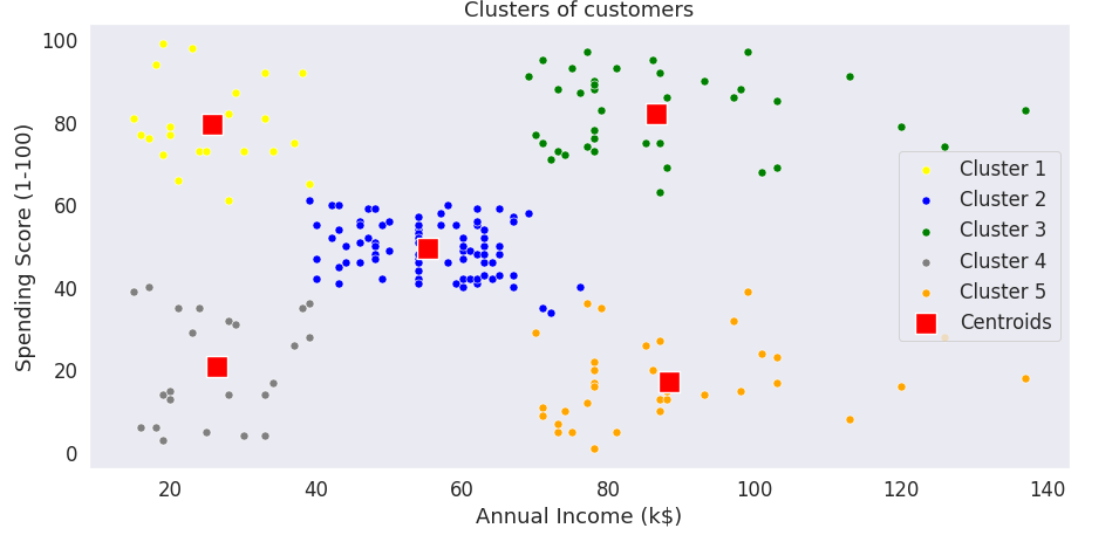
plt.grid(False)

plt.title('Clusters of customers')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.legend()plt.show()



Conclusion :

* There are clearly Five segments of Customers namely Miser, General, Target, Spendthrift, Careful based on their Annual Income and Spending Score which are reportedly the best factors/attributes to determine the segments of a customer in a Mall.
* Remember that the specific steps and techniques you use for preprocessing will depend on the nature of your dataset and the goals of your customer segmentation project. Each dataset may require custom handling to ensure the best results.