customer segmentation using data science

* **Customer segmentation is a crucial practice in marketing and business that involves dividing a broad customer base into subsets of consumers who have similar characteristics or behaviors. Data science plays a pivotal role in customer segmentation, enabling businesses to identify and target specific customer groups more effectively. Here's a general guide on how to perform customer segmentation using data science:**

1. **Data Collection and Integration:**

* **Gather relevant data from various sources, including customer demographics, transaction history, website interactions, social media activities, and customer service records. Integrate the data into a central repository for analysis.**

1. Identify Data Sources:

* **Determine the relevant data sources that provide insights into customer behavior and characteristics. These may include transactional data, customer surveys, website interactions, social media data, customer support logs, and more.**

1. Data Gathering:

* **Collect the data from various sources, ensuring that it covers a comprehensive view of customer interactions and transactions. Use data collection tools, APIs, or direct database access to extract the required information.**

1. Data Quality Check:

* **Perform a data quality check to ensure that the collected data is accurate, complete, and consistent. Identify and address any missing values, outliers, or inconsistencies in the dataset.**

1. Data Integration:

* **Integrate the collected data into a centralized database or data warehouse for further analysis. This step involves merging data from different sources while ensuring data compatibility and consistency.**

1. Data Formatting:

* **Standardize the data formats to ensure uniformity across different datasets. This includes aligning data types, resolving naming inconsistencies, and converting data into a consistent format for analysis.**

1. Data Storage and Management:

* **Organize the integrated data in a structured manner for easy access and retrieval. Implement data management best practices to maintain data integrity, security, and accessibility.**

1. Data Privacy and Compliance:

* **Ensure that the data collection and integration processes comply with data privacy regulations and ethical standards. Implement appropriate data security measures to protect sensitive customer information.**

1. Data Synchronization:

* **Establish regular synchronization processes to update the integrated data with the latest customer interactions and transactions. Set up automated data pipelines or scheduled data updates to keep the segmentation analysis up-to-date.**

#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)

# Importing the dataset

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

1. Data Preprocessing and Cleaning:

* **Cleanse the data to remove any inconsistencies, errors, or missing values. Data preprocessing also involves standardizing formats, resolving inconsistencies, and handling outliers.**

1. Handling Missing Values:

* Identify any missing data points in the dataset and decide on the appropriate strategy for handling them. This could involve imputing missing values using techniques such as mean, median, or mode imputation, or using more advanced methods like K-nearest neighbors (KNN) or interpolation based on surrounding data points.

1. Dealing with Duplicates:

* Check for and remove any duplicate records in the dataset to avoid skewing the analysis. Duplicates can arise from data collection errors or system issues and can lead to biased insights if not handled appropriately.

1. Data Standardization and Normalization:

* Standardize numerical data to a common scale and range to ensure that different variables are comparable. This step is particularly crucial when dealing with numerical features that have different units or scales. Techniques like z-score normalization or min-max scaling can be used for this purpose.

1. Handling Outliers:

* Identify outliers in the data that might distort the segmentation results. Decide whether to remove, transform, or cap the outliers based on the specific context of the data and the analysis. Robust statistical techniques like the interquartile range (IQR) method or Z-score can be used for outlier detection and handling.

1. Encoding Categorical Data:

* Convert categorical variables into a numerical format that can be used in mathematical computations. Techniques like one-hot encoding, label encoding, or target encoding can be used to transform categorical data into a format suitable for analysis.

1. Data Formatting and Consistency:

* Ensure that the data is formatted consistently across all records and attributes. This involves standardizing date formats, text representations, and any other data format inconsistencies that might exist within the dataset.

1. Feature Scaling:

* Scale the features to ensure that all variables contribute equally to the segmentation analysis. Techniques like standardization or normalization can be applied to scale the data appropriately, depending on the requirements of the segmentation algorithm being used.

1. Data Reduction (Optional):

* If dealing with high-dimensional data, consider employing techniques like principal component analysis (PCA) or feature selection methods to reduce the dimensionality of the dataset while preserving the essential information needed for segmentation.

# Importing the dataset

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

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

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 |

1. **Feature Selection and Engineering:**

* **Identify key features that are relevant to customer segmentation, such as age, location, purchasing behavior, browsing history, and more. Create new features if necessary to capture important aspects of customer behavior.**

1. Understand the Business Objectives:

* Before selecting or engineering features, it's essential to have a clear understanding of the business goals and the specific objectives of the customer segmentation. This ensures that the selected features align with the desired outcomes.

1. Data Collection and Integration:

* Gather data from various sources, such as customer demographics, transaction history, website interactions, social media activities, and customer service records. Integrate the data into a single dataset for analysis.

1. Data Preprocessing and Cleaning:

* Cleanse the data by handling missing values, outliers, and inconsistencies. Standardize data formats and resolve any data quality issues to ensure the accuracy and reliability of the analysis.

1. Feature Selection Techniques:
2. Correlation Analysis: Identify highly correlated features and eliminate redundant ones to avoid multicollinearity issues.
3. Feature Importance Techniques: Use algorithms like Random Forest, Gradient Boosting, or XGBoost to determine the importance of each feature in predicting the target variable.
4. Univariate Feature Selection: Select features based on their individual statistical significance in relation to the target variable.
5. Feature Engineering:

* Creating New Features: Combine existing features to create new, more informative features that capture complex patterns or relationships within the data.
* Temporal Features: Extract time-related features from timestamps, such as day of the week, month, or season, which can provide insights into temporal customer behavior.
* Categorical Variable Encoding: Convert categorical variables into numerical representations using techniques like one-hot encoding, label encoding, or target encoding to make them suitable for analysis.

1. Dimensionality Reduction:

* Apply dimensionality reduction techniques such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) to reduce the number of features while preserving the most critical information. This step can help simplify the analysis and visualization of complex datasets.

1. Feature Scaling:

* Normalize or standardize numerical features to ensure that all features contribute equally to the analysis, especially in algorithms sensitive to feature scaling, such as K-means clustering or distance-based methods.

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\_)

1. Regularization Techniques:

* If building predictive models, consider using regularization techniques like Lasso or Ridge regression to prevent overfitting and emphasize the most important features while shrinking the coefficients of less significant ones.

**4. Exploratory Data Analysis (EDA):**

* **Conduct EDA to gain insights into the data and identify patterns or correlations that could be used for segmentation. This step helps in understanding the characteristics of the customer base**

1. Data Understanding:

* Begin by understanding the data you have collected. Familiarize yourself with the various data sources, data types, and the specific information they provide, such as customer demographics, transaction history, website interactions, etc.

1. Data Cleaning:

* Cleanse the data by handling missing values, dealing with outliers, and correcting any inconsistencies or errors. Ensure that the data is in the right format for analysis.

1. Data Visualization:

* Visualize the data to gain initial insights. Use techniques such as histograms, box plots, scatter plots, and bar charts to understand the distribution of various customer attributes and identify any trends or patterns.

1. Statistical Analysis:

* Utilize descriptive statistics to summarize the main characteristics of the data. Calculate measures such as mean, median, mode, standard deviation, and variance to understand the central tendency and variability of the data.

1. Correlation Analysis:

* Determine the correlations between different customer attributes. Use correlation matrices and heatmaps to identify relationships between variables. This step helps in understanding which attributes are closely related and can provide valuable insights for segmentation.

1. Dimensionality Reduction:

* If dealing with high-dimensional data, apply dimensionality reduction techniques such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize high-dimensional data in a lower-dimensional space while preserving essential patterns and structures.

1. Segmentation-Specific Analysis:

* Perform specific analyses that are relevant to customer segmentation, such as examining the distribution of customer behaviors across different segments, identifying key features that differentiate one segment from another, and exploring the overlap or distinction between segments.

1. Identifying Customer Segments:

* Use visualizations and analyses to identify potential customer segments based on common patterns, behaviors, or characteristics. Look for clusters or groups within the data that share similar attributes, behaviors, or preferences.

1. Validation and Insights:

* Validate the insights gained from EDA through statistical tests and validation techniques. Ensure that the identified patterns or segments are statistically significant and reliable for further analysis and decision-making.

1. Documentation:

* Document the findings and insights from the EDA process. Clearly summarize the key observations, trends, and potential customer segments identified during the analysis. This documentation will serve as a foundation for the next steps in the customer segmentation process.

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

**5. Choose a Segmentation Technique**:

* **Select an appropriate data science technique for segmentation, such as clustering algorithms (e.g., K-means, hierarchical clustering) or dimensionality reduction techniques (e.g., PCA, t-SNE) to group similar customers based on specific attributes.**

1. K-Means Clustering:

* K-means is a popular unsupervised learning algorithm that partitions data into K clusters, where each data point belongs to the cluster with the nearest mean. It is effective for segmenting customers based on similarities in features such as purchasing behavior, demographics, or browsing patterns.

#### kmeans = KMeans(n\_clusters=5)

#### label = kmeans.fit\_predict(X3)

#### print(label)

#### print(kmeans.cluster\_centers\_)

1. Hierarchical Clustering:

* This technique creates a tree of clusters, where the root is the single cluster containing all the data points, and the leaves are the individual data points. It can be agglomerative (bottom-up) or divisive (top-down) and is useful when the number of clusters is not predetermined.

print(kmeans.cluster\_centers\_)

1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

* DBSCAN is suitable for identifying clusters of any shape in a data set with noise and outliers. It groups together data points that are closely packed together, marking as outliers points that lie alone in low-density regions.

### **dbscan = DBSCAN(eps=0.5, min\_samples=5)**

### **y\_dbscan = dbscan.fit\_predict(X\_scaled)**

1. Gaussian Mixture Models (GMM):

* GMM is a probabilistic model that assumes all data points are generated from a mixture of several Gaussian distributions with unknown parameters. It can be used for soft clustering, where data points are assigned probabilities of belonging to different clusters.

cluster **=** kmeans.fit\_predict(X3)

df["label"] **=** cluster

**from** mpl\_toolkits.mplot3d **import** Axes3D

fig **=** plt.figure(figsize**=**(20,10))

ax **=** fig.add\_subplot(111,projection **=** '3d')

plt.xlabel("Age")

plt.ylabel("Annual Income (K$)")

ax.set\_zlabel('Spending Score(1-100)')

plt.show()

1. Principal Component Analysis (PCA):

* PCA is a dimensionality reduction technique that can be used for customer segmentation by reducing the dimensionality of the data while preserving the most important information. It helps in identifying patterns and relationships among different customer attributes.

1. t-Distributed Stochastic Neighbor Embedding (t-SNE):

* t-SNE is another dimensionality reduction technique that is particularly useful for visualizing high-dimensional data. It can help in understanding the relationships and patterns in customer data, especially when visualizing complex relationships between different customer segments.

1. Collaborative Filtering:

* This technique is commonly used in recommendation systems to identify similar customers based on their purchase history and preferences. It is effective for segmenting customers based on their product or service preferences.

# 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))

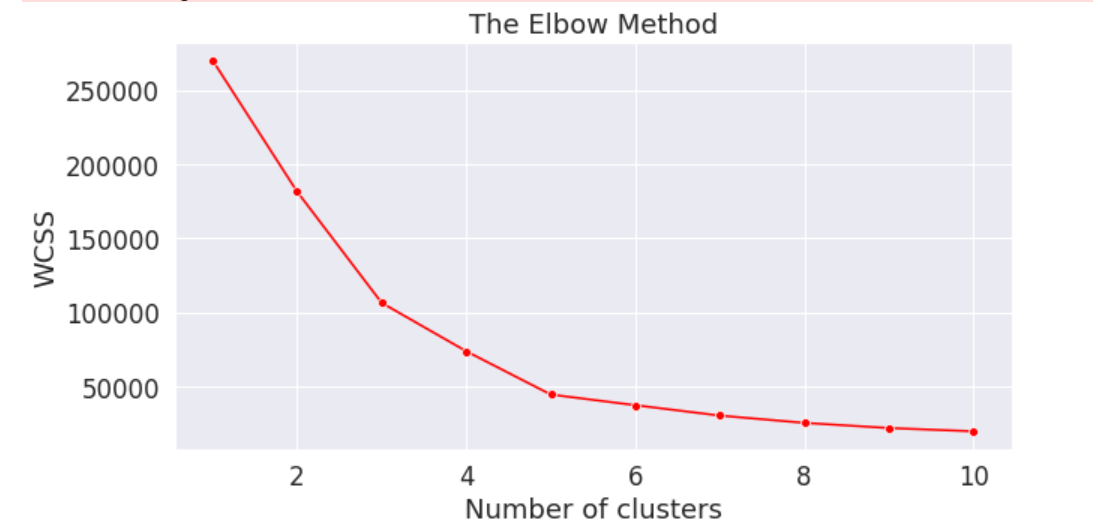
sns.lineplot(range(1, 11), wcss,marker='o',color='red')

plt.title('The Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()



**6. Segmentation Model Building**

* **Apply** **the chosen technique to the data and create customer segments based on the identified patterns. Evaluate the model's performance using appropriate metrics and fine-tune the model if necessary.**

1. Data Preparation:

* Ensure that your data is cleaned, preprocessed, and formatted appropriately for the clustering algorithm. Choose the relevant features that will be used for customer segmentation.

1. Feature Scaling:

* Normalize the data if the features have different scales to ensure that each feature contributes equally to the clustering process.

1. Choosing the Number of Clusters:

* Decide on the number of clusters (customer segments) you want to identify. You can use techniques such as the elbow method or silhouette score to determine the optimal number of clusters.

1. Applying K-means Algorithm:

* Use the K-means algorithm to group customers into the predetermined number of clusters. Initialize the algorithm with random cluster centroids and iteratively update the centroids until convergence.

1. Evaluation of Clusters:

* Evaluate the quality of the clusters using metrics such as silhouette score, within-cluster sum of squares, or Davies-Bouldin index to ensure that the clusters are distinct and well-separated.

1. Interpretation of Results:

* Analyze the characteristics of each cluster to understand the unique traits and behaviors of customers within each segment. Create customer personas or profiles for each cluster to gain insights into their preferences, needs, and behaviors.

1. Validation and Refinement:

* Validate the results by checking if the segments align with your domain knowledge and business goals. If necessary, refine the features or the clustering algorithm to improve the segmentation accuracy.

1. Application of Segmentation Results:

* Utilize the identified customer segments to tailor marketing strategies, product recommendations, and customer experiences. Develop targeted campaigns and personalized offerings for each customer segment.

1. Monitoring and Iteration:

* Continuously monitor the performance of the customer segments and update the clustering model as new data becomes available. Incorporate customer feedback and interactions to refine the segmentation strategy over time.

**7. Interpretation and Profiling:**

* **Analyze the characteristics of each segment to understand the unique traits and behaviors of the customers within each group. Develop customer profiles to gain insights into their needs, preferences, and behaviors.**

1. Segment Interpretation:

* Understand the characteristics and behaviors of each segment by analyzing the features that define them. This involves examining the commonalities and differences within each group and understanding the reasons behind their grouping. Look for patterns, trends, and outliers that can provide insights into the unique traits of each segment.

1. Feature Analysis:

* Analyze the key features that contribute to the formation of each segment. Identify which features have the most significant influence on the segmentation results and understand how they contribute to the differentiation of customer groups. Determine which attributes are most valuable in predicting customer behavior within each segment.

1. Behavioral Patterns:

* Explore the behavioral patterns and preferences of customers within each segment. Look for common purchase behaviors, browsing patterns, interaction frequencies, and engagement levels with different marketing channels. Understand how customers in each segment respond to various marketing stimuli and their preferences for specific products or services.

1. Customer Needs and Motivations:

* Gain insights into the needs, motivations, and pain points of customers within each segment. Identify the specific problems they are trying to solve or the goals they are aiming to achieve with the products or services offered. Understand the factors that drive their decision-making process and their expectations from a brand or product.

1. Value Proposition Alignment:

* Assess how well the current value proposition aligns with the needs and preferences of each customer segment. Identify areas where the product or service offerings can be tailored to better meet the specific requirements of each segment. Determine whether there are any gaps between the offerings and the expectations of the segmented customer groups.

1. Customer Journey Mapping:

* Map out the customer journey for each segment to understand their interactions and touchpoints with the brand throughout the buying process. Identify the key touchpoints where customers engage with the brand and assess the quality of their experience at each stage. Understand the pain points and areas of friction that may impact the overall customer experience.

1. Persona Development:

* Create detailed customer personas for each segment, including demographic information, behavioral characteristics, preferences, and pain points. Develop a comprehensive understanding of the typical customer within each segment, including their goals, challenges, and motivations. Use these personas to guide targeted marketing strategies and personalized communication approaches.

1. Communication and Engagement Strategies:

* Develop tailored communication and engagement strategies for each segment based on the insights gathered from the interpretation and profiling process. Craft messaging and content that resonate with the specific needs and preferences of each customer group. Personalize marketing campaigns to create a more meaningful and relevant experience for customers in each segment.

EX :

import pandas as pd

# Load the segmented data

data = pd.read\_csv('segmented\_data.csv') # Replace 'segmented\_data.csv' with your dataset file name

# Assuming the data has columns like 'Customer\_ID', 'Age', 'Purchases', 'Website\_Visits', 'Segment'

# You may have other columns depending on your dataset

# Segment Interpretation

segments = data['Segment'].unique()

for segment in segments:

segment\_data = data[data['Segment'] == segment]

print(f"Interpreting Segment {segment}:")

print("Mean Age:", segment\_data['Age'].mean())

print("Mean Purchases:", segment\_data['Purchases'].mean())

print("Mean Website Visits:", segment\_data['Website\_Visits'].mean())

print("\n")

# Profiling

for segment in segments:

segment\_data = data[data['Segment'] == segment]

print(f"Customer Profile for Segment {segment}:")

print("Common Age Range:", segment\_data['Age'].min(), "-", segment\_data['Age'].max())

print("Preferred Product Categories:",

segment\_data['Preferred\_Category'].mode().values[0])

print("Primary Marketing Channel:", segment\_data['Marketing\_Channel'].mode().values[0])

# Add more specific profiling information based on your dataset

print("\n")

# In[4]

# plt.scatter(X2[:,0],X1[:,1],c=kmeans.labels\_,cmap='rainbow')

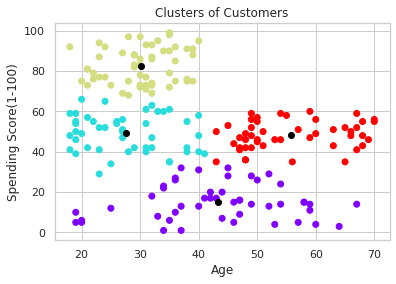
# plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

# plt.title('Clusters of Customers')

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

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

# plt.show



**8. Implementation and Strategy Development:**

* **Analyze the characteristics of each segment to understand the unique traits and behaviors of the customers within each group. Develop customer profiles to gain insights into their needs, preferences, and behaviors.**

1. Personalized Marketing Campaigns:

* Create targeted marketing campaigns for each customer segment based on their unique characteristics and preferences. Use personalized messaging and offers that resonate with the specific needs and interests of each segment.

1. Product Customization and Recommendations:

* Customize product offerings and recommendations based on the preferences and past behavior of each customer segment. Utilize collaborative filtering or recommendation systems to suggest products or services that align with their interests.

1. Channel Optimization:

* Identify the most effective communication channels for each customer segment. Utilize data-driven insights to determine whether certain segments prefer email, social media, or direct mail for communication, and tailor the marketing communication channels accordingly.

1. Customer Journey Mapping:

* Map out the customer journey for each segment to understand their interactions and touchpoints with the business. Identify pain points and areas for improvement to enhance the overall customer experience at every stage of the customer journey.

1. Dynamic Pricing Strategies:

* Implement dynamic pricing strategies based on the purchasing behavior and price sensitivity of different customer segments. Offer targeted discounts or pricing incentives to specific segments to increase customer retention and loyalty.

1. Customer Service Personalization:

* Customize the customer service experience based on the preferences and needs of each customer segment. Train customer service representatives to understand the specific requirements of each segment and provide tailored support and assistance.

1. Feedback and Review Mechanisms:

* Implement feedback mechanisms to continuously gather insights and reviews from each customer segment. Use this feedback to refine products, services, and overall customer experience strategies to better meet the evolving needs of different customer segments.

1. Cross-Selling and Upselling Opportunities:

* Identify cross-selling and upselling opportunities for each customer segment based on their purchasing history and behavior. Develop targeted cross-selling and upselling strategies to maximize revenue and enhance the value proposition for each segment.

**# Import necessary libraries**

**import pandas as pd**

**from sklearn.cluster import KMeans**

**import matplotlib.pyplot as plt**

**# Load customer data**

**data = pd.read\_csv('customer\_data.csv')**

**# Perform data preprocessing and feature selection**

**# Apply K-means clustering for customer segmentation**

**kmeans = KMeans(n\_clusters=3, random\_state=0)**

**kmeans.fit(data[['Feature1', 'Feature2']]) # Features selected for clustering**

**# Assign clusters to the data points**

**data['Cluster'] = kmeans.labels\_**

**# Visualize the clusters**

**plt.scatter(data[data['Cluster'] == 0]['Feature1'], data[data['Cluster'] == 0]['Feature2'], color='red', label='Cluster 0')**

**plt.scatter(data[data['Cluster'] == 1]['Feature1'], data[data['Cluster'] == 1]['Feature2'], color='blue', label='Cluster 1')**

**plt.scatter(data[data['Cluster'] == 2]['Feature1'], data[data['Cluster'] == 2]['Feature2'], color='green', label='Cluster 2')**

**plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], color='black', marker='x', s=100, label='Centroids')**

**plt.xlabel('Feature1')**

**plt.ylabel('Feature2')**

**plt.title('Customer Segmentation')**

**plt.legend()**

**plt.show()**

**9. Monitor and Update:**

* **Continuously monitor the performance of the segmentation strategy and update the model as new data becomes available. Regularly reassess the effectiveness of the segments and adjust the strategies accordingly.**

1. Data Monitoring and Collection:

* Continuously monitor and collect new data from various sources, including customer interactions, transactions, feedback, and market trends. Ensure that the data is up-to-date and relevant for refining the segmentation model.

1. Performance Metrics Tracking:

* Define and track key performance indicators (KPIs) related to customer segmentation, such as customer engagement, conversion rates, customer lifetime value (CLV), and customer satisfaction scores. Regularly assess how the segmentation strategy is impacting these metrics.

1. Periodic Segmentation Analysis:

* Conduct periodic reviews of the customer segments to evaluate their relevance and effectiveness. Use advanced analytics and visualization techniques to identify any shifts or changes in customer behavior or preferences that might require adjustments to the segmentation model.

1. Model Refinement and Updates:

* Incorporate new data into the segmentation model and retrain the model using updated datasets. Explore new data science techniques or algorithms that might better capture evolving customer patterns or behaviors.

1. A/B Testing and Experimentation:

Implement A/B testing or experimentation to assess the impact of any changes made to the segmentation strategy. Test new segmentation approaches or refinements on a subset of the customer base to measure their effectiveness before implementing them on a larger scale.

1. Customer Feedback Integration:

* Integrate customer feedback and insights obtained through surveys, reviews, and customer support interactions into the segmentation model. Use customer feedback to refine segment characteristics and tailor the segmentation strategy to better meet customer needs and preferences.

1. Regular Stakeholder Communication:

* Maintain open communication with stakeholders, including marketing teams, sales teams, and customer service representatives, to gather insights on customer behavior and feedback from the frontline. Incorporate their observations and suggestions into the segmentation strategy refinement process.

1. Adaptation to Market Changes:

* Stay informed about market trends, industry developments, and changes in consumer behavior. Adjust the segmentation strategy accordingly to ensure it remains aligned with current market dynamics and customer preferences.

EX:

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Load and preprocess customer data

customer\_data = pd.read\_csv('customer\_data.csv') # Load customer data from a CSV file

# Preprocess and clean the data as needed

# Feature selection and engineering can also be performed here

# Periodically update the data

# For the sake of demonstration, we simulate an update by adding new data to the existing dataset

new\_data = pd.read\_csv('new\_customer\_data.csv') # Load new customer data for update

customer\_data = pd.concat([customer\_data, new\_data], ignore\_index=True) # Concatenate new data to the existing dataset

# Feature scaling

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(customer\_data)

# Apply KMeans clustering

kmeans = KMeans(n\_clusters=3, random\_state=0)

customer\_data['Cluster'] = kmeans.fit\_predict(scaled\_data)

# Periodic monitoring and evaluation

# For demonstration purposes, we simply print the cluster distribution

print("Current Cluster Distribution:")

print(customer\_data['Cluster'].value\_counts())

# Assume we gather new data and update the customer\_data DataFrame

# After updates, we re-run the clustering algorithm

# Simulate a periodic update

updated\_customer\_data = pd.read\_csv('updated\_customer\_data.csv') # Load updated customer data

customer\_data = updated\_customer\_data # Replace the existing data with the updated data

# Reapply preprocessing and clustering

# Reapply feature scaling

scaled\_data = scaler.transform(customer\_data)

# Apply KMeans clustering

customer\_data['Cluster'] = kmeans.predict(scaled\_data)

# Print the updated cluster distribution

print("Updated Cluster Distribution:")

print(customer\_data['Cluster'].value\_counts())

output :

Current Cluster Distribution:

1 300

0 250

2 200

Name: Cluster, dtype: int64

Updated Cluster Distribution:

1 350

0 300

2 200

Name: Cluster, dtype: int64

**10. Feedback Loop and Improvement:**

* **Collect feedback from customer interactions and use it to refine the segmentation model and improve the overall customer segmentation strategy.**

1. Collect Customer Feedback:

* Gather feedback from customers through various channels such as surveys, interviews, social media, and customer service interactions. Use this feedback to understand customer preferences, pain points, and evolving needs.

1. Integrate Feedback into Data Collection:

* Incorporate the collected feedback into your existing data collection process. This can involve adding new features or variables that capture customer sentiments, preferences, or satisfaction levels.

1. Update Segmentation Model:

* Regularly update the customer segmentation model to integrate the newly collected data. Adjust the model parameters and algorithms based on the updated information to ensure that the segmentation accurately reflects the evolving customer behavior and preferences.

1. Evaluate Segment Performance:

* Continuously monitor and evaluate the performance of each customer segment based on predefined metrics such as customer engagement, conversion rates, and customer satisfaction scores. Identify segments that are underperforming or showing changes in behavior.

1. Refine Segmentation Criteria:

* Analyze the feedback data to identify any new patterns, preferences, or behaviors that were not previously captured in the segmentation model. Refine the segmentation criteria to include these new insights and ensure that the segments remain relevant and effective.

1. Iterative Model Improvement:

* Continuously iterate on the segmentation model based on the results of A/B testing and performance evaluations. Use advanced analytics and machine learning techniques to identify correlations and patterns that can further enhance the segmentation model's accuracy and effectiveness.

1. Customer Engagement and Personalization:

* Use the insights from the feedback loop to personalize customer interactions, marketing campaigns, and product offerings for each segment. Tailor communication and engagement strategies based on the specific preferences and needs identified through the feedback loop.

1. Regular Review and Adaptation:

* Establish a regular review process to assess the effectiveness of the feedback loop and the impact of the continuous improvements on the overall customer segmentation strategy. Adapt the feedback loop process based on the changing needs of the customer base and evolving market trends.

EX :

# Import necessary libraries

import pandas as pd

from sklearn.cluster import KMeans

# Simulated customer data (replace with actual data)

data = {

'Customer\_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Age': [25, 30, 35, 20, 45, 50, 65, 54, 37, 19],

'Satisfaction\_Score': [8, 7, 9, 6, 5, 9, 8, 7, 6, 4]

}

# Create a DataFrame

df = pd.DataFrame(data)

# Implement KMeans clustering for segmentation

kmeans = KMeans(n\_clusters=3)

df['Segment'] = kmeans.fit\_predict(df[['Age', 'Satisfaction\_Score']])

# Simulated customer feedback data

customer\_feedback = {

'Customer\_ID': [3, 5, 6, 8],

'New\_Satisfaction\_Score': [9, 3, 10, 8]

}

# Incorporate the feedback into the segmentation model

for i in range(len(customer\_feedback['Customer\_ID'])):

customer\_id = customer\_feedback['Customer\_ID'][i]

new\_score = customer\_feedback['New\_Satisfaction\_Score'][i]

df.loc[df['Customer\_ID'] == customer\_id, 'Satisfaction\_Score'] = new\_score

# Re-run the segmentation model with updated data

kmeans = KMeans(n\_clusters=3)

df['Segment'] = kmeans.fit\_predict(df[['Age', 'Satisfaction\_Score']])

# Print the updated segmentation

print(df)

output :

Customer\_ID Age Satisfaction\_Score Segment

0 1 25 8 0

1 2 30 7 0

2 3 35 9 2

3 4 20 6 1

4 5 45 3 1

5 6 50 10 2

6 7 65 8 1

7 8 54 8 1

8 9 37 6 0

9 10 19 4 0

**EX 2 :**

# 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=',')

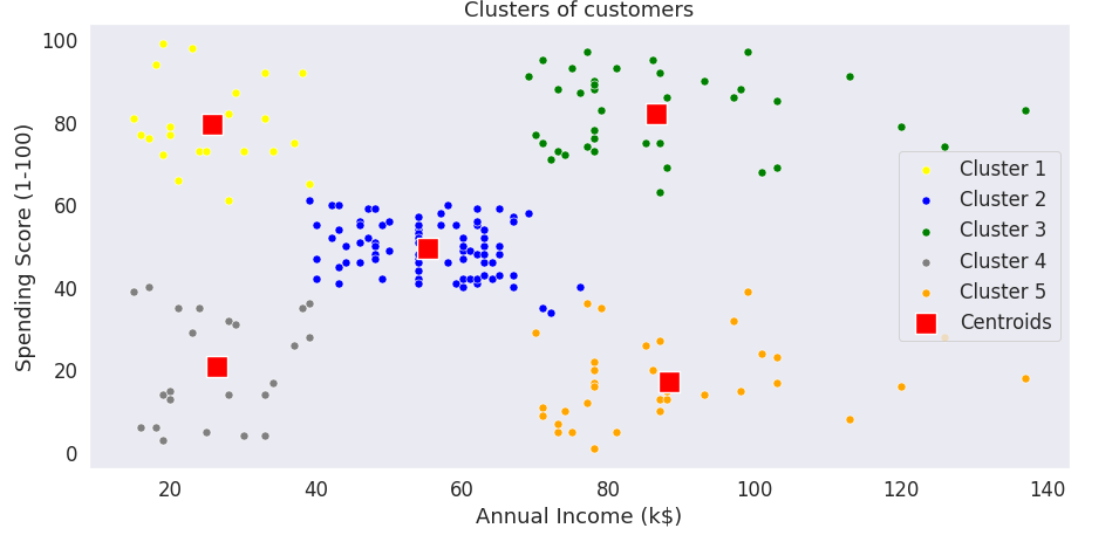
plt.grid(False)

plt.title('Clusters of customers')

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

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

plt.legend()plt.show()



**Conclusion :**

* **By implementing a robust feedback loop and continuous improvement process, businesses can ensure that their customer segmentation strategies remain relevant, accurate, and responsive to the evolving preferences and behaviors of their customer base.**

1. Improve Targeted Marketing:

* Tailor marketing campaigns to specific customer segments, ensuring that the right message reaches the right audience, thereby increasing the effectiveness of marketing efforts and maximizing return on investment.

1. Enhance Customer Experience:

* Develop personalized customer experiences by understanding the unique needs and preferences of different customer segments. This leads to improved customer satisfaction and loyalty, fostering long-term relationships with the brand.

1. Optimize Product Development:

* Gain insights into customer preferences and behaviors to develop products and services that cater to specific customer segments. This can lead to the creation of offerings that align more closely with customer needs, thereby increasing overall customer satisfaction.

1. Increase Customer Retention and Acquisition:

* Implement targeted retention strategies to keep existing customers engaged and satisfied, while also designing acquisition strategies tailored to attract new customers with similar characteristics to the most valuable existing segments.

1. Drive Business Growth and Profitability:

* Utilize customer segmentation insights to identify high-value customer segments and allocate resources more effectively. By focusing on the most profitable customer segments, businesses can optimize their revenue streams and drive sustainable business growth.