[Title:](https://en.wikipedia.org/wiki/Title" \t "_self)

Customer Segmentation using Data Science – Guidelines

## AIM:

## In this part you will continue building your project.

## Continue building the customer segmentation model by:

# Feature engineering

# Applying clustering algorithms

# Visualization

# Interpretation.

Feature engineering:

### **1 . Recency, Frequency, Monetary (RFM) Features:**

### **Calculate the recency, frequency, and monetary value for each customer based on their transaction history. These features can help identify loyal customers, recent buyers, and big spenders.**

### **2. Demographic Features:**

### **Include customer demographic information, such as age, gender, location, and income, as features for segmentation.**

**3.Behavioral Features:**

* **Create features based on customer behavior, such as the number of products viewed, time spent on the website, or the number of past purchases.**

**4. Categorical Encoding:**

* **Convert categorical variables (e.g., product categories, customer segments) into numerical representations using techniques like one-hot encoding or label encoding.**

**5.Time-Based Features:**

* **Extract features related to time, such as day of the week, month, or year of a purchase. This can help capture seasonality and trends.**

**6 . Customer Lifetime Value (CLV):**

* **Calculate the CLV of each customer, which is a measure of the predicted future value a customer will bring to the business. It can be based on historical purchase data and can be used as a feature.**

**7 . Geospatial Features:**

* **If relevant, include features related to the geographic location of customers. This could be based on ZIP codes, cities, or even longitude and latitude coordinates.**

**8.Geospatial Features:**

* **If relevant, include features related to the geographic location of customers. This could be based on ZIP codes, cities, or even longitude and latitude coordinates.**

**9.Social Media and Online Interaction**

* **If available, incorporate features related to a customer's social media activity, online reviews, or interactions with the company on social platforms.**
* **10.Text Features:**
* **Analyze text data, such as customer reviews or feedback, and extract sentiment or topic-related features that can provide insights into customer preferences.**

**11.Feature Scaling:**

* **Standardize or normalize features to ensure that they are on the same scale. This is important for algorithms like K-Means.**

**12.Dimensionality Reduction:**

* **If you have a high-dimensional dataset, consider using dimensionality reduction techniques like Principal Component Analysis (PCA) to reduce the number of features while retaining the most important information.**

**13.Interaction Features:**

* **Create interaction features by combining two or more existing features. For example, you can create a feature that represents the product category and the frequency of purchases in that category.**

**14.Domain-Specific Features:**

* **Depending on your specific business and industry, you may have domain-specific features that are highly relevant to customer segmentation. These can be invaluable in creating meaningful clusters.**

Applying clustering algorithms

1. **Choose the Right Clustering Algorithm:**

* **Select a clustering algorithm that is suitable for your dataset and problem. Common clustering algorithms include:**
* **K-Means: Divides data into K clusters, where K is a predefined number.**
* **Hierarchical Clustering: Forms a hierarchy of clusters, often represented as a dendrogram.**
* **DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Clusters data points based on their density.**
* **Gaussian Mixture Model (GMM): Assumes that data points are generated from a mixture of several Gaussian distributions.**
* **Agglomerative Clustering: A type of hierarchical clustering that starts with individual data points and progressively merges them into clusters.**

1. **Preprocess the Data:**

* **Ensure that your data is properly preprocessed. This includes handling missing values, scaling or normalizing features, and encoding categorical variables if needed.**

1. **Select the Number of Clusters (K):**

* **For K-Means and GMM, you'll need to choose the number of clusters (K) in advance. Various techniques like the elbow method, silhouette score, or cross-validation can help you determine an appropriate K.**

1. **Apply the Chosen Algorithm:**

* **Fit the selected clustering algorithm to your preprocessed data using the chosen number of clusters (K, if applicable).**

1. **Evaluate the Clustering:**

* **Assess the quality of the clustering using appropriate metrics. Common metrics for clustering include:**
* **Silhouette Score: Measures the quality of clustering by considering both the distance between clusters and the distance within clusters.**
* **Davies-Bouldin Index: Measures the average similarity ratio of each cluster with the cluster that is most similar to it.**
* **Inertia (K-Means only): Measures the within-cluster sum of squares, which should be minimized.**
* **Calinski-Harabasz Index: Measures the ratio of between-cluster variance to within**-**cluster variance.**

1. **Visualize the Clusters:**

* **Create visualizations to better understand the clusters and their relationships. Common visualization techniques include:**
* **Scatter Plots: Plot data points with different colors or symbols for each cluster.**
* **t-SNE (t-distributed Stochastic Neighbor Embedding): A dimensionality reduction technique that can help visualize high-dimensional data in two or three dimensions.**
* **Dendrogram (for hierarchical clustering): Visualizes the hierarchy of clusters.**

1. **Interpret the Results:**
   * **Assign meaningful labels to clusters based on their characteristics.**
   * **Profile each cluster to understand the common attributes of data points within it.**
   * **Interpret the business implications of the clusters. How can they be used to make decisions or drive actions?**
2. **Iterate and Refine:**
   * **Clustering can be an iterative process. You may need to adjust your feature engineering or try different clustering algorithms to improve result**.

Visualization :

1. **Scatter Plots: Scatter plots are an essential tool for visualizing clusters. They allow you to plot data points in a two-dimensional space, with different colors or markers representing different clusters. This provides a clear view of how data points are distributed and separated into clusters.**
2. **t-SNE (t-distributed Stochastic Neighbor Embedding): t-SNE is a dimensionality reduction technique that is often used to visualize high-dimensional data in two or three dimensions. It can help reveal the underlying structure of data by mapping it to a lower-dimensional space while preserving the similarity between data points. t-SNE plots can be effective for visualizing the separability of clusters.**
3. **Dendrogram (for Hierarchical Clustering): Hierarchical clustering often results in a dendrogram, which is a tree-like structure showing how data points are grouped together. Dendrograms provide a visual representation of the hierarchy of clusters and how they are merged.**
4. **Heatmaps: Heatmaps can be used to visualize the relationships between clusters and the features used for clustering. You can create heatmaps to show the average or median values of each feature within each cluster, helping you identify what defines each cluster.**
5. **Parallel Coordinate Plots: Parallel coordinate plots are useful for visualizing multivariate data, where each axis represents a different feature, and lines connect data points based on their feature values. This can be helpful for understanding the distribution of data points across different features in each cluster.**
6. **Cluster Profiles: Create visual profiles for each cluster, summarizing the key statistics and characteristics of data points within the cluster. These profiles can include histograms, box plots, and summary statistics for each feature within a cluster.**
7. **Geospatial Visualization: If your data has a geographical component, you can create maps to visualize the geographical distribution of customers or data points within different clusters. Tools like Geographic Information Systems (GIS) can be used for this purpose.**
8. **Interactive Visualizations: Use interactive tools and libraries like Plotly or D3.js to create dynamic visualizations that allow users to explore the data and cluster relationships interactively.**
9. **Data Distribution Plots: Plot the distribution of data points within each cluster, such as histograms, kernel density plots, or violin plots, to understand the underlying data distribution in each cluster.**
10. **Cluster Center Visualizations: For algorithms like K-Means, visualizing the cluster centers (centroids) can provide insights into the central tendencies of each cluster. You can use bar charts or radar plots to show the feature values at the cluster centers.**

**Eg**:

# In[1]

# ai\_0\_30 = df["Annual Income (k­-$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)]

# ai\_31\_60= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=31)& (df["Annual Income (k$)"] <=60)]

# ai\_61\_90= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=61)& (df["Annual Income (k$)"] <=90)]

# ai\_61\_90=df["Annual Income (k$)"][(df["Annual Income (k$)"] >=91)& (df["Annual Income (k$)"] <=120)]

# ai\_121\_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=121) & (df["Annual Income (k$)"] <=150)]

# aix = ["$ 0 - 30,000","$ 30,001 - 60,000","$ 60,001 - 90,000","$ 90,001 - 120,000","$ 120,001 - 150,000"]

# aiy = [len(ai\_0\_30.values),len(ai\_31\_60.values),len(ai\_61\_90.values),len(ai\_61\_90.values),len(ai\_121\_150.values)]

# plt.figure(figsize=(15,6))

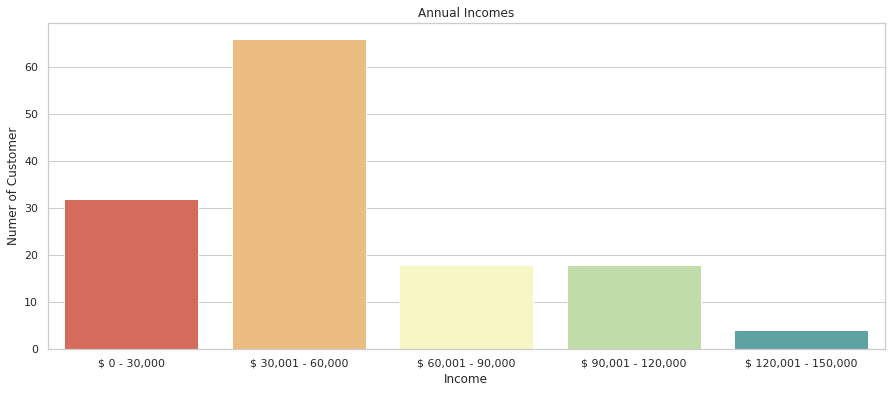
# sns.barplot(x=aix,y=aiy,palette="Spectral")

# plt.title("Annual Incomes")

# plt.xlabel("Income")

# plt.ylabel("Numer of Customer")

# plt.show()



# In[2]

# X1 = df.loc[:,["Age","Spending Score (1-100)"]].values

# from sklearn.cluster import KMeans

# wcss=[]

# for k in range(1,11):

# kmeans = KMeans(n\_clusters = k, init = "k-means++")

# kmeans.fit(X1)

# wcss.append(kmeans.inertia\_)

# plt.figure(figsize =( 12,6))

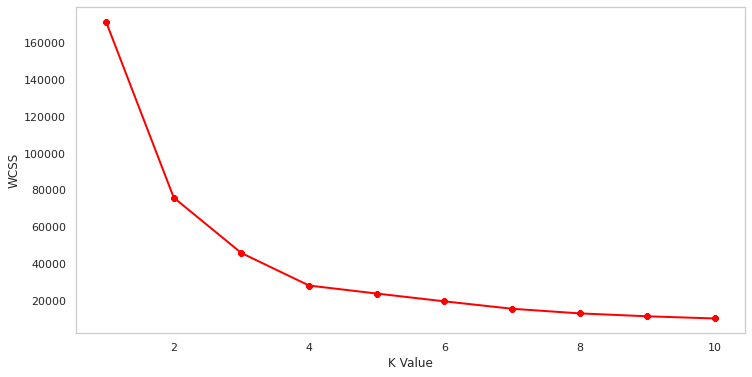
# plt.grid()

# plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")

# plt.xlabel("K Value")

# plt.ylabel("WCSS")

# plt.show()



# In[3]

# X2 = df.loc[:,["Annual Income (k$)","Spending Score (1-100)"]].values

# from sklearn.cluster import KMeans

# wcss=[]

# for k in range(1,11):

# kmeans = KMeans(n\_clusters = k, init = "k-means++")

# kmeans.fit(X2)

# wcss.append(kmeans.inertia\_)

# plt.figure(figsize =( 12,6))

# plt.grid()

# plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")

# plt.xlabel("K Value")

# plt.ylabel("WCSS")

# plt.show()

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

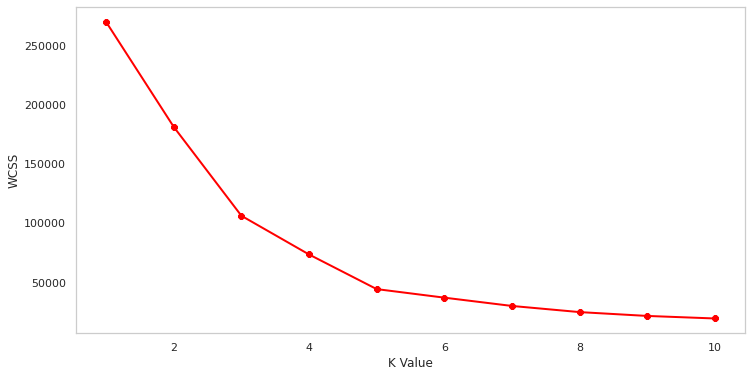
# plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

# plt.title('Clusters of Customers')

# plt.xlabel('Age')

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

# plt.show



# In[4]

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

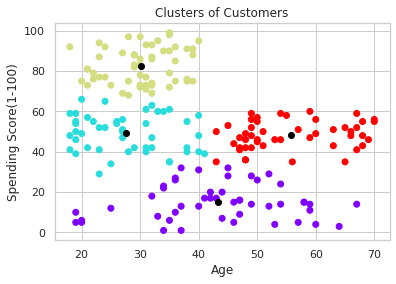
# plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

# plt.title('Clusters of Customers')

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

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

# plt.show



In[5]

# 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')

# ax.scatter(df.Age[df.label == 0 ],

# df["Annual Income (k$)"][df.label == 0],

# df["Spending Score (1-100)"][df.label == 0], c = 'blue',s=60)

# ax.scatter(df.Age[df.label == 1 ],df["Annual Income (k$)"][df.label == 1],df["Spending Score (1-100)"][df.label == 1], c = 'red',s=60)

# ax.scatter(df.Age[df.label == 2 ],df["Annual Income (k$)"][df.label == 2],df["Spending Score (1-100)"][df.label == 2], c = 'green',s=60)

# ax.scatter(df.Age[df.label == 3 ],df["Annual Income (k$)"][df.label == 3],df["Spending Score (1-100)"][df.label == 3], c = 'orange',s=60)

# ax.scatter(df.Age[df.label == 4 ],df["Annual Income (k$)"][df.label == 4],df["Spending Score (1-100)"][df.label == 4], c = 'purple',s=60)

# ax.view\_init(30,185)

# plt.xlabel("Age")

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

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

# 

Interpretation.

1. **Assign Meaningful Labels to Clusters:**

* **Give each cluster a descriptive label based on the characteristics of the data points within it. These labels should be easy to understand and should reflect the key attributes of the cluster.**

1. **Profile Each Cluster:**

* **Create detailed profiles for each cluster by summarizing the main characteristics. This includes statistical summaries, such as means, medians, and standard deviations for numerical features, and frequency distributions for categorical features.**

1. **Explore Feature Importance:**

* **Analyze the importance of each feature within clusters. Heatmaps or bar charts can help you understand which features are most relevant in defining each cluster. This can highlight the distinguishing characteristics of each group.**

1. **Compare Clusters:**

* **Compare the clusters to each other to understand the differences and similarities. Visualizations like parallel coordinate plots or radar charts can help with this. It's important to see how clusters are separated in feature space.**

1. **Business Implications:**

* **Translate the cluster characteristics into actionable insights or business implications. What do these clusters mean for your organization or project? How can you use this information to make decisions or take actions?**

1. **Customer Personas:**

* **Create customer personas for each cluster. Describe the typical customer or data point within each cluster, including demographics, behavior, preferences, and pain points.**

1. **Hypotheses Generation:**

* **Use the insights from clusters to generate hypotheses or ideas. For example, if you discover a cluster of high-value customers, you may hypothesize that offering them targeted promotions could increase their loyalty.**

1. **Validation and Iteration:**

* **Test the interpretations and hypotheses by conducting experiments or surveys. This can help validate your insights and refine your clustering model.**

1. **Feedback Loop:**

* **Maintain a feedback loop with stakeholders or domain experts. Their input and expertise can provide valuable context and refine your interpretations.**

1. **Documentation:**

* **Document your interpretations and findings for future reference. Clear documentation makes it easier to share insights with others and to revisit the analysis if needed.**

1. **Communication:**

* **Present your findings and interpretations to relevant stakeholders, team members, or decision-makers. Use effective visualizations and storytelling to convey the meaning of the clusters and their implications.**

1. **Actionable Insights:**

* **Ultimately, the goal of clustering is to turn insights into actions. Use the cluster interpretations to develop strategies, make data-driven decisions, and create targeted marketing campaigns, product recommendations, or other initiatives.**