**Que1. Business Problem**

* 1. **What is the business objective?**
  2. **Are there any constraints?**

**----🡪**

What is the business objective?

---- **Goal**: What does the business want to achieve? Keep it simple and clear.

**Example**: The business wants to increase sales by 10% in the next year.

Are there any constraints?

---- **Limits**: What could make it hard to solve the problem? Think about money, time, rules, or other things.**Example**: A limit could be "having only $20,000 to spend or following privacy rules.

**Que3. . Data Pre-processing**

**3.1 Data Cleaning, Feature Engineering, etc**

**Data Cleaning**

**What is it?**: Fixing the data to remove mistakes or fill in missing information.**Examples**: Remove wrong or repeated data, fill in missing values, or correct errors.

**Feature Engineering**

**What is it?**: Creating or changing data columns to help the analysis.**Examples**:

1)Turn dates into "day of the week."

2)Change words into numbers.

3)Make new columns, like grouping ages into "age groups."

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

**4.2 Univariate analysis.**

**4.3. Bivariate analysis**

**What is it?**: Getting an overview of the data.

**Examples**: Checking the basic details like averages, smallest and largest values, and how the data is organized.

**Univariate Analysis**

**What is it?**: Looking at one thing at a time.

**Examples**:

Finding out the average score of students.

Counting how many people fall into different age groups.

Making charts to show one feature.

**Bivariate Analysis**

**What is it?**: Looking at how two things relate to each other.

**Examples**:

1)Checking if older people earn more money.

2)Making scatter plots to see if one thing affects another.

3)Finding out if two features are related.

**6. Write about the benefits/impact of the solution - in what way does the business (client) benefit from the solution provided?**

-🡪

**. Benefits/Impact of the Solution**

**What is it?**: Explain how the solution helps the business.

**Examples**:

**More Sales**: The business can sell more products or services.

**Save Money**: The solution can cut costs or make things cheaper.

**Happy Customers**: Customers will be more satisfied and come back more often.

**Faster Work**: Employees will get things done quicker and easier.

**Better Choices**: The business can make smarter decisions with better information.

**Que.5. Model Building**

**5.1 Build the model on the scaled data (try multiple options).**

**5.2 Perform K- means clustering and obtain optimum number of clusters using scree plot.**

**5.3 Validate the clusters (try with different number of clusters) – label the clusters and derive insights (compare the results from multiple approaches).**

**--🡪Model Building.**

#### 5.1 Build the Model on Scaled Data

**What to Do**: Create and test different models using data that’s been adjusted to a common scale.

**Example**: Try models like linear regression or decision trees.

#### 5.2 Perform K-Means Clustering and Find the Best Number of Clusters

**What to Do**: Use K-Means to group data into clusters. Use a scree plot to find the best number of clusters.

**Example**: Look at a graph that shows how adding more clusters affects the results.

#### 5.3 Validate Clusters, Label Them, and Get Insights

**What to Do**: Check how well different numbers of clusters work. Name each cluster and see what you can learn from them.

**Example**: Test different cluster counts, name each group based on what they represent, and compare results to understand patterns.

**For this one I did some codes :**

**Firstly I created a numeric dataset for it with help of this code-🡪**

**import pandas as pd**

**# Create a DataFrame with sample data**

**data = {**

**'Age': [25, 30, 22, 35, 28, 40, 27, 33, 26, 31],**

**'Income': [3000, 3500, 2500, 4000, 3200, 4500, 3100, 3800, 2900, 3600],**

**'SpendingScore': [7, 5, 8, 4, 6, 3, 7, 5, 8, 4],**

**'CustomerSatisfaction': [4, 3, 5, 2, 4, 1, 4, 3, 5, 2]**

**}**

**df = pd.DataFrame(data)**

**# Save to Excel file**

**df.to\_excel("C:/Assignment-1/numericdata.xlsx", index=False)**

**################**

**After this I did this codes -🡪**

**For (5.1 Build the model on the scaled data (try multiple options)**

**import pandas as pd**

**from sklearn.preprocessing import StandardScaler**

**from sklearn.linear\_model import LinearRegression**

**from sklearn.tree import DecisionTreeRegressor**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import mean\_squared\_error**

**df = pd.read\_excel("C:/Assignment-1/numericdata.xlsx")**

**X = df.drop('CustomerSatisfaction', axis=1)**

**y = df['CustomerSatisfaction']**

**scaler = StandardScaler()**

**X\_scaled = scaler.fit\_transform(X)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.3, random\_state=42)**

**lr = LinearRegression()**

**lr.fit(X\_train, y\_train)**

**y\_pred\_lr = lr.predict(X\_test)**

**print("Linear Regression MSE:", mean\_squared\_error(y\_test, y\_pred\_lr))**

**dt = DecisionTreeRegressor()**

**dt.fit(X\_train, y\_train)**

**y\_pred\_dt = dt.predict(X\_test)**

**print("Decision Tree MSE:", mean\_squared\_error(y\_test, y\_pred\_dt))**

**###############**

**Now for (5.2 Perform K- means clustering and obtain optimum number of clusters using scree plot.)**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.cluster import KMeans**

**inertia = []**

**for k in range(1, 11):**

**kmeans = KMeans(n\_clusters=k, random\_state=42)**

**kmeans.fit(X\_scaled)**

**inertia.append(kmeans.inertia\_)**

**plt.plot(range(1, 11), inertia, marker='o')**

**plt.xlabel('Number of Clusters')**

**plt.ylabel('Inertia')**

**plt.title('Scree Plot')**

**plt.show()**

**Now for (5.3 Validate the clusters (try with different number of clusters) – label the clusters and derive insights (compare the results from multiple approaches).**

**for k in [3, 4, 5]:**

**kmeans = KMeans(n\_clusters=k, random\_state=42)**

**clusters = kmeans.fit\_predict(X\_scaled)**

**df[f'Cluster\_{k}'] = clusters**

**print(f"Cluster Analysis for {k} Clusters:")**

**print(df.groupby(f'Cluster\_{k}').mean())**

**print()**

**Problem Statements:**

Perform K means clustering on the airlines dataset to obtain optimum number of clusters. Draw the inferences from the clusters obtained.

---🡪

For this one I created a new dataset.i am not using eastewest dataset for it

I created that dataset by following code

import pandas as pd

# Sample data

data = {

'PassengerID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Age': [25, 30, 45, 35, 50, 29, 40, 32, 28, 38],

'FlightMiles': [15000, 20000, 5000, 12000, 8000, 18000, 6000, 16000, 14000, 10000],

'FlightFrequency': [10, 15, 5, 12, 8, 14, 6, 11, 13, 9],

'SpendingScore': [500, 600, 300, 550, 400, 650, 350, 520, 580, 450],

'LoyaltyPoints': [2000, 2500, 1000, 2200, 1500, 2700, 1200, 2100, 2300, 1700]

}

# Create DataFrame

df = pd.DataFrame(data)

# Save DataFrame to Excel file with the name 'problem1.xlsx'

file\_path = 'C:/Assignment-1/problem1.xlsx'

df.to\_excel(file\_path, index=False)

print(f"Dataset created and saved to {file\_path}")

**and I did this code by this dataset**

**import pandas as pd**

**from sklearn.preprocessing import StandardScaler**

**from sklearn.cluster import KMeans**

**from sklearn.metrics import silhouette\_score**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**file\_path = 'C:/Assignment-1/problem1.xlsx'**

**data = pd.read\_excel(file\_path)**

**data = data.drop(columns=['PassengerID'])**

**scaler = StandardScaler()**

**data\_scaled = scaler.fit\_transform(data)**

**distortions = []**

**K = range(1, 11)**

**for k in K:**

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

**kmeans.fit(data\_scaled)**

**distortions.append(kmeans.inertia\_)**

**plt.figure(figsize=(8, 6))**

**plt.plot(K, distortions, 'bo-')**

**plt.xlabel('Number of clusters')**

**plt.ylabel('Distortion')**

**plt.title('Elbow Method for Optimal Number of Clusters')**

**plt.show()**

**silhouette\_scores = []**

**for k in K:**

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

**clusters = kmeans.fit\_predict(data\_scaled)**

**silhouette\_avg = silhouette\_score(data\_scaled, clusters)**

**silhouette\_scores.append(silhouette\_avg)**

**plt.figure(figsize=(8, 6))**

**plt.plot(K, silhouette\_scores, 'bo-')**

**plt.xlabel('Number of clusters')**

**plt.ylabel('Silhouette Score')**

**plt.title('Silhouette Scores for Optimal Number of Clusters')**

**plt.show()**

**optimal\_k = 3**

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

**clusters = kmeans.fit\_predict(data\_scaled)**

**data['Cluster'] = clusters**

**cluster\_summary = data.groupby('Cluster').mean()**

**print(cluster\_summary)**

**plt.figure(figsize=(10, 8))**

**sns.scatterplot(x=data['FlightMiles'], y=data['SpendingScore'], hue=data['Cluster'], palette='viridis')**

**plt.title('Clusters Visualization')**

**plt.show()**

1. Perform clustering on mixed data. Convert the categorical variables to numeric by using dummies or label encoding and perform normalization techniques. The dataset has the details of customers related to their auto insurance. Refer to Autoinsurance.csv dataset.

**--🡪 for this question I take help from that dataset autoinsurance I create a new small datset and then I write a code**

**With help of this code I make that dataset :**

**import pandas as pd**

**# Define the dataset**

**data = {**

**'Months Since Policy Inception': [15, 42, 38],**

**'Number of Open Complaints': [16, 0, 0],**

**'Number of Policies': [17, 8, 2],**

**'Policy Type': ['Corporate Auto', 'Personal Auto', 'Personal Auto'],**

**'Policy': ['Corporate L3', 'Personal L3', 'Personal L3'],**

**'Renew Offer Type': ['Offer1', 'Offer3', 'Offer1'],**

**'Sales Channel': ['Agent', 'Agent', 'Agent'],**

**'Total Claim Amount': [384.81, 1131.46, 566.47],**

**'Vehicle Class': ['Two-Door Car', 'Four-Door Car', 'Two-Door Car'],**

**'Vehicle Size': ['Medsize', 'Medsize', 'Medsize']**

**}**

**# Create a DataFrame**

**df = pd.DataFrame(data)**

**# Define the file path**

**file\_path = r'C:/Assignment-1/autoinsurance.xlsx'**

**# Save the DataFrame to an Excel file**

**df.to\_excel(file\_path, index=False, engine='openpyxl')**

**print(f"Dataset saved to {file\_path}")**

**after that I write this code**

**import pandas as pd**

**from sklearn.preprocessing import StandardScaler, OneHotEncoder**

**from sklearn.compose import ColumnTransformer**

**from sklearn.pipeline import Pipeline**

**from sklearn.cluster import KMeans**

**file\_path = r'C:/Assignment-1/autoinsurance.xlsx'**

**df = pd.read\_excel(file\_path, engine='openpyxl')**

**categorical\_columns = ['Policy Type', 'Policy', 'Renew Offer Type', 'Sales Channel', 'Vehicle Class', 'Vehicle Size']**

**numeric\_columns = ['Months Since Policy Inception', 'Number of Open Complaints', 'Number of Policies', 'Total Claim Amount']**

**preprocessor = ColumnTransformer(**

**transformers=[**

**('cat', OneHotEncoder(), categorical\_columns),**

**('num', StandardScaler(), numeric\_columns)**

**])**

**pipeline = Pipeline(steps=[**

**('preprocessor', preprocessor),**

**('clusterer', KMeans(n\_clusters=3, random\_state=42))**

**])**

**df['Cluster'] = pipeline.fit\_predict(df)**

**output\_file\_path = r'C:\Assignment-1\autoinsurance\_with\_clusters.xlsx'**

**df.to\_excel(output\_file\_path, index=False, engine='openpyxl')**

**print(f"Clustered dataset saved to {output\_file\_path}")**