WEEK 3 TASKS:

Supervised Learning :

◇ What it means:

Model ko **pehle se labelled data** diya jata hai. Har input ke sath sahi answer (label) bhi hota hai.
Model "sikh kar" naye answers predict karta hai.

◇ Example:

Tumhare paas students ke marks aur pass/fail status ka data hai:

```
mathematica
CopyEdit
Input: Marks = 80 → Label: Pass
Input: Marks = 30 → Label: Fail
```

Model seekh jata hai: "Agar marks zyada hain to Pass."

◇ Algorithms:

- Linear Regression
- Decision Trees
- SVM
- Logistic Regression

☑ Unsupervised Learning:

◇ What it means:

Data ke sath **koi label nahi hota**. Model **patterns dhoondta hai** bina bataye ke kya sahi jawab hai.

Example:

Tumhare paas sirf customers ka data hai (jaise Age aur Income), magar koi label nahi ke kaun rich hai ya poor.
Model khud hi data ko **groups (clusters)** mein divide karta hai.

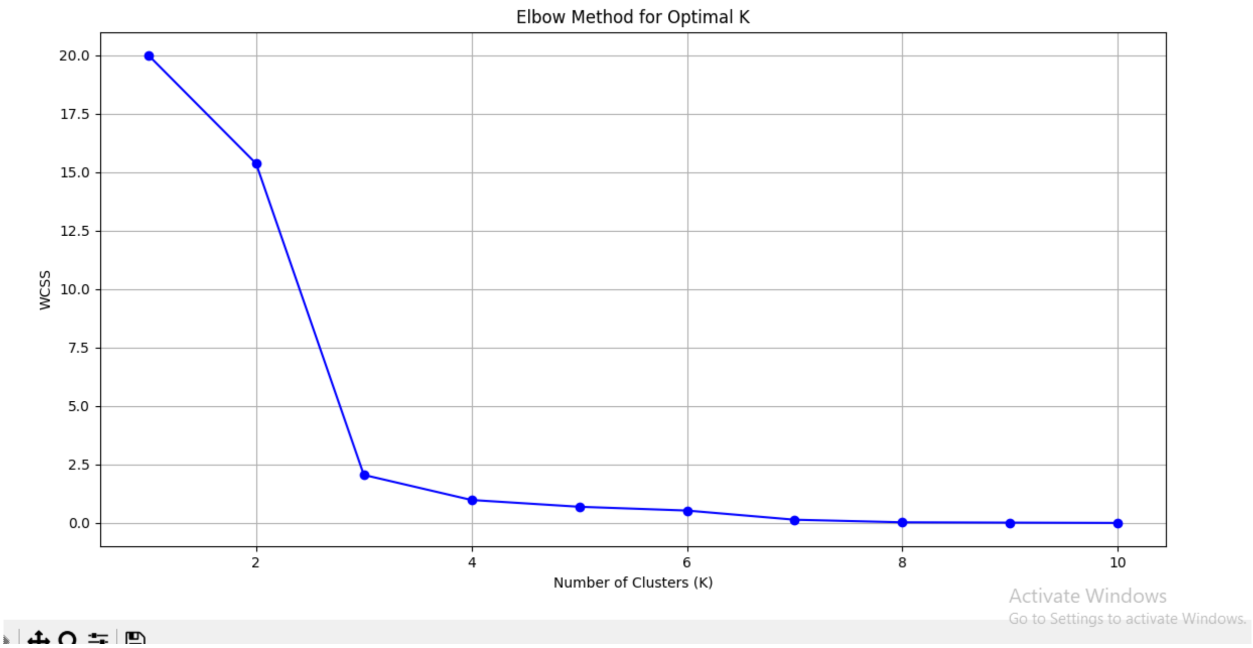
Algorithms:

- K-Means Clustering
- Hierarchical Clustering
- PCA (Dimensionality Reduction)

K_MEANS ELBOW METHOD:
SOURCE CODE:

```
import matplotlib.pyplot as plt
vfrom sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import pandas as pd

# Load dataset
df = pd.read_csv("customer_data.csv")

# We'll use Age and Spending_Score for clustering
X = df[['Age', 'Spending_Score']]

# Standardize the features

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# --- Elbow Method ---
wcss = []
K_range = range(1, 11)
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot Elbow
plt.figure(figsize=(8, 4))
plt.plot(K_range, wcss, 'bo-')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.title('Elbow Method for Optimal K')
plt.grid(True)
plt.show()
```

OUTPUT:



K_MEANS Clustering Centroids:

SOURCE CODE:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import pandas as pd

# Load dataset
```

```
df = pd.read_csv("customer_data.csv")

# Apply KMeans with K=3 based on Elbow result
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)

# Get cluster centroids
centroids = kmeans.cluster_centers_

# Plot scatter with centroids
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Spending_Score', hue='Cluster', data=df,
palette='Set2', s=100)
plt.scatter(
    scaler.inverse_transform(centroids)[:, 0],
    scaler.inverse_transform(centroids)[:, 1],
    s=300, c='black', marker='X', label='Centroids'
)
plt.title('Customer Segmentation (K-Means Clusters)')
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.legend()
plt.grid(True)
plt.show()
```

K_Means Clustering Dendgram:

SOURCE CODE:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import pandas as pd

# Load dataset
df = pd.read_csv("customer_data.csv")

# Apply KMeans with K=3 based on Elbow result
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
```

```
# Get cluster centroids
centroids = kmeans.cluster_centers_

# Plot scatter with centroids
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Spending_Score', hue='Cluster', data=df,
palette='Set2', s=100)
plt.scatter(
    scaler.inverse_transform(centroids)[:, 0],
    scaler.inverse_transform(centroids)[:, 1],
    s=300, c='black', marker='X', label='Centroids'
)
plt.title('Customer Segmentation (K-Means Clusters)')
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT: