NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

ARTIFICIAL INTELLIGENCE LAB

NAME	Ayesha Imran
Class	CS-A
Oel	09
Course	Artificial Intelligence
Date	9-December-25
Submitted To	Lec. Ijlal Haider

IN LAB TASKS

TASK 01: For the given dataset “wine-clustering.csv”, which contains information about various chemical properties of different wine samples. Your task is to apply the K-Means Clustering algorithm to group the wine samples into distinct clusters based on their feat Data Exploration and Preprocessing:

Perform exploratory data analysis (EDA) to:

- Understand the structure of the dataset (e.g., data types, missing values, and summary statistics).
- Visualize relationships between the features using scatter plots, heatmaps, or box plots.

Preprocess the data by:

- Handling missing values, if any.
- Encoding categorical variables like Condition and Location.
- Normalizing or scaling numerical features to ensure all features are on a similar scale.

Determine the Optimal Number of Clusters:

- Use the Elbow Method to determine the optimal number of clusters k.
- Plot the Within-Cluster Sum of Squares (WCSS) against the number of clusters.
- Based on the elbow plot, choose the value of k where the WCSS curve starts to level off.

Apply K-Means Clustering:

- Apply the K-Means Clustering algorithm using the chosen value of k.
- Print the cluster labels for each wine sample.
- Visualization of Clusters
- Visualize the clusters using a 2D scatter plot. Use the first two principal components
- (PCA) to reduce the dimensions for visualization.
- Use different colors to represent different clusters.
- Add labels and legends to the plot for clarity.

Cluster Analysis

- Analyze the cluster centroids and interpret the results.
- Discuss the differences between clusters based on the chemical features of the wine.
- Summarize your findings in a few sentences.

CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```python
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
df = pd.read_csv('wine-clustering.csv')

# Data Exploration
print("Dataset Shape:", df.shape)
print("\nFirst 5 rows:")
print(df.head())
print("\nDataset Info:")
print(df.info())
print("\nSummary Statistics:")
print(df.describe())
print("\nMissing Values:")
print(df.isnull().sum())

# Correlation Heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Feature Correlation Heatmap')
plt.show()

# Preprocessing: Scale the features
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

# Elbow Method to find optimal k
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(df_scaled)
    wcss.append(kmeans.inertia_)

# Plot Elbow Curve
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.grid(True)
plt.show()

# Apply K-Means with k=3 (optimal based on elbow)
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
```

```python
cluster_labels = kmeans.fit_predict(df_scaled)

# Print cluster labels
print("\nCluster Labels for each sample:")
print(cluster_labels)

# Add clusters to dataframe
df['Cluster'] = cluster_labels

# Cluster Centroids in original scale
centroids_scaled = kmeans.cluster_centers_
centroids_original = scaler.inverse_transform(centroids_scaled)
centroids_df = pd.DataFrame(centroids_original, columns=df.columns[:-1])
print("\nCluster Centroids (original feature scale):")
print(centroids_df)

# PCA for 2D Visualization
pca = PCA(n_components=2)
df_pca = pca.fit_transform(df_scaled)

# Scatter plot of clusters
plt.figure(figsize=(10, 8))
colors = ['#1f77b4', '#ff7f0e', '#2ca02c']
for i in range(3):
    plt.scatter(df_pca[cluster_labels == i, 0],
                df_pca[cluster_labels == i, 1],
                label=f'Cluster {i}',
                color=colors[i],
                alpha=0.7)

plt.title('K-Means Clusters Visualized using PCA')
plt.xlabel(f'Principal Component 1 ({pca.explained_variance_ratio_[0]:.2%}
variance)')
plt.ylabel(f'Principal Component 2 ({pca.explained_variance_ratio_[1]:.2%}
variance)')
plt.legend()
plt.grid(True)
plt.show()

# Optional: Analyze differences with box plots for key features
key_features = ['Alcohol', 'Malic_Acid', 'Flavanoids', 'Color_Intensity',
'Proline']
for feature in key_features:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Cluster', y=feature, data=df, palette='Set2')
```

```python
    plt.title(f'{feature} Distribution by Cluster')
    plt.show()

# Summary of findings
print("\nSummary:")
print("The K-Means algorithm successfully grouped the wines into 3 clusters.")
print("Cluster differences are evident in phenolic compounds (e.g., Flavanoids, Total_Phenols),")
print("color intensity, hue, acidity, and proline content.")
```

Output:

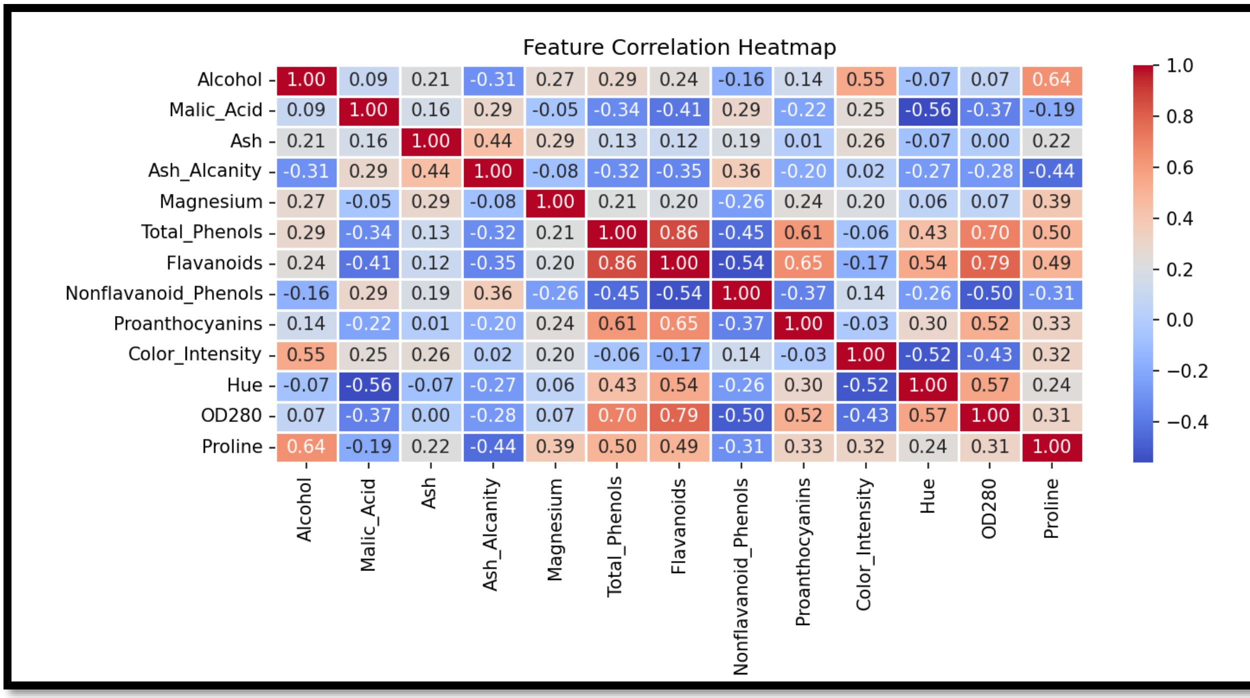
```
Dataset Shape: (178, 13)

First 5 rows:
   Alcohol  Malic_Acid   Ash  ...   Hue  OD280  Proline
0    14.23        1.71  2.43  ...  1.04   3.92     1065
1    13.20        1.78  2.14  ...  1.05   3.40     1050
2    13.16        2.36  2.67  ...  1.03   3.17     1185
3    14.37        1.95  2.50  ...  0.86   3.45     1480
4    13.24        2.59  2.87  ...  1.04   2.93      735

[5 rows x 13 columns]

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
```

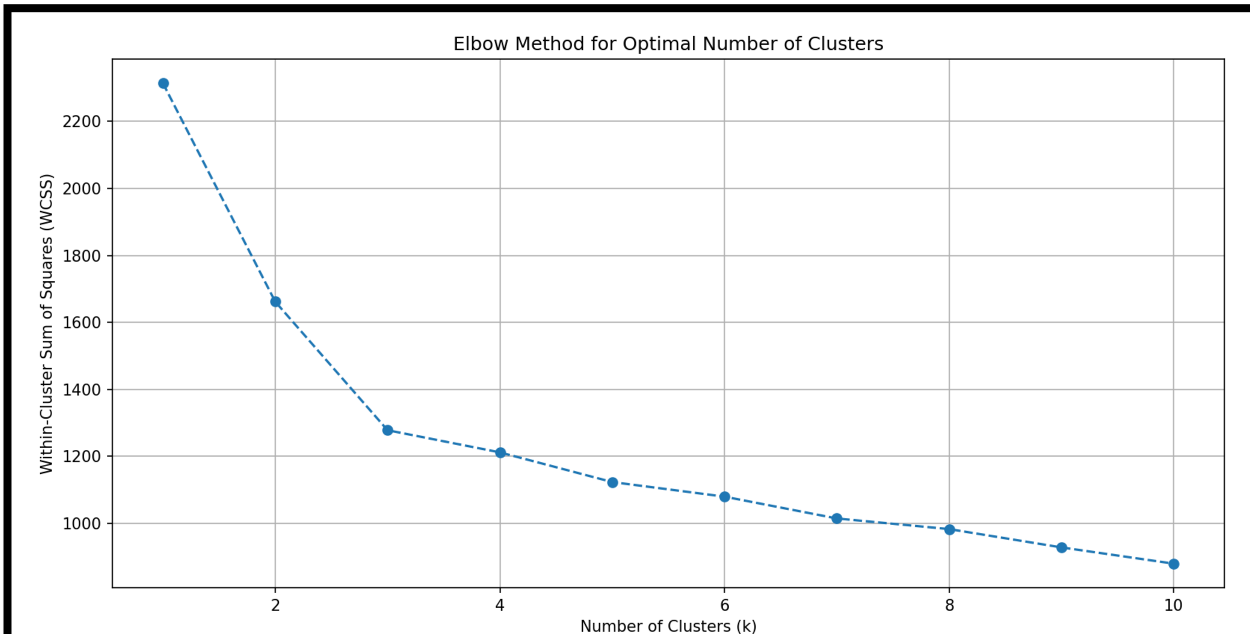
```
 0   Alcohol               178 non-null    float64
 1   Malic_Acid            178 non-null    float64
 2   Ash                   178 non-null    float64
 3   Ash_Alcanity          178 non-null    float64
 4   Magnesium             178 non-null    int64
 5   Total_Phenols         178 non-null    float64
 6   Flavanoids            178 non-null    float64
 7   Nonflavanoid_Phenols  178 non-null    float64
 8   Proanthocyanins       178 non-null    float64
 9   Color_Intensity       178 non-null    float64
 10  Hue                   178 non-null    float64
 11  OD280                 178 non-null    float64
 12  Proline               178 non-null    int64
dtypes: float64(11), int64(2)
memory usage: 18.2 KB
None

Summary Statistics:
```

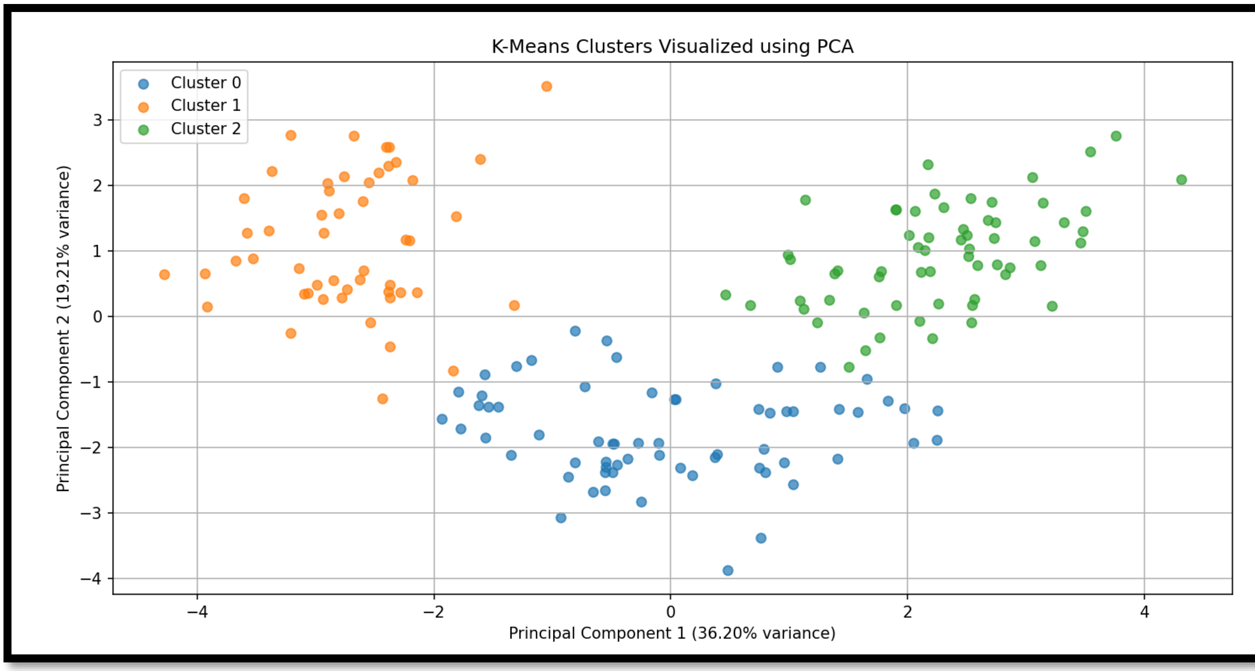
```
Summary Statistics:
          Alcohol  Malic_Acid  ...       OD280      Proline
count  178.000000  178.000000  ...  178.000000   178.000000
mean    13.000618    2.336348  ...    2.611685   746.893258
std      0.811827    1.117146  ...    0.709990   314.907474
min     11.030000    0.740000  ...    1.270000   278.000000
25%     12.362500    1.602500  ...    1.937500   500.500000
50%     13.050000    1.865000  ...    2.780000   673.500000
75%     13.677500    3.082500  ...    3.170000   985.000000
max     14.830000    5.800000  ...    4.000000  1680.000000

[8 rows x 13 columns]

Missing Values:
Alcohol                 0
Malic_Acid              0
Ash                     0
Ash_Alcanity            0
```

```
Ash_Alcanity            0
Magnesium               0
Total_Phenols           0
Flavanoids              0
Nonflavanoid_Phenols    0
Proanthocyanins         0
Color_Intensity         0
Hue                     0
OD280                   0
Proline                 0
dtype: int64
```

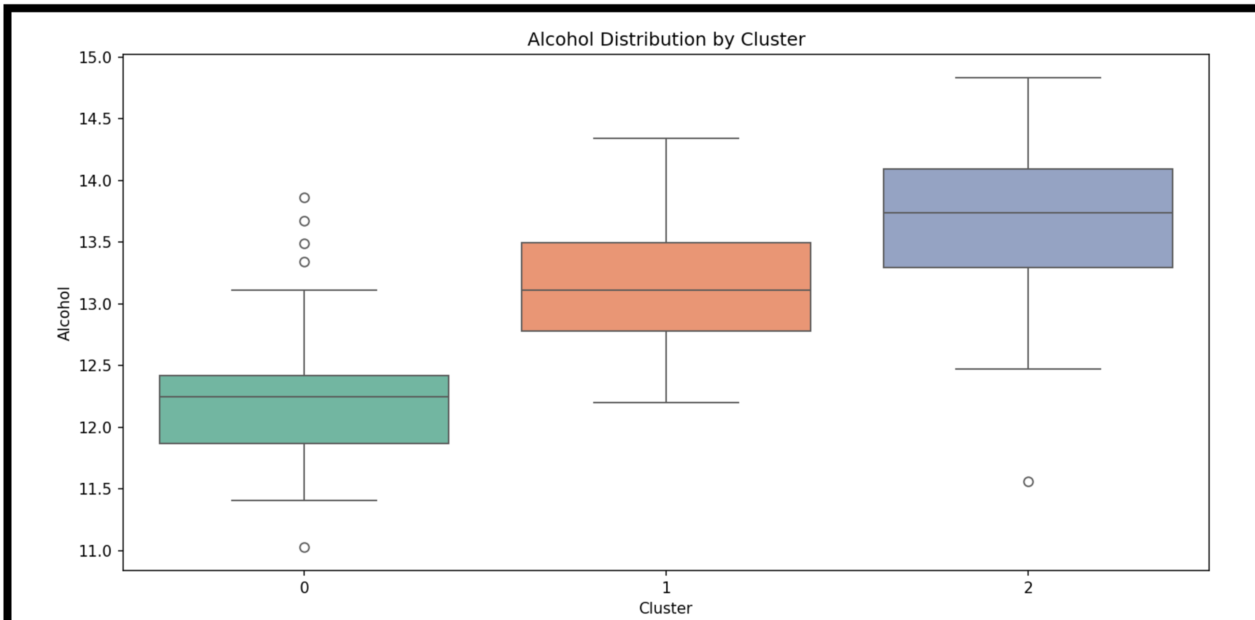
```
Proline                 0
dtype: int64

Cluster Labels for each sample:
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Cluster Centroids (original feature scale):
     Alcohol  Malic_Acid       Ash  ...       Hue     OD280      Proline
0  12.250923    1.897385  2.231231  ...  1.062708  2.803385   510.169231
1  13.134118    3.307255  2.417647  ...  0.691961  1.696667   619.058824
2  13.676774    1.997903  2.466290  ...  1.065484  3.163387  1100.225806
```


Feature Correlation Heatmap
Alcohol 1.00 0.09 0.21 -0.31 0.27 0.29 0.24 -0.16 0.14 0.55 -0.07 0.07 0.64
Malic_Acid 0.09 1.00 0.16 0.29 -0.05 -0.34 -0.41 0.29 -0.22 0.25 -0.56 -0.37 -0.19
Ash 0.21 0.16 1.00 0.44 0.29 0.13 0.12 0.19 0.01 0.26 -0.07 0.00 0.22
Ash_Alcanity -0.31 0.29 0.44 1.00 -0.08 -0.32 -0.35 0.36 -0.20 0.02 -0.27 -0.28 -0.44
Magnesium 0.27 -0.05 0.29 -0.08 1.00 0.21 0.20 -0.26 0.24 0.20 0.06 0.07 0.39
Total_Phenols 0.29 -0.34 0.13 -0.32 0.21 1.00 0.86 -0.45 0.61 -0.06 0.43 0.70 0.50
Flavanoids 0.24 -0.41 0.12 -0.35 0.20 0.86 1.00 -0.54 0.65 -0.17 0.54 0.79 0.49
Nonflavanoid_Phenols -0.16 0.29 0.19 0.36 -0.26 -0.45 -0.54 1.00 -0.37 0.14 -0.26 -0.50 -0.31
Proanthocyanins 0.14 -0.22 0.01 -0.20 0.24 0.61 0.65 -0.37 1.00 -0.03 0.30 0.52 0.33
Color_Intensity 0.55 0.25 0.26 0.02 0.20 -0.06 -0.17 0.14 -0.03 1.00 -0.52 -0.43 0.32
Hue -0.07 -0.56 -0.07 -0.27 0.06 0.43 0.54 -0.26 0.30 -0.52 1.00 0.57 0.24
OD280 0.07 -0.37 0.00 -0.28 0.07 0.70 0.79 -0.50 0.52 -0.43 0.57 1.00 0.31
Proline 0.64 -0.19 0.22 -0.44 0.39 0.50 0.49 -0.31 0.33 0.32 0.24 0.31 1.00
Alcohol Malic_Acid Ash Ash_Alcanity Magnesium Total_Phenols Flavanoids Nonflavanoid_Phenols Proanthocyanins Color_Intensity Hue OD280 Proline
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4

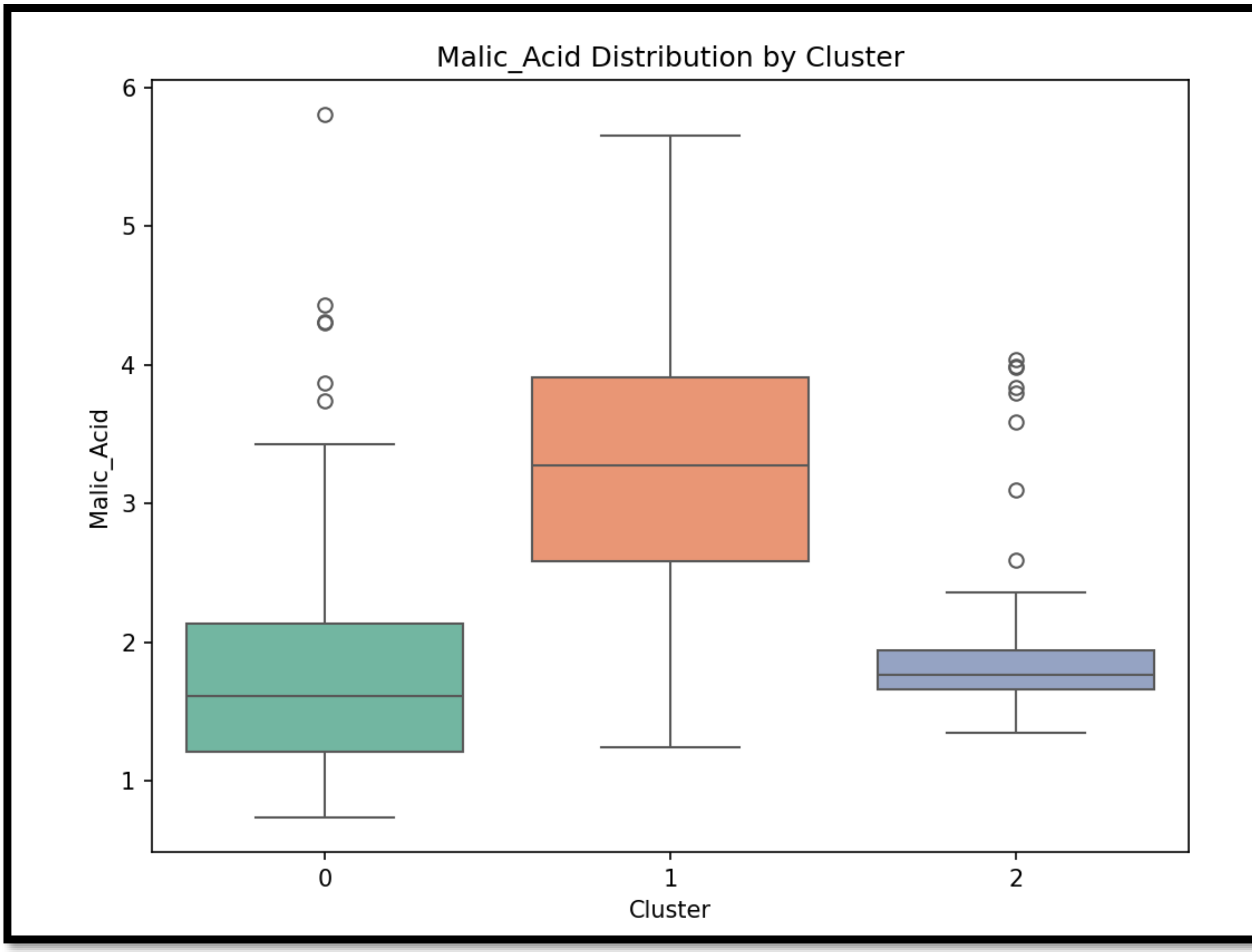

Elbow Method for Optimal Number of Clusters
Within-Cluster Sum of Squares (WCSS)
2200
2000
1800
1600
1400
1200
1000
2 4 6 8 10
Number of Clusters (k)



K-Means Clusters Visualized using PCA
Cluster 0
Cluster 1
Cluster 2
Principal Component 2 (19.21% variance)
Principal Component 1 (36.20% variance)



Alcohol Distribution by Cluster
Alcohol
Cluster




```
Summary:
The K-Means algorithm successfully grouped the wines into 3 clusters.
Cluster differences are evident in phenolic compounds (e.g., Flavanoids, Total_Phenols)
color intensity, hue, acidity, and proline content.
```

Task 1

For the give dataset “Mall_Customers.csv”, which contains customer data, including their Customer ID, Gender, Age, Annual Income (k$), and Spending Score (1-100). Your task is to use the K-Means Clustering algorithm to group the customers into distinct clusters based on their Annual Income and Spending Score.

Data Exploration and Preprocessing:

- Perform exploratory data analysis (EDA) to:

- Understand the structure of the dataset (e.g., data types, missing values,and summary statistics).
- Visualize relationships between the features using scatter plots, heatmaps, or box plots.

 Preprocess the data by:

- Handling missing values, if any.
- Encoding categorical variables like Condition and Location.
- Normalizing or scaling numerical features to ensure all features are on a similar scale.

 Extract the Annual Income and Spending Score columns for clustering.
Determine the Optimal Number of Clusters:

- Use the Elbow Method to determine the optimal number of clusters k.
- Plot the Within-Cluster Sum of Squares (WCSS) against the number of clusters.
- Based on the elbow plot, choose the value of k where the WCSS curve starts to level off.

Apply K-Means Clustering:

- Apply the K-Means Clustering algorithm to the selected features (Annual Income and Spending Score) using the optimal value of k.
- Assign cluster labels to each customer and display the resulting data. Visualization of Clusters
- Plot a 2D scatter plot of the clusters, with Annual Income on the x-axis and Spending

Score on the y-axis.

- Use different colors to represent different clusters.
- Mark the cluster centroids on the plot for better visualization. Cluster Analysis
- Analyze the clusters and interpret the results.
- Describe the characteristics of each cluster (e.g., high spenders, low-income customers, etc.).
- Provide insights into customer segments that could help businesses target their

- customers effectively.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Load the dataset (adjust path if necessary)
# Dataset available on Kaggle:
https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-
python
df = pd.read_csv('Mall_Customers.csv')

# Data Exploration and Preprocessing
print("Dataset Shape:", df.shape)
print("\nFirst 5 rows:")
print(df.head())
print("\nDataset Info:")
df.info()
print("\nSummary Statistics:")
print(df.describe())
print("\nMissing Values:")
print(df.isnull().sum())

# Note: No missing values in this dataset.
# Gender is categorical, but not used for clustering here.
# No 'Condition' or 'Location' columns (likely a copy-paste error from another
task).

# Extract features for clustering
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]

# Scale the features (important since income and score have different ranges)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Determine Optimal Number of Clusters - Elbow Method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
```

```python
# Plot Elbow Curve
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.grid(True)
plt.show()

# Optimal k = 5 (elbow point where curve starts to level off)

# Apply K-Means with k=5
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=42)
cluster_labels = kmeans.fit_predict(X_scaled)

# Add cluster labels to original dataframe
df['Cluster'] = cluster_labels

# Display the resulting data (first 20 rows as example)
print("\nData with Cluster Labels (first 20 rows):")
print(df.head(20))

# Full cluster assignment counts
print("\nCluster Sizes:")
print(df['Cluster'].value_counts())

# Visualization of Clusters
plt.figure(figsize=(10, 8))
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd']

for i in range(5):
    plt.scatter(df[df['Cluster'] == i]['Annual Income (k$)'],
                df[df['Cluster'] == i]['Spending Score (1-100)'],
                s=100, c=colors[i], label=f'Cluster {i}', alpha=0.7)

# Plot centroids (in original scale)
centroids_original = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids_original[:, 0], centroids_original[:, 1],
            s=300, c='black', marker='X', label='Centroids')

plt.title('Customer Clusters based on Annual Income and Spending Score')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.grid(True)
```

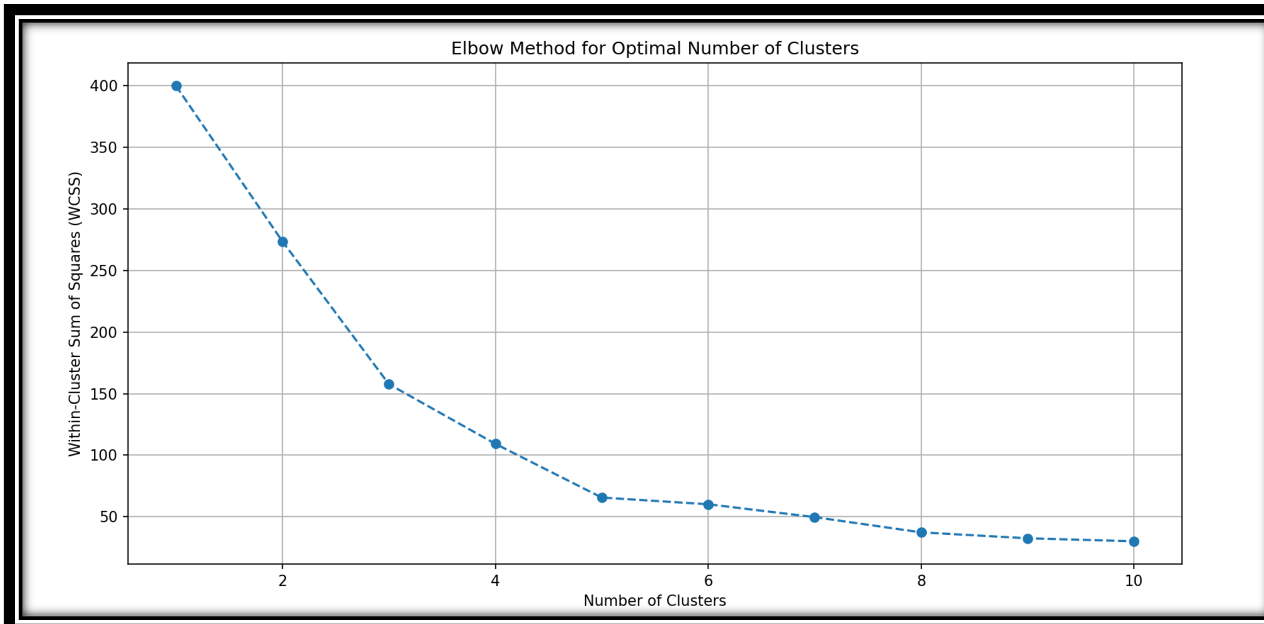
```
plt.show()
```

Output:

```
Dataset Shape: (200, 5)

First 5 rows:
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
```

```
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB

Summary Statistics:
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```

```
Missing Values:
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```



```
Data with Cluster Labels (first 20 rows):
    CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)  Cluster
0            1    Male   19                  15                      39        4
1            2    Male   21                  15                      81        2
2            3  Female   20                  16                       6        4
3            4  Female   23                  16                      77        2
4            5  Female   31                  17                      40        4
5            6  Female   22                  17                      76        2
6            7  Female   35                  18                       6        4
7            8  Female   23                  18                      94        2
8            9    Male   64                  19                       3        4
9           10  Female   30                  19                      72        2
10          11    Male   67                  19                      14        4
11          12  Female   35                  19                      99        2
12          13  Female   58                  20                      15        4
13          14  Female   24                  20                      77        2
14          15    Male   37                  20                      13        4
```

END