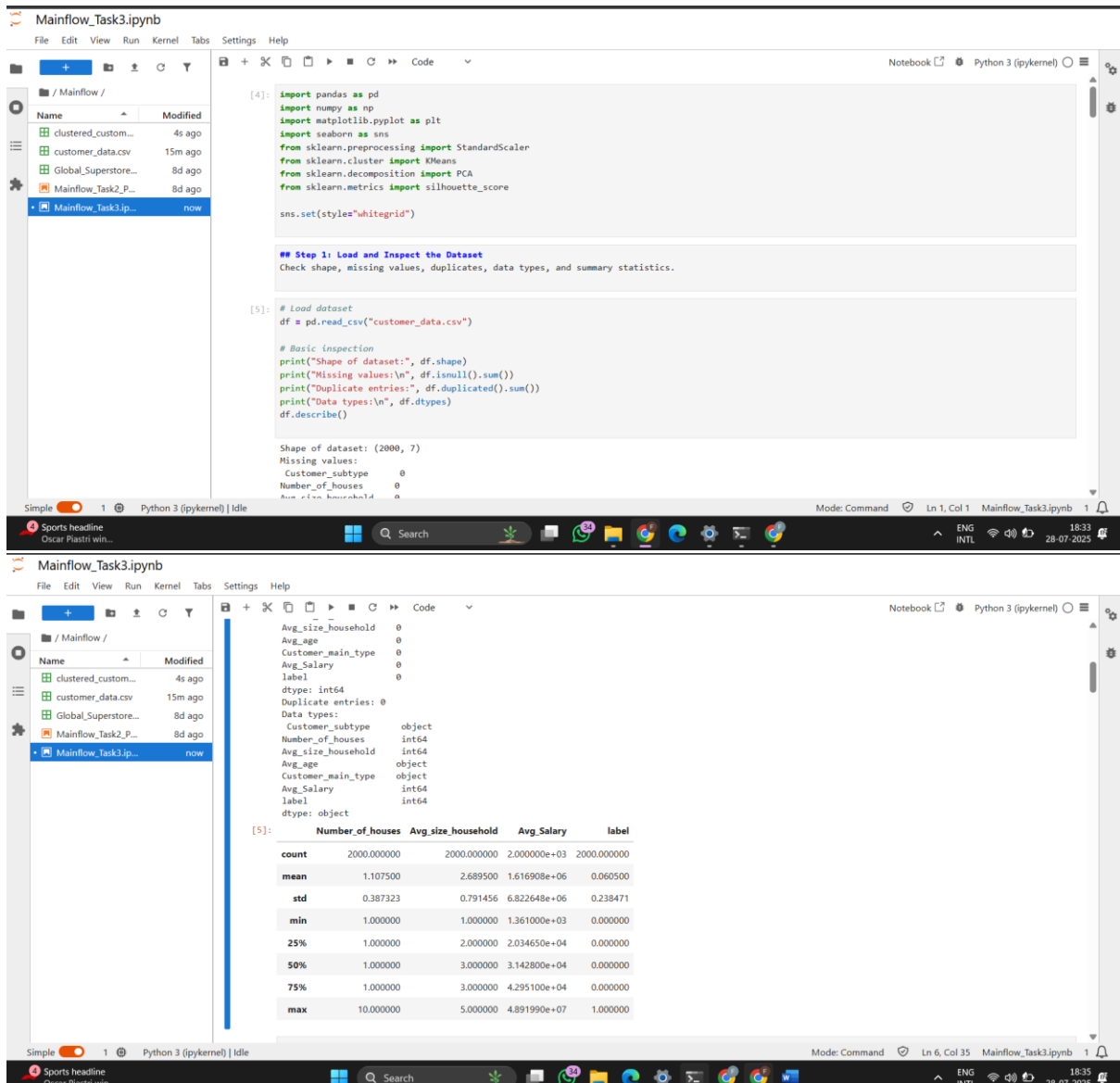


Name:-Faizan Sarfaraz Dandu

Mainflow Task3



```
[4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

sns.set(style="whitegrid")

## Step 1: Load and Inspect the Dataset
Check shape, missing values, duplicates, data types, and summary statistics.

[5]: # Load dataset
df = pd.read_csv("customer_data.csv")

# Basic inspection
print("Shape of dataset:", df.shape)
print("Missing values:\n", df.isnull().sum())
print("Duplicate entries:", df.duplicated().sum())
print("Data types:\n", df.dtypes)
df.describe()
```

Shape of dataset: (2000, 7)
Missing values:
Customer_subtype 0
Number_of_houses 0
Avg_size_household 0

```
[5]:
```

	Number_of_houses	Avg_size_household	Avg_Salary	label
count	2000.000000	2000.000000	2.000000e+03	2000.000000
mean	1.107500	2.689500	1.616908e+06	0.060500
std	0.387323	0.791456	6.822648e+06	0.238471
min	1.000000	1.000000	1.361000e+03	0.000000
25%	1.000000	2.000000	2.034650e+04	0.000000
50%	1.000000	3.000000	3.142800e+04	0.000000
75%	1.000000	3.000000	4.295100e+04	0.000000
max	10.000000	5.000000	4.891990e+07	1.000000

Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Python 3 (ipykernel)

Simple 1 Python 3 (ipykernel) | Idle

Upcoming Earnings

Search

ENG INTL 18:35 28-07-2025

Mode: Command Ln 6, Col 35 Mainflow_Task3.ipynb 1

Elbow Method

6000

Elbow Method

Standardize the selected features before applying KMeans.

```
[10]: scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Determine the optimal number of clusters using the Elbow method.

```
[12]: wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, markers='o')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```

Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Python 3 (ipykernel)

Simple 1 Python 3 (ipykernel) | Idle

Upcoming Earnings

Search

ENG INTL 18:35 28-07-2025

Mode: Command Ln 6, Col 35 Mainflow_Task3.ipynb 1

Standardize features to bring them to the same scale.

```
[ ]: features = ['Age', 'Annual Income', 'Spending Score']
X = df[features]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Find Optimal Number of Clusters using Elbow Method

```
[8]: # Convert Avg_age to numeric if needed
if df['Avg_age'].dtype == 'object':
    df['Avg_age'] = pd.to_numeric(df['Avg_age'].str.extract(r'(\d+)', expand=False), errors='coerce')

# Drop rows with missing values
df.dropna(inplace=True)
```

Feature Selection

We will select relevant numerical features for clustering.

```
[9]: features = ['Avg_age', 'Avg_Salary', 'Avg_size_household']
X = df[features]
```

Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Number of Clusters

Step 8: Silhouette Score
Evaluate clustering performance for different values of 'k'.

```
[13]: for k in range(2, 11):
      kmeans = KMeans(n_clusters=k, random_state=42)
      labels = kmeans.fit_predict(X_scaled)
      score = silhouette_score(X_scaled, labels)
      print(f'Silhouette Score for k={k}: {score:.4f}')
```

Silhouette Score for k=2: 0.4105
Silhouette Score for k=3: 0.3915
Silhouette Score for k=4: 0.4732
Silhouette Score for k=5: 0.5816
Silhouette Score for k=6: 0.6579
Silhouette Score for k=7: 0.6569
Silhouette Score for k=8: 0.6829
Silhouette Score for k=9: 0.6983
Silhouette Score for k=10: 0.7439

Step 9: Apply KMeans Clustering
We apply KMeans with the optimal number of clusters found earlier.

```
[14]: optimal_k = 4 # Adjust this if a different k is found optimal
      kmeans = KMeans(n_clusters=optimal_k, random_state=42)
      df['Cluster'] = kmeans.fit_predict(X_scaled)
```

Simple 1 Python 3 (ipykernel) | Idle

Hot days ahead 29°C

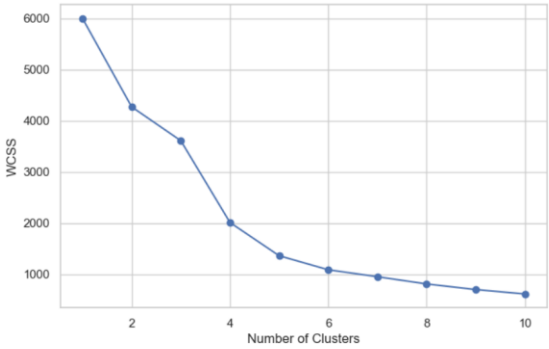
Mode: Command Ln 6, Col 35 Mainflow_Task3.ipynb 1

ENG INTL 18:37 28-07-2025

Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Elbow Method



Number of Clusters	WCSS (approx.)
2	6000
3	4300
4	3600
5	2000
6	1400
7	1200
8	1100
9	1050
10	1000

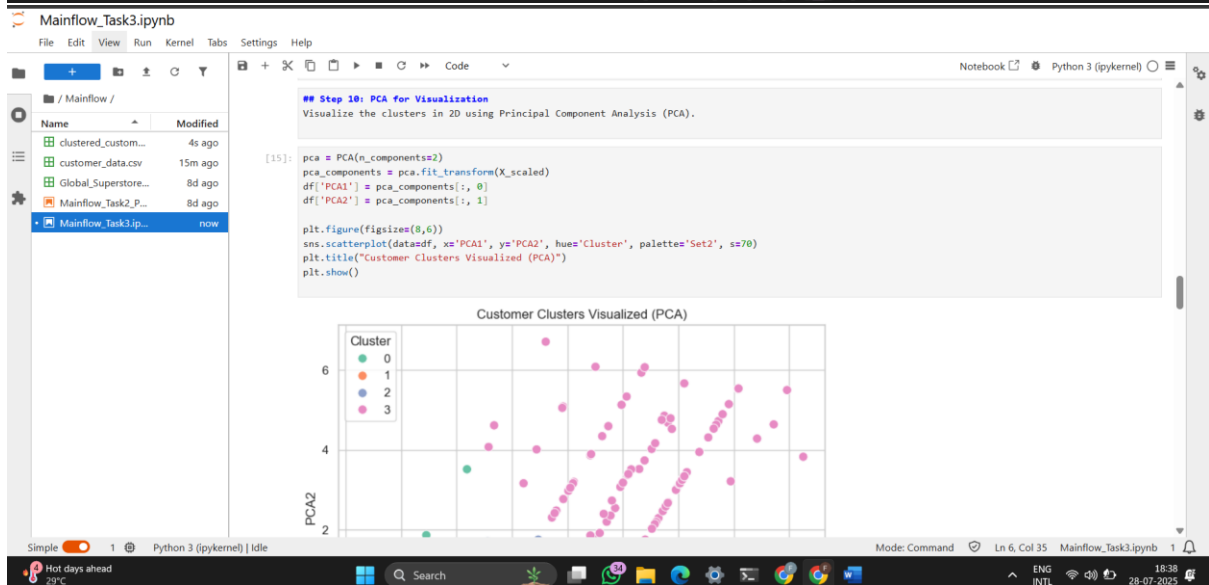
Step 8: Silhouette Score
Evaluate clustering performance for different values of 'k'.

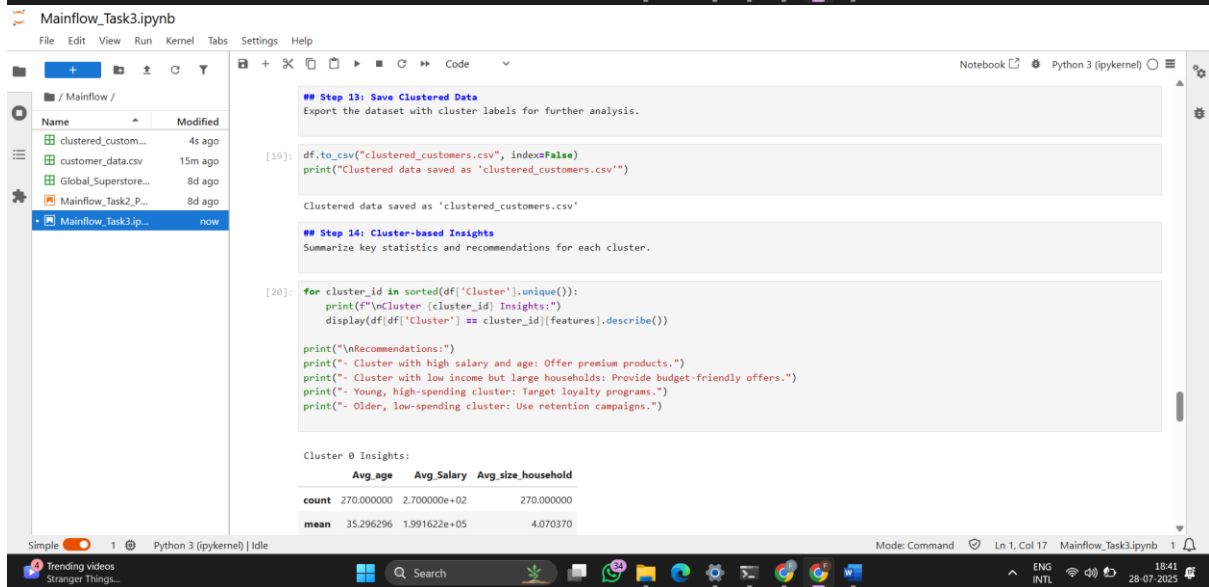
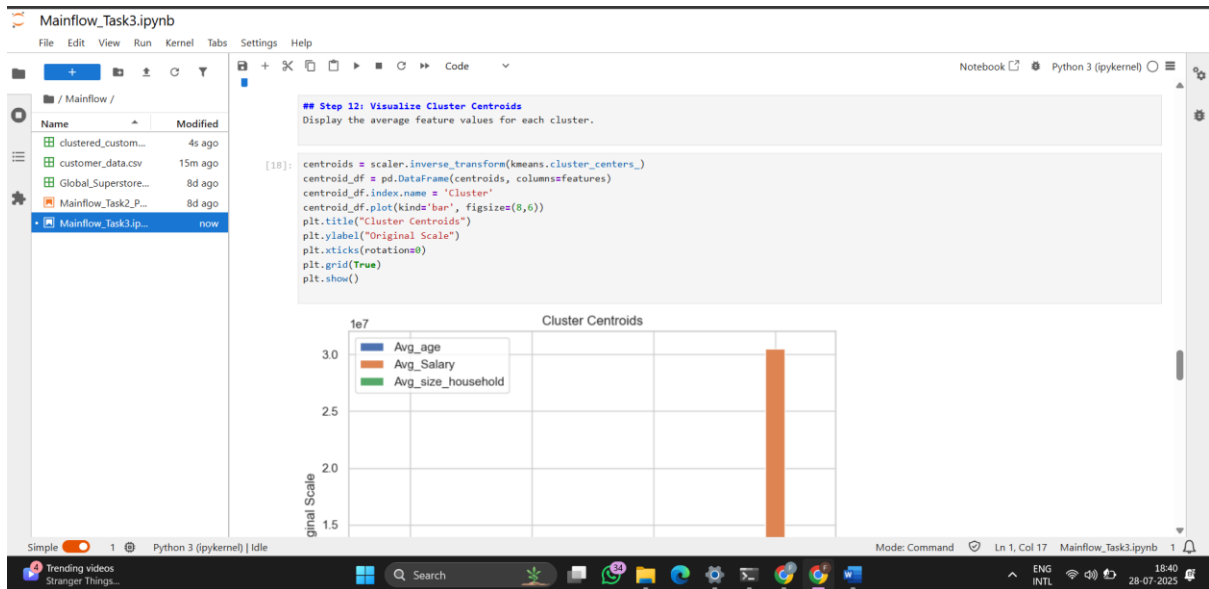
Simple 1 Python 3 (ipykernel) | Idle

Upcoming Earnings

Mode: Command Ln 6, Col 35 Mainflow_Task3.ipynb 1

ENG INTL 18:36 28-07-2025





Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Python 3 (ipykernel)

Cluster 0 Insights:

	Avg_age	Avg_Salary	Avg_size_household
count	270.000000	2.700000e+02	270.000000
mean	35.296296	1.991622e+05	4.070370
std	5.147019	1.329219e+06	0.256245
min	30.000000	2.208000e+03	4.000000
25%	30.000000	2.077100e+04	4.000000
50%	40.000000	3.253850e+04	4.000000
75%	40.000000	4.290000e+04	4.000000
max	50.000000	1.865756e+07	5.000000

Cluster 1 Insights:

	Avg_age	Avg_Salary	Avg_size_household
count	678.000000	6.780000e+02	678.000000
mean	46.519174	3.008033e+05	1.985251
std	7.371274	1.587379e+06	0.441316
min	30.000000	1.361000e+03	1.000000
25%	40.000000	1.923525e+04	2.000000
50%	50.000000	3.052550e+04	2.000000
75%	50.000000	4.234900e+04	2.000000

Simple Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 17 Mainflow_Task3.ipynb 18:41 28-07-2025

Mainflow_Task3.ipynb

File Edit View Run Kernel Tabs Settings Help

Python 3 (ipykernel)

Cluster 2 Insights:

	Avg_age	Avg_Salary	Avg_size_household
count	958.000000	9.580000e+02	958.000000
mean	35.824635	1.157722e+05	2.824635
std	5.551934	8.421835e+05	0.422139
min	20.000000	1.750000e+03	1.000000
25%	30.000000	1.978025e+04	3.000000
50%	40.000000	2.931600e+04	3.000000
75%	40.000000	4.036400e+04	3.000000
max	40.000000	1.374522e+07	3.000000

Simple Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 17 Mainflow_Task3.ipynb 18:41 28-07-2025

