

← →

Untitled (Workspace)

⚙️ 🗑️ 📄 📄

ExpFour.ipynb ×

CL IV - BI Codes > ExpFour > ExpFour.ipynb > import pandas as pd

🔍 Generate + Code + Markdown ▶ Run All 🧹 Clear All Outputs | 📄 Outline ...

Select Kernel

⌵

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import silhouette_score
import plotly.express as px
import nbformat
```

[1] Python

⌵

```
df = pd.read_csv('Malls_Customers.csv')
df.head()
```

[2] Python

...

	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

⌵

```
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
```

[3] Python

⌵

```
X = df[['Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']]
```

[4] Python

⌵

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

[5] Python

0 18 Java: Ready Spaces: 4 Cell 1 of 12 Go Live Prettier

← →

Untitled (Workspace)

⚙️ 🗑️ 📄 📄

ExpFour.ipynb ×

CL IV - BI Codes > ExpFour > ExpFour.ipynb > import pandas as pd

🔍 Generate + Code + Markdown ▶ Run All 🧹 Clear All Outputs | 📄 Outline ...

Select Kernel

⌵

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
```

[6] Python

⌵

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

[7] Python

...

Elbow Method

Number of clusters	WCSS
1	80
2	54
3	38
4	28
5	20
6	15
7	12
8	9
9	7
10	6

0 18 Java: Ready Spaces: 4 Cell 1 of 12 Go Live

