Pycaret Assignment

❖ Clustering

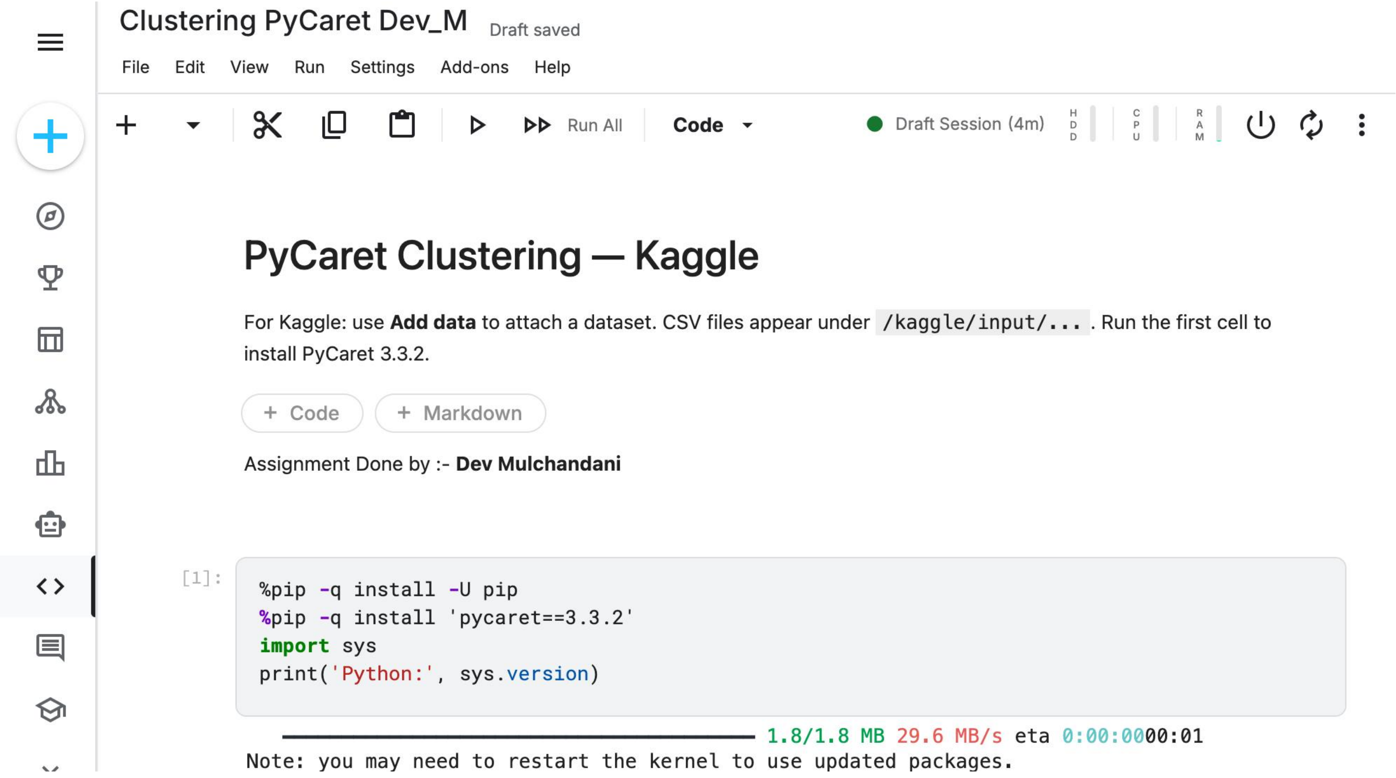
❖ Kaggle Notebook :- Link

❖ Submitted By :- Dev Mulchandani

❖ Overview :-

In my clustering notebook, I used PyCaret's clustering module to group similar data points without using labels. After loading and preprocessing the dataset, I initialized the environment using the setup() function, which automatically handled scaling and normalization. Then I created a K-Means model with create_model('kmeans') to identify natural clusters within the data. I assigned cluster labels using assign_model() and visualized the results with a 2-D cluster plot to see how the data was grouped. Finally, I saved both the trained model and the labeled dataset for future analysis. This workflow demonstrated how clustering can reveal hidden patterns and group structures in data with just a few lines of code using PyCaret.

❖ Screenshots :-



PyCaret Clustering — Kaggle

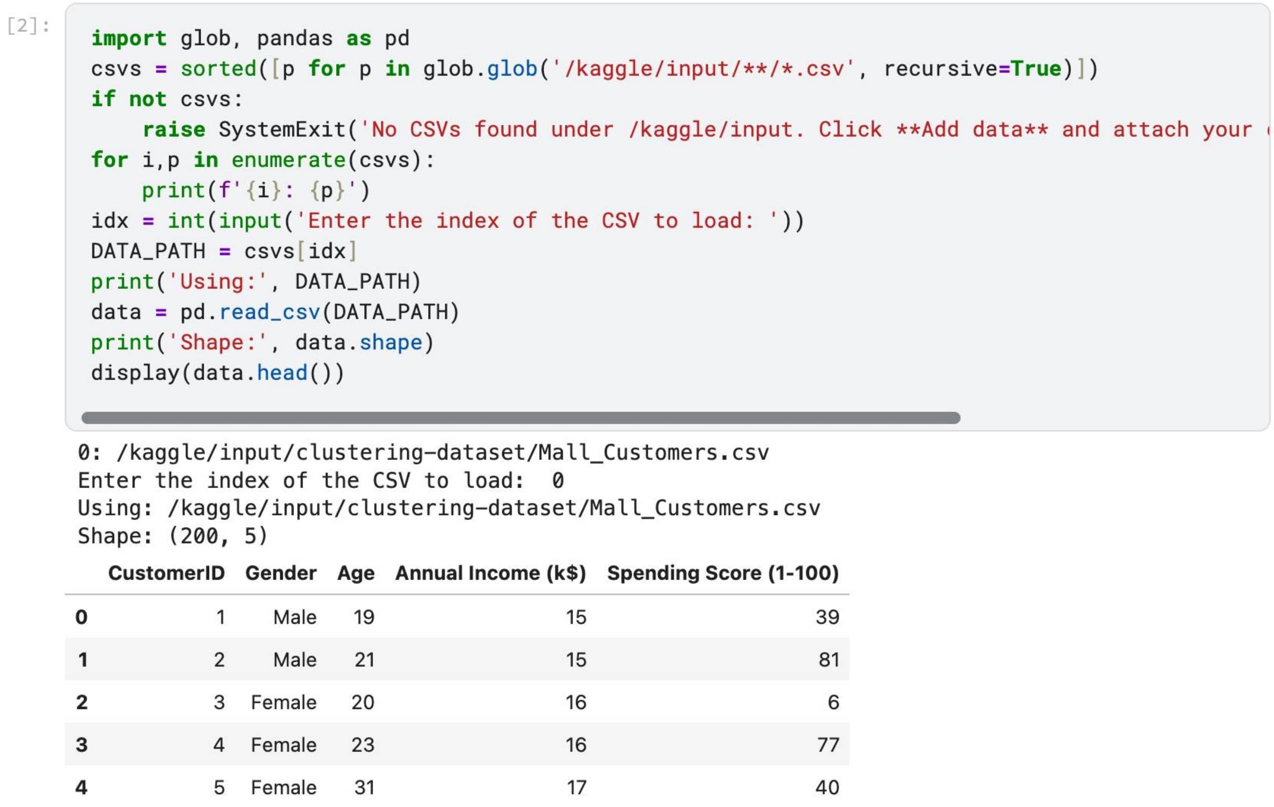
For Kaggle: use **Add data** to attach a dataset. CSV files appear under `/kaggle/input/...`. Run the first cell to install PyCaret 3.3.2.

Assignment Done by :- **Dev Mulchandani**

[1]:

```
%pip -q install -U pip
%pip -q install 'pycaret==3.3.2'
import sys
print('Python:', sys.version)
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 29.6 MB/s eta 0:00:0000:01
Note: you may need to restart the kernel to use updated packages.
```

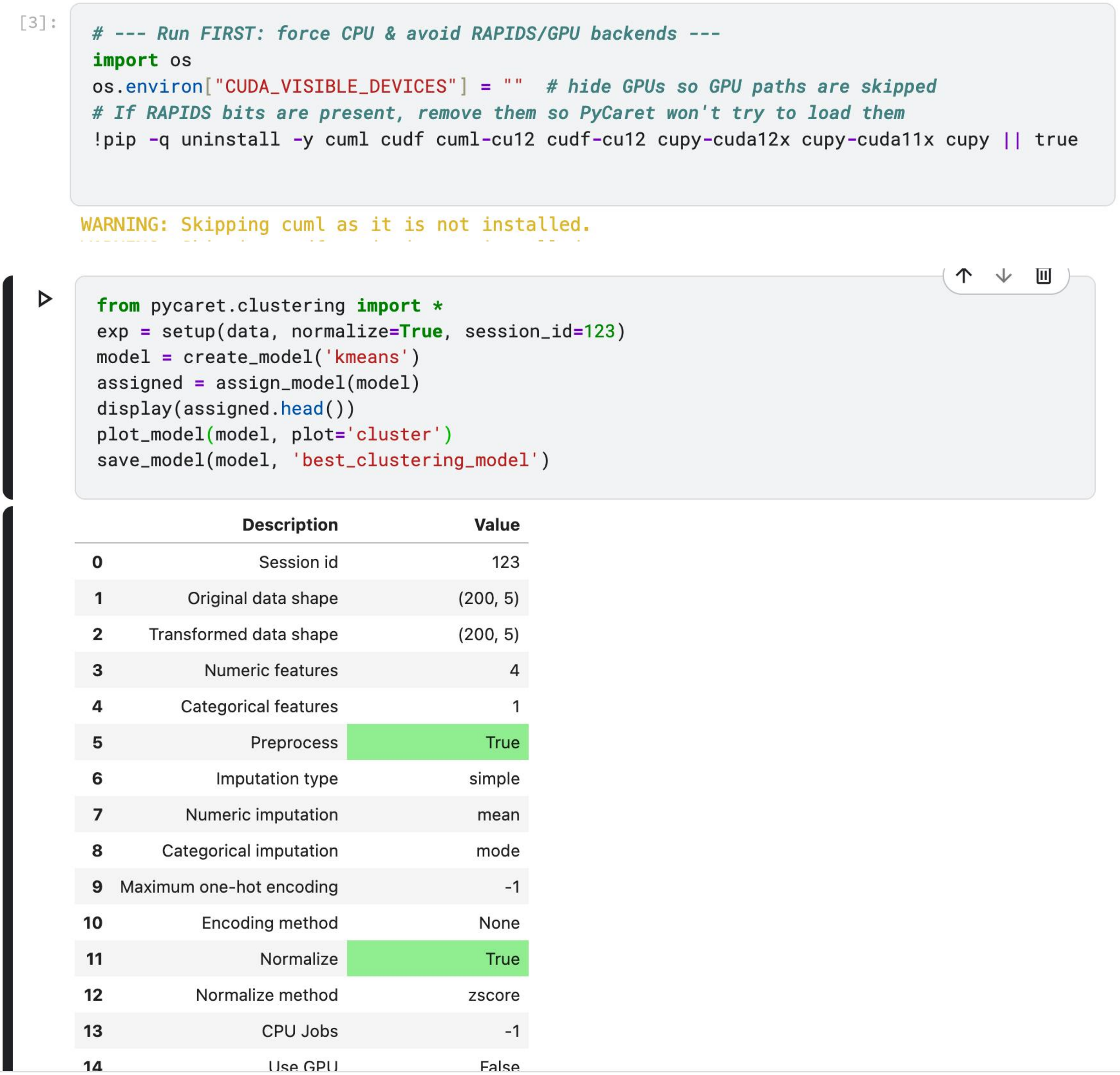


[2]:

```
import glob, pandas as pd
csvs = sorted([p for p in glob.glob('/kaggle/input/**/*.csv', recursive=True)])
if not csvs:
    raise SystemExit('No CSVs found under /kaggle/input. Click **Add data** and attach your
for i,p in enumerate(csvs):
    print(f'{i}: {p}')
idx = int(input('Enter the index of the CSV to load: '))
DATA_PATH = csvs[idx]
print('Using:', DATA_PATH)
data = pd.read_csv(DATA_PATH)
print('Shape:', data.shape)
display(data.head())
```

```
0: /kaggle/input/clustering-dataset/Mall_Customers.csv
Enter the index of the CSV to load:  0
Using: /kaggle/input/clustering-dataset/Mall_Customers.csv
Shape: (200, 5)
```

	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



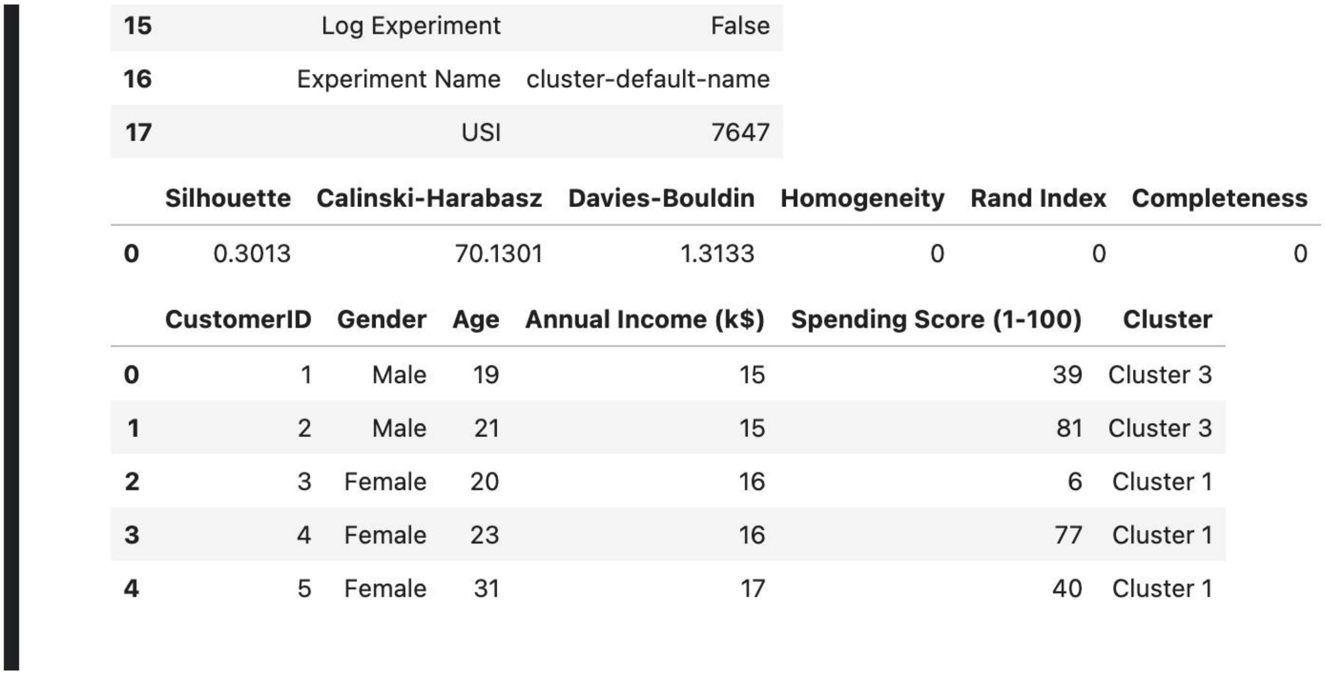
[3]:

```
# --- Run FIRST: force CPU & avoid RAPIDS/GPU backends ---
import os
os.environ["CUDA_VISIBLE_DEVICES"] = ""  # hide GPUs so GPU paths are skipped
# If RAPIDS bits are present, remove them so PyCaret won't try to load them
!pip -q uninstall -y cuml cudf cuml-cu12 cudf-cu12 cupy-cuda12x cupy-cuda11x cupy || true
```

```
WARNING: Skipping cuml as it is not installed.
```

```
from pycaret.clustering import *
exp = setup(data, normalize=True, session_id=123)
model = create_model('kmeans')
assigned = assign_model(model)
display(assigned.head())
plot_model(model, plot='cluster')
save_model(model, 'best_clustering_model')
```

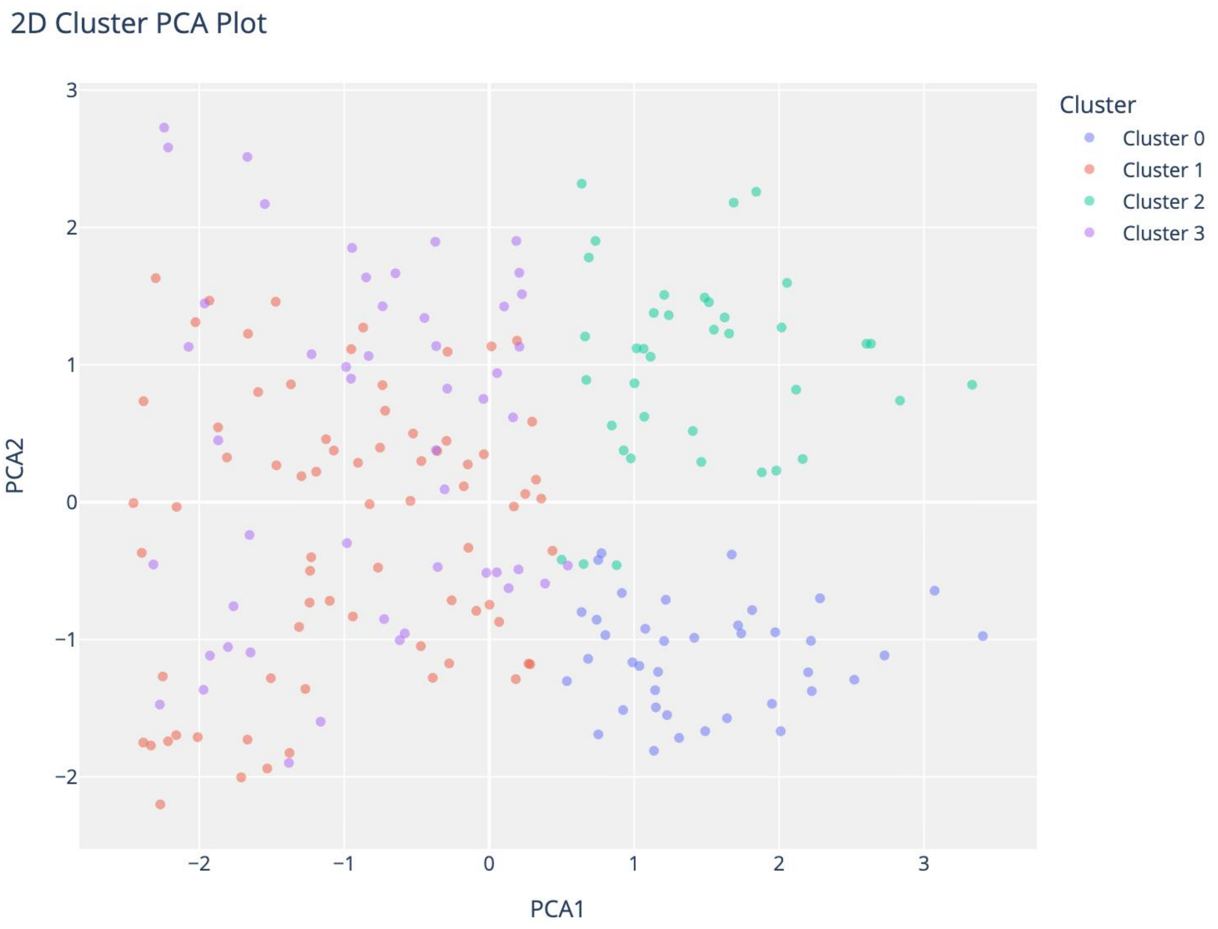
	Description	Value
0	Session id	123
1	Original data shape	(200, 5)
2	Transformed data shape	(200, 5)
3	Numeric features	4
4	Categorical features	1
5	Preprocess	True
6	Imputation type	simple
7	Numeric imputation	mean
8	Categorical imputation	mode
9	Maximum one-hot encoding	-1
10	Encoding method	None
11	Normalize	True
12	Normalize method	zscore
13	CPU Jobs	-1
14	Use GPU	False



15	Log Experiment	False
16	Experiment Name	cluster-default-name
17	USI	7647

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3013	70.1301	1.3133	0	0	0

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


2D Cluster PCA Plot
Cluster
Cluster 0
Cluster 1
Cluster 2
Cluster 3
PCA2
3
2
1
0
−1
−2
PCA1
−2
−1
0
1
2
3