In [30]:
```
import numpy as np
import pandas as pd
import random
random.seed(42)
np.random.seed(42)
```

In [31]:
```
df = pd.read_csv("Credit_card/GENERAL.csv")
```

In [32]:
```
df.head()
```

Out[32]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTA
0	C10001	40.900749	0.818182	95.40	0.00	
1	C10002	3202.467416	0.909091	0.00	0.00	
2	C10003	2495.148862	1.000000	773.17	773.17	
3	C10004	1666.670542	0.636364	1499.00	1499.00	
4	C10005	817.714335	1.000000	16.00	16.00	

In [33]:
```
df1 = df.drop("CUST_ID", axis=1)
```

In [34]:
```
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

In [35]:
```
num_cols = df1.columns.tolist()
```

In [36]:
```
num_pipe = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler",  StandardScaler())
])

preprocessor = ColumnTransformer(transformers=[
    ("num", num_pipe, num_cols)
],remainder="drop")


X = preprocessor.fit_transform(df1)

results = []

for k in range(2, 11):
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = km.fit_predict(X)
    sil = silhouette_score(X, labels)
```

```
    results.append((k, sil))
    print(f"k={k:2d}  silhouette={sil:.4f}")

# pick best k by silhouette
best_k = max(results, key=lambda t: t[1])[0]
print("Best k by silhouette:", best_k)
```

```
k= 2  silhouette=0.2100
k= 3  silhouette=0.2510
k= 4  silhouette=0.1977
k= 5  silhouette=0.1931
k= 6  silhouette=0.2029
k= 7  silhouette=0.2077
k= 8  silhouette=0.2217
k= 9  silhouette=0.2260
k=10  silhouette=0.2204
Best k by silhouette: 3
```

In [37]:
```
# 6) Final pipeline (preprocess + model)
final_model = Pipeline([
    ("preprocess", preprocessor),
    ("kmeans", KMeans(n_clusters=best_k, random_state=42, n_init=10))
])

clusters = final_model.fit_predict(df)
df1["Cluster"] = clusters

print(df1["Cluster"].value_counts().sort_index())
```

```
Cluster
0    1275
1    6114
2    1561
Name: count, dtype: int64
```

In [38]:
```
submission = pd.DataFrame({
    "CustomerID": df["CUST_ID"],   # or original ID if you kept it
    "Cluster": df1["Cluster"]
})

submission.to_csv("submission.csv", index=False)
```