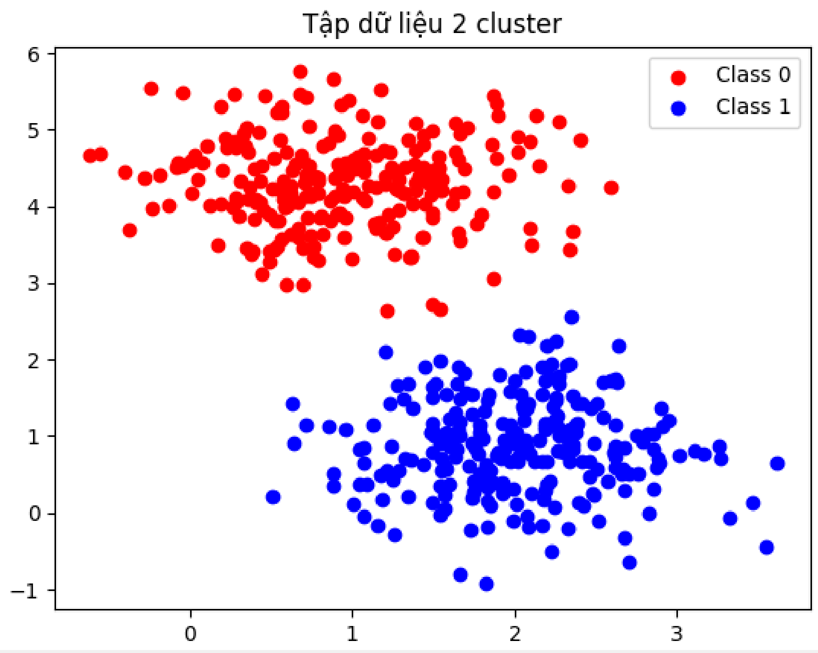
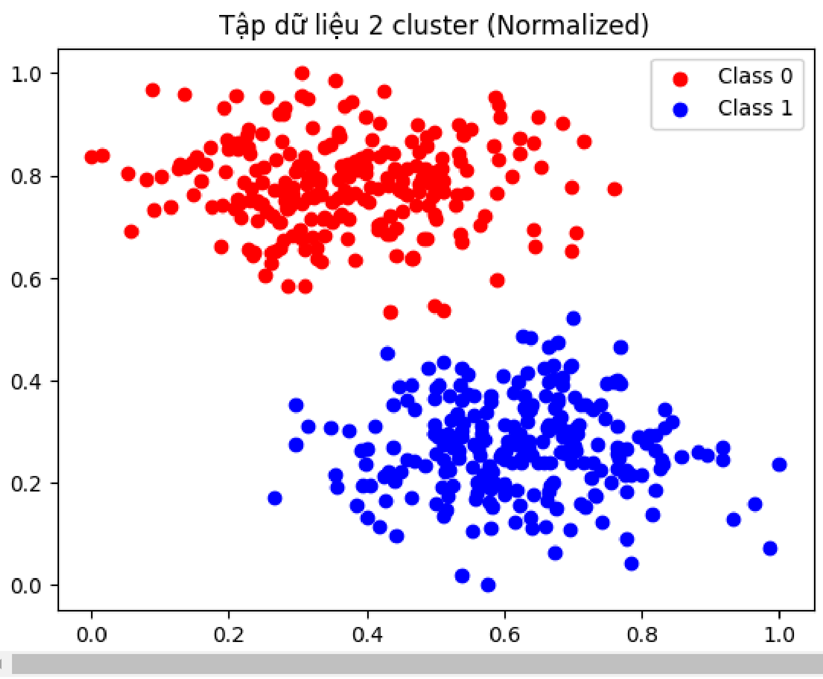
Bài 1

```
from sklearn.datasets import make_blobs
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
```

```
X, y = make_blobs(n_samples=500, centers=2, cluster_std=0.6, random_state=0)

colors = {0: 'r', 1: 'b'}

for i in np.unique(y):
    plt.scatter(X[y == i, 0], X[y == i, 1], label=f'Class {i}', c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster')
plt.show()
```



```
X_min = np.min(X, axis=0)
X_max = np.max(X, axis=0)
X_normalized = (X - X_min) / (X_max - X_min)

colors = {0: 'r', 1: 'b'}

for i in np.unique(y):
    plt.scatter(X_normalized[y == i, 0], X_normalized[y == i, 1], label=f'Class {i}', c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster (Normalized)')
plt.show()
```

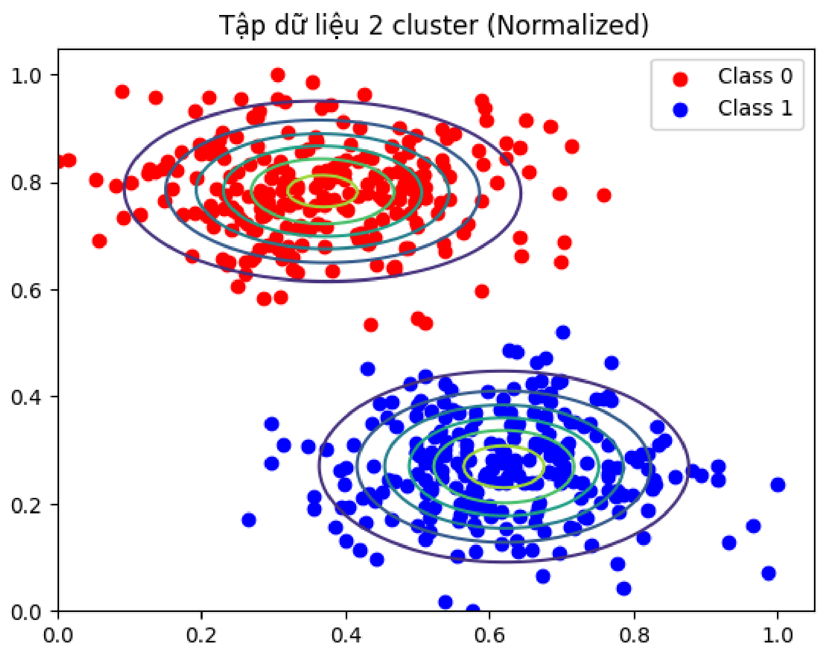


```
means = []
covariances = []

for class_value in np.unique(y):
    class_point = X_normalized[y == class_value, :]
    means.append(np.mean(class_point, axis=0))
    covariances.append(np.cov(class_point.T))

    print(f'Class {class_value}')
    print(f'Mean: {means[-1]}')
    print(f'Covariance: {covariances[-1]}')
```

```
Class 0
Mean: [0.3687902  0.78255391]
Covariance: [[ 0.0205023  -0.00031041]
 [-0.00031041  0.00763981]]
Class 1
Mean: [0.620373  0.2692462]
Covariance: [[ 1.74447092e-02 -9.11635959e-05]
 [-9.11635959e-05  8.40133638e-03]]
```

```
g1 = multivariate_normal(mean=means[0], cov=covariances[0])
g2 = multivariate_normal(mean=means[1], cov=covariances[1])
```

```
a, b = np.mgrid[0:1:0.01, 0:1:0.01]
pos = np.dstack((a, b))

plt.contour(a, b, g1.pdf(pos))
plt.contour(a, b, g2.pdf(pos))

for i in np.unique(y):
    plt.scatter(X_normalized[y == i, 0], X_normalized[y == i, 1], label=f'Class {i}', c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster (Normalized)')
plt.show()
```



Bài 2

```
df = pd.read_csv('/content/Social_Network_Ads.csv')
df.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

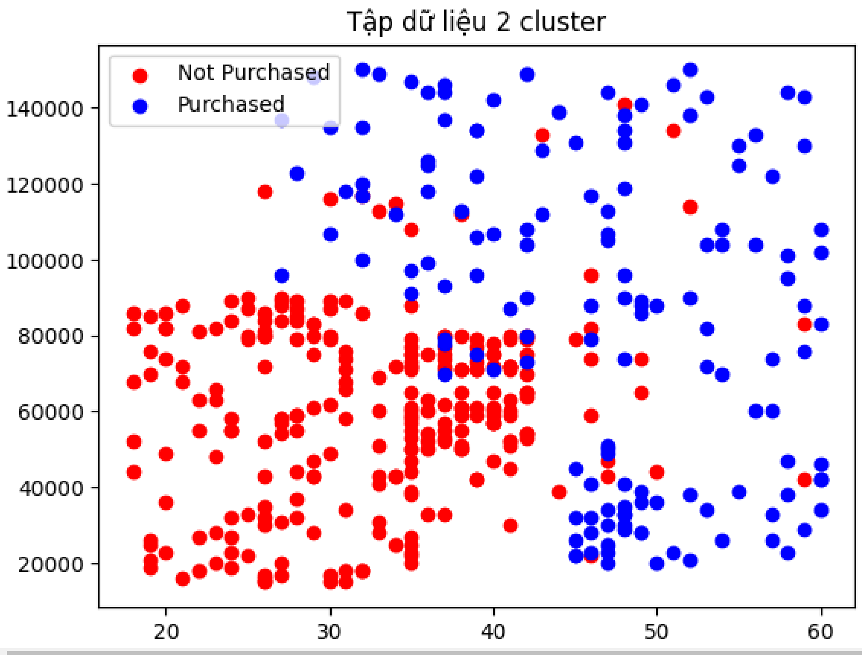
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.info()
```

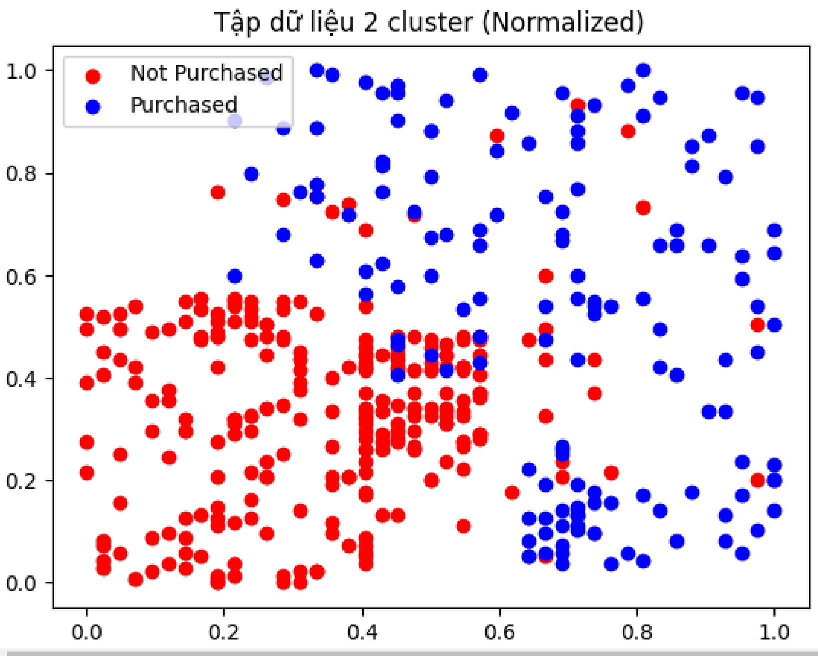
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   User ID          400 non-null    int64
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64
 3   EstimatedSalary  400 non-null    int64
 4   Purchased        400 non-null    int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

```
X = df.iloc[:, [2, 3]].values
y = df.iloc[:, -1].values
```

```
colors = {0: 'r', 1: 'b'}
labels = {0: 'Not Purchased', 1: 'Purchased'}

for i in np.unique(y):
    plt.scatter(X[y == i, 0], X[y == i, 1], label=labels[i], c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster')
plt.show()
```



```
X_min = np.min(X, axis=0)
X_max = np.max(X, axis=0)
X_normalized = (X - X_min) / (X_max - X_min)

for i in np.unique(y):
    plt.scatter(X_normalized[y == i, 0], X_normalized[y == i, 1], label=labels[i], c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster (Normalized)')
plt.show()
```

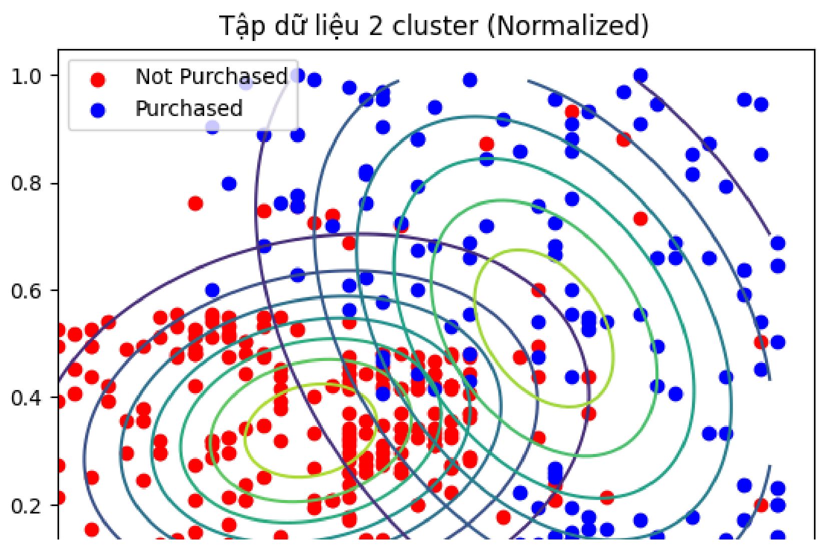


```
means = []
covariances = []

for class_value in np.unique(y):
    class_point = X_normalized[y == class_value, :]
    means.append(np.mean(class_point, axis=0))
    covariances.append(np.cov(class_point.T))

    print(f'Class {class_value}')
    print(f'Mean: {means[-1]}')
    print(f'Covariance: {covariances[-1]}')
```

```
Class 0
Mean: [0.35223272 0.3373685 ]
Covariance: [[0.0361529  0.00618653]
 [0.00618653 0.03253767]]
Class 1
Mean: [0.67599068 0.52794613]
Covariance: [[ 0.04204621 -0.0235776 ]
 [-0.0235776   0.09708625]]
```

```
g1 = multivariate_normal(mean=means[0], cov=covariances[0])
g2 = multivariate_normal(mean=means[1], cov=covariances[1])
```

```
a, b = np.mgrid[0:1:0.01, 0:1:0.01]
pos = np.dstack((a, b))

plt.contour(a, b, g1.pdf(pos))
plt.contour(a, b, g2.pdf(pos))

for i in np.unique(y):
    plt.scatter(X_normalized[y == i, 0], X_normalized[y == i, 1], label=labels[i], c=colors[i])
plt.legend()
plt.title('Tập dữ liệu 2 cluster (Normalized)')
plt.show()
```


Tập dữ liệu 2 cluster (Normalized)
Not Purchased
Purchased