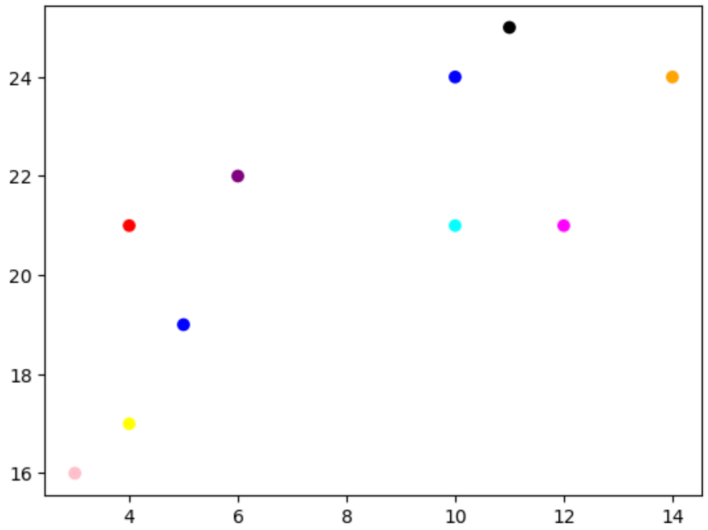
BÀI TẬP THỰC HÀNH
MÔN TRÍ TUỆ NHÂN TẠO
BÀI TOÁN GOM CỤM VỚI K-MEANS

Bài tập 1: Cho tập hợp các điểm có tọa độ sau:

(4, 21), (5, 19), (10, 24), (4, 17), (3, 16), (11, 25), (14, 24), (6, 22), (10, 21), (12, 21)

a. Hiển thị lên đồ thị với tập hợp màu như sau:

color = np.array(["red", "blue","blue", "yellow", "pink", "black", "orange", "purple", "cyan", "magenta"])



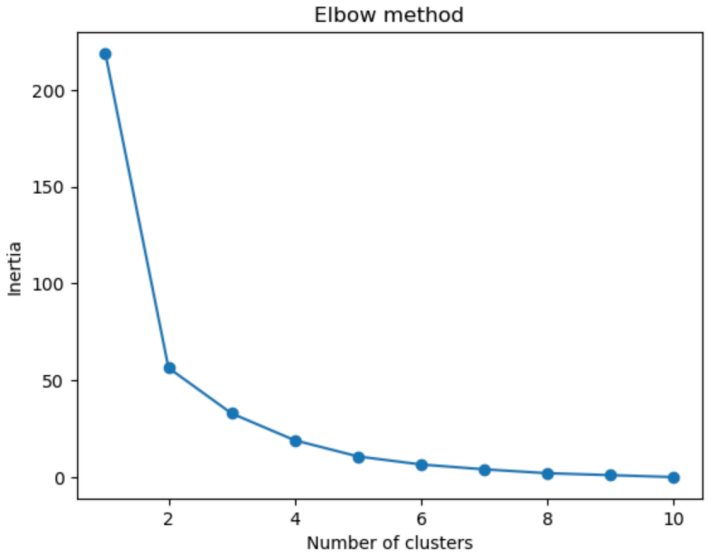
b. Dùng phương pháp Elbow tính số cụm K-means:

```
from sklearn.cluster import KMeans
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

Kết quả:

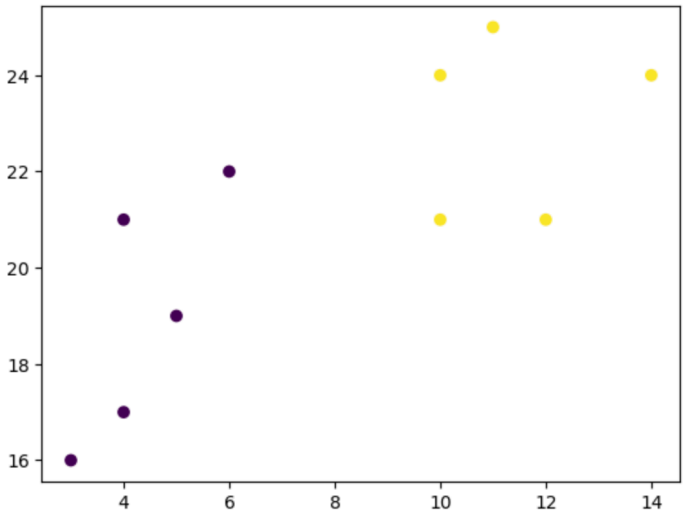


Chọn số phân cụm là 2:

```
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels_)
plt.show()
```

Kết quả:



c. **Tính các center point của từng cụm:**

```
print('Centers found by scikit-learn:')
print(kmeans.cluster_centers_)
```

Kiểm chứng kết quả:

```
Centers found by scikit-learn:
[[ 4.4 19. ]
 [11.4 23. ]]
```

Bài tập 2: Cho dữ liệu 'housing.csv' như sau:

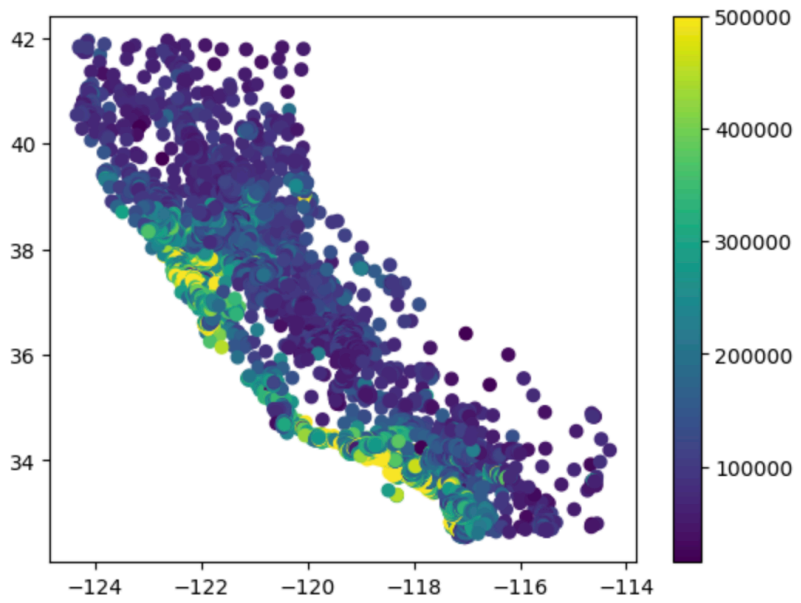
housing

	ongitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
2	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
3	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
4	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
5	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
6	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY
7	-122.25	37.85	52.0	919.0	213.0	413.0	193.0	4.0368	269700.0	NEAR BAY
8	-122.25	37.84	52.0	2535.0	489.0	1094.0	514.0	3.6591	299200.0	NEAR BAY
9	-122.25	37.84	52.0	3104.0	687.0	1157.0	647.0	3.12	241400.0	NEAR BAY
10	-122.26	37.84	42.0	2555.0	665.0	1206.0	595.0	2.0804	226700.0	NEAR BAY
11	-122.25	37.84	52.0	3549.0	707.0	1551.0	714.0	3.6912	261100.0	NEAR BAY
12	-122.26	37.85	52.0	2202.0	434.0	910.0	402.0	3.2031	281500.0	NEAR BAY
13	-122.26	37.85	52.0	3503.0	752.0	1504.0	734.0	3.2705	241800.0	NEAR BAY
14	-122.26	37.85	52.0	2491.0	474.0	1098.0	468.0	3.075	213500.0	NEAR BAY
15	-122.26	37.84	52.0	696.0	191.0	345.0	174.0	2.6736	191300.0	NEAR BAY
16	-122.26	37.85	52.0	2643.0	626.0	1212.0	620.0	1.9167	159200.0	NEAR BAY
17	-122.26	37.85	50.0	1120.0	283.0	697.0	264.0	2.125	140000.0	NEAR BAY
18	-122.27	37.85	52.0	1966.0	347.0	793.0	331.0	2.775	152500.0	NEAR BAY

a. **Tiến hành đọc housing.csv vào DataFrame df**
b. **Hiển thị df**
c. **Hiển thị tên các columns trong df**
d. **Hiển thị số records dữ liệu trong df**
e. **Dựa trên 'longitude' và 'latitude' để gom cụm các giá nhà 'median_house_value', tiến hành dùng matplotlib để hiển thị với x = 'longitude', y = 'latitude' và màu sắc sẽ phụ thuộc vào 'median_house_value':**

```
import numpy as np
x = df['longitude'].to_numpy()
y = df['latitude'].to_numpy()
color = df['median_house_value'].to_numpy()
plt.scatter(x, y, c=color)
plt.colorbar()
plt.show()
```

Kết quả thu được:

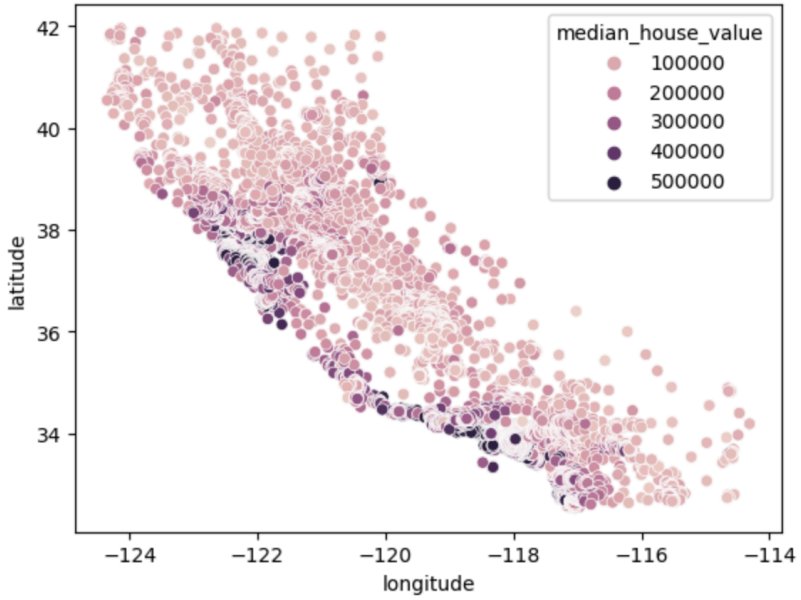


Thư viện seaborn để hiển thị giá trị:

```
import seaborn as sns

sns.scatterplot(data = df, x = 'longitude', y = 'latitude', hue = 'median_house_value')
```

Kết quả thu được:



f. K-means trong scikitlearn

Chuẩn bị chia dữ liệu thành tỉ lệ train-test: 70%-30%

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df[['latitude', 'longitude']], df[['median_house_value']],
                                                    test_size=0.3, random_state=0)
print(X_train.shape)
print(X_test.shape)
print(df.shape)
```

Tiền xử lý dữ liệu:

```
from sklearn import preprocessing

X_train_norm = preprocessing.normalize(X_train)
X_test_norm = preprocessing.normalize(X_test)
print(X_train_norm.shape)
print(X_test_norm.shape)
```

Quan sát kết quả trước và sau tiền xử lý.

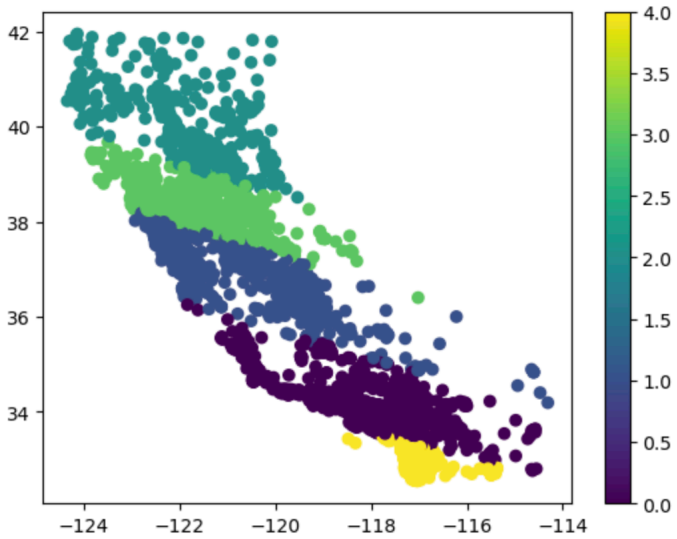
Tiến hành huấn luyện bằng K-Means với số cụm = 5.

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters = 5, random_state = 0, n_init='auto')
kmeans.fit(X_train_norm)
```

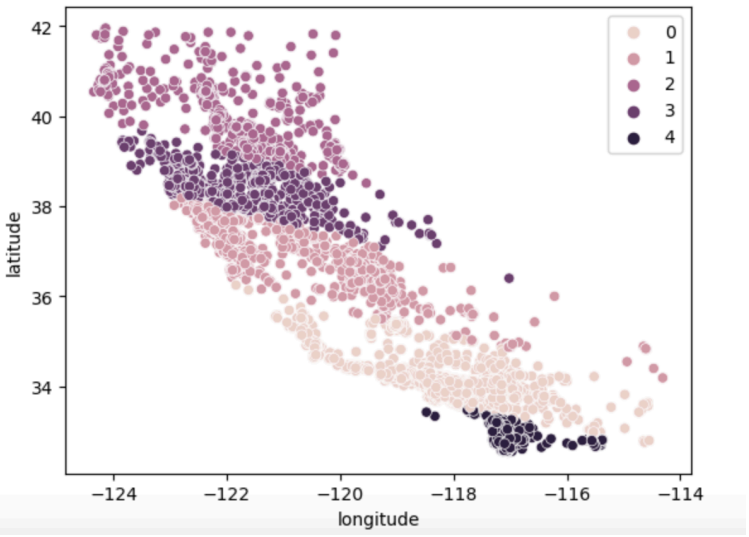
Hiển thị kết quả huấn luyện:

```
plt.scatter(X_train['longitude'].to_numpy(), X_train['latitude'].to_numpy(), c=kmeans.labels_)
plt.colorbar()
plt.show()
```



Hoặc bằng seaborn:

```
sns.scatterplot(data = X_train, x = 'longitude', y = 'latitude', hue = kmeans.labels_)
```



g. Tiến hành kiểm thử:

```
y_pred = kmeans.predict(X_test_norm)
sns.scatterplot(data = X_test, x = 'longitude', y = 'latitude', hue = y_pred)
```

Hiển thị kết quả phân cụm với tập kiểm thử bằng matplotlib:

```
plt.scatter(X_test['longitude'].to_numpy(), X_test['latitude'].to_numpy(), c=y_pred)
plt.colorbar()
plt.show()
```

Bằng seaborn:

```
y_pred = kmeans.predict(X_test_norm)
sns.scatterplot(data = X_test, x = 'longitude', y = 'latitude', hue = y_pred)
```

Quan sát kết quả kiểm tra với cụm phân chia.

h. Xuất kết quả giá trị trung tâm mà K-Means tính được:

```
print('Centers found by scikit-learn:')
print(kmeans.cluster_centers_)
```

Kết quả tính:

```
Centers found by scikit-learn:
[[ 0.27698887 -0.96087076]
 [ 0.29345769 -0.95596888]
 [ 0.31133094 -0.95029171]
 [ 0.30047293 -0.95378665]
 [ 0.27026045 -0.96278599]]
```

Bài tập 3: Với dataset ở bài tập 2, hãy tiến hành chia lại tập train-test theo tỉ lệ 80-20 và đảm bảo dữ liệu các class đều có trong tỉ lệ train-test này. Sau đó so sánh lại với kết quả bài 2 và kết quả mô phỏng giải thuật trên giấy.

--Hết--