CLUSTERING STATES INTO DIFFERENT COVID-19 ZONES

Hoàng-Nguyên Vũ

1. Mô tả

Bài tập thực hành KMeans nhằm phân vùng bệnh Covid-19 trong bài toán thực tế về dịch bệnh Sar-Cov2 năm 2021 là một ứng dụng quan trọng trong phân tích dữ liệu y tế. Mục tiêu của bài tập là sử dụng thuật toán KMeans để phân nhóm các khu vực hoặc quốc gia dựa trên các chỉ số quan trọng như số ca nhiễm, số ca hồi phục và số ca tử vong. Qua việc phân cụm, ta có thể xác định các khu vực có đặc điểm dịch tễ học tương đồng, từ đó hỗ trợ các nhà quản lý y tế đưa ra quyết định hiệu quả hơn trong việc phân bổ nguồn lực và triển khai các biện pháp phòng chống dịch phù hợp. Đây là một ví dụ điển hình về cách sử dụng machine learning để giải quyết các vấn đề thực tiễn trong y tế cộng đồng.

2. Nội dung

Chúng ta sẽ sử dụng thuật toán K-Means, đây là một thuật toán phân cụm không giám sát. Thuật toán k-means là một thuật toán lặp đi lặp lại, cố gắng chia tập dữ liệu thành K cụm (nhóm con) riêng biệt không chồng chéo, trong đó mỗi điểm dữ liệu chỉ thuộc về một cụm. Thuật toán cố gắng làm cho các điểm dữ liệu trong cùng một cụm càng giống nhau càng tốt, đồng thời giữ cho các cụm càng xa nhau càng tốt. Nó gán các điểm dữ liệu vào một cụm sao cho tổng khoảng cách bình phương giữa các điểm dữ liệu và tâm cụm là nhỏ nhất.

Bước 1: Load dataset: Tải tại đây

```python
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt

df_india = # Your coder here to load dataset#
print(df_india)
```

	State	Confirmed	Recovered	Deaths
0	Maharashtra	41642	11726	1454
1	Tamil Nadu	13967	6282	95
2	Gujarat	12910	5488	773
3	Delhi	11659	5567	194
4	Rajasthan	6227	3485	151
5	Madhya Pradesh	5981	2844	271

Bước 2: Thực hiện tính tỉ lệ hồi phục và tử vòng cho mỗi khu vực:

Chúng ta sẽ thực hiện tính tỉ lệ hồi phục và tử vong từ tập dữ liệu trên theo công thức sau:

$$\text{Tỉ lệ hồi phục} = \frac{\text{Hồi phục}}{\text{Số ca xác nhận}} * 100\% \tag{1}$$

$$\text{Tỉ lệ tử vong} = \frac{\text{Tử vong}}{\text{Số ca xác nhận}} * 100\% \tag{2}$$

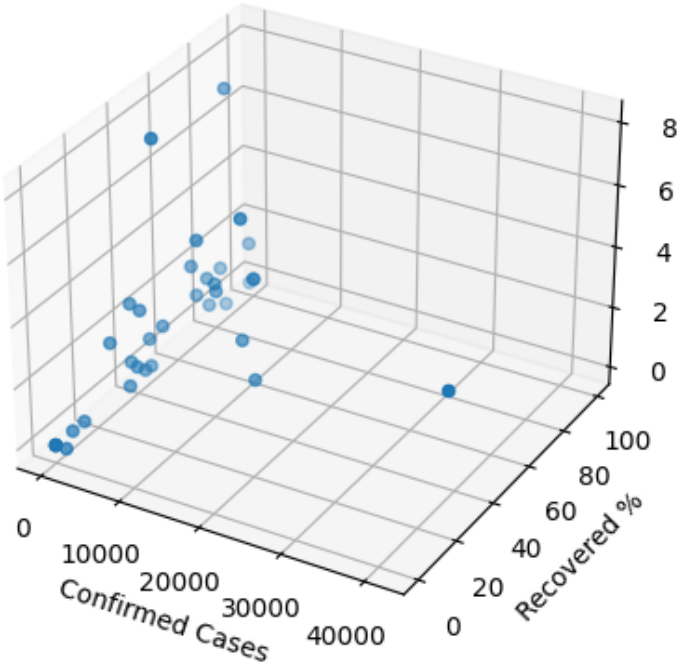
```
# Calculate the 'Recovered' and 'Deaths' percentages, ensuring 'Confirmed' is not zero
df_india['Recovered'] = # Your code here #
df_india['Deaths'] = # Your code here #
print(df_india)
```

	State	Confirmed	Recovered	Deaths
0	Maharashtra	41642	28.159070	3.491667
1	Tamil Nadu	13967	44.977447	0.680175
2	Gujarat	12910	42.509682	5.987607
3	Delhi	11659	47.748520	1.663951
4	Rajasthan	6227	55.965955	2.424924
5	Madhya Pradesh	5981	47.550577	4.531015

Hình 1: Kết quả sau khi tính tỉ lệ

Bước 3: Trực quan hóa dữ liệu trước khi áp dụng K-Means:

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df_india['Confirmed'], df_india['Recovered'], df_india['Deaths'])
ax.set_xlabel('Confirmed Cases')
ax.set_ylabel('Recovered %')
ax.set_zlabel('Deaths %')

plt.show()
```

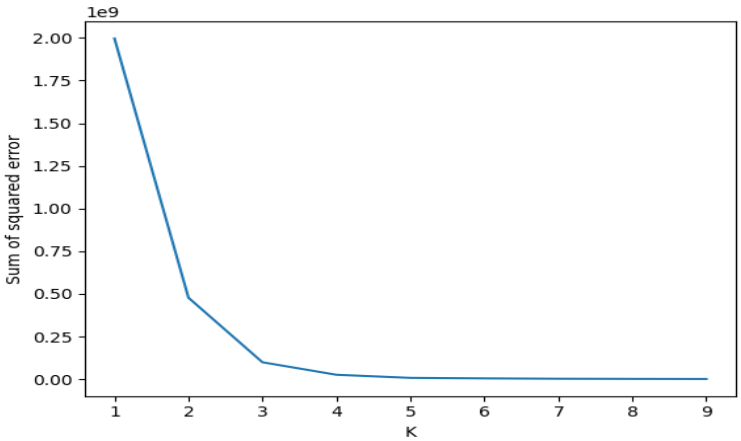


Hình 2: Phân bố data covid trước khi phân cụm

Bước 4: Cài đặt K-Means:

Chúng ta sẽ áp dụng mô hình K-Means và áp dụng phương pháp Elbow để chọn K tốt nhất trong tập dữ liệu này:

```
sse = []
k_rng = range(1,10)
# Your code here #
```



Hình 3: Kết quả Elbow - Qua đồ thị này dễ thấy K = 4 là kết quả tối ưu

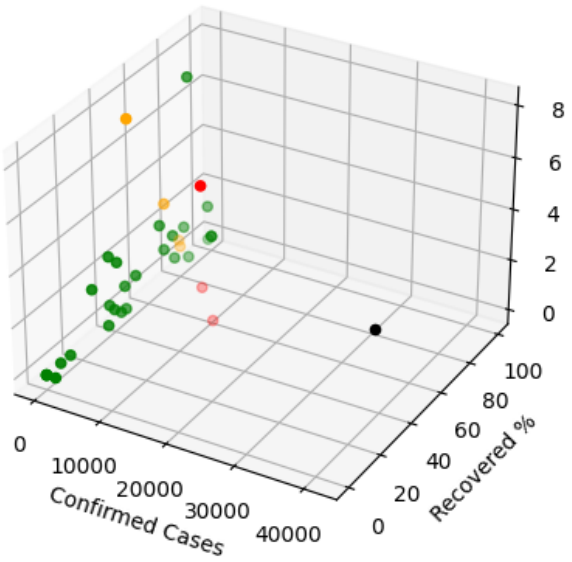
Bước 5: Mapping kết quả cluster vào dữ liệu để trực quan hóa trên bản đồ thế giới:

Chúng ta sẽ mapping kết quả cluster trên với K = 4 vào dữ liệu với cột mới có tên là **Zone**, giá trị cột này là giá trị clustering từ mô hình K-Mean với K = 4

```
# Your code here #
print(df_india)
```

	State	Confirmed	Recovered	Death	Zone
0	Maharashtra	41642	28.159070	3.491667	3
1	Tamil Nadu	13967	44.977447	0.680175	2
2	Gujarat	12910	42.509682	5.987607	2
3	Delhi	11659	47.748520	1.663951	2
4	Rajasthan	6227	55.965955	2.424924	1
5	Madhya Pradesh	5981	47.550577	4.531015	1

Hình 4: Kết quả mapping zone



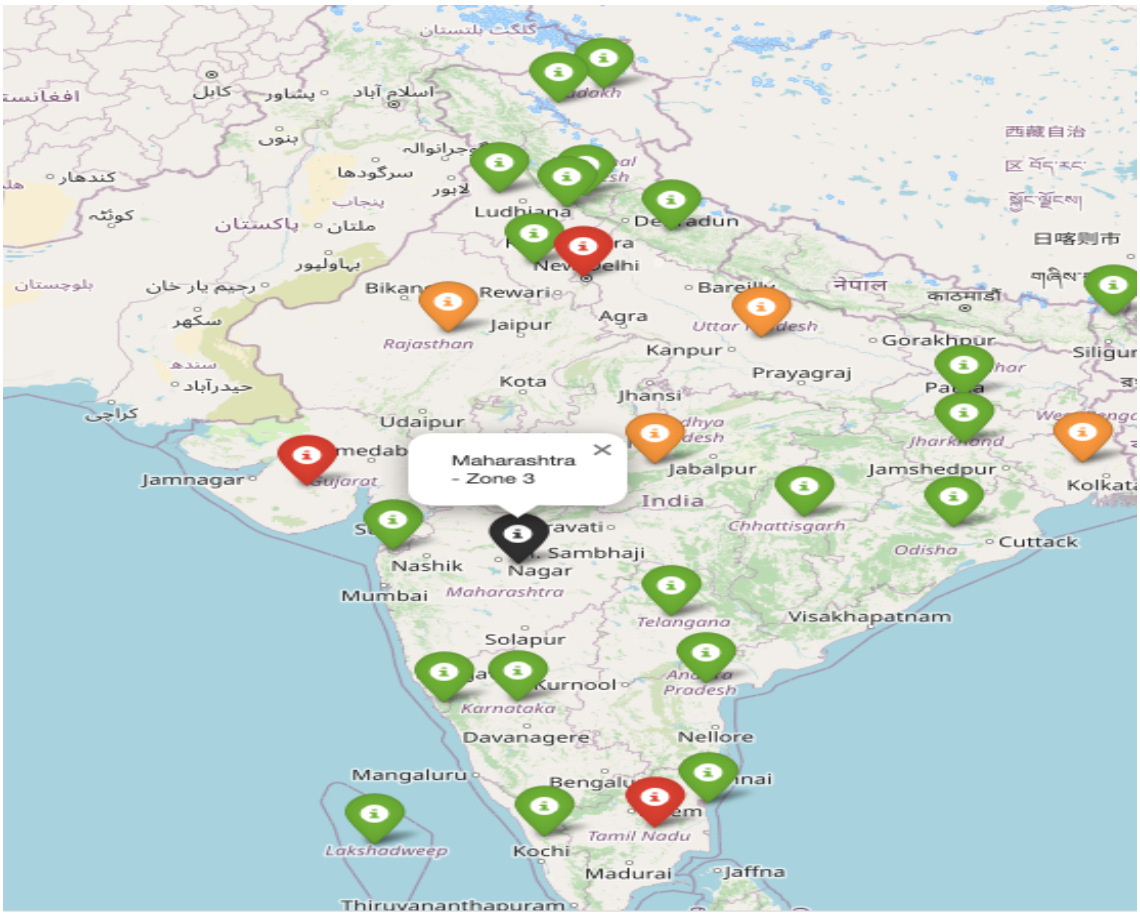
Hình 5: Kết quả phân bố sau khi thực hiện phân cụm

Bước 6: Trực quan hóa phân cụm trên bản đồ thế giới:

Để thực hiện trực quan hóa trên bản đồ thế giới, chúng ta cần cài đặt thêm thư viện Folium để giúp vẽ bản đồ thế giới. Cũng như các có dữ liệu kinh tuyến và vĩ tuyến của các bang của Ấn Độ. Trong project này, chúng ta đã được cung cấp dữ liệu các tọa độ của các bang, tuy nhiên đối với bài toán thực tế các bạn có thể lấy dữ liệu tọa độ theo dữ liệu GeoJSON Tại đây. Chúng ta sẽ thực hiện trực quan hóa toàn bộ phân cụm lên bản đồ thế giới với Folium như sau:

```
!pip install folium
import folium

# Assuming you have a dataframe called df_india with 'State' and 'Zone' columns
# and a dictionary called state_coords with state names as keys and latitude, longitude tuples as values.

# Example state_coords dictionary (replace with your actual data)
state_coords = {
    "Andhra Pradesh": (15.9129, 79.7399),
    "Arunachal Pradesh": (28.2180, 94.7278),
    "Assam": (26.2006, 92.9376),
    "Bihar": (25.0961, 85.3131),
    "Chhattisgarh": (21.2787, 81.8661),
    "Goa": (15.2993, 74.1240),
    "Gujarat": (22.2587, 71.1924),
    "Haryana": (29.0588, 76.0856),
    "Himachal Pradesh": (31.1048, 77.1734),
    "Jharkhand": (23.6102, 85.2799),
    "Karnataka": (15.3173, 75.7139),
    "Kerala": (10.8505, 76.2711),
    "Madhya Pradesh": (22.9734, 78.6569),
    "Maharashtra": (19.7515, 75.7139),
    "Manipur": (24.6637, 93.9063),
    "Meghalaya": (25.4670, 91.3662),
    "Mizoram": (23.1645, 92.9376),
    "Nagaland": (26.1584, 94.5624),
    "Odisha": (20.9517, 85.0985),
    "Punjab": (31.1471, 75.3412),
    "Rajasthan": (27.0238, 74.2179),
    "Sikkim": (27.5330, 88.5122),
    "Tamil Nadu": (11.1271, 78.6569),
    "Telangana": (18.1124, 79.0193),
    "Tripura": (23.9408, 91.9882),
    "Uttar Pradesh": (26.8467, 80.9462),
    "Uttarakhand": (30.0668, 79.0193),
    "West Bengal": (22.9868, 87.8550),
    "Andaman and Nicobar Islands": (11.7401, 92.6586),
    "Chandigarh": (30.7333, 76.7794),
    "Dadra and Nagar Haveli and Daman and Diu": (20.2270, 73.0169),
    "Delhi": (28.7041, 77.1025),
    "Jammu and Kashmir": (33.7782, 76.5762),
    "Ladakh": (34.1526, 77.5806),
```

```
    "Lakshadweep": (10.5593, 72.6358),
    "Puducherry": (11.9416, 79.8083),
    "Orissa": (20.9517, 85.0985)  # Added Orissa for consistency
}


# Create a map centered on India
map_india = folium.Map(location=[20.5937, 78.9629], zoom_start=4)

# Add markers for each state with color based on cluster
for index, row in df_india.iterrows():
    state = row['State']
    zone = row['Zone']
    if state in state_coords:
        lat, lon = state_coords[state]
        if zone == 0:
            color = "green"
        elif zone == 1:
            color = "orange"
        elif zone == 2:
            color = "red"
        else:
            color = "black"
        folium.Marker(
            location=[lat, lon],
            popup=f"{state} - Zone {zone}",
            icon=folium.Icon(color=color)
        ).add_to(map_india)

# Display the map
map_india
```



Hình 6: Kết quả phân vùng các mức độ bệnh Covid ở Ấn Độ

- Hết -