

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

---□ □ □---



**BÁO CÁO MÔN HỌC
KHAI PHÁ DỮ LIỆU**

Đề tài

KHAI PHÁ DỮ LIỆU BILLIONAIRES STATISTICS 2023

Nhóm 8:

Trương Tường Vi - 2151050542 (Nhóm trưởng)

Huỳnh Nguyễn Bảo Trân - 2151010390

Huỳnh Nguyễn Bảo Châu - 2151010036

Lớp: DH21CS02

Giảng viên: Nguyễn Văn Bảy

Tháng 16 năm 2024

Mục lục

| | |
|--|-----------|
| 1. Mô tả dữ liệu | 1 |
| 2. Tiền xử lý dữ liệu | 2 |
| 3. Môi quan hệ tương quan | 11 |
| 4. Gom cụm K-means | 11 |
| a. Gom cụm K-means trên Weka | 11 |
| b. Gom cụm K-means trên Colab | 17 |
| 5. Luật kết hợp | 21 |
| 6. Thuật toán KNN & Naive Bayes | 24 |
| 7. Cây quyết định | 27 |
| 8. Trục quan hóa dữ liệu | 31 |
| 9. Nhận xét và đánh giá | 37 |
| 10. Tổng kết | 37 |

1. Mô tả dữ liệu

Dữ liệu chứa số liệu thống kê về các tỷ phú trên thế giới năm 2023, bao gồm:

1. Thứ hạng (rank)
2. Giá trị tài sản ròng của tỷ phú (finalWorth)
3. Ngành mà doanh nghiệp của tỷ phú hoạt động (category)
4. Tên đầy đủ (personName)
5. Tuổi (age)
6. Quốc gia (country)
7. Thành phố (city)
8. Nguồn thu nhập của tỷ phú (source)
9. Các ngành nghề gắn liền với lợi ích kinh doanh của tỷ phú (industries)
10. Quốc tịch (countryOfCitizenship)
11. Tên tổ chức liên kết với tỷ phú (organization)
12. Cho biết tỷ phú có phải người tự lập hay không (selfMade)
13. Trạng thái của tỷ phú (status):
 - + D: tự sáng lập doanh nghiệp
 - + U: thừa kế tài sản từ gia đình
14. Giới tính (gender)
15. Ngày sinh (birthDate)

16. Họ của tỷ phú (lastName)
17. Tên của tỷ phú (firstName)
18. Chức vụ của tỷ phú (title)
19. Ngày thu thập dữ liệu (Date)
20. Nơi cư trú của tỷ phú (state)
21. Khu vực cư trú của tỷ phú (residenceStateRegion)
22. Năm sinh (birthYear)
23. Tháng sinh (birthMonth)
24. Ngày sinh (birthDay)
25. Chỉ số giá tiêu dùng (CPI) của quốc gia tỷ phú (cpi_country)
26. Thay đổi CPI của quốc gia tỷ phú (cpi_change_country)
27. Tổng sản phẩm trong nước (GDP) của quốc gia tỷ phú (gdp_country)
28. Tuyển sinh đại học ở quốc gia tỷ phú (gross_tertiary_education_enrollment)
29. Tuyển sinh tiểu học ở quốc gia tỷ phú (gross_primary_education_enrollment_country)
30. Tuổi thọ ở quốc gia tỷ phú (life_expectancy_country)
31. Doanh thu thuế ở quốc gia của tỷ phú (tax_revenue_country_country)
32. Tổng thuế suất tại quốc gia của tỷ phú (total_tax_rate_country)
33. Dân số ở đất nước của tỷ phú (population_country)
34. Tọa độ vĩ độ của đất nước tỷ phú (latitude_country)
35. Tọa độ kinh độ của đất nước tỷ phú (longitude_country)

2. Tiền xử lý dữ liệu

Vì dữ liệu lớn, nên không thể dùng Microsoft Excel để xử lý dữ liệu. Do vậy, bài toán này sử dụng thư viện panda trong python để hỗ trợ. Cụ thể chúng em dùng google colab để giải quyết bài toán này.

- Sử dụng thư viện pandas, xác định đường dẫn đến file dữ liệu và đọc tệp vào Pandas DataFrame

```
import pandas as pd

df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/Billionaires Statistics Dataset.csv")
df.head()
```

Hình. Đường dẫn tới file và đọc tệp

- Khi chạy xong đoạn lệnh trên, ta sẽ xem được dữ liệu gồm 2640 dòng và 35 cột, lưu thông tin gồm 2500 tỷ phú trên toàn thế giới vào năm 2023.

- Dữ liệu tỷ phú của 5 dòng dữ liệu đầu trong tệp dữ liệu tỷ phú năm 2023.

| rank | finalWorth | category | personName | age | country | city | source | industries | countryOfCitizenship | organization | selfMade | status |
|------|------------|-----------------------|--------------------------|-----|---------------|--------|--------------------|-----------------------|----------------------|----------------------------------|----------|--------|
| 1 | 211000 | Fashion & Retail | Bernard Arnault & family | 74 | France | Paris | LVMH | Fashion & Retail | France | LVMH Moët Hennessy Louis Vuitton | FALSE | U |
| 2 | 180000 | Automotive | Elon Musk | 51 | United States | Austin | Tesla, SpaceX | Automotive | United States | Tesla | TRUE | D |
| 3 | 114000 | Technology | Jeff Bezos | 59 | United States | Medina | Amazon | Technology | United States | Amazon | TRUE | D |
| 4 | 107000 | Technology | Larry Ellison | 78 | United States | Lanai | Oracle | Technology | United States | Oracle | TRUE | U |
| 5 | 106000 | Finance & Investments | Warren Buffett | 92 | United States | Omaha | Berkshire Hathaway | Finance & Investments | United States | Berkshire Hathaway Inc. (CI A) | TRUE | D |

| gender | birthDate | lastName | firstName | title | date | state | residenceStateRegion | birthYear | birthMonth | birthDay | cpi_country |
|--------|----------------|----------|-----------|----------------------|---------------|------------|----------------------|-----------|------------|----------|-------------|
| M | 3/5/1949 0:00 | Arnault | Bernard | Chairman and CEO | 4/4/2023 5:01 | | | 1949 | 3 | 5 | 110.05 |
| M | 6/28/1971 0:00 | Musk | Elon | CEO | 4/4/2023 5:01 | Texas | South | 1971 | 6 | 28 | 117.24 |
| M | 1/12/1964 0:00 | Bezos | Jeff | Chairman and Founder | 4/4/2023 5:01 | Washington | West | 1964 | 1 | 12 | 117.24 |
| M | 8/17/1944 0:00 | Ellison | Larry | CTO and Founder | 4/4/2023 5:01 | Hawaii | West | 1944 | 8 | 17 | 117.24 |
| M | 8/30/1930 0:00 | Buffett | Warren | CEO | 4/4/2023 5:01 | Nebraska | Midwest | 1930 | 8 | 30 | 117.24 |

| cpi_change_country | gdp_country | gross_tertiary_education_enrollment | gross_primary_education_enrollment_country | life_expectancy_country |
|--------------------|----------------------|-------------------------------------|--|-------------------------|
| 1.1 | \$2,715,518,274,227 | 65.6 | 102.5 | 82.5 |
| 7.5 | \$21,427,700,000,000 | 88.2 | 101.8 | 78.5 |
| 7.5 | \$21,427,700,000,000 | 88.2 | 101.8 | 78.5 |
| 7.5 | \$21,427,700,000,000 | 88.2 | 101.8 | 78.5 |
| 7.5 | \$21,427,700,000,000 | 88.2 | 101.8 | 78.5 |

| tax_revenue_country_country | total_tax_rate_country | population_country | latitude_country | longitude_country |
|-----------------------------|------------------------|--------------------|------------------|-------------------|
| 24.2 | 60.7 | 67059887 | 46.227638 | 2.213749 |
| 9.6 | 36.6 | 328239523 | 37.09024 | -95.712891 |
| 9.6 | 36.6 | 328239523 | 37.09024 | -95.712891 |
| 9.6 | 36.6 | 328239523 | 37.09024 | -95.712891 |
| 9.6 | 36.6 | 328239523 | 37.09024 | -95.712891 |

Hình. Đây là 5 dữ liệu thô ban đầu, chưa được xử lý.

- Xem chiều dài và kích thước của dataframe
 - Len: 2640
 - Total 35 columns
 - Dtypes: bool(1), float64(14), int64(2), object(18)
- Xóa đi những cột không cần thiết, để dataframe gọn hơn. Cụ thể là các cột: `['category', 'countryOfCitizenship', 'organization', 'birthDate', 'title', 'date', 'state', 'residenceStateRegion', 'lastName', 'firstName', 'gross_tertiary_education_enrollment', 'gross_primary_education_enrollment_country', 'life_expectancy_country', 'latitude_country', 'longitude_country', 'status']`
- Đặt lại tên các cột để đảm bảo đầy đủ các cột và dễ nhìn các dòng dữ liệu. Tập dữ liệu mới chứa các cột, gồm: `['NO', 'RANK', 'NETWORTH', 'NAME', 'AGE', 'COUNTRY', 'CITY', 'SOURCE', 'INDUSTRY', 'SELFMADE', 'STATUS', 'GENDER', 'BIRTHYEAR', 'BIRTHMONTH', 'BIRTHDAY', 'CPI_COUNTRY', 'CPI_CHANGE_COUNTRY', 'GPD_COUNTRY', 'TAX_REVENUE_COUNTRY', 'TOTAL_TAX_RATE_COUNTRY', 'POPULATION_COUNTRY']`
- Đổi tên các cột thuộc tính thành in hoa, để dễ dàng tra cứu dữ liệu cũng như nhìn đẹp hơn cho DataFrame.

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY |
|-----|------|----------|--------------------------|------|---------------|--------|--------------------|-----------------------|----------|--------|-----------|------------|----------|-------------|
| 0 | 1 | 211000 | Bernard Arnault & family | 74.0 | France | Paris | LVMH | Fashion & Retail | False | M | 1949.0 | 3.0 | 5.0 | 110.05 |
| 1 | 2 | 180000 | Elon Musk | 51.0 | United States | Austin | Tesla, SpaceX | Automotive | True | M | 1971.0 | 6.0 | 28.0 | 117.24 |
| 2 | 3 | 114000 | Jeff Bezos | 59.0 | United States | Medina | Amazon | Technology | True | M | 1964.0 | 1.0 | 12.0 | 117.24 |
| 3 | 4 | 107000 | Larry Ellison | 78.0 | United States | Lanai | Oracle | Technology | True | M | 1944.0 | 8.0 | 17.0 | 117.24 |
| 4 | 5 | 106000 | Warren Buffett | 92.0 | United States | Omaha | Berkshire Hathaway | Finance & Investments | True | M | 1930.0 | 8.0 | 30.0 | 117.24 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Hình. Hình ảnh của một số cột sau khi được đổi tên

- Ta nhận thấy cột NAME có những tên tỷ phú bị lỗi cụ thể là có chữ '& family' ví dụ như tên tỷ phú 'Bernard Arnault & family'. Ta sẽ làm sạch lại dữ liệu cột NAME bằng cách bỏ đi phần '& family'. Sau khi bỏ, ta sẽ lưu tệp thành file csv.

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY |
|-----|------|----------|-----------------|------|---------------|--------|--------------------|-----------------------|----------|--------|-----------|------------|----------|-------------|
| 0 | 1 | 211000 | Bernard Arnault | 74.0 | France | Paris | LVMH | Fashion & Retail | False | M | 1949.0 | 3.0 | 5.0 | 110.05 |
| 1 | 2 | 180000 | Elon Musk | 51.0 | United States | Austin | Tesla, SpaceX | Automotive | True | M | 1971.0 | 6.0 | 28.0 | 117.24 |
| 2 | 3 | 114000 | Jeff Bezos | 59.0 | United States | Medina | Amazon | Technology | True | M | 1964.0 | 1.0 | 12.0 | 117.24 |
| 3 | 4 | 107000 | Larry Ellison | 78.0 | United States | Lanai | Oracle | Technology | True | M | 1944.0 | 8.0 | 17.0 | 117.24 |
| 4 | 5 | 106000 | Warren Buffett | 92.0 | United States | Omaha | Berkshire Hathaway | Finance & Investments | True | M | 1930.0 | 8.0 | 30.0 | 117.24 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Hình. Đây là dòng dữ liệu đã được làm sạch.

- Kiểm tra tính hợp lệ của cột RANK, NAME, COUNTRY và áp dụng hàm kiểm tra tính hợp lệ cho từng dòng dữ liệu và chia dữ liệu thành df_valid và df_invalid dựa trên kết quả kiểm tra.

```
# Tạo DataFrame mới lưu trữ dòng dữ liệu không hợp lệ ban đầu
df_invalid = df_combined.copy()

# Hàm kiểm tra tính hợp lệ của cột RANK, NAME, COUNTRY
def is_valid_data(row):
    if row['RANK'] == '0' or row['NAME'] == '0' or row['COUNTRY'] == '0':
        return False
    return True

# Áp dụng hàm kiểm tra tính hợp lệ cho từng dòng dữ liệu
mask = df_combined.apply(lambda row: is_valid_data(row), axis=1)

# Chia dữ liệu thành df_valid và df_invalid dựa trên kết quả kiểm tra
df_valid = df_combined[mask]
df_invalid = df_combined[~mask]

df_valid
```

Hình. Các lệnh kiểm tra và chia dữ liệu

- Nhờ đây ta có thể biết được cột nào và dòng dữ liệu nào gặp vấn đề, có thể giải quyết kịp thời vấn đề gặp phải. Chia dữ liệu thành hai phần dữ liệu lỗi và dữ liệu hoàn chỉnh không gộp chung vào một tệp tránh mất thời gian xử lý tệp.
- Đồng thời lưu hai tệp thành file excel, để dễ dàng sử dụng khi cần.

```
# Lưu DataFrame df_valid thành file Excel
df_valid.to_excel('/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df_valid.xlsx', index=False)

# Lưu DataFrame df_invalid thành file Excel
df_invalid.to_excel('/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df_invalid.xlsx', index=False)
```

Hình. Các lệnh lưu hai dataframe thành file excel.

- Tiếp theo, ta lưu dataframe df_valid thành file excel để dễ sử dụng cho các bước tiếp theo.


```
import pandas as pd

file = "/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df_valid.xlsx"
df_Billionaires2023 = pd.read_excel(file)

# Lưu DataFrame df_Billionaires2023 thành file excel
df_Billionaires2023.to_excel('/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df_Billionaires2023.xlsx', index=False)
```

Hình. Các lệnh lưu.

- Kiểm tra dữ liệu trùng lặp. Tránh dữ liệu giống nhau xuất hiện từ hai lần trở lên, gây nhiễu dữ liệu, đưa kết quả phân tích không chính xác.

 Không có dữ liệu trùng lặp.

- Kiểm tra dữ liệu thiếu. Tránh dữ liệu rỗng, gây nhiễu dữ liệu, đưa kết quả phân tích không chính xác. Hiển thị dữ liệu thiếu theo từng cột, để tiện quan sát dữ liệu. Sau cùng, tổng hợp tất cả các dữ liệu thiếu của cả DataFrame.
 - Code: `df1_Billionaires2023.isnull().sum()`. Với đoạn code kiểm tra dữ liệu trống lần đầu tiên, ta thu được kết quả sau:

| | |
|--|----|
| <input checked="" type="checkbox"/> RANK | 0 |
| <input checked="" type="checkbox"/> NETWORTH | 0 |
| <input checked="" type="checkbox"/> NAME | 0 |
| <input type="checkbox"/> AGE | 65 |
| <input type="checkbox"/> COUNTRY | 38 |
| <input type="checkbox"/> CITY | 72 |
| <input checked="" type="checkbox"/> SOURCE | 0 |
| <input checked="" type="checkbox"/> INDUSTRY | 0 |
| <input checked="" type="checkbox"/> SELFMADE | 0 |
| <input checked="" type="checkbox"/> GENDER | 0 |

| | |
|---|-----|
| <input type="checkbox"/> BIRTHYEAR | 76 |
| <input type="checkbox"/> BIRTHMONTH | 76 |
| <input type="checkbox"/> BIRTHDAY | 76 |
| <input type="checkbox"/> CPI_COUNTRY | 184 |
| <input type="checkbox"/> cpi_change_country | 184 |
| <input type="checkbox"/> GPD_COUNTRY | 164 |
| <input type="checkbox"/> TAX_REVENUE_COUNTRY | 183 |
| <input type="checkbox"/> TOTAL_TAX_RATE_COUNTRY | 182 |
| <input type="checkbox"/> POPULATION_COUNTRY | 164 |

dtype: int64

- Ta thấy còn một số cột có dữ liệu trống. Tiến hành lấp đầy dữ liệu trống. Code: `df1 = df1_Billionaires2023.fillna(df1_Billionaires2023.mean(), inplace = True)`, ta thu được kết quả sau: Đây là cách điền giá trị trung bình vào dữ liệu trống.

| | |
|--|-----|
| <input checked="" type="checkbox"/> RANK | 0 |
| <input checked="" type="checkbox"/> NETWORTH | 0 |
| <input checked="" type="checkbox"/> NAME | 0 |
| <input checked="" type="checkbox"/> AGE | 0 |
| <input type="checkbox"/> COUNTRY | 38 |
| <input type="checkbox"/> CITY | 72 |
| <input checked="" type="checkbox"/> SOURCE | 0 |
| <input checked="" type="checkbox"/> INDUSTRY | 0 |
| <input checked="" type="checkbox"/> SELFMADE | 0 |
| <input checked="" type="checkbox"/> GENDER | 0 |
| <input checked="" type="checkbox"/> BIRTHYEAR | 0 |
| <input checked="" type="checkbox"/> BIRTHMONTH | 0 |
| <input checked="" type="checkbox"/> BIRTHDAY | 0 |
| <input checked="" type="checkbox"/> CPI_COUNTRY | 0 |
| <input checked="" type="checkbox"/> cpi_change_country | 0 |
| <input type="checkbox"/> GPD_COUNTRY | 164 |
| <input checked="" type="checkbox"/> TAX_REVENUE_COUNTRY | 0 |
| <input checked="" type="checkbox"/> TOTAL_TAX_RATE_COUNTRY | 0 |
| <input checked="" type="checkbox"/> POPULATION_COUNTRY | 0 |

dtype: int64

- Ta thấy vẫn còn dữ liệu trống nên ta tiếp tục xử lý bằng cách xóa các hàng có dữ liệu trống . Code: `df1 = df1_Billionaires2023.dropna(subset = ['COUNTRY', 'CITY', 'GPD_COUNTRY'])`, ta thu được kết quả sau:

| | |
|--|---|
| <input checked="" type="checkbox"/> RANK | 0 |
| <input checked="" type="checkbox"/> NETWORTH | 0 |
| <input checked="" type="checkbox"/> NAME | 0 |
| <input checked="" type="checkbox"/> AGE | 0 |
| <input checked="" type="checkbox"/> COUNTRY | 0 |
| <input checked="" type="checkbox"/> CITY | 0 |
| <input checked="" type="checkbox"/> SOURCE | 0 |
| <input checked="" type="checkbox"/> INDUSTRY | 0 |
| <input checked="" type="checkbox"/> SELFMADE | 0 |
| <input checked="" type="checkbox"/> GENDER | 0 |
| <input checked="" type="checkbox"/> BIRTHYEAR | 0 |
| <input checked="" type="checkbox"/> BIRTHMONTH | 0 |
| <input checked="" type="checkbox"/> BIRTHDAY | 0 |
| <input checked="" type="checkbox"/> CPI_COUNTRY | 0 |
| <input checked="" type="checkbox"/> epi_change_country | 0 |
| <input checked="" type="checkbox"/> GPD_COUNTRY | 0 |
| <input checked="" type="checkbox"/> TAX_REVENUE_COUNTRY | 0 |
| <input checked="" type="checkbox"/> TOTAL_TAX_RATE_COUNTRY | 0 |
| <input checked="" type="checkbox"/> POPULATION_COUNTRY | 0 |

dtype: int64

- Ta thấy, tất cả dữ liệu đã được lấp đầy và không còn dữ liệu trống, hỗ trợ tối đa việc khai phá dữ liệu.
- Cuối cùng, ta lưu dataframe đã hoàn chỉnh thành một file mới để hỗ trợ cho các công tác về sau dễ dàng hơn. Tên file: “**Billionaires2023.xlsx**”
- Ta tiếp tục đưa dataframe mới lưu ra, ta tạo một dataframe mới từ dataframe trên với hai cột ['NAME', 'AGE']. Code: “`df_Old = df2[['NAME', 'AGE']]`”. Thống kê tần suất xuất hiện của các tuổi và hiển thị bảng thống kê tần suất xuất hiện của các tuổi. Ta cho hiển thị bảng bảng thống kê tần suất xuất hiện của các tuổi.

```
# Thống kê tần suất xuất hiện của các tuổi
frequency_table = df2['AGE'].value_counts().sort_index().reset_index()
frequency_table.columns = ['Old', 'Frequency']

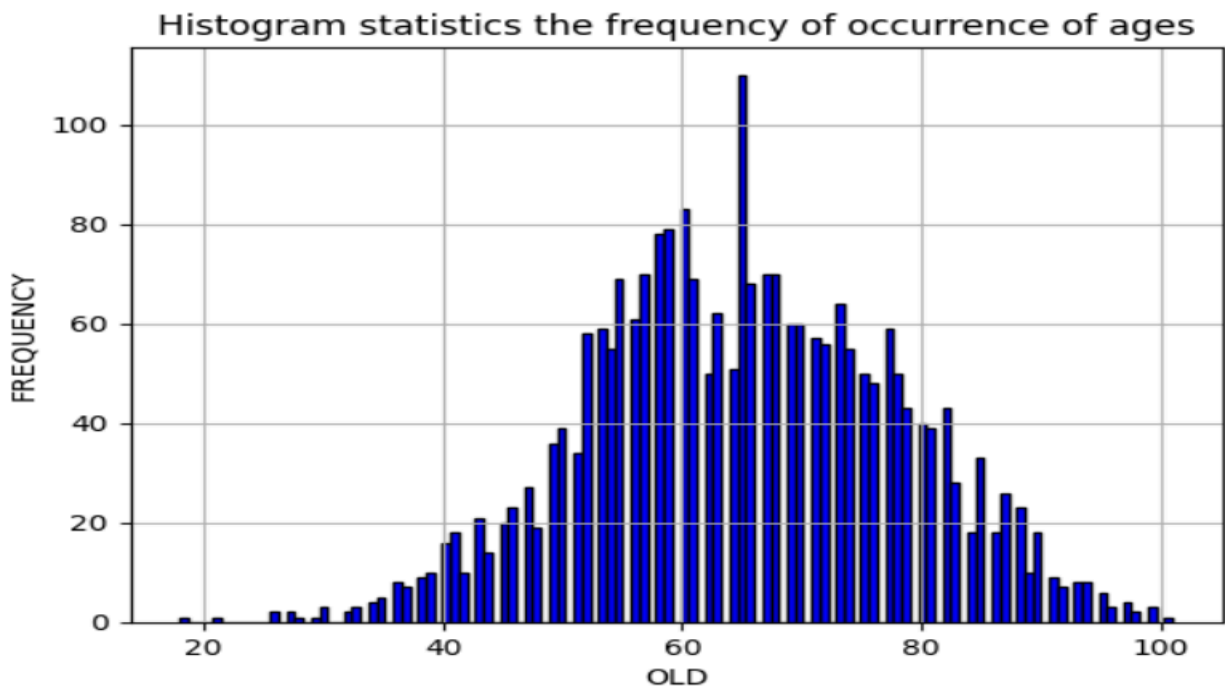
df_BillionaireOld = frequency_table
# Hiển thị bảng thống kê tần suất xuất hiện của các tuổi
print("Bảng thống kê tần suất xuất hiện của các giá trị tài sản:")
df_BillionaireOld
```

- Sau đó ta thống kê lại bằng cách biểu diễn bằng biểu đồ
 - Code:

```
import matplotlib.pyplot as plt
# Khoảng điểm
bins = 121

# Vẽ histogram từ DataFrame
plt.hist(df_BillionaireOld['Old'], bins=bins, weights=df_BillionaireOld['Frequency'], color='blue', edgecolor='black')
plt.xlabel('OLD')
plt.ylabel('FREQUENCY')
plt.title('Histogram statistics the frequency of occurrence of ages')
plt.grid(True)
plt.show()
```

- Biểu đồ:



Hình. Biểu đồ thống kê tần suất xuất hiện của các lứa tuổi

- o Hình ảnh biểu đồ thể hiện thống kê các số tuổi của các tỷ phú. Ta thấy nhóm tuổi từ 55 tuổi đến 80 tuổi có nhiều tỷ phú hơn các tuổi còn lại.
- o Bên cạnh đó ta cũng có thể thấy, nhóm tuổi từ 18 tuổi đến dưới 50 tuổi đã trở thành những tỷ phú. Dù khách quan hay chủ quan, điều này vẫn cho ta thấy được sự thành công của giới trẻ hiện nay.
- o Cuối cùng là nhóm tuổi từ 80 trở lên.

Hình. Đây là thông tin tổng quan về phân phối và phân tán độ tuổi các tỷ phú.

| | AGE |
|-------|----------|
| count | 2447.000 |
| mean | 65.006 |
| std | 12.962 |
| min | 18.000 |
| 25% | 56.000 |
| 50% | 65.000 |
| 75% | 74.000 |
| max | 101.000 |

- Tính tuổi trung bình, tuổi trung vị, mode và độ lệch chuẩn.

- Code:

```
# Tính tuổi trung bình, tuổi trung vị, mode và độ lệch chuẩn
import pandas as pd
import numpy as np

mean_age = np.mean(df_Old['AGE']).round(3)
print(f"Tuổi trung bình: {mean_age}")

median_age = np.median(df_Old['AGE'])
print(f"Tuổi trung vị: {median_age}")

std_dev_age = np.std(df_Old['AGE']).round(3)
print(f"Độ lệch chuẩn: {std_dev_age}")

mode_age = df_Old['AGE'].mode()
print(f"Mode (tuổi xuất hiện nhiều nhất): {mode_age[0]}")
```

- Ta có kết quả thu được từ dữ liệu tuổi trên là:

- o Tuổi trung bình: 65.006
- o Tuổi trung vị: 65.0
- o Độ lệch chuẩn: 12.959
- o Mode (tuổi xuất hiện nhiều nhất): 60.0

3. Mối quan hệ tương quan



Hình. Mối quan hệ giữa các thông tin của các tỷ phú năm 2023

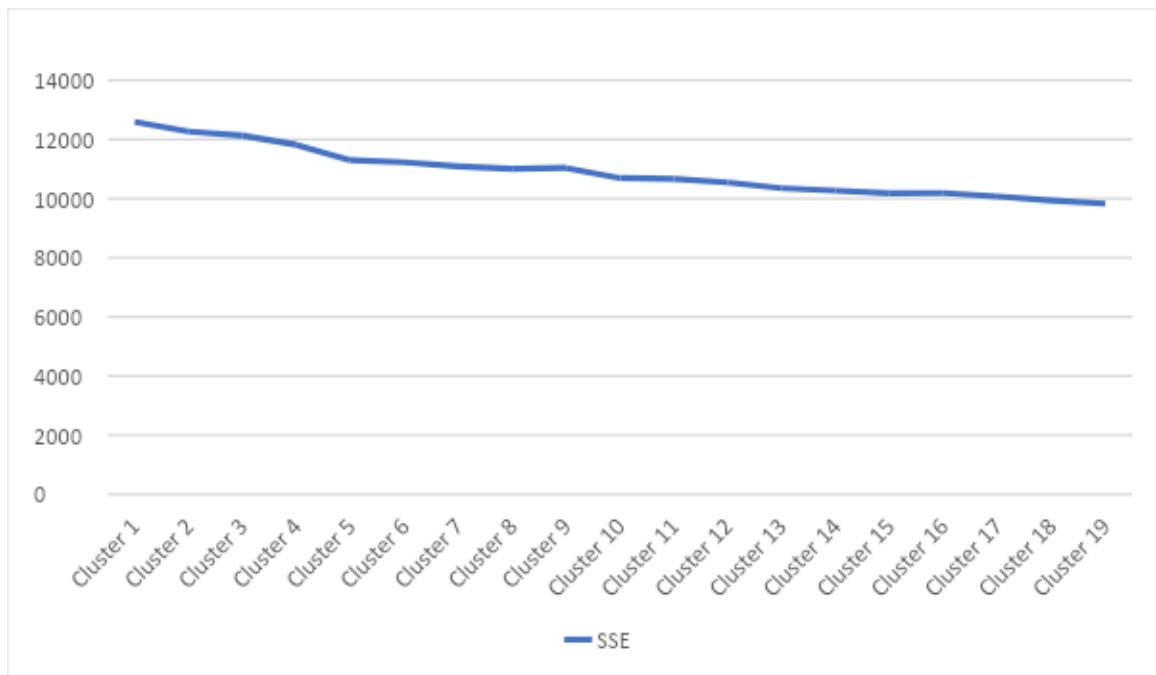
Chỉ số về mối quan hệ tương quan dao động từ $[-1.00]$ đến $[1.00]$. Tổng quan ta thấy, đối với các cặp thuộc tính có mức tương quan gần 1, có mối quan hệ mạnh mẽ và đồng biến giữa chúng. Mức tương quan gần -1, mối quan hệ đồng biến nhưng âm, tức là khi một thuộc tính tăng, thuộc tính kia giảm. Các cặp thuộc tính với mức tương quan dương gần 1 thường có sự đồng biến mạnh mẽ. Các cặp thuộc tính với mức tương quan âm gần -1 thường có mối quan hệ nghịch biến mạnh mẽ. Giá trị tương quan ở mức trung bình từ 0.75-0.50, có thể có mối quan hệ tương đối mạnh nhưng không đến mức rất mạnh. Khi giá trị tương quan gần 0, có thể cho thấy sự độc lập giữa các thuộc tính, không có mối quan hệ tuyến tính rõ ràng.

4. Gom cụm Kmeans

a. Trên Weka

Sử dụng tập dữ liệu thu được từ bước tiền xử lý dữ liệu trên, tiến hành dùng phần mềm Weka để phân cụm thuật toán Kmeans.

- Dùng phương pháp Elbow để xác định số cụm (Cluster) thông qua số tuổi của các tỷ phú, ta thu được biểu đồ sau:



Hình. Số cluster tương ứng để đề xuất cho bài toán phân cụm dữ liệu.

Như vậy, ta có thể kết luận:

- Số cluster tương ứng có thể là 17
- Số lượng instance có trong dữ liệu là 9825.785
- Thiết lập số lần lặp là 37
- Ta thu được kết quả sau:

```

kMeans
=====

Number of iterations: 37
Within cluster sum of squared errors: 9825.785796377386

Initial starting points (random):

Cluster 0: 1434,2100,'Venu Srinivasan',70,India,Chennai,Two-wheelers,Automotive,1,M,1952,12,11,180.44,7.7,'$2,611,000,000,000 ',11.2,49.7,1366417754
Cluster 1: 2259,1200,'Vadim Yakunin',60,Russia,Moscow,Pharmacy,Healthcare,1,M,1963,1,5,180.75,4.5,'$1,699,876,578,871 ',11.4,46.2,144373535
Cluster 2: 1104,2700,'Nadir Godrej',72,India,Mumbai,'Consumer goods',Diversified,1,M,1951,1,1,180.44,7.7,'$2,611,000,000,000 ',11.2,49.7,1366417754
Cluster 3: 1027,2900,'Martin Lorentzon',54,Sweden,Stockholm,Spotify,Technology,1,M,1969,4,1,110.51,1.8,'$530,832,908,738 ',27.9,49.1,10285453
Cluster 4: 1647,1800,'O. Francis Biondi',58,'United States','New York','Hedge funds','Finance & Investments',1,M,1964,7,4,117.24,7.5,'$21,427,700,000,000 ',9.6,36.6,328239
Cluster 5: 2405,1100,'Zhang Xuansong',51,China,Fuzhou,Supermarkets,'Fashion & Retail',1,M,1971,10,9,125.08,2.9,'$19,910,000,000,000 ',9.4,59.2,1397715000
Cluster 6: 1368,2200,'Henry Swieca',65,'United States','New York','Hedge funds','Finance & Investments',1,M,1957,5,9,117.24,7.5,'$21,427,700,000,000 ',9.6,36.6,328239
Cluster 7: 1905,1500,'Li Jiaquan',59,China,Chengdu,Chemicals,Manufacturing,1,M,1963,9,6,125.08,2.9,'$19,910,000,000,000 ',9.4,59.2,1397715000
Cluster 8: 1516,2000,'Larry Tanenbaum',77,Canada,Toronto,Sports,Sports,1,M,1945,7,1,116.76,1.9,'$1,736,425,629,520 ',12.8,24.5,36991981
Cluster 9: 2405,1100,'Huang Xiaofen ',61,China,Shenzhen,'Printed circuit boards',Technology,1,F,1962,1,1,125.08,2.9,'$19,910,000,000,000 ',9.4,59.2,1397715000
Cluster 10: 1027,2900,'God Nisanov',50,Russia,Moscow,'Real estate','Real Estate',1,M,1972,4,24,180.75,4.5,'$1,699,876,578,871 ',11.4,46.2,144373535
Cluster 11: 2259,1200,'Danna Azrieli',55,Israel,Herzliya,'Real estate','Real Estate',1,F,1967,6,3,108.15,0.8,'$395,098,666,122 ',23.1,25.3,9053300
Cluster 12: 31,38300,'John Mars',87,'United States',Jackson,'Candy, pet food','Food & Beverage',1,M,1935,10,15,117.24,7.5,'$21,427,700,000,000 ',9.6,36.6,328239
Cluster 13: 1027,2900,'Bernard Ecclestone ',92,'United Kingdom',London,'Formula One',Sports,1,M,1930,10,28,119.62,1.7,'$2,827,113,184,696 ',25.5,30.6,66834405
Cluster 14: 268,7900,'Orlando Bravo',52,'United States','Miami Beach','Private equity','Finance & Investments',1,M,1970,9,23,117.24,7.5,'$21,427,700,000,000 ',9.6,36.6,328239
Cluster 15: 1575,1900,'James Leprino',85,'United States','Indian Hills',Cheese,'Food & Beverage',1,M,1937,11,22,117.24,7.5,'$21,427,700,000,000 ',9.6,36.6,328239
Cluster 16: 1104,2700,'Zhu Yi',59,China,Chengdu,Pharmaceuticals,Healthcare,1,M,1963,12,1,125.08,2.9,'$19,910,000,000,000 ',9.4,59.2,1397715000
Cluster 17: 2259,1200,'Karl Knauf',65.140194,Germany,Iphofen,'Building materials',Manufacturing,1,M,1957.183307,5.74025,12.099844,112.85,1.4,'$3,845,630,030,824
Cluster 18: 1905,1500,'Federico De Nora',55,Italy,Milan,Electrodes,Manufacturing,1,M,1968,3,23,110.62,0.6,'$2,001,244,392,042 ',24.3,59.1,60297396
Cluster 19: 2133,1300,'Rustem Sulteev',69,Russia,Kazan,'Refinery, chemicals',Energy,1,M,1954,1,4,180.75,4.5,'$1,699,876,578,871 ',11.4,46.2,144373535

```

Các cluster:

Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Clusterer

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 20 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

Use training set

Supplied test set

Set...

Percentage split

% 66

Classes to clusters evaluation

(Num) POPULATION_COUNTRY

Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

19:08:04 - SimpleKMeans

19:08:39 - SimpleKMeans

19:09:01 - SimpleKMeans

19:09:19 - SimpleKMeans

19:09:34 - SimpleKMeans

19:09:53 - SimpleKMeans

19:10:07 - SimpleKMeans

19:10:21 - SimpleKMeans

19:10:40 - SimpleKMeans

19:10:55 - SimpleKMeans

19:11:09 - SimpleKMeans

19:11:22 - SimpleKMeans

19:11:34 - SimpleKMeans

19:11:56 - SimpleKMeans

19:12:12 - SimpleKMeans

19:12:26 - SimpleKMeans

19:12:43 - SimpleKMeans

19:13:01 - SimpleKMeans

19:13:19 - SimpleKMeans

Clusterer output

Cluster 18: 1905,1500,'Federico De Nora',55,Italy,Milan,Electrodes,Manufacturing,1,M,1968,3,23,110.62,0.6,'\$2,001,244,392,042 ',24.3,59.1,60297396

Cluster 19: 2133,1300,'Rustem Sulteev',69,Russia,Kazan,'Refinery, chemicals',Energy,1,M,1954,1,4,180.75,4.5,'\$1,699,876,578,871 ',11.4,46.2,144373535

Missing values globally replaced with mean/mode

Final cluster centroids:

| Attribute | 0 | 1 | 2 | 3 |
|------------------------|-----------------------|---------------------|---------------------|---------------------|
| | (2447.0) | (104.0) | (70.0) | (125.0) |
| RANK | 1285.8018 | 1495.0288 | 1118.1 | 1288.176 |
| NETWORK | 4713.5268 | 3099.0385 | 4327.1429 | 4381.6 |
| NAME | Wang Yangming | Shiv Nadar | Vladimir Potanin | Mukesh Ambani |
| AGE | 65.006 | 67.8942 | 63.9714 | 69.9394 |
| COUNTRY | United States | India | Russia | India |
| CITY | New York | Delhi | Moscow | Mumbai |
| SOURCE | Real estate | Pharmaceuticals | Pharmaceuticals | Diversified |
| INDUSTRY | Finance & Investments | Healthcare | Healthcare | Diversified |
| SELFMADE | 1 | 1 | 1 | 1 |
| GENDER | M | M | M | M |
| BIRTHYEAR | 1957.3141 | 1954.3462 | 1958.6429 | 1952.4124 |
| BIRTHMONTH | 5.7634 | 6.7596 | 3.8571 | 5.8258 |
| BIRTHDAY | 12.2759 | 13.2173 | 14.5143 | 12.6504 |
| CPI_COUNTRY | 127.8713 | 180.44 | 150.6713 | 166.8742 |
| cpi_change_country | 4.2853 | 7.7 | 3.1304 | 6.1127 |
| GDP_COUNTRY | \$21,427,700,000,000 | \$2,611,000,000,000 | \$1,699,876,578,871 | \$2,611,000,000,000 |
| TAX_REVENUE_COUNTRY | 12.516 | 11.2 | 13.5185 | 13.2171 |
| TOTAL_TAX_RATE_COUNTRY | 44.0363 | 49.7 | 42.8666 | 44.6879 |
| POPULATION_COUNTRY | 515231897.9264 | 1366417754 | 94114092.4714 | 631536909.008 |

Time taken to build model (full training data) : 0.19 seconds

=== Model and evaluation on training set ===

Status

OK

Log

x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Cluster

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 20 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

☒ Use training set

☐ Supplied test set Set...

☐ Percentage split % 66

☐ Classes to clusters evaluation

(Num) POPULATION_COUNTRY

☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

190804 - SimpleKMeans

190839 - SimpleKMeans

190901 - SimpleKMeans

190919 - SimpleKMeans

190934 - SimpleKMeans

190953 - SimpleKMeans

191007 - SimpleKMeans

191021 - SimpleKMeans

191040 - SimpleKMeans

191055 - SimpleKMeans

191109 - SimpleKMeans

191122 - SimpleKMeans

191134 - SimpleKMeans

191156 - SimpleKMeans

191212 - SimpleKMeans

191226 - SimpleKMeans

191243 - SimpleKMeans

191301 - SimpleKMeans

191319 - SimpleKMeans

Clusterer output

| 3 (65.0) | 4 (170.0) | 5 (76.0) | 6 (218.0) | 7 (201.0) |
|------------------|-----------------------|----------------------|-----------------------|----------------------|
| 1373.1077 | 1802.0882 | 1575.7237 | 1011.289 | 1583.2189 |
| 3296.9231 | 1879.4118 | 2631.5789 | 7026.6055 | 2523.3831 |
| Stefan Persson | Donald Newhouse | Lu Xiangyang | Jeff Bezos | Wang Yanqing |
| 56.7714 | 64.9588 | 60.7782 | 69.4502 | 60.7255 |
| Sweden | United States | China | United States | China |
| Stockholm | New York | Guangzhou | New York | Shanghai |
| Software | Investments | Retail | Investments | Chemicals |
| Technology | Finance & Investments | Fashion & Retail | Finance & Investments | Manufacturing |
| 1 | 1 | 1 | 1 | 1 |
| M | M | M | M | M |
| 1965.5413 | 1957.0471 | 1961.4517 | 1953.221 | 1961.3954 |
| 4.1037 | 9.3412 | 5.2958 | 2.53 | 4.9257 |
| 13.9092 | 12.7 | 13.3053 | 17.6381 | 8.0144 |
| 114.4718 | 117.3615 | 123.8284 | 118.0197 | 124.7837 |
| 2.2477 | 6.0224 | 2.5632 | 6.7538 | 2.8542 |
| 5530,832,908,738 | \$21,427,700,000,000 | \$19,910,000,000,000 | \$21,427,700,000,000 | \$19,910,000,000,000 |
| 21.3431 | 10.7329 | 10.4711 | 10.2601 | 9.4363 |
| 44.3815 | 38.0053 | 54.7316 | 37.0492 | 58.4299 |
| 40905111.6769 | 269369311.7706 | 1149282238.7763 | 294078116.4266 | 1358134041.1393 |

Status OK

Log

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Cluster

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 20 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

☒ Use training set

☐ Supplied test set Set...

☐ Percentage split % 66

☐ Classes to clusters evaluation

(Num) POPULATION_COUNTRY

☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

190804 - SimpleKMeans

190839 - SimpleKMeans

190901 - SimpleKMeans

190919 - SimpleKMeans

190934 - SimpleKMeans

190953 - SimpleKMeans

191007 - SimpleKMeans

191021 - SimpleKMeans

191040 - SimpleKMeans

191055 - SimpleKMeans

191109 - SimpleKMeans

191122 - SimpleKMeans

191134 - SimpleKMeans

191156 - SimpleKMeans

191212 - SimpleKMeans

191226 - SimpleKMeans

191243 - SimpleKMeans

191301 - SimpleKMeans

191319 - SimpleKMeans

Clusterer output

| 7 (201.0) | 8 (85.0) | 9 (134.0) | 10 (96.0) | 11 (67.0) |
|----------------------|---------------------|----------------------|---------------------|------------------------------|
| 1583.2189 | 1452.3765 | 1242.4478 | 1451.0313 | 1347.2687 |
| 2523.3831 | 5969.2353 | 5375.3731 | 4351.0417 | 9245.6716 |
| Wang Yanqing | Bernard Arnault | Zhang Yiming | Carlos Slim Helu | Francoise Bettencourt Meyers |
| 60.7255 | 71.568 | 55.9488 | 64.4167 | 64.9764 |
| China | Canada | China | Russia | Israel |
| Shanghai | Toronto | Shenzhen | Moscow | Tel Aviv |
| Chemicals | Real estate | Software | Real estate | Diversified |
| Manufacturing | Fashion & Retail | Technology | Real Estate | Diversified |
| 1 | 1 | 1 | 1 | 1 |
| M | M | M | M | P |
| 1961.3954 | 1951.0867 | 1966.7626 | 1957.6979 | 1957.456 |
| 4.9257 | 3.7468 | 3.4831 | 7.7083 | 5.3018 |
| 8.0144 | 7.5553 | 5.4187 | 15.7083 | 10.7209 |
| 124.7837 | 120.8785 | 125.08 | 164.5905 | 123.7005 |
| 2.8542 | 2.0199 | 2.9 | 4.4576 | 2.1413 |
| \$19,910,000,000,000 | \$1,736,425,629,520 | \$19,910,000,000,000 | \$1,699,976,578,871 | \$395,098,666,122 |
| 9.4363 | 13.3722 | 9.4 | 13.7749 | 20.2634 |
| 58.4299 | 32.1716 | 59.2 | 45.1534 | 37.7278 |
| 1358134041.1393 | 50654013.9176 | 1397715000 | 105005572.7188 | 44037086.9851 |

Status OK

Log

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Cluster

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 20 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

☒ Use training set

☐ Supplied test set Set...

☐ Percentage split % 66

☐ Classes to clusters evaluation (Num) POPULATION, COUNTRY

☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

190804 - SimpleKMeans

190839 - SimpleKMeans

190901 - SimpleKMeans

190919 - SimpleKMeans

190934 - SimpleKMeans

190953 - SimpleKMeans

191007 - SimpleKMeans

191021 - SimpleKMeans

191040 - SimpleKMeans

191055 - SimpleKMeans

191109 - SimpleKMeans

191122 - SimpleKMeans

191134 - SimpleKMeans

191156 - SimpleKMeans

191212 - SimpleKMeans

191226 - SimpleKMeans

191243 - SimpleKMeans

191301 - SimpleKMeans

191319 - SimpleKMeans

Cluster output

| | 12 (174.0) | 13 (91.0) | 14 (173.0) | 15 (169.0) | 16 (132.0) |
|----------------------|-----------------------|-----------------------|----------------------|----------------------|---------------|
| 1072.477 | 1267.978 | 723.7225 | 971 | 1489.6136 | |
| 5093.1034 | 4360.4396 | 9398.2659 | 6130.7692 | 3270.4545 | |
| Larry Ellison | Len Blavatnik | Elon Musk | Bill Gates | Li Li | |
| 67.9335 | 64.102 | 60.0058 | 73.3964 | 59.027 | |
| United States | United Kingdom | United States | United States | China | |
| Atlanta | London | San Francisco | Palm Beach | Beijing | |
| Cargill | Diversified | Private equity | Real estate | Pharmaceuticals | |
| Food & Beverage | Finance & Investments | Finance & Investments | Food & Beverage | Healthcare | |
| 1 | 1 | 1 | 1 | 1 | |
| M | M | M | M | M | |
| 1954.6526 | 1958.2128 | 1962.0809 | 1948.6686 | 1963.2115 | |
| 3.0174 | 5.8794 | 8.0289 | 8.2308 | 6.5886 | |
| 6.4224 | 11.9363 | 13.3237 | 22.1006 | 5.5568 | |
| 118.4224 | 121.4752 | 119.2612 | 118.4211 | 125.08 | |
| 6.2716 | 1.8044 | 6.7281 | 6.6485 | 2.9 | |
| \$21,427,700,000,000 | \$2,827,113,184,696 | \$21,427,700,000,000 | \$21,427,700,000,000 | \$19,910,000,000,000 | |
| 10.7905 | 24.9835 | 9.9306 | 10.3663 | 9.4 | |
| 38.2711 | 31.9396 | 36.3983 | 37.1036 | 59.2 | |
| 274617097.3563 | 65891339.2308 | 290193762.6243 | 285554386.2485 | 1397715000 | |

Status OK

Log x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Cluster

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 20 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

☒ Use training set

☐ Supplied test set Set...

☐ Percentage split % 66

☐ Classes to clusters evaluation (Num) POPULATION, COUNTRY

☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

190804 - SimpleKMeans

190839 - SimpleKMeans

190901 - SimpleKMeans

190919 - SimpleKMeans

190934 - SimpleKMeans

190953 - SimpleKMeans

191007 - SimpleKMeans

191021 - SimpleKMeans

191040 - SimpleKMeans

191055 - SimpleKMeans

191109 - SimpleKMeans

191122 - SimpleKMeans

191134 - SimpleKMeans

191156 - SimpleKMeans

191212 - SimpleKMeans

191226 - SimpleKMeans

191243 - SimpleKMeans

191301 - SimpleKMeans

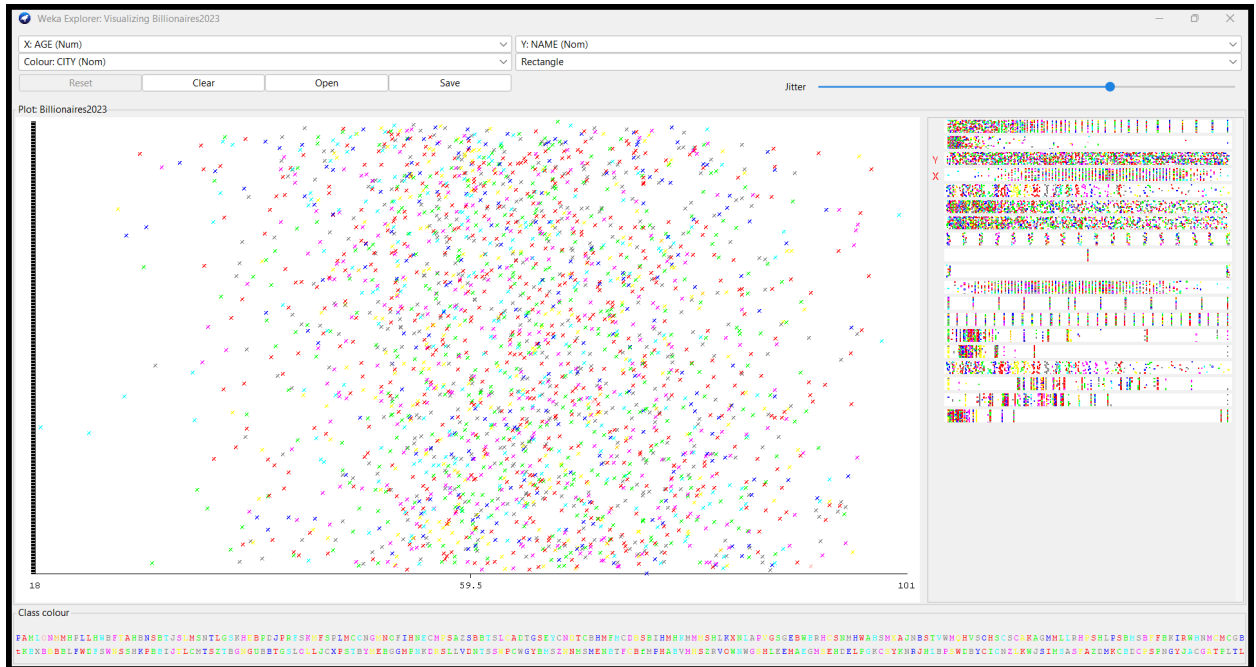
191319 - SimpleKMeans

Cluster output

| | 14 (173.0) | 15 (169.0) | 16 (132.0) | 17 (136.0) | 18 (91.0) | 19 (70.0) |
|----------------|----------------------|----------------------|---------------------|---------------------|-------------------|--------------|
| 723.7225 | 971 | 1489.6136 | 1310.625 | 1271.5495 | 1373.4857 | |
| 9398.2659 | 6130.7692 | 3270.4545 | 4265.4412 | 4342.8571 | 3402.8571 | |
| Elon Musk | Bill Gates | Li Li | Dieter Schwarz | Amancio Ortega | Low Tuck Kwong | |
| 60.0058 | 73.3964 | 59.027 | 64.8958 | 68.1334 | 67.3714 | |
| United States | United States | China | Germany | Italy | Singapore | |
| San Francisco | Palm Beach | Beijing | Munich | Milan | Singapore | |
| Private equity | Real estate | Pharmaceuticals | Medical technology | Luxury goods | Real estate | |
| Investments | Food & Beverage | Healthcare | Manufacturing | Fashion & Retail | Energy | |
| 1 | 1 | 1 | 1 | 1 | 1 | |
| M | M | M | M | M | M | |
| 1962.0809 | 1948.6686 | 1963.2115 | 1957.3865 | 1954.1339 | 1954.874 | |
| 8.0289 | 8.2308 | 6.5886 | 5.3115 | 6.1373 | 5.3249 | |
| 13.3237 | 22.1006 | 5.5568 | 12.922 | 16.7703 | 9.0871 | |
| 119.2612 | 118.4211 | 125.08 | 118.3706 | 114.0563 | 124.2886 | |
| 6.7281 | 6.6485 | 2.9 | 1.7203 | 1.0238 | 2.1629 | |
| 0,000,000 | \$21,427,700,000,000 | \$19,910,000,000,000 | \$3,845,630,030,824 | \$2,001,244,392,042 | \$372,062,527,489 | |
| 9.9306 | 10.3663 | 9.4 | 12.5945 | 22.8375 | 13.4114 | |
| 36.3983 | 37.1036 | 59.2 | 46.1946 | 54.6468 | 27.7457 | |
| 93762.6243 | 285554386.2485 | 1397715000 | 82394055.2132 | 55677400.6813 | 37239509.5857 | |

Status OK

Log x0



Hình. Với trục X:AGE và Y:NAME.

- Thời gian xây dựng model: 0.019 giây
- Thành phần của các cluter:
 - o Cluster 0: 104 (4%)
 - o Cluster 1: 70 (3%)
 - o Cluster 2: 125 (5%)
 - o Cluster 3: 65 (3%)
 - o Cluster 4: 170 (7%)
 - o Cluster 5: 76 (3%)
 - o Cluster 6: 218 (9%)
 - o Cluster 7: 201 (8%)
 - o Cluster 8: 85 (3%)
 - o Cluster 9: 134 (5%)
 - o Cluster 10: 96 (4%)
 - o Cluster 11: 67 (3%)
 - o Cluster 12: 174 (7%)
 - o Cluster 13: 91 (4%)

- o Cluster 14: 173 (7%)
- o Cluster 15: 169 (7%)
- o Cluster 16: 132 (5%)
- o Cluster 17: 136 (6%)
- o Cluster 18: 91 (4%)
- o Cluster 19: 70 (3%)

b. Trên Colab

Đọc tệp excel 'Billionaires2023.xlsx' thu được từ bước tiền xử lý dữ liệu trên vào dataframe df2.

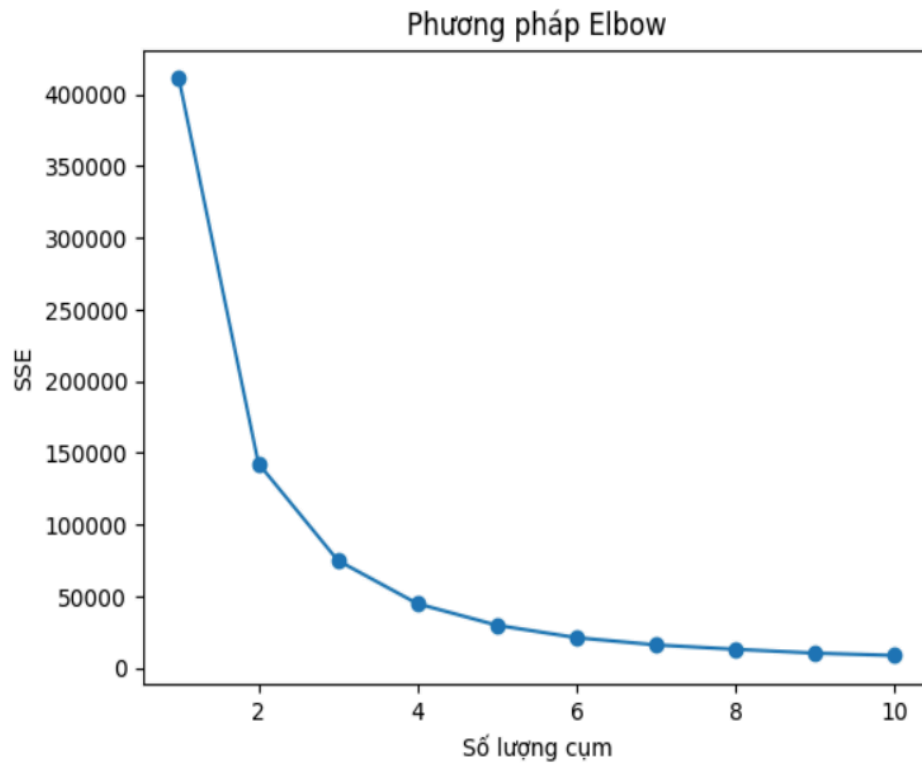
- Dùng phương pháp Elbow để tìm số cụm tối ưu (Cluster) thông qua số tuổi của các tỷ phú

```
[ ] from sklearn.cluster import KMeans
    from sklearn.metrics import silhouette_score

[ ] # Tìm số cụm tối ưu sử dụng phương pháp Elbow
max_clusters = 10 # Số cụm tối đa muốn xem
sse = []
for k in range(1, max_clusters + 1):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df2[['AGE']])
    sse.append(kmeans.inertia_)
```

[illegible]

- Ta thu được biểu đồ sau:



Hình. Số cluster tương ứng để đề xuất cho bài toán phân cụm dữ liệu.

Như vậy, ta có:

- Số cluster tối ưu dựa trên biểu đồ Elbow là 5

Áp dụng thuật toán Kmeans để phân cụm theo độ tuổi và khối tài sản của các tỷ phú

```
# Áp dụng thuật toán K-means
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(df2[['AGE']])

# Gán nhãn cụm cho dữ liệu
df2['Cluster'] = kmeans.labels_

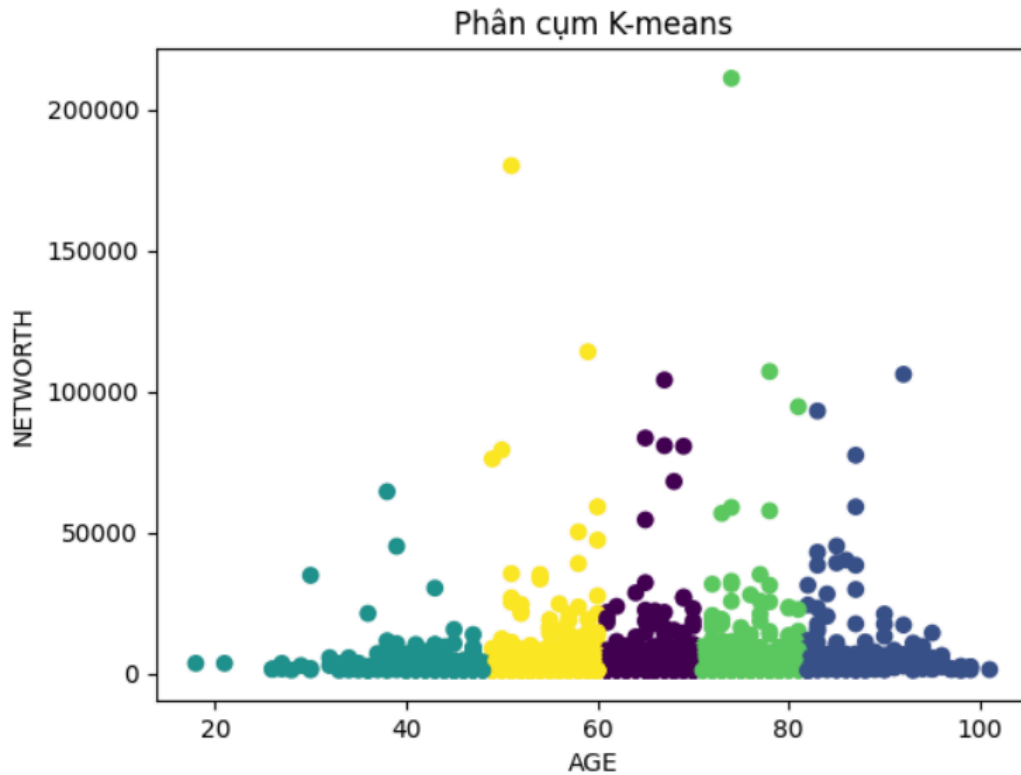
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of 'n_init' will change from 10 to 'auto' in 1.4. Set the value of 'n_init' explicitly to silence this warning.
```

Tiếp theo, ta trực quan hóa các cụm bằng cách vẽ biểu đồ

```
# Vẽ biểu đồ phân cụm
plt.scatter(df2['AGE'], df2['NETWORTH'], c=df2['Cluster'], cmap='viridis')
plt.xlabel('AGE')
plt.ylabel('NETWORTH')
plt.title('Phân cụm K-means')
plt.show()

df2[['AGE', 'NETWORTH']].round(2)
```

Ta thu được biểu đồ sau:



Hình. Phân cụm dữ liệu theo K-means.

Lọc các cụm để dễ quan sát hơn

- Các cụm có nhãn là 0 (cụm màu tím)

```
# lọc các cụm
filtered_df = df2[(df2['cluster'] == 0.0)]
filtered_df
```

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_C |
|------|------|----------|------------------------------|------|---------------|---------------|----------------------------|------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------|
| 5 | 6 | 104000 | Bill Gates | 67.0 | United States | Medina | Microsoft | Technology | True | M | 1955.0 | 10.0 | 28.0 | 117.24 | 7.5 | \$21,427,700,0 |
| 8 | 9 | 83400 | Mukesh Ambani | 65.0 | India | Mumbai | Diversified | Diversified | False | M | 1957.0 | 4.0 | 19.0 | 180.44 | 7.7 | \$2,611,000,0 |
| 9 | 10 | 80700 | Steve Ballmer | 67.0 | United States | Hunts Point | Microsoft | Technology | True | M | 1956.0 | 3.0 | 24.0 | 117.24 | 7.5 | \$21,427,700,0 |
| 10 | 11 | 80500 | Francoise Bettencourt Meyers | 69.0 | France | Paris | L'Oréal | Fashion & Retail | False | F | 1953.0 | 7.0 | 10.0 | 110.05 | 1.1 | \$2,715,518,2 |
| 14 | 15 | 68000 | Zhong Shanshan | 68.0 | China | Hangzhou | Beverages, pharmaceuticals | Food & Beverage | True | M | 1954.0 | 12.0 | 1.0 | 125.08 | 2.9 | \$19,910,000,0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2423 | 2540 | 1000 | Neerja Sethi | 68.0 | United States | Fisher Island | IT consulting, outsourcing | Technology | True | F | 1955.0 | 1.0 | 16.0 | 117.24 | 7.5 | \$21,427,700,0 |
| 2428 | 2540 | 1000 | Wijono Tanoko | 70.0 | Indonesia | Surabaya | Paints | Manufacturing | False | M | 1952.0 | 8.0 | 28.0 | 151.18 | 3.0 | \$1,119,190,7 |
| 2433 | 2540 | 1000 | Shigefumi Wada | 70.0 | Japan | Nakano, Tokyo | Software | Technology | True | M | 1952.0 | 8.0 | 30.0 | 105.48 | 0.5 | \$5,081,769,5 |
| 2441 | 2540 | 1000 | Yi Xianzhong | 63.0 | China | Guangzhou | Pharmaceuticals | Healthcare | True | M | 1959.0 | 5.0 | 1.0 | 125.08 | 2.9 | \$19,910,000,0 |

-> Các tỷ phú có độ tuổi từ 60 đến dưới 70 tuổi có khối tài sản dao động từ khoảng 100 tỷ trở xuống.

- Các cụm có nhãn là 1 (cụm màu xanh dương)

```
filtered_df = df2[(df2['cluster'] == 1)]
filtered_df
```

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_COUNTRY |
|------|------|----------|-----------------------------|------|---------------|-------------|--------------------|-----------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------------|
| 4 | 5 | 106000 | Warren Buffett | 92.0 | United States | Omaha | Berkshire Hathaway | Finance & Investments | True | M | 1930.0 | 8.0 | 30.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 7 | 8 | 93000 | Carlos Slim Helu | 83.0 | Mexico | Mexico City | Telecom | Telecom | True | M | 1940.0 | 1.0 | 28.0 | 141.54 | 3.6 | \$1,258,286,717,121 |
| 12 | 13 | 77300 | Amancio Ortega | 87.0 | Spain | La Coruna | Zara | Fashion & Retail | True | M | 1936.0 | 3.0 | 28.0 | 110.96 | 0.7 | \$1,394,116,310,761 |
| 16 | 17 | 59000 | Charles Koch | 87.0 | United States | Wichita | Koch Industries | Diversified | False | M | 1935.0 | 11.0 | 1.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 24 | 25 | 45100 | Phil Knight | 85.0 | United States | Hillsboro | Nike | Fashion & Retail | True | M | 1938.0 | 2.0 | 24.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2344 | 2405 | 1100 | Yao Hsiao Tung | 83.0 | Singapore | Singapore | Manufacturing | Manufacturing | True | M | 1940.0 | 1.0 | 1.0 | 114.41 | 0.6 | \$372,062,527,481 |
| 2361 | 2540 | 1000 | Ana Maria Brescia Cafferata | 98.0 | Peru | Lima | Mining, banking | Diversified | False | F | 1924.0 | 4.0 | 19.0 | 129.78 | 2.1 | \$226,848,050,821 |
| 2395 | 2540 | 1000 | Morris Kahn | 93.0 | Israel | Beit Yanay | Software | Technology | True | M | 1930.0 | 1.0 | 1.0 | 108.15 | 0.8 | \$395,098,666,121 |

-> Các tỷ phú có độ tuổi trên 80 tuổi có khối tài sản dao động từ khoảng 100 tỷ trở xuống.

- Các cụm có nhãn là 2 (cụm màu xanh lá đậm)

```
filtered_df = df2[(df2['cluster'] == 2)]
filtered_df
```

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_COUNTRY |
|------|------|----------|-------------------|------|---------------|-------------|---------------------|------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------------|
| 15 | 16 | 64400 | Mark Zuckerberg | 38.0 | United States | Palo Alto | Facebook | Technology | True | M | 1984.0 | 5.0 | 14.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 25 | 26 | 45000 | Zhang Yiming | 39.0 | China | Beijing | TikTok | Technology | True | M | 1984.0 | 1.0 | 1.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 35 | 37 | 34700 | Mark Mateschitz | 30.0 | Austria | Salzburg | Red Bull | Food & Beverage | False | M | 1992.0 | 5.0 | 7.0 | 118.06 | 1.5 | \$446,314,739,528 |
| 43 | 45 | 30200 | Colin Zheng Huang | 43.0 | China | Shanghai | E-commerce | Technology | True | M | 1980.0 | 2.0 | 2.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 71 | 74 | 21200 | Lukas Walton | 36.0 | United States | Chicago | Walmart | Fashion & Retail | False | M | 1986.0 | 9.0 | 19.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2390 | 2540 | 1000 | Daniilo Iervolino | 45.0 | Italy | Rome | Online universities | Technology | True | M | 1978.0 | 4.0 | 2.0 | 110.62 | 0.6 | \$2,001,244,392,042 |
| 2393 | 2540 | 1000 | LeBron James | 38.0 | United States | Los Angeles | Basketball | Sports | True | M | 1984.0 | 12.0 | 30.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 2401 | 2540 | 1000 | Kim Jung-woong | 48.0 | South Korea | Seoul | Cosmetics | Fashion & Retail | True | M | 1975.0 | 1.0 | 1.0 | 115.16 | 0.4 | \$2,029,000,000,000 |
| 2409 | 2540 | 1000 | Li Wanqiang | 45.0 | China | Beijing | Smartphones | Technology | True | M | 1977.0 | 8.0 | 1.0 | 125.08 | 2.9 | \$19,910,000,000,000 |

-> Các tỷ phú có độ tuổi dưới 50 tuổi có khối tài sản dao động từ khoảng 70 tỷ trở xuống.

- Các cụm có nhãn là 3 (cụm màu xanh lá chuối)

```
filtered_df = df2[(df2['cluster'] == 3)]
filtered_df
```

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_COUNTRY |
|------|------|----------|------------------------|------|---------------|-------------|------------------------|-----------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------------|
| 0 | 1 | 211000 | Bernard Arnault | 74.0 | France | Paris | LVMH | Fashion & Retail | False | M | 1949.0 | 3.0 | 5.0 | 110.05 | 1.1 | \$2,715,518,274,227 |
| 3 | 4 | 107000 | Larry Ellison | 78.0 | United States | Lanai | Oracle | Technology | True | M | 1944.0 | 8.0 | 17.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 6 | 7 | 94500 | Michael Bloomberg | 81.0 | United States | New York | Bloomberg LP | Media & Entertainment | True | M | 1942.0 | 2.0 | 14.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 18 | 19 | 58800 | Jim Walton | 74.0 | United States | Bentonville | Walmart | Fashion & Retail | False | M | 1948.0 | 6.0 | 7.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 19 | 20 | 57600 | Rob Walton | 78.0 | United States | Bentonville | Walmart | Fashion & Retail | False | M | 1944.0 | 10.0 | 27.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2430 | 2540 | 1000 | David Tran | 77.0 | United States | Arcadia | Hot sauce | Food & Beverage | True | M | 1945.0 | 11.0 | 19.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 2432 | 2540 | 1000 | Murat Vargi | 75.0 | Turkey | Istanbul | Telecom | Telecom | True | M | 1947.0 | 11.0 | 14.0 | 234.44 | 15.2 | \$754,411,708,203 |
| 2436 | 2540 | 1000 | V. Prem Watsa | 72.0 | Canada | Toronto | Insurance, Investments | Finance & Investments | True | M | 1950.0 | 8.0 | 5.0 | 116.76 | 1.9 | \$1,736,425,629,520 |
| 2443 | 2540 | 1000 | Richard Yuengling, Jr. | 80.0 | United States | Pottsville | Beer | Food & Beverage | False | M | 1943.0 | 3.0 | 10.0 | 117.24 | 7.5 | \$21,427,700,000,000 |

-> Các tỷ phú có độ tuổi từ 70 đến 80 tuổi có khối tài sản dao động từ khoảng 200 tỷ trở xuống.

- Các cụm có nhãn là 4 (cụm màu vàng)

```
[ ] filtered_df = df2[(df2['cluster'] == 4)]
filtered_df
```

| | RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_COUNTRY |
|------|------|----------|---------------|------|---------------|-----------|------------------------------|------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------------|
| 1 | 2 | 180000 | Elon Musk | 51.0 | United States | Austin | Tesla, SpaceX | Automotive | True | M | 1971.0 | 6.0 | 28.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 2 | 3 | 114000 | Jeff Bezos | 59.0 | United States | Medina | Amazon | Technology | True | M | 1964.0 | 1.0 | 12.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 11 | 12 | 79200 | Larry Page | 50.0 | United States | Palo Alto | Google | Technology | True | M | 1973.0 | 3.0 | 26.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 13 | 14 | 76000 | Sergey Brin | 49.0 | United States | Los Altos | Google | Technology | True | M | 1973.0 | 8.0 | 21.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 17 | 17 | 59000 | Julia Koch | 60.0 | United States | New York | Koch Industries | Diversified | False | F | 1962.0 | 4.0 | 12.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2438 | 2540 | 1000 | Xie Bingzheng | 54.0 | China | Guangzhou | Apparel | Fashion & Retail | True | M | 1969.0 | 1.0 | 1.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 2439 | 2540 | 1000 | Xu Gang | 59.0 | China | Jiaozuo | Chemicals | Manufacturing | True | M | 1963.0 | 4.0 | 17.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 2440 | 2540 | 1000 | Yan Junxu | 53.0 | China | Taichang | Manufacturing | Manufacturing | True | M | 1969.0 | 7.0 | 17.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 2442 | 2540 | 1000 | Yu Rong | 51.0 | China | Shanghai | Health clinics | Healthcare | True | M | 1971.0 | 12.0 | 14.0 | 125.08 | 2.9 | \$19,910,000,000,000 |
| 2444 | 2540 | 1000 | Zhang Gongyun | 60.0 | China | Gaomi | Tyre manufacturing machinery | Manufacturing | True | M | 1962.0 | 12.0 | 18.0 | 125.08 | 2.9 | \$19,910,000,000,000 |

-> Các tỷ phú có độ tuổi từ 50 đến 60 tuổi có khối tài sản dao động từ khoảng 1 tỷ đến dưới 200 tỷ.

-> Từ các cụm trên, ta thấy được khối tài sản tích lũy của các tỷ phú trên thế giới dù ở độ tuổi nào thì mức chênh lệch không nhiều.

Lưu dataframe df2 có chứa cột Cluster thành file excel

```
[ ] # Lưu DataFrame df2 thành file excel
df2.to_excel('/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df2_Billionaires2023.xlsx', index=False)
```

5. Luật kết hợp

Đọc tệp excel 'df2_Billionaires2023.xlsx' thu được từ bước phân cụm dữ liệu trên vào dataframe data.

Tiếp theo, chuyển đổi dữ liệu thành dạng One-Hot Encoding.

- Code:

```
# Chuyển đổi dữ liệu thành dạng One-Hot Encoding
data_encoded = data.drop('Cluster', 1).applymap(lambda x: True if x == 1 else False)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` and `should_run_async(code)`
```

Hình. Dòng lệnh thực hiện chuyển đổi dạng dữ liệu

Sử dụng thuật toán Apriori để tìm tập phổ biến (frequent itemsets) từ dữ liệu đã được chuẩn hóa. Sau đó sử dụng frequent itemsets đã được tìm thấy để tạo ra các luật kết hợp.

Ngưỡng tối thiểu cho độ tin cậy là 0.5, chỉ giữ lại các luật có độ tin cậy lớn hơn hoặc bằng 0.5.

```
# Áp dụng thuật toán Apriori để tìm các luật kết hợp
frequent_itemsets = apriori(data_encoded, min_support=0.1, use_colnames=True)

# Tìm các luật kết hợp dựa trên frequent itemsets và độ tin cậy
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)
# Có thể điều chỉnh 'min_threshold' tùy vào nhu cầu! Trong khoảng 0 -> 1 tương ứng với độ tin cậy của kết quả đầu ra
# Một giá trị ngưỡng cao sẽ tạo ra ít luật hơn nhưng có độ tin cậy cao, trong khi một giá trị ngưỡng thấp sẽ tạo ra nhiều luật hơn nhưng có độ tin cậy thấp.
```

Hình. Các dòng lệnh thực hiện yêu cầu trên

In kết quả:

- Code:

```
# In kết quả
print("Tập phổ biến:")
print(frequent_itemsets)

print("\nLuật kết hợp:")
print(rules)
```

- Kết quả:

Tập phổ biến:

| | support | itemsets |
|---|----------|------------------------|
| 0 | 0.696772 | (SELFMADE) |
| 1 | 0.206784 | (BIRTHMONTH) |
| 2 | 0.255415 | (BIRTHDAY) |
| 3 | 0.145893 | (BIRTHMONTH, SELFMADE) |
| 4 | 0.192889 | (SELFMADE, BIRTHDAY) |
| 5 | 0.140172 | (BIRTHMONTH, BIRTHDAY) |

Luật kết hợp:

| | antecedents | consequents | antecedent support | consequent support | \ |
|---|--------------|--------------|--------------------|--------------------|---|
| 0 | (BIRTHMONTH) | (SELFMADE) | 0.206784 | 0.696772 | |
| 1 | (BIRTHDAY) | (SELFMADE) | 0.255415 | 0.696772 | |
| 2 | (BIRTHMONTH) | (BIRTHDAY) | 0.206784 | 0.255415 | |
| 3 | (BIRTHDAY) | (BIRTHMONTH) | 0.255415 | 0.206784 | |

| | support | confidence | lift | leverage | conviction | zhangs_metric |
|---|----------|------------|----------|----------|------------|---------------|
| 0 | 0.145893 | 0.705534 | 1.012575 | 0.001812 | 1.029756 | 0.015657 |
| 1 | 0.192889 | 0.755200 | 1.083856 | 0.014923 | 1.238678 | 0.103908 |
| 2 | 0.140172 | 0.677866 | 2.653979 | 0.087356 | 2.311412 | 0.785671 |
| 3 | 0.140172 | 0.548800 | 2.653979 | 0.087356 | 1.758015 | 0.836986 |

Hình. Kết quả sau khi sử dụng thuật toán Apriori để tìm các luật kết hợp

Lưu dataframe rules có chứa danh sách các luật kết hợp thành file excel.

```
# Lưu ra file excel
rules.to_excel("test_res0.5.xlsx")
```

Sau khi sử dụng thuật toán Apriori lên bộ dữ liệu, nhóm em rút ra được 4 luật:

| antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction | zhangs_metric |
|---------------------------|---------------------------|--------------------|--------------------|-------------|-------------|-------------|-------------|-------------|---------------|
| frozenset({'BIRTHMONTH'}) | frozenset({'SELFMADE'}) | 0.206783817 | 0.696771557 | 0.14589293 | 0.705533597 | 1.012575197 | 0.001811848 | 1.029755652 | 0.015656546 |
| frozenset({'BIRTHDAY'}) | frozenset({'SELFMADE'}) | 0.255414794 | 0.696771557 | 0.192889252 | 0.7552 | 1.083855953 | 0.014923489 | 1.23867828 | 0.103907747 |
| frozenset({'BIRTHDAY'}) | frozenset({'BIRTHMONTH'}) | 0.255414794 | 0.206783817 | 0.140171639 | 0.5488 | 2.653979447 | 0.087355993 | 1.75801459 | 0.836985916 |
| frozenset({'BIRTHMONTH'}) | frozenset({'BIRTHDAY'}) | 0.206783817 | 0.255414794 | 0.140171639 | 0.677865613 | 2.653979447 | 0.087355993 | 2.311411745 | 0.785671478 |

Từ bảng trên ta nhận xét các luật đã rút ra được:

-[BIRTHMONTH] -> [SELFMADE] với Confidence = 0.7

-[BIRTHDAY] -> [SELFMADE] với Confidence = 0.75

-[BIRTHDAY] -> [BIRTHMONTH] với Confidence = 0.55

=> Từ đó, chúng ta thấy được rằng antecedents tăng lên thì consequents giữ nguyên hoặc giảm xuống nên độ tin cậy của nó không cao.

6. KNN Và Naive-Bayes

Cũng từ file dữ liệu đã có, ta xác định đường dẫn và đọc lại tệp.

Hình. Dữ liệu tệp.

- Tiếp theo, xóa cột 'RANK' vì không cần thiết cho việc phân loại. Ta chia chia dữ liệu thành features (X) và nhãn (y) với cột 'Cluster'.

| RANK | NETWORTH | NAME | AGE | COUNTRY | CITY | SOURCE | INDUSTRY | SELFMADE | GENDER | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | GPD_COUNTRY |
|------|----------|-------------------|------|---------------|-------------|--------------------|-----------------------|----------|--------|-----------|------------|----------|-------------|--------------------|----------------------|
| 1 | 211000 | Bernard Arnault | 74.0 | France | Paris | LVMH | Fashion & Retail | False | M | 1949.0 | 3.0 | 5.0 | 110.05 | 1.1 | \$2,715,518,274,227 |
| 2 | 180000 | Elon Musk | 51.0 | United States | Austin | Tesla, SpaceX | Automotive | True | M | 1971.0 | 6.0 | 28.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 3 | 114000 | Jeff Bezos | 59.0 | United States | Medina | Amazon | Technology | True | M | 1964.0 | 1.0 | 12.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 4 | 107000 | Larry Ellison | 78.0 | United States | Lanai | Oracle | Technology | True | M | 1944.0 | 8.0 | 17.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 5 | 106000 | Warren Buffett | 92.0 | United States | Omaha | Berkshire Hathaway | Finance & Investments | True | M | 1930.0 | 8.0 | 30.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 6 | 104000 | Bill Gates | 67.0 | United States | Medina | Microsoft | Technology | True | M | 1955.0 | 10.0 | 28.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 7 | 94500 | Michael Bloomberg | 81.0 | United States | New York | Bloomberg LP | Media & Entertainment | True | M | 1942.0 | 2.0 | 14.0 | 117.24 | 7.5 | \$21,427,700,000,000 |
| 8 | 93000 | Carlos Slim Helu | 83.0 | Mexico | Mexico City | Telecom | Telecom | True | M | 1940.0 | 1.0 | 28.0 | 141.54 | 3.6 | \$1,258,286,717,125 |
| 9 | 83400 | Mukesh Ambani | 65.0 | India | Mumbai | Diversified | Diversified | False | M | 1957.0 | 4.0 | 19.0 | 180.44 | 7.7 | \$2,611,000,000,000 |
| 10 | 80700 | Steve Ballmer | 67.0 | United States | Hunts Point | Microsoft | Technology | True | M | 1956.0 | 3.0 | 24.0 | 117.24 | 7.5 | \$21,427,700,000,000 |

- Code:

```
# Xóa cột "RANK" vì không cần thiết cho việc phân loại
data.drop(["RANK", axis=1, inplace=True])

# Chia dữ liệu thành features (X) và nhãn (y)
X = data.drop("Cluster", axis=1)
y = data["Cluster"]

# Chuyển đổi các cột dữ liệu dạng văn bản sang dạng số hóa
X = pd.get_dummies(X)
# X.to_excel('/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/demo_Billionaires2023.xlsx', index=False)
X.to_csv("/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/demo_Billionaires2023.csv")
```

Hình. Các dòng lệnh thực hiện yêu cầu trên

- Tiếp theo nữa, ta sử dụng KNN để phân loại. Chia dữ liệu thành hai phần huấn luyện và kiểm tra (test data – train data)
 - o Khởi tạo mô hình với KNN = 4
 - o Dự đoán nhãn cho tập kiểm tra
 - o Đánh giá mô hình KNN

Khởi tạo mô hình KNN với k = 4

- knn_model =
KNeighborsClassifier(n_neighbors=3)
- knn_model.fit(X_train, y_train)

Dự đoán nhãn cho tập kiểm tra

- y_pred = knn_model.predict(X_test)

Đánh giá mô hình KNN

- print("KNN Model Evaluation:")
- print(confusion_matrix(y_test, y_pred))
- print(classification_report(y_test, y_pred))

KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)

```
KNN Model Evaluation:
[[38  4  2 11 11]
 [ 6  9  4 12  1]
 [ 7  0 10  1 15]
 [20  6  2 15  3]
 [29  2  2  5 30]]
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.38 | 0.58 | 0.46 | 66 |
| 1 | 0.43 | 0.28 | 0.34 | 32 |
| 2 | 0.50 | 0.30 | 0.38 | 33 |
| 3 | 0.34 | 0.33 | 0.33 | 46 |
| 4 | 0.50 | 0.44 | 0.47 | 68 |
| accuracy | | | 0.42 | 245 |
| macro avg | 0.43 | 0.39 | 0.40 | 245 |
| weighted avg | 0.43 | 0.42 | 0.41 | 245 |

Nhờ đây ta thu được kết quả, với accuracy = 0.42 (hoặc 42%) với k = 4 điều này chỉ ra rằng mô hình dự đoán đúng 42% trên tổng số lượng mẫu được sử dụng để đánh giá hiệu suất của nó.

- Tách dữ liệu thành hai phần tập dữ liệu huấn luyện và tập dữ liệu kiểm tra với tỷ lệ 90% - 10% và đảm bảo tỷ lệ các nhãn ngang nhau (nếu bên tập huấn luyện thì nhãn các nhãn ngang nhau, tập kiểm tra cũng vậy)
- Kiểm tra kết quả phân lớp – kiểm tra mô hình KNN trên tập dữ liệu mới (Test data).
 - o Code:

```

# Kiểm tra mô hình KNN trên tập dữ liệu mới (TestData)
# Đọc dữ liệu từ tập tin TrainData.csv và TestData.csv
train_data = pd.read_csv("TrainData.csv")
test_data = pd.read_csv("TestData.csv")

# Chia dữ liệu huấn luyện và kiểm tra thành features (X) và nhãn (y)
X_train = train_data.drop("Cluster", axis=1)
y_train = train_data["Cluster"]
X_test = test_data.drop("Cluster", axis=1)

# Chuyển đổi các cột dữ liệu dạng văn bản sang dạng số hóa
X_train = pd.get_dummies(X_train)
X_test = pd.get_dummies(X_test)

# Khởi tạo mô hình KNN với k=5 và huấn luyện trên toàn bộ dữ liệu huấn luyện
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
print(y_test)
# Dự đoán nhãn cho tập dữ liệu mới (TestData)
y_pred = knn_model.predict(X_test)

# In kết quả dự đoán
print("KNN Classification Results on New Data:")
print(y_pred)

```

○ Kết quả:

```

2036    0
521      3
2022     1
433      0
834      1
..
2         4
746      0
233      3
518      0
1424     0
Name: Cluster, Length: 245, dtype: int64
KNN Classification Results on New Data:
[4 0 3 0 3 0 3 3 0 3 0 0 0 1 0 3 0 3 4 2 0 0 4 3 4 4 4 0 3 0 0 0 3 4 0 4 0
 3 0 3 0 4 4 0 3 0 0 0 0 4 0 0 3 4 1 4 0 0 0 4 2 3 4 3 3 0 3 4 3 4 0 0 3 0
 2 0 4 0 2 4 0 4 4 3 0 4 0 3 3 0 3 0 1 4 1 0 0 3 0 1 3 2 4 4 3 3 0 3 0 1 0
 0 4 1 3 4 0 4 4 1 3 4 0 4 0 3 3 3 3 4 4 3 4 3 3 0 0 1 4 3 4 0 4 0 3 0 0 0
 0 4 3 4 0 1 4 0 0 0 4 3 4 4 3 4 3 4 1 2 4 3 4 3 0 0 3 2 4 2 3 0 0 4 1 3 2
 4 2 3 3 3 2 0 1 3 0 4 0 0 4 1 0 3 3 3 0 0 4 3 4 0 4 0 4 3 3 0 3 3 0 3 4 0
 0 0 4 3 4 0 0 4 2 1 0 0 0 2 3 0 3 4 0 0 0 1 4]

```

Hình. Đây là kết quả ta thu được khi phân loại KNN trên dữ liệu mới

- Sử dụng Naive-Bayes để phân loại.

Khởi tạo mô hình Naive-Bayes

- `nb_model = GaussianNB()`
- `nb_model.fit(X_train, y_train)`

Dự đoán nhãn cho tập kiểm tra

- `y_pred_nb = nb_model.predict(X_test)`

Đánh giá mô hình Naive-Bayes

- `print("Naive-Bayes Model Evaluation on Test Data:")`
- `print(confusion_matrix(y_test, y_pred_nb))`
- `print(classification_report(y_test, y_pred_nb))`

Naive-Bayes Model Evaluation on Test Data:

```
[[50  0  0  0 16]
 [28  0  0  2  2]
 [20  0  0  0 13]
 [34  0  0  1 11]
 [39  0  0  2 27]]
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.29 | 0.76 | 0.42 | 66 |
| 1 | 0.00 | 0.00 | 0.00 | 32 |
| 2 | 0.00 | 0.00 | 0.00 | 33 |
| 3 | 0.20 | 0.02 | 0.04 | 46 |
| 4 | 0.39 | 0.40 | 0.39 | 68 |
| accuracy | | | 0.32 | 245 |
| macro avg | 0.18 | 0.24 | 0.17 | 245 |
| weighted avg | 0.22 | 0.32 | 0.23 | 245 |

Nhờ đây ta thu được kết quả, với accuracy = 0.32 (hoặc 32%), điều này chỉ ra rằng mô hình dự đoán đúng 32% trên tổng số lượng mẫu được sử dụng để đánh giá hiệu suất của nó.

7. Cây quyết định

- Đọc dữ liệu file

```
# Đọc dữ liệu từ file df1_Billionaire23.xls
data = pd.read_excel("/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/df2_Billionaires2023.xlsx")
```

- Chọn các cột đặc trưng và cột nhãn

- Như là:

```
["RANK", "NETWORTH", "NAME", "AGE", "COUNTRY", "CITY", "SOURCE",
 "INDUSTRY", "GENDER", "BIRTHYEAR", "BIRTHMONTH", "BIRTHDAY",
 "CPI_COUNTRY", "cpi_change_country", "GPD_COUNTRY",
```

"TAX_REVENUE_COUNTRY", "TOTAL_TAX_RATE_COUNTRY",
"POPULATION_COUNTRY", "SELFMADE"]

- Chuyển đổi các cột dữ liệu dạng văn bản thành dạng số (phải thực hiện để huấn luyện mô hình)
 - Code: `features = pd.get_dummies(features)`
`features`
 - Kết quả:

| | RANK | NETWORTH | AGE | BIRTHYEAR | BIRTHMONTH | BIRTHDAY | CPI_COUNTRY | cpi_change_country | TAX_REVENUE_COUNTRY |
|------|------|----------|------|-----------|------------|----------|-------------|--------------------|---------------------|
| 0 | 1 | 211000 | 74.0 | 1949.0 | 3.0 | 5.0 | 110.05 | 1.1 | 24.2 |
| 1 | 2 | 180000 | 51.0 | 1971.0 | 6.0 | 28.0 | 117.24 | 7.5 | 9.6 |
| 2 | 3 | 114000 | 59.0 | 1964.0 | 1.0 | 12.0 | 117.24 | 7.5 | 9.6 |
| 3 | 4 | 107000 | 78.0 | 1944.0 | 8.0 | 17.0 | 117.24 | 7.5 | 9.6 |
| 4 | 5 | 106000 | 92.0 | 1930.0 | 8.0 | 30.0 | 117.24 | 7.5 | 9.6 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2442 | 2540 | 1000 | 51.0 | 1971.0 | 12.0 | 14.0 | 125.08 | 2.9 | 9.4 |
| 2443 | 2540 | 1000 | 80.0 | 1943.0 | 3.0 | 10.0 | 117.24 | 7.5 | 9.6 |
| 2444 | 2540 | 1000 | 60.0 | 1962.0 | 12.0 | 18.0 | 125.08 | 2.9 | 9.4 |
| 2445 | 2540 | 1000 | 71.0 | 1951.0 | 8.0 | 21.0 | 125.08 | 2.9 | 9.4 |
| 2446 | 2540 | 1000 | 66.0 | 1956.0 | 11.0 | 1.0 | 129.61 | 2.5 | 14.0 |

Hình. Dữ liệu đã được chuyển đổi.

- Thay thế ký tự không hợp lệ
 - Như là: `[^a-zA-Z0-9]`
- Chia tập dữ liệu thành tập train (80%) và tập test (20%):
 - Xây dựng mô hình Cây quyết định (J48) và định nghĩa các giá trị tham số cần thử nghiệm, tạo mô hình Cây quyết định (J48) và sử dụng GridSearchCV để thử nghiệm các tham số và lựa chọn mô hình tốt nhất.
 - Code:

```
# Xây dựng mô hình Cây quyết định (J48) và định nghĩa các giá trị tham số cần thử nghiệm
param_grid = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Tạo mô hình Cây quyết định (J48)
clf = DecisionTreeClassifier()

# Sử dụng GridSearchCV để thử nghiệm các tham số và lựa chọn mô hình tốt nhất
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring='accuracy')
grid_search.fit(train_features, train_labels)
```

Hình. Các dòng lệnh thực hiện yêu cầu.

- Kết quả thu được:

```
GridSearchCV
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                          'max_depth': [None, 10, 20, 30],
                          'min_samples_leaf': [1, 2, 4],
                          'min_samples_split': [2, 5, 10],
                          'splitter': ['best', 'random']}},
             scoring='accuracy')
  estimator: DecisionTreeClassifier
    DecisionTreeClassifier()
      DecisionTreeClassifier
        DecisionTreeClassifier()
```

Hình. Sử dụng GridSearchCV

- Lựa chọn mô hình tốt nhất sau khi thử nghiệm.

```
best_clf = grid_search.best_estimator_
```

- In ra dòng thông tin tốt nhất cho mô hình

```
Best Parameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf':
1, 'min_samples_split': 2, 'splitter': 'best'}
```

```
Best Accuracy: 1.0
```

- Tạo biểu đồ biểu diễn cây quyết định

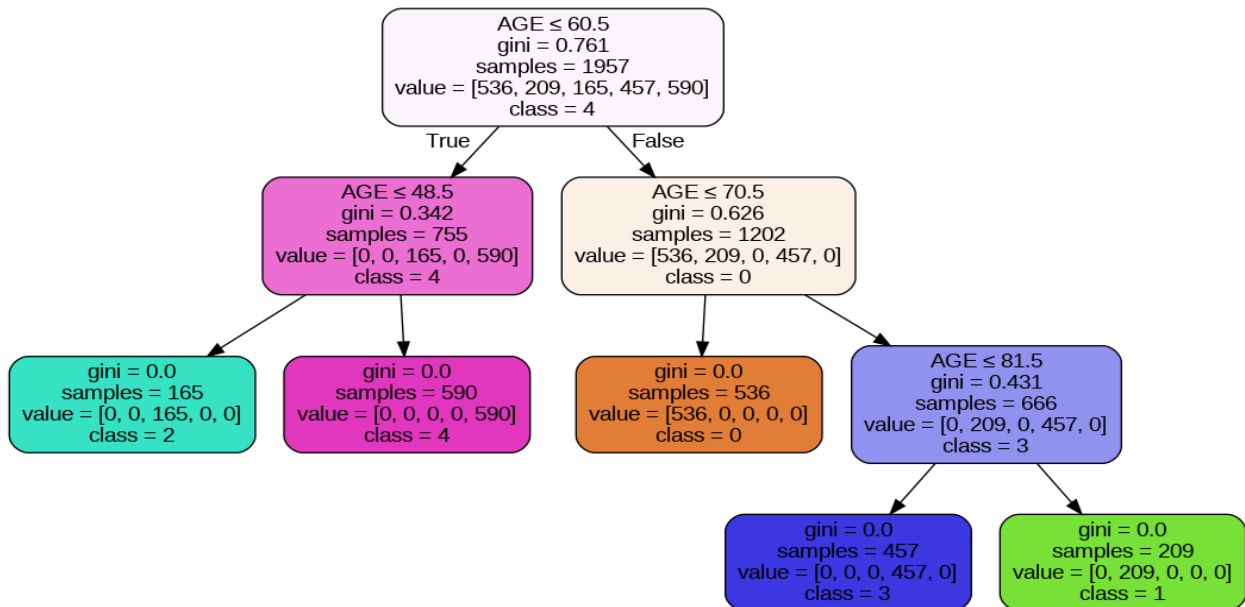
- Code:

```
# Tạo biểu diễn đồ thị cho cây quyết định
dot_data = export_graphviz(best_clf, out_file=None, feature_names=list(train_features.columns),
                           class_names=list(map(str, best_clf.classes_)), filled=True, rounded=True,
                           special_characters=True)

# Hiển thị đồ thị
graph = graphviz.Source(dot_data)
graph.render(filename='/content/drive/MyDrive/Colab Notebooks/KPDL_BTL/Res/ds01', format='png', cleanup=True)
graph.view()
```

Hình. Các dòng lệnh thực hiện yêu cầu.

- Biểu đồ:



- Dự đoán kết quả trên tập Test bằng mô hình tốt nhất đã chọn

```
# Dự đoán kết quả trên tập Test bằng mô hình tốt nhất đã lựa chọn
predictions = best_clf.predict(test_features)
```

- Đánh giá hiệu suất của mô hình tốt nhất trên tập Test

```
# Đánh giá hiệu suất của mô hình tốt nhất trên tập Test
accuracy = accuracy_score(test_labels, predictions)
print("Accuracy on Test Set:", accuracy)
```

➤ Accuracy on Test Set: 1.0

8. Trục quan hóa dữ liệu

- Chuyển hóa dữ liệu cột 'GDP_COUNTRY'

```
# Dữ liệu có ký tự đặc biệt
currency_string = df_visualize['GPD_COUNTRY']

# Loại bỏ ký tự đặc biệt
cleaned_string = currency_string.replace('$', '').replace(',', '')

# Convert to Numeric
try:
    numeric_value = pd.to_numeric(cleaned_string)
    print("Numeric Value:", numeric_value)
except ValueError as e:
    print("Error:", e)
```

Hình. Các dòng lệnh loại bỏ ký tự đặc biệt và chuyển đổi kiểu ký tự

Ta nhận thấy cột dữ liệu 'GDP_COUNTRY' có ký tự đặc biệt và dữ liệu khác dữ liệu số. Ta loại bỏ ký tự đó và chuyển dạng dữ liệu trở về thành dạng dữ liệu số (numeric). Quá trình này quan trọng để có thể thực hiện phân tích dữ liệu tiếp theo.

- So sánh độ tuổi của các tỷ phú của các nước:

- Code:

```
# So sánh độ tuổi tỷ phú của các nước
Country = list(df_visualize.COUNTRY.unique())

Age = []

for i in Country:
    x = df_visualize[df_visualize.COUNTRY == i]
    Age.append(x["AGE"].mean())

d1 = pd.DataFrame({"COUNTRY":Country, "AGE":Age})
d1.sort_values("AGE", ascending=True, inplace=True)
```

```
#visuzal ization

plt.figure(figsize=(10,15))

sns.set(style="white")

sns.barplot(x="AGE", y="COUNTRY", data=d1, palette= "colorblind")
```

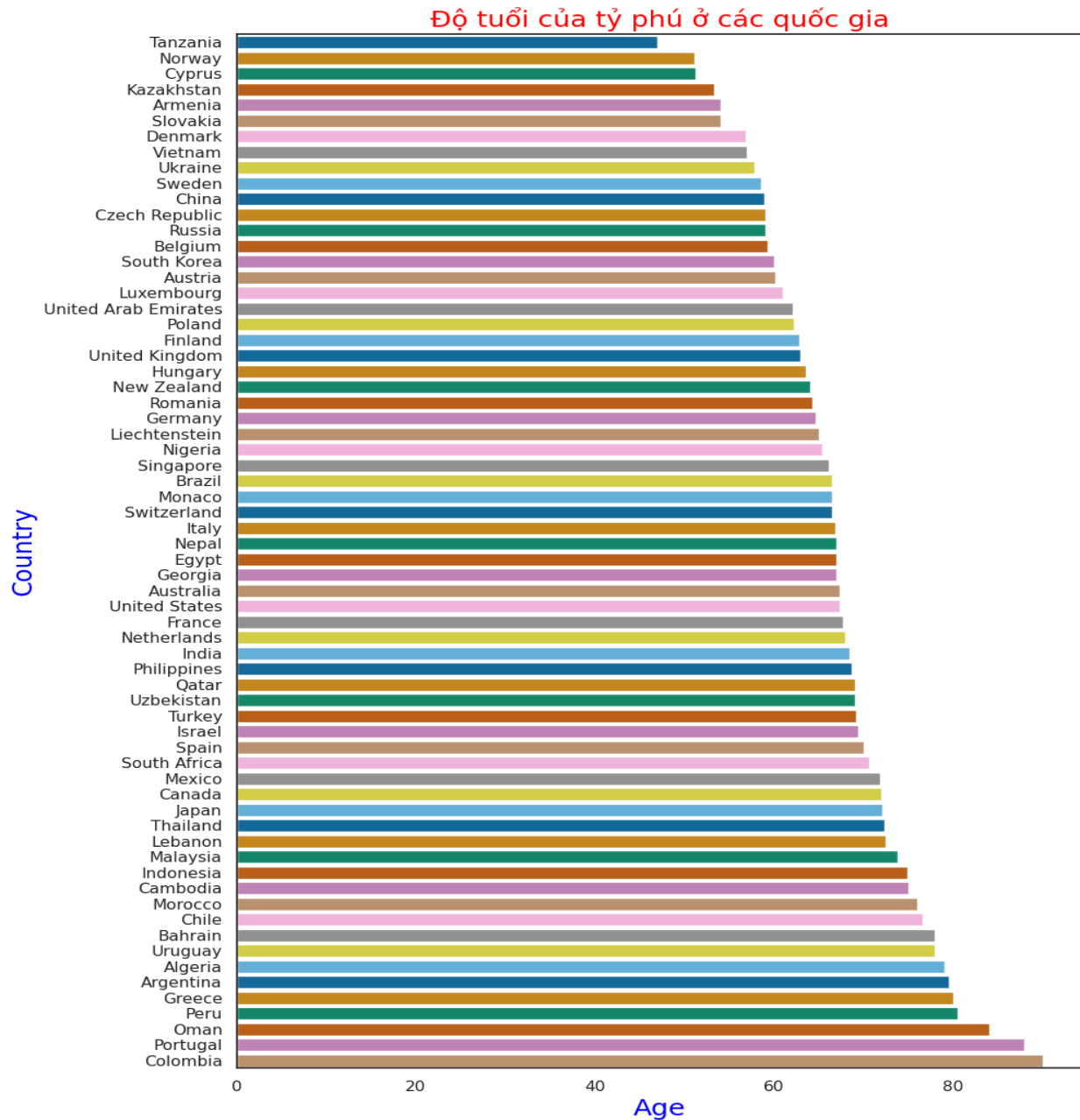


```
#sms.despine(left True, right True)

plt.xlabel("Age", fontsize=18, color="blue")
plt.ylabel("Country", fontsize=18, color="blue")
plt.title("Độ tuổi của tỷ phú ở các quốc gia", fontsize=18, color="red")

plt.show()
```

- Biểu đồ:



➤ Nhận xét:

- Ở Tanzania tỷ phú có tuổi cao nhất rơi vào khoảng 50 tuổi đến 60 tuổi, cho thấy ở quốc gia này phần lớn là các tỷ phú có độ tuổi khá trẻ và trung niên.
- Quốc gia có tỷ phú tuổi cao nhất là Colombia với độ tuổi rơi vào khoảng 90 tuổi đến 100 tuổi. (1)

- Chuyển hóa dữ liệu cột 'NETWORTH'

```
# Dữ liệu có ký tự đặc biệt
currency_string = df_visualize['NETWORTH']

# Loại bỏ ký tự đặc biệt
cleaned_string = currency_string.replace('$', '').replace(',', '')

# Convert to Numeric
try:
    numeric_value = pd.to_numeric(cleaned_string)
    print("Numeric Value:", numeric_value)
except ValueError as e:
    print("Error:", e)
```

Hình. Các dòng lệnh loại bỏ ký tự đặc biệt và chuyển đổi kiểu ký tự

Ta nhận thấy cột dữ liệu 'NETWORTH' có ký tự đặc biệt và dữ liệu khác dữ liệu số. Ta loại bỏ ký tự đó và chuyển dạng dữ liệu trở về thành dạng dữ liệu số (numeric). Quá trình này quan trọng để có thể thực hiện phân tích dữ liệu tiếp theo.

- So sánh khối tài sản của các tỷ phú giữa các ngành nghề
- Code:

```
# So sánh khối tài sản của tỷ phú giữa các ngành nghề
Industry = list(df_visualize.INDUSTRY.unique())

Networth = []

for i in Industry:
    x = df_visualize[df_visualize.INDUSTRY == i]
    Networth.append(x["NETWORTH"].mean())

d1 = pd.DataFrame({"INDUSTRY":Industry, "NETWORTH":Networth})

d1.sort_values("NETWORTH", ascending=True, inplace=True)

#visuzal ization

plt.figure(figsize=(10,15))

sns.set(style="white")

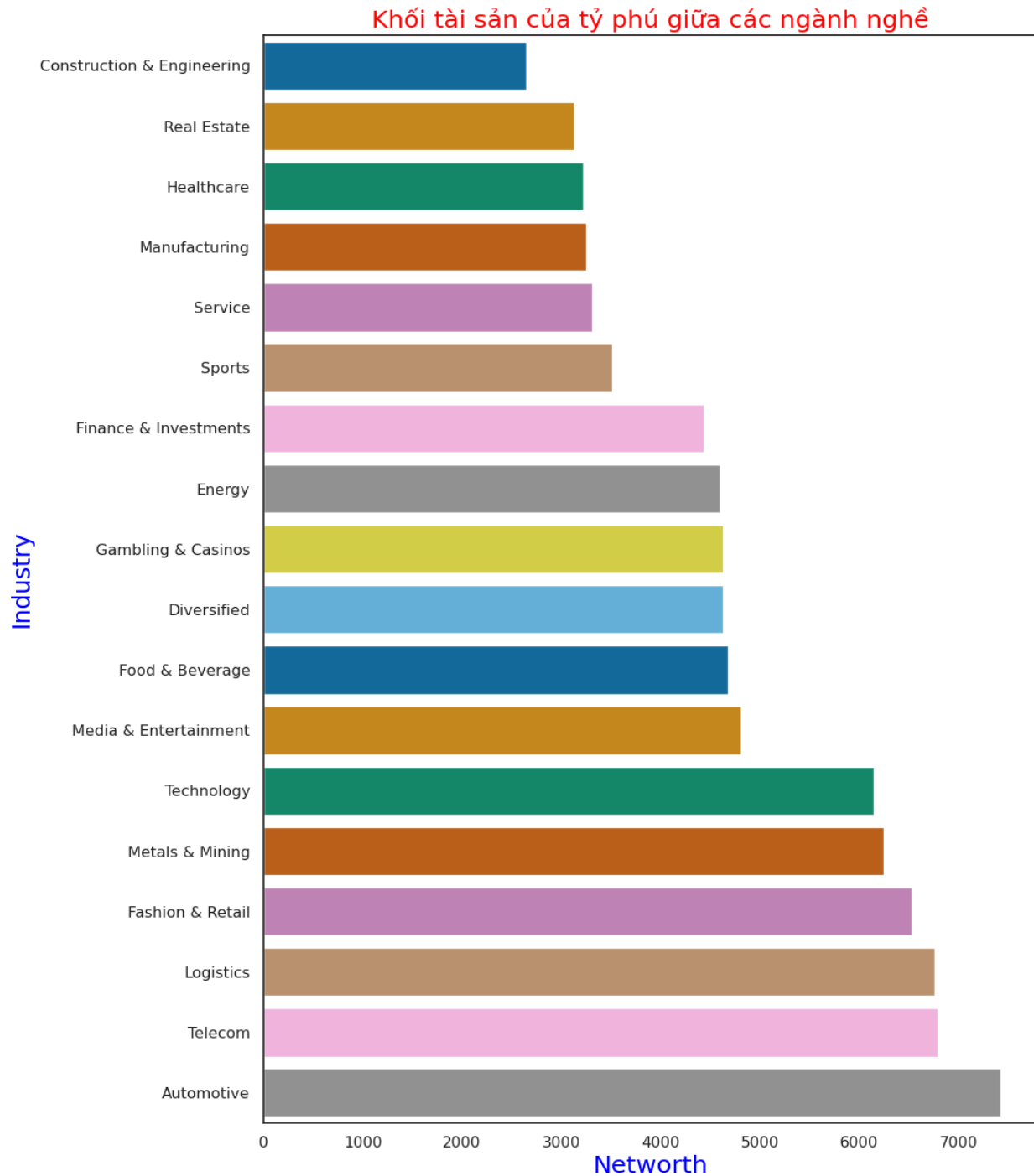
sns.barplot(x="NETWORTH", y="INDUSTRY", data=d1, palette= "colorblind")

#sns.despine(left True, right True)

plt.xlabel("Networth", fontsize=18, color="blue")
plt.ylabel("Industry", fontsize=18, color="blue")
plt.title("Khối tài sản của tỷ phú giữa các ngành nghề", fontsize=18, color="red")

plt.show()
```

- Biểu đồ:



➤ Nhận xét:

- Các tỷ phú tham gia lĩnh vực Construction & Engineering có khối tài sản thấp nhất với mức dao động vào khoảng 3000.

- Các tỷ phú tham gia lĩnh vực Automotive có khối tài sản cao nhất với mức dao động vào khoảng 7000 đến 8000. (2)
- Do ở trên đã chuyển đổi dữ liệu ở hai cột thuộc tính này nên không cần thực hiện lại bước này.
- So sánh khối tài sản của các tỷ phú ở các quốc gia
- Code:

```
# So sánh khối tài sản của tỷ phú ở các quốc gia
Country = list(df_visualize.COUNTRY.unique())

Networth = []

for i in Country:
    x = df_visualize[df_visualize.COUNTRY == i]
    Networth.append(x["NETWORTH"].mean())

d1 = pd.DataFrame({"COUNTRY": Country, "NETWORTH":Networth})

d1.sort_values("NETWORTH", ascending=True, inplace=True)

#visuzal ization

plt.figure(figsize=(10,15))

sns.set(style="white")

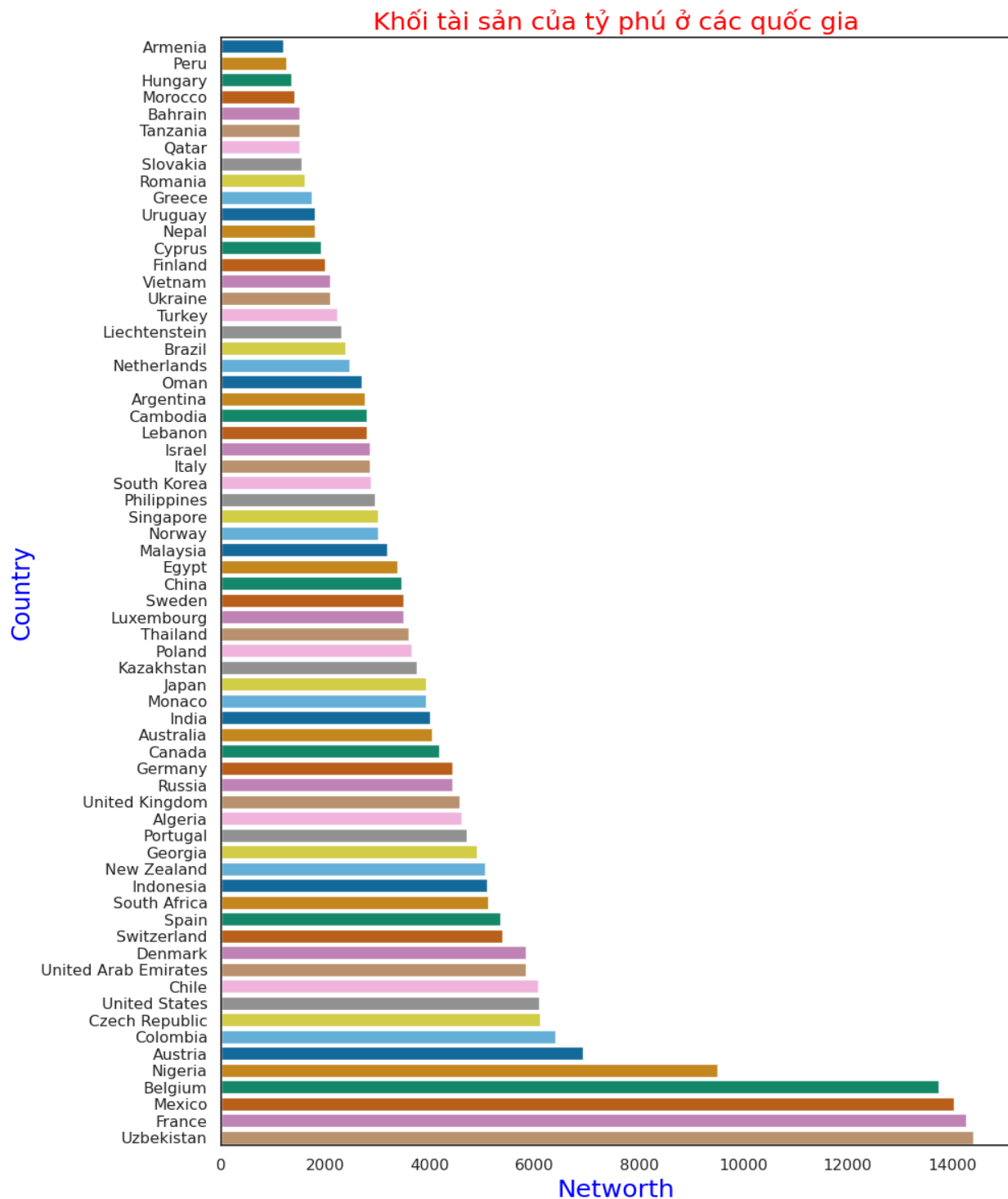
sns.barplot(x="NETWORTH", y="COUNTRY", data=d1, palette= "colorblind")

#sns.despine(left True, right True)

plt.xlabel("Networth", fontsize=18, color="blue")
plt.ylabel("Country", fontsize=18, color="blue")
plt.title("Khối tài sản của tỷ phú ở các quốc gia", fontsize=18, color="red")

plt.show()
```

- Biểu đồ:



➤ **Nhận xét:**

- Đối với quốc gia có khối tài sản của các tỷ phú thấp nhất là Armenia, quốc gia nằm kín trong lục địa tại nam Caucasus.

- Quốc gia chiếm tỷ lệ khối tài sản của các tỷ phú cao nhất là quốc gia Uzbekistan nằm ở khu vực trung đông với nền kinh tế tăng trưởng nhanh thứ 2 trên thế giới và sở hữu trữ lượng vàng lớn thứ tư toàn cầu. (3)

9. Nhận xét và đánh giá

Từ các nhận xét (1), (2) và (3), có thể thấy được các tỷ phú trên thế có độ tuổi trung bình từ 50 tuổi trở lên. Ngoài ra từ các nhận xét cũng cho thấy ngành Automotive là ngành có nhiều tỷ phú tham gia và số tài sản tích lũy của các tỷ phú tham gia rất lớn. Automotive có thể là một ngành được ưa chuộng đối với giới trẻ và những người muốn trở thành tỷ phú trong tương lai.

Một nhận xét khác là các tỷ phú ở các quốc gia có nền kinh tế phát triển sẽ có khối tài sản tích lũy lớn hơn so với các tỷ phú ở các quốc gia có nền kinh tế kém phát triển.

10. Tổng kết

Trong quá trình thực hiện bài báo cáo, mục tiêu của chúng em đã là cố gắng phân tích và hiểu rõ hơn về dữ liệu. Dùng công cụ Weka, Colab và các thuật toán tương ứng, chúng em đã cố gắng áp dụng kiến thức về khai thác dữ liệu và máy học để tìm ra các mô hình, mẫu và quy luật có thể được áp dụng cho thực tế.

Tuy nhiên, chúng em nhận thấy rằng việc xử lý và phân tích dữ liệu không phải lúc nào cũng dễ dàng như ta tưởng. Trong quá trình thực hiện, em đã gặp phải nhiều khó khăn. Việc lựa chọn các thuộc tính phù hợp, xử lý dữ liệu thiếu, đưa ra nhận xét và giải thích ý nghĩa thực sự của các mô hình là một thách thức không nhỏ.

Dù đã cố gắng hết sức, nhưng chúng em nhận thấy rằng kết quả từ các phân tích vẫn còn nhiều hạn chế. Có một số dữ liệu không đầy đủ hoặc không đủ thông tin để đưa ra các nhận xét chính xác. Một số thuật toán cũng không thể phân tích một số lớp mẫu một cách hiệu quả.

Tóm lại, dù có những khó khăn và hạn chế, quá trình thực hiện bài báo cáo đã giúp chúng em có cái nhìn tổng quan về việc áp dụng máy học vào việc khai thác dữ liệu. Chúng em

tin rằng, những kiến thức và kinh nghiệm học được từ bài tập lớn này sẽ góp phần vào việc tìm hiểu và ứng dụng các phương pháp này trong thực tế.

11. Tài liệu tham khảo

[1] Rob LaFranco và Chase Peterson-Withorn, "World's Billionaires List", 2023. [Trực tiếp].

Địa chỉ: Tỷ phú Forbes 2023: Những người giàu nhất thế giới. [Truy cập 13/01/2024].

[2] NIDULA ELGIRIYEWITHANA, "Billionaires Statistics Dataset (2023)", 2023. [Trực tiếp].

Địa chỉ: [Billionaires Statistics Dataset \(2023\) \(kaggle.com\)](#). [Truy cập 27/12/2023]

[3] Jemxyk, "Trực quan hóa dữ liệu với Tableau", 2023. [Trực tiếp]. Địa

chỉ: <https://bis.net.vn/forums/t/1815.aspx>. [Truy cập 10/01/2024]

[4] T.Bay , "Code mẫu và các bài thực hành", 2023. [Trực tiếp].