MỤC LỤC

CHƯƠNG 1: MÔ TẢ DỮ LIỆU

1.1.Bộ dữ liệu World Happiness Report 2015-2022.

Gồm các file: 2015.csv, 2016.csv, 2017.csv, 2018.csv, 2019.csv, 2020.csv, 2021.csv, 2022.csv.

Gồm các thuộc tính đặc trưng sau được giữ lại:

- **Country:** Tên quốc gia trong báo cáo.
- **Happiness Score:** điểm hạnh phúc của các quốc gia.
- **Economy (GDP per Capita):** Kinh tế (GDP bình quân đầu người của quốc gia).
- **Social support:** Hỗ trợ xã hội của quốc gia.
- **Health (Life Expectancy):** Sức khỏe (tuổi thọ) người dân của quốc gia.
- **Freedom:** sự tự do của quốc gia.
- **Trust (Government Corruption):** Niềm tin đối với chính phủ.
- **Generosity:** sự rộng lượng.
- **Year:** Năm báo cáo

Thông tin các dataset:

World Happiness Report 2015 - df15

```
RangeIndex: 158 entries, 0 to 157
Data columns (total 13 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        158 non-null    object
 1   Region                         158 non-null    object
 2   Happiness Rank                 158 non-null    int64
 3   Happiness Score                158 non-null    float64
 4   Standard Error                 158 non-null    float64
 5   Economy (GDP per Capita)       158 non-null    float64
 6   Family                         158 non-null    float64
 7   Health (Life Expectancy)       158 non-null    float64
 8   Freedom                        158 non-null    float64
 9   Trust (Government Corruption)  158 non-null    float64
 10  Generosity                     158 non-null    float64
 11  Dystopia Residual              158 non-null    float64
 12  Year                           158 non-null    int64
dtypes: float64(9), int64(2), object(2)
memory usage: 16.2+ KB
None
```

World Happiness Report 2016 - df16

```
Data columns (total 14 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        157 non-null    object
 1   Region                         157 non-null    object
 2   Happiness Rank                 157 non-null    int64
 3   Happiness Score                157 non-null    float64
 4   Lower Confidence Interval      157 non-null    float64
 5   Upper Confidence Interval      157 non-null    float64
 6   Economy (GDP per Capita)       157 non-null    float64
 7   Family                         157 non-null    float64
 8   Health (Life Expectancy)       157 non-null    float64
 9   Freedom                        157 non-null    float64
 10  Trust (Government Corruption)  157 non-null    float64
 11  Generosity                     157 non-null    float64
 12  Dystopia Residual              157 non-null    float64
 13  Year                           157 non-null    int64
dtypes: float64(10), int64(2), object(2)
```

World Happiness Report 2017 - df17

```
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Country                       155 non-null    object
 1   Happiness.Rank                155 non-null    int64
 2   Happiness.Score               155 non-null    float64
 3   Whisker.high                  155 non-null    float64
 4   Whisker.low                   155 non-null    float64
 5   Economy..GDP.per.Capita.      155 non-null    float64
 6   Family                        155 non-null    float64
 7   Health..Life.Expectancy.      155 non-null    float64
 8   Freedom                       155 non-null    float64
 9   Generosity                    155 non-null    float64
 10  Trust..Government.Corruption. 155 non-null    float64
 11  Dystopia.Residual             155 non-null    float64
 12  Year                          155 non-null    int64
dtypes: float64(10), int64(2), object(1)
```

World Happiness Report 2018 - df18

```
Data columns (total 10 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Overall rank                    156 non-null    int64
 1   Country or region               156 non-null    object
 2   Score                           156 non-null    float64
 3   GDP per capita                  156 non-null    float64
 4   Social support                  156 non-null    float64
 5   Healthy life expectancy         156 non-null    float64
 6   Freedom to make life choices    156 non-null    float64
 7   Generosity                      156 non-null    float64
 8   Perceptions of corruption       155 non-null    float64
 9   Year                            156 non-null    int64
dtypes: float64(7), int64(2), object(1)
```

World Happiness Report 2019 - df19

```
Data columns (total 10 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Overall rank                    156 non-null    int64
 1   Country or region               156 non-null    object
 2   Score                           156 non-null    float64
 3   GDP per capita                  156 non-null    float64
 4   Social support                  156 non-null    float64
 5   Healthy life expectancy         156 non-null    float64
 6   Freedom to make life choices    156 non-null    float64
 7   Generosity                      156 non-null    float64
 8   Perceptions of corruption       156 non-null    float64
 9   Year                            156 non-null    int64
dtypes: float64(7), int64(2), object(1)
```

World Happiness Report 2020 - df20

```
Data columns (total 21 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   Country name                              153 non-null    object 
 1   Regional indicator                        153 non-null    object 
 2   Ladder score                              153 non-null    float64
 3   Standard error of ladder score            153 non-null    float64
 4   upperwhisker                              153 non-null    float64
 5   lowerwhisker                              153 non-null    float64
 6   Logged GDP per capita                     153 non-null    float64
 7   Social support                            153 non-null    float64
 8   Healthy life expectancy                   153 non-null    float64
 9   Freedom to make life choices              153 non-null    float64
 10  Generosity                                153 non-null    float64
 11  Perceptions of corruption                 153 non-null    float64
 12  Ladder score in Dystopia                  153 non-null    float64
 13  Explained by: Log GDP per capita          153 non-null    float64
 14  Explained by: Social support              153 non-null    float64
 15  Explained by: Healthy life expectancy     153 non-null    float64
 16  Explained by: Freedom to make life choices  153 non-null    float64
 17  Explained by: Generosity                  153 non-null    float64
 18  Explained by: Perceptions of corruption   153 non-null    float64
 19  Dystopia + residual                       153 non-null    float64
 20  Year                                      153 non-null    int64  
dtypes: float64(18), int64(1), object(2)
```

World Happiness Report 2021 - df21

```
Data columns (total 21 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   Country name                              149 non-null    object 
 1   Regional indicator                        149 non-null    object 
 2   Ladder score                              149 non-null    float64
 3   Standard error of ladder score            149 non-null    float64
 4   upperwhisker                              149 non-null    float64
 5   lowerwhisker                              149 non-null    float64
 6   Logged GDP per capita                     149 non-null    float64
 7   Social support                            149 non-null    float64
 8   Healthy life expectancy                   149 non-null    float64
 9   Freedom to make life choices              149 non-null    float64
 10  Generosity                                149 non-null    float64
 11  Perceptions of corruption                 149 non-null    float64
 12  Ladder score in Dystopia                  149 non-null    float64
 13  Explained by: Log GDP per capita          149 non-null    float64
 14  Explained by: Social support              149 non-null    float64
 15  Explained by: Healthy life expectancy     149 non-null    float64
 16  Explained by: Freedom to make life choices  149 non-null    float64
 17  Explained by: Generosity                  149 non-null    float64
 18  Explained by: Perceptions of corruption   149 non-null    float64
 19  Dystopia + residual                       149 non-null    float64
 20  Year                                      149 non-null    int64  
dtypes: float64(18), int64(1), object(2)
```

World Happiness Report 2022 - df22

```
Data columns (total 13 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   RANK                                          147 non-null    int64
 1   Country                                       147 non-null    object
 2   Happiness score                               146 non-null    object
 3   Whisker-high                                  146 non-null    object
 4   Whisker-low                                   146 non-null    object
 5   Dystopia (1.83) + residual                    146 non-null    object
 6   Explained by: GDP per capita                  146 non-null    object
 7   Explained by: Social support                  146 non-null    object
 8   Explained by: Healthy life expectancy         146 non-null    object
 9   Explained by: Freedom to make life choices    146 non-null    object
 10  Explained by: Generosity                      146 non-null    object
 11  Explained by: Perceptions of corruption       146 non-null    object
 12  Year                                          147 non-null    int64
dtypes: int64(2), object(11)
```

1.2.Dữ liệu World Happiness Report 2023.

Tên file: WHR2023.csv.

Gồm các thuộc tính đặc trưng sau được giữ lại:

- **Country:** Tên quốc gia trong báo cáo.
- **Happiness Score:** điểm hạnh phúc của các quốc gia.
- **Economy (GDP per Capita):** Kinh tế (GDP bình quân đầu người của quốc gia).
- **Social support:** Hỗ trợ xã hội của quốc gia.
- **Health (Life Expectancy):** Sức khỏe (tuổi thọ) người dân của quốc gia.
- **Freedom:** Sự tự do của quốc gia.
- **Trust (Government Corruption):** Niềm tin đối với chính phủ.
- **Generosity:** Sự rộng lượng.

CHƯƠNG 2: LÀM SẠCH & CHUẨN HÓA DỮ LIỆU

Đầu tiên, ta thêm cột Year vào từng bảng Data.

```
df15['Year'] = 2015
df16['Year'] = 2016
df17['Year'] = 2017
df18['Year'] = 2018
df19['Year'] = 2019
df20['Year'] = 2020
df21['Year'] = 2021
df22['Year'] = 2022
```

Tiếp theo ta kiểm tra xem tên các cột đã đồng nhất hay chưa, nếu chưa ta cần đổi tên để các cột trong bảng dữ liệu được đồng nhất.

```
#Kiểm tra dữ liệu đã đồng nhất tên cột hay chưa?
sets = [df15, df16, df17, df18, df19, df20, df21, df22]
for i in sets:


    print(i.info())
```

Để tiếp tục cho việc chuẩn hóa dữ liệu, ta cần xác định các cột giá trị để chuẩn hóa. Đối với tên cột cần lấy của từng bảng dữ liệu, ta thực hiện chuẩn hóa như sau.

Năm 2015, cột 'Family' được đổi tên thành 'Social support'.

```
df15 = df15.rename(columns = {
    'Family' : 'Social support'})
```

Năm 2016, cột 'Family' tương đương với 'Social support'.

```
df16 = df16.rename(columns = {
    'Family' : 'Social support'})
```

Năm 2017:

- Cột 'Happiness.Score' tương ứng với 'Happiness Score'.
- Cột 'Economy..GDP.per.Capita.' tương ứng với 'Economy (GDP per Capita)'.
- Cột 'Health..Life.Expectancy.' tương ứng với cột 'Health (Life Expectancy)'.
- Cột'Trust..Government.Corruption.' tương ứng với cột 'Trust (Government Corruption)'.

- Cột ‘Family’ tương ứng với cột ‘Social support’.
- Cột ‘Dystopia.Residual’ tương ứng với cột ‘Dystopia Residual’.

```
df17 = df17.rename(columns = {
    'Happiness.Score' : 'Happiness Score',
    'Economy..GDP.per.Capita.' : 'Economy (GDP per Capita)',
    'Health..Life.Expectancy.' : 'Health (Life Expectancy)',
    'Trust..Government.Corruption.' : 'Trust (Government Corruption)',
    'Family' : 'Social support',
    'Dystopia.Residual' : 'Dystopia Residual'})
```

Năm 2018, các tên cột gốc được đổi thành các tên cột mới như sau:

- ‘Country or region’ đổi thành ‘Country’.
- ‘Score’ đổi thành ‘Happiness Score’.
- ‘GDP per capita’ đổi thành ‘Economy (GDP per Capita)’.
- ‘Healthy life expectancy’ đổi thành ‘Health (Life Expectancy)’.
- ‘Freedom to make life choices’ đổi thành ‘Freedom’.
- ‘Perceptions of corruption’ đổi thành ‘Trust (Government Corruption)’.

```
df18 = df18.rename(columns = {
    'Country or region' : 'Country',
    'Score' : 'Happiness Score',
    'GDP per capita' : 'Economy (GDP per Capita)',
    'Healthy life expectancy' : 'Health (Life Expectancy)',
    'Freedom to make life choices' : 'Freedom',
    'Perceptions of corruption' : 'Trust (Government Corruption)'})
```

Năm 2019 đổi tên tương tự với năm 2018.

```
df19 = df19.rename(columns = {
    'Country or region' : 'Country',
    'Score' : 'Happiness Score',
    'GDP per capita' : 'Economy (GDP per Capita)',
    'Healthy life expectancy' : 'Health (Life Expectancy)',
    'Freedom to make life choices' : 'Freedom',
    'Perceptions of corruption' : 'Trust (Government Corruption)'})
```

Năm 2020, Các tên cột gốc được đổi thành các tên cột mới như sau:

- ‘Country name’ đổi thành ‘Country’.
- ‘Ladder score’ đổi thành ‘Happiness Score’.
- ‘Freedom to make life choices’ đổi thành ‘Freedom’.
- ‘Perceptions of corruption’ đổi thành ‘Trust (Government Corruption)’.

```
df20 = df20.rename(columns = {
    'Country name' : 'Country',
    'Ladder score' : 'Happiness Score',
    'Freedom to make life choices' : 'Freedom',
    'Perceptions of corruption' : 'Trust (Government Corruption)'})
```

```
df20['Economy (GDP per Capita)'] = df20['Logged GDP per capita'] / 10
```

Dòng này thực hiện việc tạo một cột mới trong DataFrame df20 có tên ‘Economy (GDP per Capita)’. Giá trị của cột mới này được tính bằng cách chia giá trị trong cột ‘Logged GDP per capita’ cho 10.

```
df20['Health (Life Expectancy)'] = df20['Healthy life expectancy'] / 100
```

Dòng này tạo một cột mới trong DataFrame df20 có tên ‘Health (Life Expectancy)’. Giá trị của cột mới này được tính bằng cách chia giá trị trong cột ‘Healthy life expectancy’ cho 100.

Năm 2021, các tên cột gốc được đổi thành các tên cột mới như sau:

- ‘Country name’ đổi thành ‘Country’.
- ‘Ladder score’ đổi thành ‘Happiness Score’.
- ‘Freedom to make life choices’ đổi thành ‘Freedom’.
- ‘Perceptions of corruption’ đổi thành ‘Trust (Government Corruption)’.

```
df21 = df21.rename(columns = {
    'Country name' : 'Country',
    'Ladder score' : 'Happiness Score',
    'Freedom to make life choices' : 'Freedom',
    'Perceptions of corruption' : 'Trust (Government Corruption)'})
```

```
df21['Economy (GDP per Capita)'] = df21['Logged GDP per capita'] / 10
```

Dòng này tạo một cột mới trong DataFrame df21 có tên 'Economy (GDP per Capita)'. Giá trị của cột mới này được tính bằng cách chia giá trị trong cột 'Logged GDP per capita' cho 10.

```
df21['Health (Life Expectancy)'] = df21['Healthy life expectancy'] / 100
```

Dòng này tạo một cột mới trong DataFrame df21 có tên 'Health (Life Expectancy)'. Giá trị của cột mới này được tính bằng cách chia giá trị trong cột 'Healthy life expectancy' cho 100.

Năm 2022, các tên cột gốc được đổi thành các tên cột mới như sau:

- 'Happiness score' đổi thành 'Happiness Score'.
- 'Explained by: GDP per capita' đổi thành 'Economy (GDP per Capita)'.
- 'Explained by: Social support' đổi thành 'Social support'.
- 'Explained by: Healthy life expectancy' đổi thành 'Health (Life Expectancy)'.
- 'Explained by: Freedom to make life choices' đổi thành 'Freedom'.
- 'Explained by: Generosity' đổi thành 'Generosity'.
- 'Explained by: Perceptions of corruption' đổi thành 'Trust (Government Corruption)'.

```
df22 = df22.rename(columns = {
    'Happiness score' : 'Happiness Score',
    'Explained by: GDP per capita' : 'Economy (GDP per Capita)',
    'Explained by: Social support' : 'Social support',
    'Explained by: Healthy life expectancy' : 'Health (Life Expectancy)',
    'Explained by: Freedom to make life choices' : 'Freedom',
    'Explained by: Generosity' : 'Generosity',
    'Explained by: Perceptions of corruption' : 'Trust (Government Corruption)'})
```

Sau khi đã chuẩn hóa tên cột cho các bảng dữ liệu, ta gộp dữ liệu ở các bảng lại.

```
df = pd.concat([df15, df16, df17, df18, df19, df20, df21, df22])[[
    'Country',
    'Happiness Score',
    'Economy (GDP per Capita)',
    'Social support',
    'Health (Life Expectancy)',
    'Freedom',
    'Trust (Government Corruption)',
    'Generosity',
    'Year']]
```

Và xóa đi những dữ liệu không hợp lệ.

```
df = df.dropna(axis=0,thresh=3)
```

Tiếp theo ta tiến hành chuẩn hóa dữ liệu.

```
df['Happiness Score'] = df['Happiness Score'].apply(lambda x: str(x).replace(',', '.'))
```

Dòng này thay thế dấu phẩy bằng dấu chấm trong cột 'Happiness Score'. Ta sử dụng hàm apply để áp dụng một hàm xử lý cho từng giá trị trong cột. Trong trường hợp này, hàm xử lý là một hàm lambda thực hiện việc chuyển giá trị thành chuỗi, sau đó thay thế dấu phẩy bằng dấu chấm.

Tương tự, các dòng sau thực hiện việc thay thế dấu phẩy bằng dấu chấm trong các cột khác nhau: 'Economy (GDP per Capita)', 'Social support', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity'.

```
df['Economy (GDP per Capita)'] = df['Economy (GDP per Capita)'].apply(lambda x: str(x).replace(',', '.'))
df['Social support'] = df['Social support'].apply(lambda x: str(x).replace(',', '.'))
df['Health (Life Expectancy)'] = df['Health (Life Expectancy)'].apply(lambda x: str(x).replace(',', '.'))
df['Freedom'] = df['Freedom'].apply(lambda x: str(x).replace(',', '.'))
df['Trust (Government Corruption)'] = df['Trust (Government Corruption)'].apply(lambda x: str(x).replace(',', '.'))
df['Generosity'] = df['Generosity'].apply(lambda x: str(x).replace(',', '.'))
```

Công việc tiếp theo, ta sử dụng phương thức .astype() để thay đổi kiểu dữ liệu của các cột trong DataFrame ‘df’ .Phương thức .astype() có công dụng thay đổi kiểu dữ liệu của các cột trong DataFrame df theo đúng kiểu dữ liệu mà ta đã chỉ định. Ở đây ta đang chuyển đổi các cột thành kiểu dữ liệu float32, đây là một kiểu số thực (floating-point) với 32-bit, tiết kiệm bộ nhớ hơn so với kiểu float64 (64-bit).

Các cột được chuyển đổi kiểu dữ liệu gồm:

- ‘Happiness Score’ thành kiểu float32.
- ‘Economy (GDP per Capita)’ thành kiểu float32.
- ‘Social support’ thành kiểu float32.
- ‘Health (Life Expectancy)’ thành kiểu float32.
- ‘Freedom’ thành kiểu float32.
- ‘Trust (Government Corruption)’ thành kiểu float32.
- ‘Generosity’ thành kiểu float32.

Cuối cùng, ta xuất dữ liệu đã được làm sạch và chuẩn hóa để sử dụng trong mô hình huấn luyện và dự đoán dữ liệu.

```
df.to_csv('df1522.csv', index=False)
```

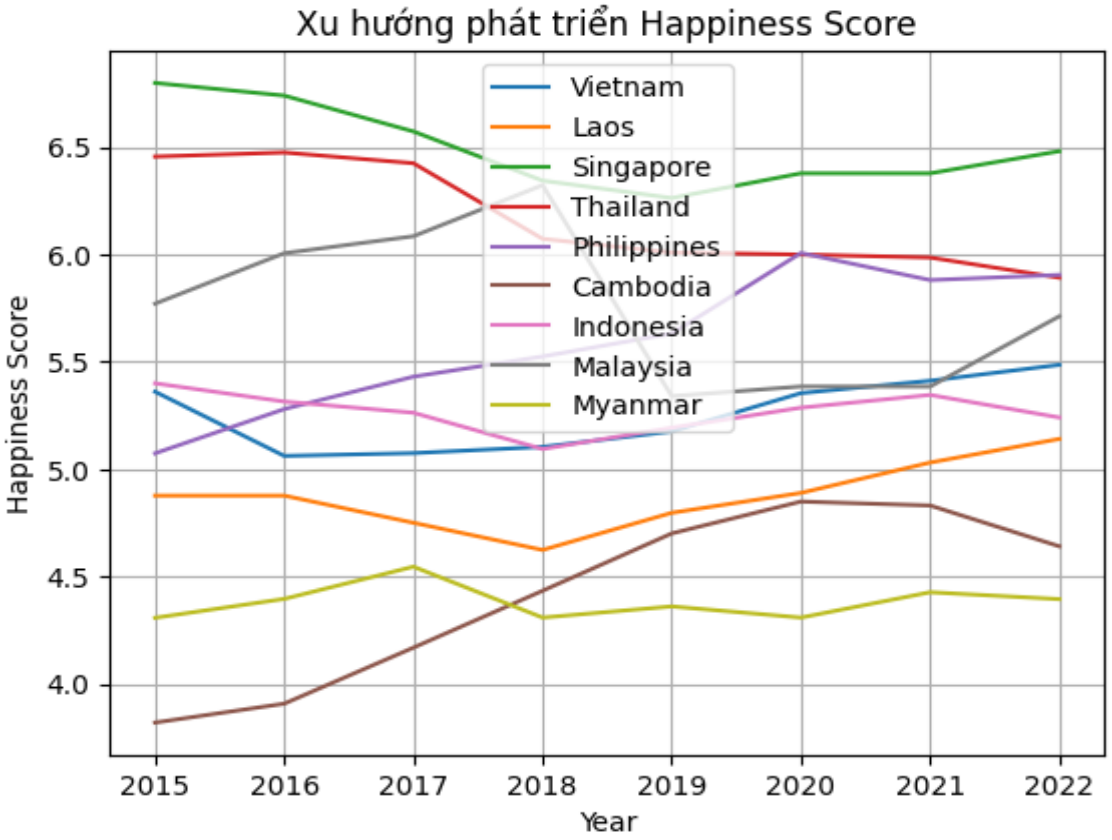
CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU

3.1.Phân tích dữ liệu và sử dụng các biểu đồ để mô tả dữ liệu

```
#Thay đổi màu sắc để dễ quan sát biểu đồ
color=["#f94144","#f3722c","#f8961e","#f9c74f","#90be6d","#43aa8b","#577590"]
sns.palplot(color)
```

3.1.1.Biểu đồ xem xét độ tăng trưởng Happiness Score của Việt Nam trong Đông Nam Á từ 2015 - 2022

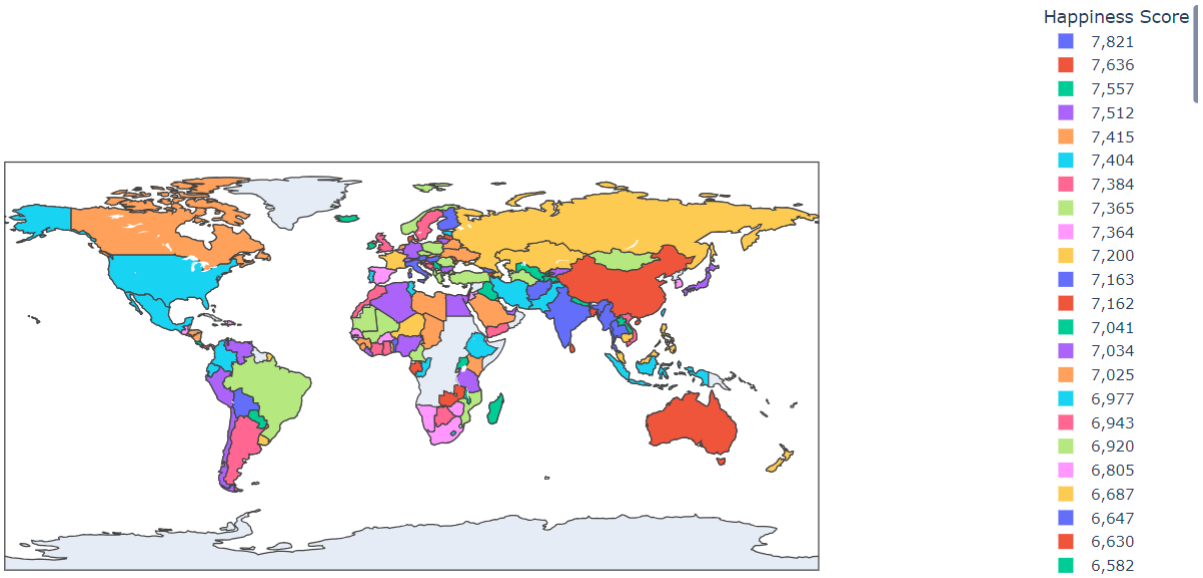
```
#Vẽ biểu đồ xem xét độ tăng trưởng Happiness Score của Việt Nam trong Đông Nam Á
columns_to_plot = df[['Country', 'Year', 'Happiness Score']]
grouped_data = df.groupby(['Country', 'Year'])['Happiness Score'].mean().reset_index()
list_of_countries = ['Vietnam', 'Laos', 'Singapore', 'Thailand', 'Philippines', 'Cambodia', 'Indonesia', 'Malaysia', 'Myanmar']
for country in list_of_countries:
    country_data = grouped_data[grouped_data['Country'] == country]
    plt.plot(country_data['Year'], country_data['Happiness Score'], label=country)
plt.xlabel('Year')
plt.ylabel('Happiness Score')
plt.title('Xu hướng phát triển Happiness Score')
plt.legend()
plt.grid(True)
plt.show()
```



➢ **Từ biểu đồ trên, ta thấy Việt Nam đang có chỉ số hạnh phúc được đánh giá dao động trong mức trung bình tại Đông Nam Á với số điểm Happiness Score từ 5.0 - 5.5. Tuy nhiên, Việt Nam đang có sự phát triển rõ rệt qua các năm từ 2016 - 2022.**

3.1.2.Bản đồ thế giới về Happiness Score Report 2022

```
fig = px.choropleth(data_frame = df22,
                    locations= 'Country',
                    locationmode='country names',
                    color= 'Happiness Score',
                    hover_name= "Country",
                    color_continuous_scale= 'RdYlGn',
                    width=1200,
                    height=600,)
fig.show()
```

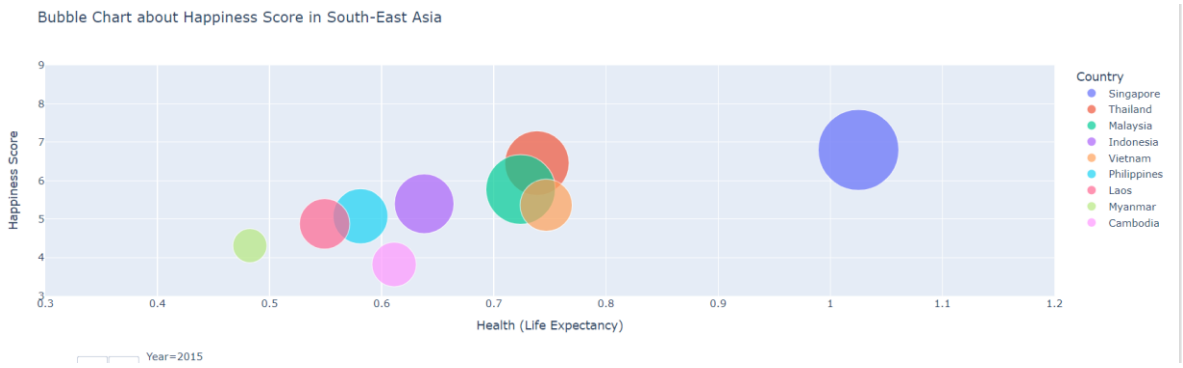


- **Từ bản đồ trên, ta thấy các nước trong khu vực Tây Âu và Bắc Mỹ có chỉ số Happiness Score cao trong khoảng 6.0 - 7.2.**
- **Những khu vực như Tây Âu và Bắc Mỹ là những khu vực đáng sống.**

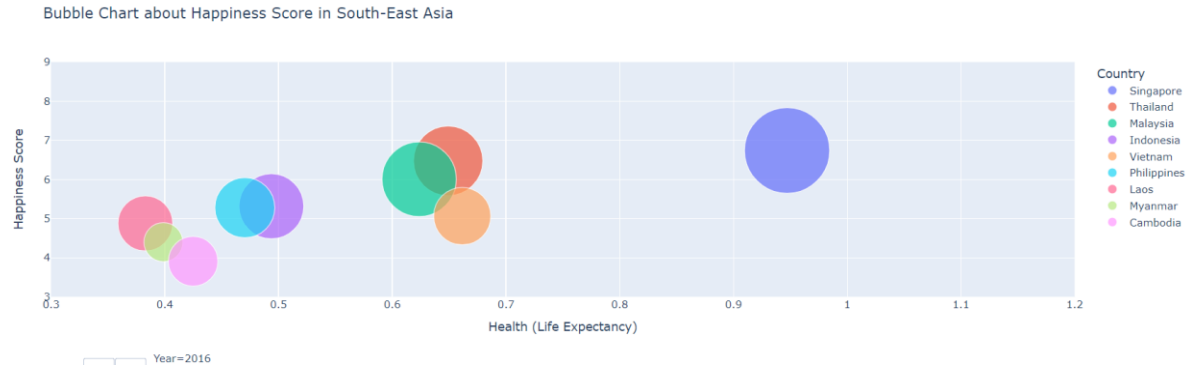
3.1.3.Biểu đồ bong bóng về Tuổi thọ (Life expectancy) và Kinh tế (GDP per Capita) với chỉ số Happiness Score của các nước trong khu vực Đông Nam Á.

```
short = df[df['Country'].isin (list_of_countries)]
fig = px.scatter(short.query("Year==2022"), x="Health (Life Expectancy)", y="Happiness Score",
      size="Economy (GDP per Capita)",
      color="Country",
      hover_name="Country", log_x=True, size_max=60)
fig.show()
```

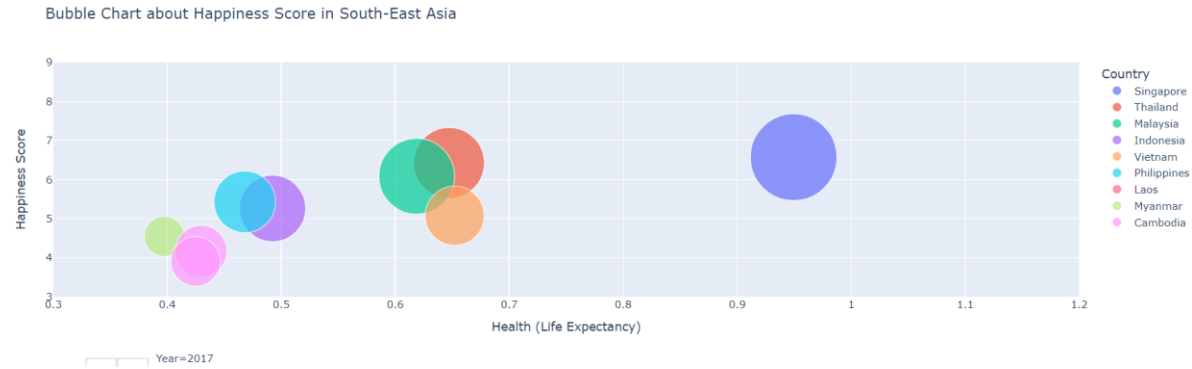
Năm 2015



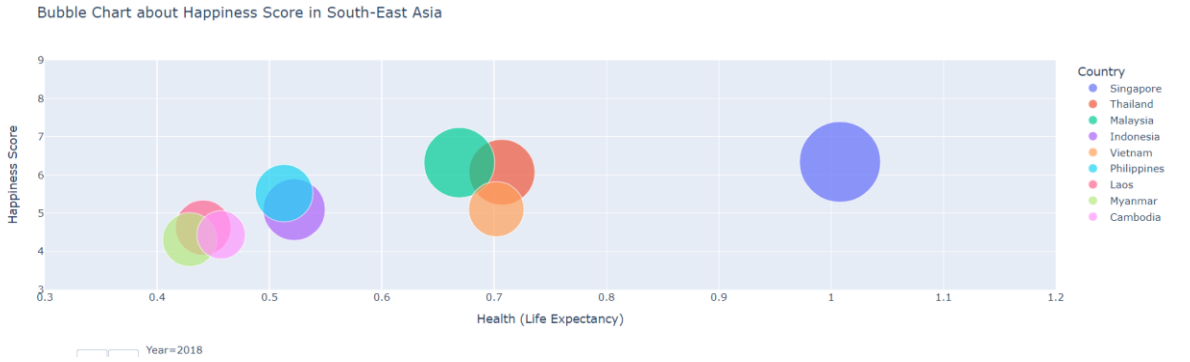
Năm 2016



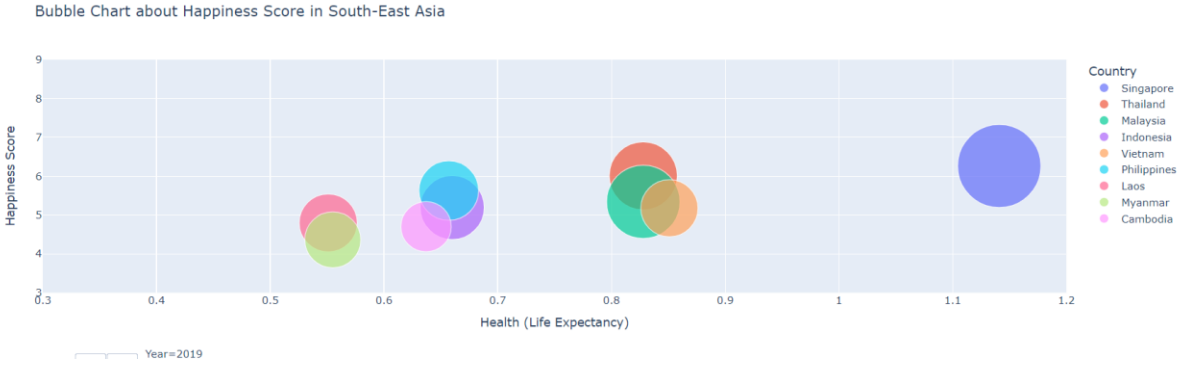
Năm 2017



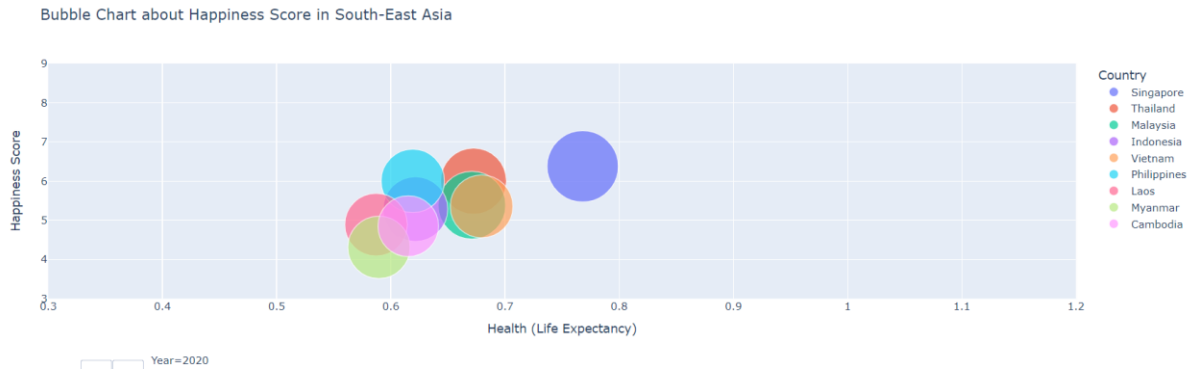
Năm 2018



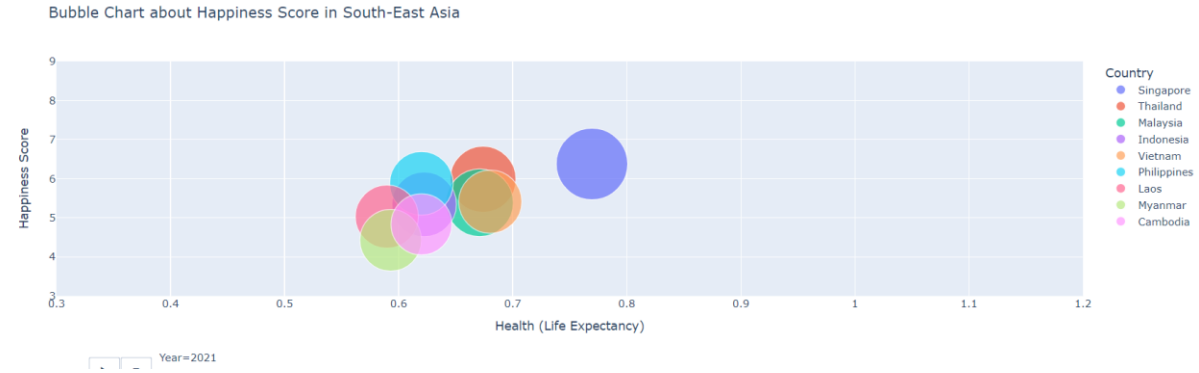
Năm 2019



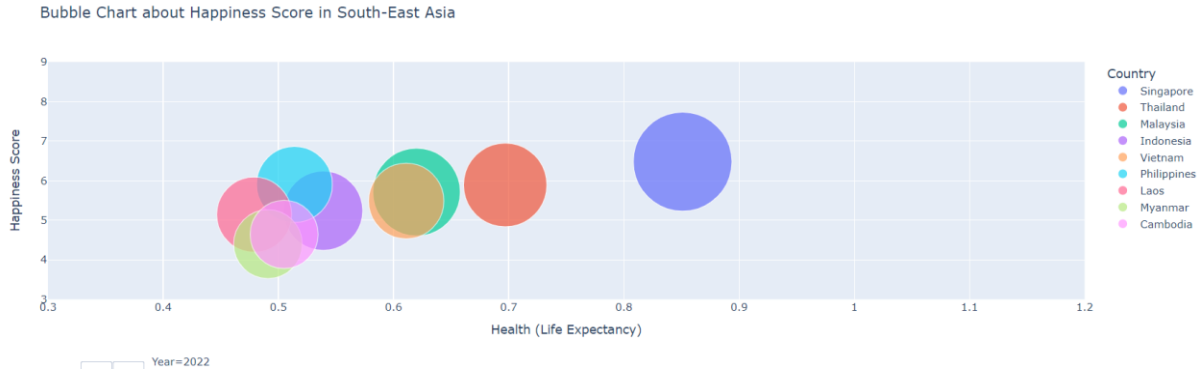
Năm 2020



Năm 2021



Năm 2022



- **Qua các biểu đồ từ năm 2015 đến năm 2022, ta thấy GDP của Việt Nam đang tăng dần theo từng năm.**
- **Trong năm 2022, Việt Nam đang thể hiện chỉ số Happiness Score, GDP và Health nằm trong nhóm trung bình tại Đông Nam Á.**

CHƯƠNG 4: XÂY DỰNG MÔ HÌNH

4.1.Mô tả thuật toán

Với các đặc trưng đã xác định cùng kiểu dữ liệu bao gồm các số, ta có thể sử dụng phương pháp hồi quy Lasso để áp dụng trong mô hình huấn luyện để đưa ra kết quả có độ tin cậy cao và sử dụng trong dự đoán chỉ số Happiness Score của các quốc gia trong tương lai.

Trong đó, mô hình hồi quy Lasso (Least Absolute Shrinkage and Selection Operator) là một phương pháp hồi quy tuyến tính trong thống kê và máy học, được sử dụng để giảm thiểu overfitting và thực hiện lựa chọn biến đầu vào trong mô hình. Mô hình Lasso là một biến thể của hồi quy tuyến tính thông thường, nhưng có một yếu tố chính khác biệt: nó thêm một hạng stricte số tuyệt đối của các trọng số hồi quy vào hàm mất mát.

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\|\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}\|_2^2 + \alpha\|\mathbf{w}\|_1$$
$$\text{subject} \quad : \|\mathbf{w}\|_1 < C, C > 0$$

Với hàm trên, Thành phần điều chuẩn norm bậc 1 cũng có tác dụng như một sự kiểm soát áp đặt lên hệ số ước lượng. Khi muốn gia tăng sự kiểm soát, chúng ta sẽ gia tăng hệ số α để mô hình trở nên bớt phức tạp hơn. Cũng tương tự như hồi qui Ridge chúng ta cùng phân tích tác động của α:

- Trường hợp α=0, thành phần điều chuẩn bị tiêu giảm và chúng ta quay trở về bài toán hồi qui tuyến tính.
- Trường hợp α nhỏ thì vai trò của thành phần điều chuẩn trở nên ít quan trọng. Mức độ kiểm soát quá khớp của mô hình sẽ trở nên kém hơn.
- Trường hợp α lớn chúng ta muốn gia tăng mức độ kiểm soát lên độ lớn của các hệ số ước lượng.

4.2.Xác định target và các giá trị ảnh hưởng

Kiểm tra các giá trị có sự ảnh hưởng với chỉ số target là Happiness Score.

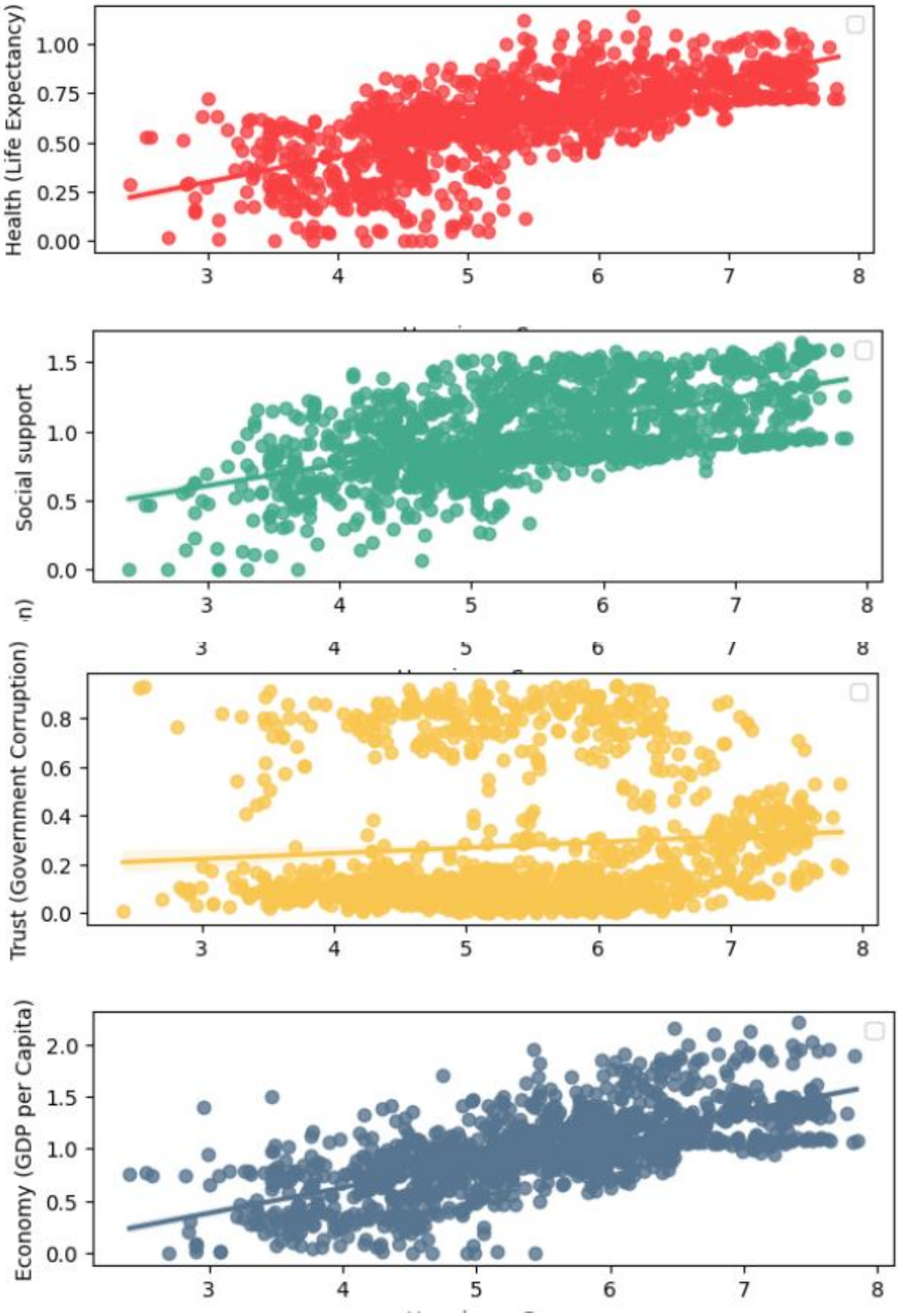
Tạo DataFrame dfCompare để gộp các cột Economy (GDP per Capita), Social support, Health (Life Expectancy), Freedom, Trust (Government Corruption), Generosity và cột Happiness Score. Sau đó dùng phương thức .corr() để tính toán ma

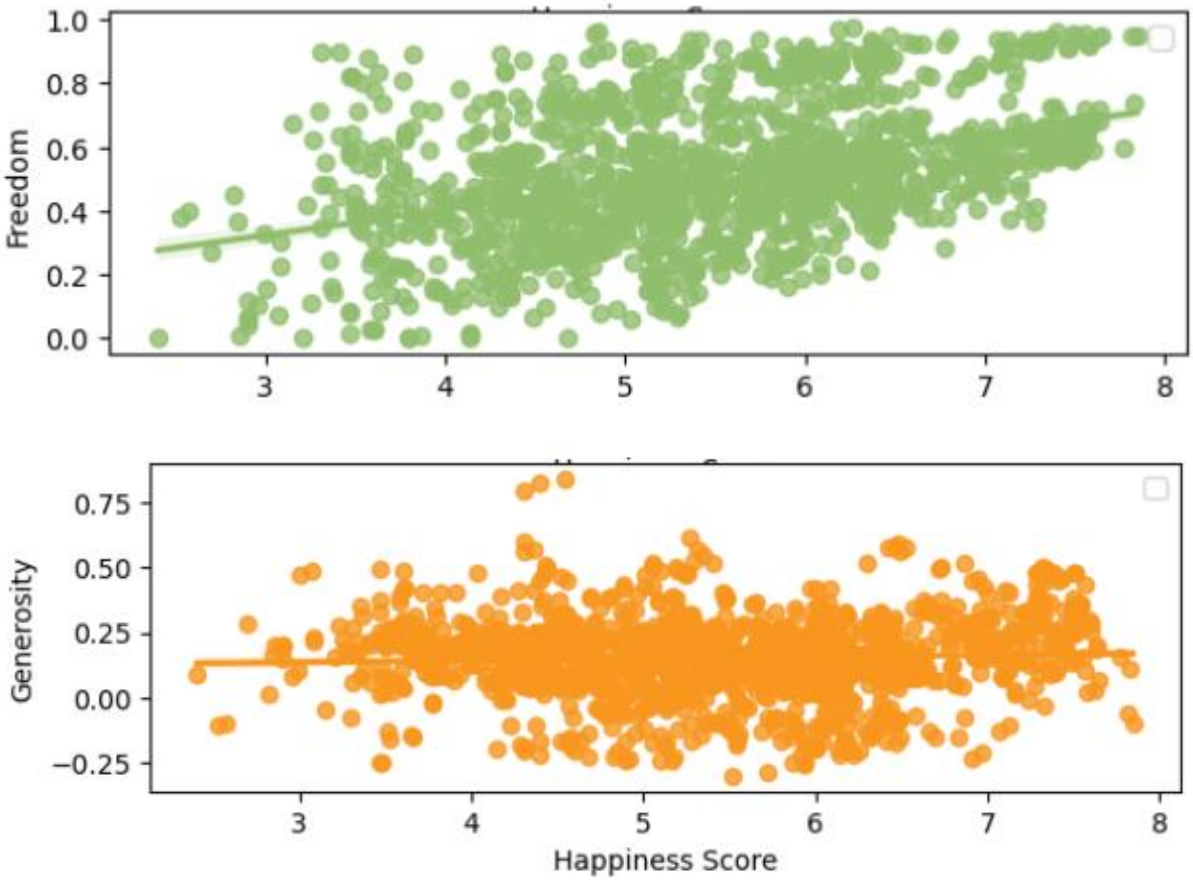
trận tương quan dựa trên Target là cột Happiness Score, từ đó sắp xếp các giá trị tương quan theo thứ tự giảm dần. Rồi gán vào cols.

```
dfCompare = df[[
    'Happiness Score',
    'Economy (GDP per Capita)',
    'Social support',
    'Health (Life Expectancy)',
    'Freedom',
    'Trust (Government Corruption)',
    'Generosity']]
cols=dfCompare.corr()['Happiness Score'].sort_values(ascending=False)
```

4.2.1.Các biểu đồ so sánh ảnh hưởng đến giá trị Happiness Score

```
fig=plt.figure(figsize=(15,10))
plt.suptitle("So sánh ảnh hưởng đến giá trị Happiness Score",family='Serif', weight='bold', size=20)
j=0
for i in cols.index[1:]:

    ax=plt.subplot(421+j)
    ax=sns.regplot(data=df, x='Happiness Score',y=i, color=color[-j])
    ax.legend('')
    j=j+1

plt.legend('')
```

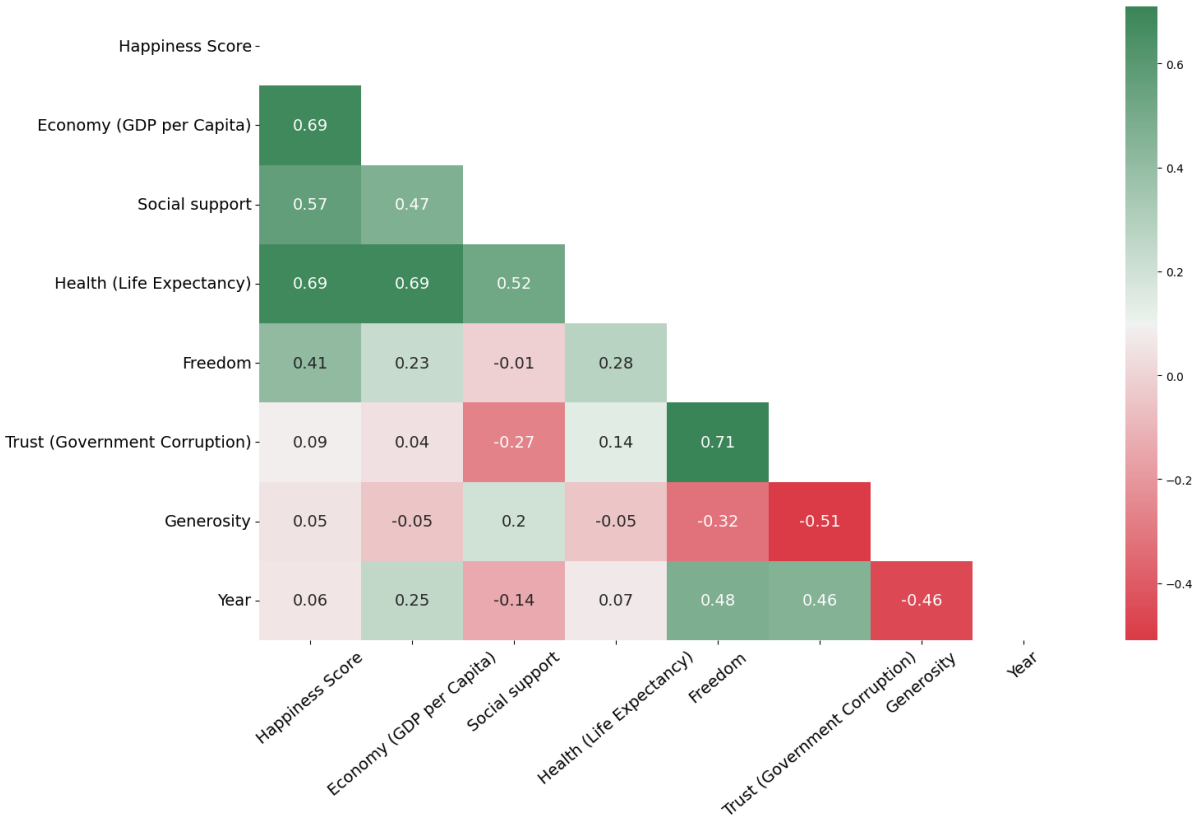

Health (Life Expectancy)
1.00
0.75
0.50
0.25
0.00
3
4
5
6
7
8
Social support
1.5
1.0
0.5
0.0
Trust (Government Corruption)
0.8
0.6
0.4
0.2
0.0
Economy (GDP per Capita)
2.0
1.5
1.0
0.5
0.0



➢ **Từ bảng so sánh trên, ta nhận thấy các giá trị Economy (GDP per Capita), Social support, Health (Life Expectancy), Freedom, Generosity có sự tương quan rõ rệt đến điểm số Happiness Score, còn giá trị Trust (Government Corruption) thì có sự tương quan chưa rõ rệt đến Happiness Score.**

4.2.2.Biểu đồ Heatmap trực quan hoá ma trận tương quan giữa các giá trị

```
#Tạo biểu đồ Heatmap
mask = np.zeros_like(df.corr())
triangle = np.triu_indices_from(mask)
mask[triangle] = True

#Vẽ biểu đồ Heatmap
cmap = sns.diverging_palette(10, 500, as_cmap=True)

plt.figure(figsize=(16, 10))
sns.heatmap(df.corr().round(2), mask = mask, cmap=cmap, annot= True, annot_kws={'size':14})
sns.set_style('white')
plt.xticks(fontsize = 14, rotation = 40)
plt.yticks(fontsize = 14);
```



➢ **Biểu đồ Heatmap trên thể hiện sự tương quan giữa các giá trị, trong đó các giá trị có sự ảnh hưởng lớn đến Happiness Score là Freedom, Health, Social Support và Economy.**

Nhận xét: Từ 2 biểu đồ trên, ta rút ra các dữ liệu có sự tương quan và ảnh hưởng lớn đến chỉ số Happiness Score gồm: Economy, Social Support, Health, Freedom và Generosity.

4.3.Phát triển mô hình

```
#PHÂN TÍCH VÀ TRAIN DỮ LIỆU ĐỂ DỰ ĐOÁN HAPPINESS SCORE (SỬ DỤNG PHƯƠNG PHÁP HỒI QUY LASSO)

# Xác định các đặc trưng và biến mục tiêu

features = ['Economy (GDP per Capita)',

'Social support',

'Health (Life Expectancy)',

'Freedom',

'Generosity',]

target = 'Happiness Score'


# Tạo X và y từ dữ liệu

X = df[features]

y = df[target]


# Chia tập dữ liệu thành huấn luyện và kiểm tra

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Tạo mô hình hồi quy LassoCV

alphas = [0.001, 0.01, 0.1, 1, 10, 100] # Các giá trị alpha cần thử nghiệm

lasso_cv_model = LassoCV(alphas=alphas, cv=5) # Sử dụng 5-fold cross-validation


# Huấn luyện mô hình trên tập huấn luyện

lasso_cv_model.fit(X_train, y_train)
```

```
# In giá trị alpha tối ưu
print(f"Alpha tối ưu: {lasso_cv_model.alpha_}")
# Lấy hệ số tối ưu
optimal_coeffs = lasso_cv_model.coef_
# In hệ số Lasso tối ưu
print("Hệ số Lasso tối ưu:")
for feature, coeff in zip(features, optimal_coeffs):
print(f"{feature}: {coeff}")
```

Trong đó:

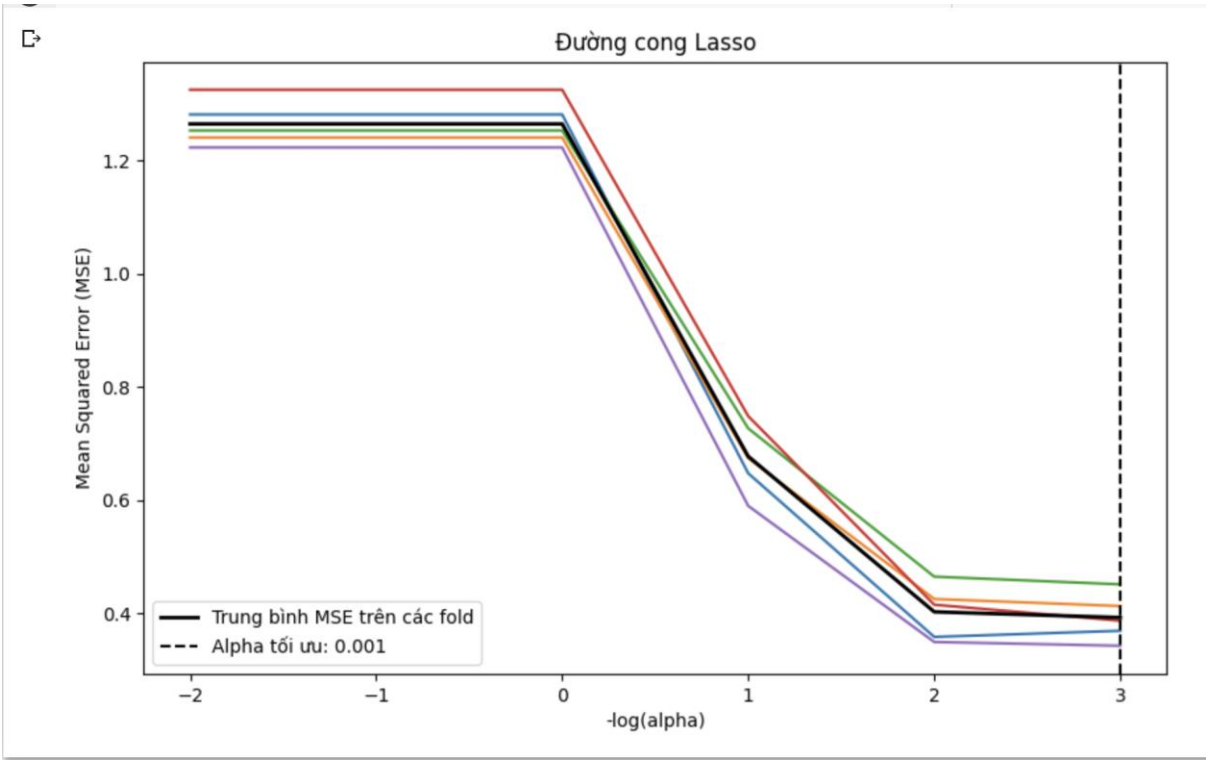
- ❖ feartures là các đặc trưng đã được xác định có sự ảnh hưởng và tương quan cao đối với Happiness Score.
- ❖ target là đối tượng giá trị được hướng đến để hiển thị kết quả và dự đoán trong tương lai.
- ❖ X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42): Đây là phần chính của mã. Hàm train_test_split được gọi với các đối số sau:
  - X: Ma trận chứa các biến đầu vào (features).
  - y: Vector chứa kết quả cần dự đoán (target).
  - test_size: Tỷ lệ dữ liệu dành cho tập kiểm tra, ở đây là 20% (0.2). Điều này có nghĩa là 20% dữ liệu sẽ được chia thành tập kiểm tra, còn lại (80%) là tập huấn luyện.
  - random_state: Giá trị này là 42 ở đây, giúp đảm bảo việc chia dữ liệu luôn giống nhau mỗi khi chạy mã.

Sau khi chạy thử nghiệm trên tập các giá trị alpha, ta sẽ có kết quả được xuất ra gồm hệ số alpha tối ưu và hệ số Lasso tối ưu cho từng đặc trưng như kết quả bên dưới.

```
Alpha tối ưu: 0.001
Hệ số Lasso tối ưu:
Economy (GDP per Capita): 0.9141866871635898
Social support: 1.0150876901110386
Health (Life Expectancy): 1.2447500214769232
Freedom: 1.6661628217464457
Generosity: 0.7976988928464914
```

4.4.Tối ưu hoá mô hình

```
# Vẽ biểu đồ đường cong Lasso
plt.figure(figsize=(10, 6))
plt.plot(-np.log10(lasso_cv_model.alphas_), lasso_cv_model.mse_path_)
plt.plot(-np.log10(lasso_cv_model.alphas_),
lasso_cv_model.mse_path_.mean(axis=-1), 'k',
label='Trung bình MSE trên các fold', linewidth=2)
plt.axvline(-np.log10(lasso_cv_model.alpha_), linestyle='--', color='k',
label='Alpha tối ưu: %s' % np.round(lasso_cv_model.alpha_, 4))
plt.legend()
plt.xlabel('-log(alpha)')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('Đường cong Lasso')
plt.axis('tight')
plt.show()
```



➢ **Với biểu đồ đường cong Lasso, ta sẽ có cái nhìn trực quan về hệ số alpha và chỉ số MSE (Mean Squared Error) để từ đó có được chỉ số Alpha tối ưu nhất là 0,001 với MSE là 0,4.**

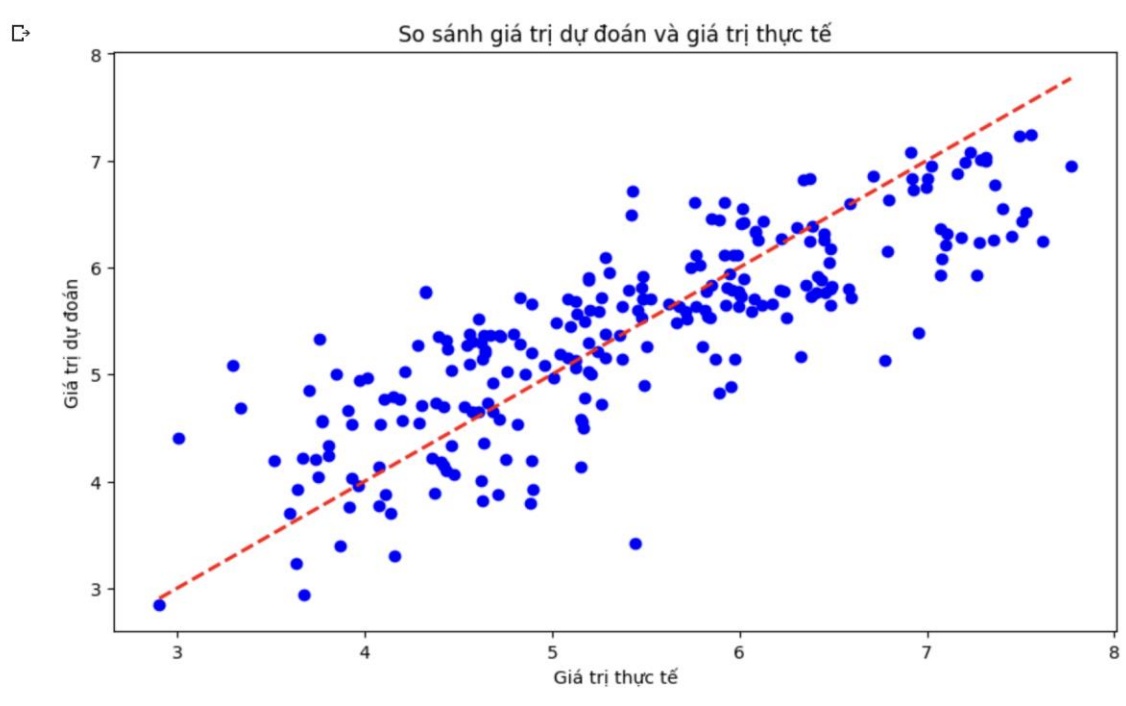
```
# Dự đoán trên tập kiểm tra
y_pred = lasso_cv_model.predict(X_test)


# Tính toán Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error trên tập kiểm tra: {mse:.2f}")
```

```
Mean Squared Error trên tập kiểm tra: 0.40
```

➢ **Để kiểm tra lại chỉ số MSE, ta sử dụng hàm mean_squared_error để tính toán và kết quả được trả về tương ứng với chỉ số MSE đã nhận xét dựa trên biểu đồ đường cong Lasso.**

4.5.So sánh kết quả

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], linestyle='--', color='red', linewidth=2)
plt.title('So sánh giá trị dự đoán và giá trị thực tế')
plt.xlabel('Giá trị thực tế')
plt.ylabel('Giá trị dự đoán')
plt.show()
```



➢ **Từ biểu đồ so sánh giá trị dự đoán và giá trị thực tế, ta có thể thấy biểu đồ đang thể hiện được mô hình có khả năng phù hợp cao để ứng dụng dự đoán kết quả trong tương lai vì mô hình đang không có gặp các vấn đề như *Overfitting* hoặc *Underfitting*.**

4.6.Sử dụng dữ liệu 2023 để dự đoán và phân tích kết quả.

```
# Nhập các giá trị của các đặc trưng theo mẫu để dự đoán
#[GDP_value, social_support_value, life_expectancy_value, freedom_value, generosity_value]
data_Vietnam2023 = np.array([[1.51, 0.836 , 0.468, 0.882, -0.041]])

# Dự đoán Happiness Score cho số năm trong tương lai
future_years = 2023
predicted_scores = lasso_cv_model.predict(data_Vietnam2023)

# Hiển thị kết quả dự đoán
print(f"Dự đoán Happiness Score của Việt Nam trong năm {future_years}: {predicted_scores[0]:.2f}")
Happiness_value = df23.loc[64, 'Happiness Score']
print(f"Happiness Score thực tế của Việt Nam trong năm {future_years}: {Happiness_value}")
```

```
Dự đoán Happiness Score của Việt Nam trong năm 2023: 6.02
Happiness Score thực tế của Việt Nam trong năm 2023: 5.763
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
  warnings.warn(
```

➢ **Hàm sử dụng các giá trị có sẵn trong Dataset 2023 về các chỉ số tại Việt Nam để dự đoán chỉ số Happiness Score của Việt Nam trong 2023. Với kết quả dự đoán trả về là 6.02 so sánh với giá trị thực tế 5.763, ta thấy được quá trình train và test đã mang lại thành quả khá tốt khi có mức sai số nằm trong khả năng cho phép.**

```
def predict_happiness_scores(model, features, df23):
X23 = df23[features]
predicted_scores = model.predict(X23)
return predicted_scores

# Gọi hàm để dự đoán Happiness Score
predicted_scores = predict_happiness_scores(lasso_cv_model, features, df23)

# Tính MSE giữa giá trị dự đoán và Happiness Score thực tế
mse = mean_squared_error(df23['Happiness Score'], predicted_scores)

print("Dự đoán Happiness Score và giá trị thực tế:")
for i, (predicted_score, real_score, country) in enumerate(zip(predicted_scores, df23['Happiness Score'], df23['Country'])):
print(f"Quốc gia: {country}")
print(f"Happiness Score dự đoán: {predicted_score:.2f}")
print(f"Happiness Score thực tế: {real_score:.2f}")
print(f"Sai số: {real_score - predicted_score:.2f}")
print("-----------")
print(f"MSE: {mse:.2f}")
```

- ❖ Hàm predict_happiness_scores nhận ba đối số: mô hình hồi quy Lasso, danh sách các đặc trưng, và DataFrame df23 chứa dữ liệu World Happiness Score cho năm 2023.
- ❖ Chúng ta tính sai số bình phương trung bình (MSE) bằng cách sử dụng hàm mean_squared_error từ sklearn.metrics. Đối số thứ nhất của hàm này là danh sách các giá trị thực tế, và đối số thứ hai là danh sách các giá trị dự đoán.
- ❖ Trong vòng lặp, chúng ta in ra thông tin về giá trị dự đoán và Happiness Score thực tế cùng với tên quốc gia tương ứng. Chúng ta cũng tính và in ra sai số (chênh lệch) giữa hai giá trị.
- ❖ Cuối cùng, chúng ta in ra giá trị MSE để đánh giá tổng sai số của mô hình.

Kết quả trả về:

```
Quốc gia: Indonesia
Happiness Score dự đoán: 5.25
Happiness Score thực tế: 5.31
Sai số: 0.06
-----------
Quốc gia: Jordan
Happiness Score dự đoán: 5.03
Happiness Score thực tế: 5.30
Sai số: 0.27
-----------
Quốc gia: Azerbaijan
Happiness Score dự đoán: 4.89
Happiness Score thực tế: 5.29
Sai số: 0.41
-----------
Quốc gia: Philippines
Happiness Score dự đoán: 5.08
Happiness Score thực tế: 5.28
Sai số: 0.20
-----------
Quốc gia: China
Happiness Score dự đoán: 5.21
Happiness Score thực tế: 5.25
Sai số: 0.04
-----------
Quốc gia: Bhutan
Happiness Score dự đoán: 5.25
Happiness Score thực tế: 5.20
Sai số: -0.05
-----------
```

➢ **Sự sai số giữa chỉ số dự đoán và chỉ số thực tế còn chưa đồng đều giữa các nước vì chỉ số Happiness Score được tính chỉ dựa trên các đặc trưng nổi bật mà chưa bao quát các đặc trưng khác.**

CHƯƠNG 5: TỰ ĐÁNH GIÁ

Trong quá trình thực hiện bài tập lớn của nhóm 18, chúng em đã được trải nghiệm và tự học hỏi thêm nhiều kiến thức quý giá và sẽ đồng hành cùng chúng em trong thời gian sắp tới. Với những sai sót hoặc thiếu xót, chúng em hy vọng thầy/cô giảng viên có thể để lại những nhận xét để chúng em tiếp tục học hỏi và cải thiện bản thân.

Qua đó, chúng em xin được gửi lời cảm ơn chân thành đến thầy **Nguyễn Văn Bảy** và thầy **Nguyễn Tiến Đạt** đã hướng dẫn chúng em tận tình để chúng em có thể hoàn thành được báo cáo cho bài tập lớn ngày hôm nay!