Project 3: Linear Regression - Life Expectancy

20127479 – Lê Nhất Duy

MTH051 - FIT-HCMUS

1. Giới thiệu và yêu cầu:

Trong đồ án này, sinh viên được yêu cầu thực hiện:

- Yêu cầu 1a: Sử dụng toàn bộ 10 đặc trưng đề bài cung cấp (2 điểm)
  - Huấn luyện 1 lần duy nhất cho 10 đặc trưng trên toàn bộ tập huấn luyện (train.csv)
  - Thể hiện công thức cho mô hình hồi quy (tính y theo 10 đặc trưng trong X)
  - Báo cáo **1 kết quả trên tập kiểm tra (test.csv)** cho mô hình vừa huấn luyện được
- Yêu cầu 1b: Xây dựng mô hình sử dụng duy nhất 1 đặc trưng, tìm mô hình cho kết quả tốt nhất (2 điểm)
  - Thử nghiệm trên toàn bộ (10) đặc trưng đề bài cung cấp
  - Yêu cầu **sử dụng phương pháp 5-fold Cross Validation** để tìm ra đặc trưng tốt nhất
  - Báo cáo **10 kết quả tương ứng cho 10 mô hình** từ 5-fold Cross Validation (lấy trung bình)

STT	Mô hình với 1 đặc trưng	RMSE
1	Adult Mortality	
2	BMI	
	...	
10	Schooling	

  - Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất (tính y theo đặc trưng tốt nhất tìm được)
  - Báo cáo **1 kết quả trên tập kiểm tra (test.csv)** cho mô hình tốt nhất tìm được
- Yêu cầu 1c: Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất (3 điểm)
  - Xây dựng m mô hình khác nhau (tối thiểu 3), đồng thời khác mô hình ở 1a và 1b
    - Mô hình có thể là sự kết hợp của 2 hoặc nhiều đặc trưng
    - Mô hình có thể sử dụng đặc trưng đã được chuẩn hóa hoặc biến đổi (bình phương, lập phương...)
    - Mô hình có thể sử dụng đặc trưng được tạo ra từ 2 hoặc nhiều đặc trưng khác nhau (cộng 2 đặc trưng, nhân 2 đặc trưng...)
    - ...

- Yêu cầu **sử dụng phương pháp 5-fold Cross Validation** để tìm ra mô hình tốt nhất
- Báo cáo **m kết quả tương ứng cho m mô hình** từ 5-fold Cross Validation (lấy trung bình)

STT	Mô hình	RMSE
1	Sử dụng 2 đặc trưng (a, b)	
...	...	
m	...	

- Thể hiện công thức cho mô hình hồi quy tốt nhất mà sinh viên tìm được
- Báo cáo **1 kết quả trên tập kiểm tra (test.csv)** cho mô hình tốt nhất tìm được

- Lưu ý:
  - Tập test.csv chỉ được sử dụng 3 lần như được đề cập bên trên
  - Kết quả báo cáo là độ đo RMSE.

2. Môi trường thực hiện:

Đồ án sử dụng Google Colab để chạy và test.

3. Liệt kê các thư viện đã sử dụng:

Thư viện	**Lý do**
1. pandas	- Để đọc cũng như in ra dữ liệu dạng data frame
2. numpy	- Để thao tác trên ma trận, array,… như phương thức append..
3. math	- Để sử dụng các hàm tính toán, như sqrt(), ..
4. sklearn	- Để được sử dụng thuật toán linear regression, k_fold cross validation và để tính mean square error (mse)

4. Liệt kê và mô tả các hàm đã sử dụng: Nhiều hàm (phương thức) cùng thuộc một class, và những hàm này được tham khảo ngay trên web chính của thư viện mà chúng ta đang dùng, nên để tránh việc cứ mỗi một hàm nhỏ ta lại phải gắn link tham khảo thì chỉ gắn link tham khảo đến ngay trang chính của thư viện mà ta đang dùng. Đối với trường hợp nếu là một hàm thực hiện thuật toán quan trọng nào đó thì sẽ được gắn link cụ thể riêng.

4.1 Hàm thuộc thư viện math:

- *math.sqrt(n):* hàm dùng để tính căn bậc 2
  - ✓ input: n: một số thực
  - ✓ output: số thực đã lấy căn
  - ✓ Hàm được sử dụng để tính căn bậc 2 của MSE, để lấy RMSE.

4.2 Hàm thuộc thư viện numpy:

- *numpy.mean():* hàm dùng để tính trung bình của một mảng
  - ✓ input: một mảng các tham số
  - ✓ output: giá trị trung bình.
  - ✓ Hàm được sử dụng để tính trung bình của RMSE trong thuật toán 5_Fold Cross Validation.
- *numpy.absolute():* hàm dùng để lấy trị tuyệt đối.
  - ✓ input: một mảng các tham số
  - ✓ output: giá trị tuyệt đối của tham số
  - ✓ Hàm được sử dụng để lấy trị tuyệt đối trước khi lấy căn trong RMSE.

4.3 Hàm thuộc thư viện pandas: [1]

- *pandas.DataFrame.sample(frac = 1):* là một hàm của pandas cho phép ta xáo trộn các dòng dữ liệu của một dataframe.
  - ✓ input: frac = 1
  - ✓ output: dataframe đã được xáo
  - ✓ Tham số frac = 1 truyền vào để hàm trả về đúng 100% khung dữ liệu mà không lượt bỏ bất cứ dữ liệu nào. Hàm này được sử dụng để xáo tập data train trong câu 1b, 1c để thực hiện thuật toán 5_Fold Cross Validation.
- *pandas.read_csv(name of file):* là một hàm của pandas cho phép ta đọc dữ liệu từ một file csv.
  - ✓ input: tên file csv.
  - ✓ output: dữ liệu kiểu dataframe.
- *pandas.iloc[]:* là một hàm của pandas cho phép ta truy xuất dữ liệu từ của một dataframe theo chỉ mục.
  - ✓ input: chỉ mục index.
  - ✓ output: dữ liệu kiểu dataframe (dòng hoặc cột ta cần truy xuất).
- *pandas.copy(deep = True):* là một hàm của pandas cho phép ta copy một dữ liệu kiểu dataframe sang một dataframe khác. Nếu không copy mà gán luôn các dataframe thì python sẽ warning.
  - ✓ input: deep (True or False), mặc định là True, có 2 kiểu copy của hàm này, nếu deep = true thì bản sao tạo ra sẽ là sự sao chép toàn bộ dữ liệu của bản gốc, sự thay đổi của cả 2 hoàn toàn không ảnh hưởng đến nhau. Nếu deep = false thì bản sao sẽ tham chiếu đến bản gốc, khi đó bản gốc thay đổi thì bản sao cũng sẽ ảnh hưởng, và ngược lại. Ở trong đồ án này, ta sử dụng chế độ mặc định của hàm.
  - ✓ output: dữ liệu kiểu dataframe. (là bản sao của dataframe gốc truyền vào)
- *pandas.DataFrame(data):* là một hàm của pandas cho phép ta tạo một dataframe từ một dữ liệu có sẵn, có thể là list, object,..
  - ✓ input: data
  - ✓ output: dữ liệu kiểu dataframe
  - ✓ Được sử dụng để in ra các kết quả đẹp hơn.

- *pandas.corr():* là một hàm của pandas cho phép ta tính toán hệ số tương quan giữa các cặp đặc trưng khác nhau trong một dataframe **[2]**
  - ✓ input: có thể truyền vào method, có 3 thuật toán tính hệ số tương quan đó là *pearson , kendall* và *spearman* .
  - ✓ output: một ma trận với các phần tử tại vị trí ij thể hiện hệ số tương quan giữa đặc trưng i và đặc trưng j, hệ số càng cao chứng tỏ 2 đặc trưng đang xét tương quan càng mạnh (tuy nhiên nếu kết quả tương quan âm thì nó sẽ tương quan nghịch, chi tiết sẽ được đề cập ở phần tính toán 1c).
  - ✓ Được sử dụng để tính mối tương quan giữa các đặc trưng khác với đặc trưng life expectancy.

4.4 Hàm thuộc thư viện sklearn:

- *sklearn.linear_model.LinearRegression():* là một class cho phép ta thực hiện thuật toán linear regression để xây dựng mô hình hồi quy tuyến tính. **[3]**
  - ✓ Input: hàm có thể truyền vào nhiều tham số tuy nhiên trong đồ án này chúng ta không truyền tham số nào vào cả.
  - ✓ Output: một đối tượng thuộc class LinearRegression.
- *LinearRegression.fit(X, y):* là một hàm (method) thuộc class LinearRegression giúp ta huấn luyện cho các đặc trưng trong tập huấn luyện
  - ✓ Input:
    - X: một mảng hay ma trận gồm dữ liệu của các đặc trưng cần huấn luyện. (X được truyền vào phải là một ma trận 2D, nếu không thì phương thức sẽ báo lỗi)
    - y: các giá trị mục tiêu của tập huấn luyện đang xét.
  - ✓ Output: không có output
  - ✓ Hàm này được sử dụng thường xuyên để huấn luyện cho các đặc trưng mà ta xét ở cả 3 câu 1a, 1b, 1c
- *LinearRegression.predict(X_test):* là một hàm hay còn gọi là method thuộc class LinearRegression giúp ta dự đoán kết quả dựa trên mô hình hồi quy tuyến tính đã tìm được sau khi fit ở trên.
  - ✓ Input: X_test: một mảng hay ma trận gồm dữ liệu của các đặc trưng cần dự đoán, lưu ý số lượng đặc trưng của X_test phải hoàn toàn tương thích với số lượng đặc trưng ta đã fit trước đó. Và một *lưu ý* nhỏ nữa X_test phải là một mảng 2D, nếu X_test là một single feature thì phải reshape(1, -1) thì phương thức predict mới nhận, nếu không thì sẽ báo lỗi.
  - ✓ Output: mảng các giá trị y_pred mà dự đoán dựa trên dữ liệu truyền vào và mô hình hồi quy tuyến tính đã có.
  - ✓ Hàm này cũng được sử dụng thường xuyên để tính giá trị dự đoán cho nhiều mô hình khác nhau ở 3 câu 1a, 1b, 1c.
- *LinearRegression.coef_:* đây không phải là một hàm mà là một thuộc tính (property) thuộc class LinearRegression, tuy nhiên trong đồ án sử dụng khá nhiều nên sẽ giới thiệu luôn tại đây. Phương thức trả ra mảng các hệ số ước lượng cho

mô hình hồi quy tuyến tính của chúng ta (sau khi fit xong), số lượng hệ số sẽ tương đương với số đặc trưng ta dùng để huấn luyện mô hình.

- ✓ Input: không có
- ✓ Output: mảng các hệ số tương ứng với đặc trưng
- ✓ Rõ ràng thuộc tính này sử dụng cũng rất thường xuyên, đi chung với việc huấn luyện xong thì ta cần biết mô hình đã huấn luyện xong là gì, thì thuộc tính này cho ra biết hệ số của các đặc trưng mà mô hình có (được sử dụng trong cả 3 câu 1a, 1b, 1c).

- *sklearn.model_selection:* là một class cho phép ta thực hiện các thuật toán thiết lập, phân tích dữ liệu và sau đó sử dụng nó để đo lường dữ liệu mới. Ở trong đồ án này ta sử dụng một phương thức của lớp model_selection đó là phương thức cross_val_scrore. **[4]**
  - ✓ Để thực hiện thuật toán 5_Fold Cross Validation
- *sklearn.model_selection.cross_val_score(estimator, X, y, cv = 5, scoring):* là một hàm thực hiện thuật toán k_Fold Cross Validation **[5]**
  - ✓ Giải thích một chút về thuật toán: thuật toán k_Fold Cross Validation chia dữ liệu của ta thành k nhóm, với mỗi nhóm thuật toán sẽ sử dụng nhóm đó để đánh giá mô hình, các nhóm còn lại thì tiếp tục huấn luyện mô hình, thay phiên nhau lần lượt vậy ta có k lần thực hiện. Sau đó tổng hợp lại kết quả lấy trung bình và đánh giá. Có thể đánh giá dựa trên nhiều tiêu chuẩn, ở đây đồ án yêu cầu sử dụng tiêu chuẩn là RMSE.
    - *Lưu ý nhỏ*, thuật toán k_Fold Cross Validation yêu cầu xáo trộn dữ liệu trước khi thực hiện chia nhỏ để đạt kết quả tốt nhất. Mà dường như cross_val_score mặc định sẽ không xáo dữ liệu trừ khi ta config nó ngay trong tham số truyền vào. Điều này sẽ dẫn đến khi ta gọi phương thức mỗi lần cho các mô hình khác nhau thì phương thức lại xáo trộn dữ liệu lại một cách khác nhau. Tuy nhiên ta cần xáo trộn dữ liệu chỉ một lần *duy nhất* và gọi cross_val_score trên nhiều mô hình với dữ liệu đã xáo, nên do đó trước khi gọi phương thức này, ta cần phải xáo dữ liệu trước để tránh việc dữ liệu không nhất quán.
  - ✓ Input:
    - estimator: ta cần truyền vào một đối tượng có thể thực hiện fit (có thể train), ở đây ta truyền vào đối tượng thuộc class LinearRegression, nhờ đó phương thức cross_val_score mới có thể biết: sử dụng mô hình hồi quy tuyến tính và đánh giá mô hình.
    - X: một mảng hay ma trận gồm dữ liệu của các đặc trưng cần huấn luyện và test
    - y: một mảng các giá trị mục tiêu mà mô hình cố gắng đạt được.
    - cv: tương ứng với chữ "k" trong k_Fold Cross Validation, cv bằng bao nhiêu thì phương thức sẽ chia tập dữ liệu của ta có ra thành bấy nhiêu phần, vì yêu cầu đồ án là sử dụng 5_Fold Cross Validation nên cv = 5.
    - scoring: (kiểu là string), ở phần giới thiệu trên ta biết thuật toán sẽ đánh giá mô hình theo một tiêu chuẩn nào đó, Do đó ở đây ta định nghĩa tiêu

chuẩn cần xét là MSE, do đó ta truyền tham số có dữ liệu truyền vào là "neg_mean_squared_error". Còn rất nhiều tiêu chuẩn khác mà chúng ta có thể tham khảo. **[6]**

- ✓ Output: giá trị của tiêu chuẩn ta đang xét, trong trường hợp này phương thức trả về giá trị MSE của mô hình. Tuy nhiên, là mảng MSE bởi vì cv = 5 nên hàm sẽ lặp 5 lần thay phiên nên sẽ có 5 giá trị MSE, vì vậy output sẽ là một mảng MSE có 5 phần tử.
- ✓ Như đã nêu trên hàm được sử dụng để thực hiện thuật toán 5_Fold Cross Validation.

- *sklearn.metrics:* là một class cho phép ta truy xuất các tiêu chuẩn, các số liệu đánh giá **[7]**
  - ✓ Trong đồ án này ta chỉ sử dụng phương thức mean_squared_error để tính MSE.

- *sklearn.metrics.mean_squared_error(y, y_pred):* là một phương thức thuộc metrics cho phép tính MSE của 2 tập dữ liệu. **[8]**

  - ✓ input:
    - y: một mảng các dữ liệu đích.
    - y_pred: một mảng các dữ liệu dự đoán.
  - ✓ output: giá trị MSE.

5. Ý tưởng thực hiện, giải thích, đánh giá và nhận xét các mô hình:

5.1 Yêu cầu 1a: Sử dụng toàn bộ 10 đặc trưng đề bài cung cấp:

- Vì câu 1a sử dụng các hàm cơ bản nhất để thực hiện xây dựng mô hình tuyến tính, các câu sau sẽ sử dụng lại thường xuyên các hàm cũng như cách viết code này nên ta sẽ giải thích code và ý tưởng của câu 1a, khi sang các câu còn lại thì ta sẽ không đi lại phần giải thích này.

```
# Phần code cho yêu cầu 1a
train1 = pd.read_csv('train.csv')
test1 = pd.read_csv('test.csv')

reg1 = LinearRegression().fit(train1.iloc[:, :-1], train1.iloc[:, -
1])

y_test_pred = reg1.predict(test1.iloc[:, :-1])
prediction = pd.DataFrame({'Actual': test1.iloc[:, -
1], 'Predicted': y_test_pred})
print(prediction)

coef = pd.DataFrame({'Feature': train1.iloc[:, :-
1].columns, 'Coefficient': reg1.coef_})
print(coef)
```

- Đầu tiên ta tạo một đối tượng thuộc class LinearRegression, như đã giải thích về thư viện này ở trên, thì nhờ vào thư viện này ta có thể huấn luyện mô hình hồi quy tuyến tính cũng như dự đoán kết quả. Ta gọi hàm fit để train mô hình, đầu vào truyền vào 2 tham số: 1 là tập các đặc trưng để xây dựng mô hình (tất cả 10 đặc trưng, không bao gồm đặc trưng life expectancy), 2 là tập giá trị đích đó chính là dữ liệu của đặc trưng life expectancy (cả 2 phần dữ liệu này đều nằm trong file train.csv)

- Tiếp theo ta test lại mô hình với dữ liệu đầu vào là tập dữ liệu của file test, gọi hàm predict để test. Đồng thời ta gọi thuộc tính coef_ của lớp để xem các hệ số của mô hình, nhờ vào điều này ta tìm được công thức cho mô hình hồi quy tuyến tính.

- Khi đó ta đạt được kết quả bên dưới, dựa vào hệ số ta có được công thức sau:

$$\begin{aligned} \textit{Life expectancy} = & -0.016674.\textit{Adult Mortality} + 0.045724.\textit{BMI} + 0.001458.\textit{Polio} + 0.025288.\textit{Diphtheria} \\ & + -0.498039.\textit{HIV / AIDS} + 0.000063.\textit{GDP} + -0.040337.\textit{Thinness age } 10-19 \\ & + -0.056481.\textit{Thinness age } 5-9 + 12.705896.\textit{Income composition of resources} + 0.707695.\textit{Schooling} \end{aligned}$$

```
    Actual  Predicted
0     65.0  62.123959
1     59.9  61.837164
2     77.8  76.307811
3     77.5  77.371350
4     75.4  76.482085
..     ...        ...
90    78.0  72.930919
91    76.8  77.806096
92    69.2  71.006486
93    61.1  63.981038
94    59.2  60.041279

[95 rows x 2 columns]
                          Feature  Coefficient
0                 Adult Mortality    -0.016674
1                             BMI     0.045724
2                           Polio     0.001458
3                      Diphtheria     0.025288
4                        HIV/AIDS    -0.498039
5                             GDP     0.000063
6             Thinness age 10-19    -0.040337
7               Thinness age 5-9    -0.056481
8  Income composition of resources    12.705896
9                       Schooling     0.707695
```

```
print("RMSE: ", math.sqrt(mean_squared_error(test.iloc[:, -
1], y_test_pred)))
```

- Ta gọi hàm mean_squared_error để tính giá trị MSE của tập giá trị dự đoán và tập đích. Sau đó tiếp tục gọi hàm sqrt() để tính RMSE. Khi đó kết quả RMSE là: `3.4610550812964633`

- Nhận xét: vì mô hình được huấn luyện trên tất cả các đặc trưng nên cho ra kết quả khá tốt trên tập test, dẫn đến RMSE khá nhỏ. Tuy nhiên, chúng ta chỉ mới test trên mô hình đầu tiên nên ta chưa thấy rõ sự khác biệt của nó với các mô hình khác, chính vì vậy hãy tiếp tục phân tích dữ liệu từ các câu sau để xem thử liệu kết quả này có thật sự tốt hay không.

5.2 Yêu cầu 1b: Xây dựng mô hình sử dụng duy nhất 1 đặc trưng, tìm mô hình cho kết quả tốt nhất:

```
train_f = pd.read_csv('train.csv')
train_f = train_f.sample(frac=1)
y_df = train_f.iloc[:, -1]

reg = LinearRegression()

feature = []
for i in range(10):
  feature.append(train_f.iloc[:, i].values.reshape(-1, 1))

name = train_f.columns
scores = []
cross = []
for i in range(10):
  scores.append(cross_val_score(reg, feature[i], y_df, scoring="neg
_mean_squared_error", cv=5))

RMSE = []
for i in range(10):
  RMSE.append(np.sqrt(np.absolute(scores[i])).mean())
  cross.append([name[i], RMSE[i]])

test = pd.DataFrame(cross, columns=['Feature','Average RMSE'])
print(test)
print()

best_index = np.argmin(RMSE)
print("Min average RMSE is: ", name[best_index], "(", RMSE[best_ind
ex], ")")
```

- Code khá đơn giản về mặt thuật toán, bởi vì nhờ vào các hàm có sẵn nên phần thực hiện cũng nhẹ đi phần nào, ta chỉ cần xử lý trước dữ liệu truyền vào, còn lại nhờ thuật toán xử lý:

- Đầu tiên ta dùng hàm sample để xáo trộn dữ liệu của tập train một lần duy nhất. Đương nhiên rồi, như ta đề cập trên, nếu config cho hàm cross_val_score tự xáo dữ liệu thì mỗi lần gọi hàm lại xáo theo một cách khác nhau, việc này gây nên việc mất tính nhất quán của dữ liệu. Do đó ta phải gọi hàm xáo trộn dữ liệu ở phần đầu.
- Tiếp theo ta chia nhỏ các đặc trưng vào mảng feature, có 10 đặc trưng tất cả, ta tách riêng biệt để có thể đánh giá trên từng mô hình riêng biệt.
- Sau đó thì lặp 10 lần tương ứng với 10 đặc trưng, trong vòng lặp này ta thực hiện thao tác chính đó là thuật toán 5_Fold Cross Validation, với mỗi lần như vậy

mảng các tiêu chuẩn (hay còn gọi là MSE) được trả về ta sẽ lưu vào một mảng scores để dễ tính.

- Còn lại thì ta chỉ cần lấy căn để tính RMSE và lấy trung bình bằng hàm mean(), tuy nhiên vì không biết scores âm hay dương, nên trước đó ta dùng hàm absolute của numpy để lấy trị tuyệt đối.

- Sau khi chạy ta được kết quả sau:

```
                          Feature  Average RMSE
0                 Adult Mortality      6.576250
1                             BMI      7.150261
2                           Polio      8.413400
3                      Diphtheria      8.258391
4                         HIV/AIDS      7.339054
5                             GDP      7.891992
6               Thinness age 10-19      7.829015
7                 Thinness age 5-9      7.850341
8  Income composition of resources      5.667831
9                        Schooling      5.924835

Min average RMSE is:  Income composition of resources ( 5.667831302542259 )
```

STT	Mô hình với một đặc trưng	RMSE
1	Adult Mortality	6.576250
2	BMI	7.150261
3	Polio	8.413400
4	Diphtheria	8.258391
5	HIV/AIDS	7.339054
6	GDP	7.891992
7	Thinness age 10-19	7.829015
8	Thinness age 5-9	7.850341
9	Income composition of resources	5.667831
10	Schooling	5.924835

- Nhận xét:

- Ta thấy Polio (Tỉ lệ tiêm chủng bại liệt (Pol3) ở trẻ 1 tuổi (%)) và Diphtheria (Tỉ lệ tiêm chủng giải độc uốn ván và ho gà (DTP3) ở trẻ 1 tuổi (%)), hiện nay 2 tỷ lệ này đã giảm đi rất nhiều, cùng với trường hợp tuổi thọ bị suy giảm do không tiêm chủng chỉ xảy ra ở một số nước và khu vực nghèo nàn, lạc hậu hay điều kiện y tế không đủ, do đó 2 đặc trưng này không thể phản ánh hoàn toàn tuổi thọ trung bình được, chính vì vậy khi xây dựng mô hình khi chỉ sử dụng 2 đặc trưng này thì sai số lớn nhất trong tất cả các sai số (RMSE)
- Tiếp theo ta thấy các đặc trưng sau cũng cho giá RMSE cao không kém: BMI, HIV/AIDS, GDP, Thinness age 10-19, Thinness age 5-9. Rõ ràng ta thấy các đặc trưng này cũng không thể phản ánh hết toàn bộ tuổi thọ trung bình được, ví dụ như HIV/AIDS vốn dĩ đã giảm xuống rất nhiều ở các nước phát triển (ví dụ như Việt Nam), giờ đây HIV/AIDS vẫn còn nhưng chỉ số rất nhỏ, và chỉ

phản ánh ở một số các khu vực cụ thể. Do đó không thể sử dụng HIV/AIDS để làm mô hình cho toàn bộ dữ liệu được. Cũng như tỉ lệ gầy ốm, tỷ lệ gầy ốm ảnh hưởng rất ít đến tuổi thọ trung bình, chỉ trừ một số người bị bệnh hiểm nghèo mới trở nên gầy ốm và gây ra chết trẻ. Vì vậy sử dụng những đặc trưng này làm cho dữ liệu dự đoán vẫn còn lệch nhiều so với giá trị thực.

- Còn lại các đặc trưng như: Adult mortality, Income composition of resources, Schooling thì cho ra kết quả tốt hơn, Rõ ràng ta thấy thời buổi bây giờ tỷ lệ tử vong ở người trưởng thành không hề thấp: tai nạn, tệ nạn xã hội, … việc này ảnh hưởng đến tuổi thọ trung bình ở nhiều nơi trên thế giới, và giờ đây tỷ lệ này vẫn không hề giảm ở nhiều nước, dù cho có là nước có phát triển, vì vậy đặc trưng này mang lại kết quả tốt hơn so với các đặc trưng trên. Số năm đi học thì sẽ ảnh hưởng đến tư tưởng, có khả năng vì thiếu giáo dục nên nhiều người mắc nhiều các tệ nạn xã hội cũng như không biết cách cải thiển chăm sóc sức khỏe, điều này cũng dẫn đến tuổi thọ trung bình bị ảnh hưởng, tuy nhiên đây chỉ là giả thiết của tôi, đến tôi cũng không nghĩ rằng đặc trưng này lại ảnh hưởng đến tuổi thọ cao như vậy.
- Tiếp theo ta xét chỉ số phát triển dựa trên thu nhập tài nguyên, không gì lạ khi thu nhập ảnh hưởng đến rất nhiều đến cuộc sống của mỗi người, chất lượng cuộc sống thể hiện thông qua thu nhập, có thu nhập thì mỗi người mới có khả năng cải thiện cuộc sống, điều này ảnh hưởng rất lớn đến tuổi thọ trung bình (có bệnh thì có tiền chữa bệnh,…) những lý do vô cùng bình thường lại ảnh hưởng rất nhiều đến chất lượng cuộc sống, sức khỏe cũng như tuổi thọ trung bình, chính vì vậy đặc trưng này mang kết quả tốt nhất trong số các đặc trưng. Vậy thì tiếp theo ta sẽ xây dựng mô hình cho đặc trưng tốt nhất này.

- Huấn luyện lại và test cho mô hình sử dụng đặc trưng tốt nhất.

```
# Huấn luyện lại mô hình best_feature_model với đặc trưng tốt nhất
trên toàn bộ tập huấn luyện
train2 = pd.read_csv('train.csv')
test2 = pd.read_csv('test.csv')

X_k_train = train2.iloc[:, best_index:best_index+1]
y_k_train = train2.iloc[:, -1]

X_k_test = test2.iloc[:, best_index:best_index+1]
y_k_test = test2.iloc[:, -1]

reg2 = LinearRegression().fit(X_k_train, y_k_train)

y_k_test_pred = reg2.predict(X_k_test)

k_prediction = pd.DataFrame({'Actual': y_k_test, 'Predicted': y_k_t
est_pred})
print(k_prediction)
```

```
k_coef = pd.DataFrame({'Feature': name[best_index], 'Coefficient':
reg2.coef_})
print(k_coef)
```

- Code không có gì khó hiểu, ta chỉ tách cột dữ liệu của đặc trưng Income composition of resources ra thành một dataframe, sau đó huấn luyện lại mô hình trên tập train sau đó sử dụng tập test để dự đoán tuổi thọ trung bình, về mặt cơ bản code không khác gì so với câu 1a, ta chỉ phải tiền xử lý dữ liệu đầu vào thôi. Nên ta nhận xét luôn kết quả và công thức:

$$Life\ expectancy = 39.497808.Income\ composition\ of\ \ resources$$

```
    Actual  Predicted
0     65.0  63.116331
1     59.9  62.997838
2     77.8  74.294211
3     77.5  74.254713
4     75.4  73.464757
..     ...        ...
90    78.0  73.662246
91    76.8  75.439647
92    69.2  71.450369
93    61.1  66.710632
94    59.2  63.866789

[95 rows x 2 columns]
                          Feature  Coefficient
0  Income composition of resources    39.497808
```

```
RMSE:  4.1592992820505374
```

- Nhận xét:

- Rõ ràng ta thấy được việc sử dụng đặc trưng này cho ra kết quả khá tốt so với các kết quả khác, sau khi tính toán ta được RMSE ~ 4.1 (có thể gọi là tốt).
- Tuy nhiên khi so sánh với RMSE ở mô hình sử dụng 10 đặc trưng (~ 3.46) thì RMSE của mô hình này vẫn còn khá cao, vậy thì ta có hiểu được rằng, tuổi thọ trung bình phụ thuộc vào rất nhiều các yếu tố khác nhau, có thể có một vài yếu tố không ảnh hướng lớn, nhưng việc sử dụng duy nhất chỉ một đặc trưng và loại bỏ hết các đặc trưng còn lại vẫn không thể phản ảnh được hoàn toàn tuổi thọ trung bình. Vì tính chất của mô hình tuổi thọ trung bình là như vậy, tuổi thọ trung bình bị ảnh hướng bởi nhiều yếu tố nên xây dựng trên nhiều đặc trưng nhất vẫn mang cho chúng ta kết quả tốt nhất (đến hiện tại là vậy)
- Sử dụng một đặc trưng cho dù nó phản ảnh tốt nhất trong tất cả các đặc trưng vẫn không đủ để đánh giá dữ liệu, do đó ta phải tăng số đặc trưng lên. Chính vì vậy ta cần đặt câu hỏi liệu có thể nào xây dựng một mô hình sử dụng số đặc trưng

ít hơn 10 mà lại cho kết quả tốt hơn hay không? Đi vào phần tiếp theo chúng ta sẽ thử tạo ra một mô hình cho ra kết quả tốt hơn.

5.3 Yêu cầu 1c: Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất:

- Để thử tạo ra mô hình tốt hơn, đầu tiên ta phải tìm cách để chọn các đặc trưng phù hợp. Sau đó ta sẽ sử dụng những đặc trưng đó để xây dựng mô hình. (Một hi vọng nhỏ rằng ra có thể tìm được mô hình với ít đặc trưng hơn 10 nhưng lại cho ra kết quả tốt nhất). Thế câu hỏi đặt ra là như thế nào là phù hợp? Có nhiều tiêu chuẩn để đánh giá, tuy nhiên trong đồ án này chúng ta sẽ chọn một tiêu chuẩn đó là hệ số tương quan. Phương pháp này tính toán hệ số tương quan giữa 2 đặc trưng khác nhau. Dựa vào phương pháp này ta có thể đánh giá được đặc trưng nào ảnh hướng đến mô hình của chúng ta nhiều hay ít.

- Trước khi bắt đầu, giải thích một chút về hệ số tương quan, để chúng ta dễ hình dung: **[9]**

- Thông thường mà nói ta có 3 thuật toán tính hệ số tương quan đó là: pearson, spearman, kendall.
- Hệ số tương quan nhận giá trị từ [-1, 1]
  - ✓ r = 0, rõ ràng tức là 2 đặc trưng không liên quan gì nhau, hoàn toàn độc lập.
  - ✓ 0 < |r| < 1: tức là 2 đặc trưng có tương quan.
    - Nếu 0 < |r| < 0.3 thì gọi là tương quan yếu.
    - Nếu 0.3 < |r| < 0.6 gọi là tương quan trung bình.
    - Nếu 0.6 < |r| < 1 thì gọi là tương quan mạnh.
    - Nếu r dương thì gọi là tương quan thuận (cái này tăng thì cái kia tăng theo).
    - Nếu r âm gọi là tương quan nghịch (cái này tăng thì cái kìa giảm) .
  - ✓ |r| = 1: gọi là tương quan hoàn hảo, với 2 đặc trưng khác nhau thì hiếm khi xảy ra việc này.
- Ở một số nguồn trên mạng, một số người không quan tâm đến con số này có ý nghĩa hay không. Hệ số tương quan *chỉ có ý nghĩa* khi *signification* (có thể gọi là sai số hay p_value) *nhỏ hơn* 0.1, một số trường hợp lấy 0.05. Nếu sai số lớn hơn khoảng này thì rõ ràng hệ số tương quan này không có ý nghĩa bởi độ tin cậy không đủ cao (nếu ai đã học xác suất thống kê ở kỳ trước thì có lẽ sẽ hiểu được nhanh phần này). Ta có thể hình dung, việc tính hệ số tương quan này cũng giống như ta đang khảo sát 2 biến ngẫu nhiên vậy, việc khảo sát này ở mức tương đối nên phải có sai số, nếu sai số nhỏ ở mức chấp nhận được (< 0.1 hoặc < 0.05) thì kết quả khảo sát mới có ý nghĩa, nếu sai số quá cao thì việc khảo sát cho ra kết quả không còn tin cậy nữa, nếu vẫn sử dụng thì có thể dẫn đến sai.

- Vậy thì đi vào cài đặt, ta sử dụng hàm *corr()* của pandas để tính hệ số tương quan giữa từng cặp đặc trưng. Đầu tiên ta tính hệ số tương quan của từng cặp đôi một trong 11 đặc trưng để xem thử độ tương quan như thế nào.

```
train_corr = pd.read_csv('train.csv')
correlations = train_corr.corr()
print(correlations)
```

```
                                 Adult Mortality       BMI     Polio  \
Adult Mortality                         1.000000 -0.377211 -0.222643
BMI                                    -0.377211  1.000000  0.260919
Polio                                  -0.222643  0.260919  1.000000
Diphtheria                             -0.227694  0.264848  0.621924
HIV/AIDS                                0.556881 -0.249780 -0.178266
GDP                                    -0.289969  0.323323  0.160011
Thinness age 10-19                      0.266782 -0.511107 -0.253642
Thinness age 5-9                        0.282005 -0.519249 -0.266599
Income composition of resources        -0.467279  0.595546  0.341070
Schooling                              -0.439463  0.607242  0.380066
Life expectancy                        -0.685290  0.610687  0.365705

                                 Diphtheria  HIV/AIDS       GDP  \
Adult Mortality                  -0.227694  0.556881 -0.289969
BMI                               0.264848 -0.249780  0.323323
Polio                             0.621924 -0.178266  0.160011
Diphtheria                        1.000000 -0.193837  0.169308
HIV/AIDS                         -0.193837  1.000000 -0.132489
GDP                               0.169308 -0.132489  1.000000
Thinness age 10-19               -0.325448  0.194401 -0.304619
Thinness age 5-9                 -0.313764  0.210497 -0.305818
Income composition of resources   0.369208 -0.319300  0.500820
Schooling                         0.393962 -0.229626  0.549445
Life expectancy                   0.406798 -0.586578  0.486761

                                 Thinness age 10-19  Thinness age 5-9  \
Adult Mortality                            0.266782          0.282005
BMI                                       -0.511107         -0.519249
Polio                                     -0.253642         -0.266599
Diphtheria                                -0.325448         -0.313764
HIV/AIDS                                   0.194401          0.210497
GDP                                       -0.304619         -0.305818
Thinness age 10-19                         1.000000          0.901653
Thinness age 5-9                           0.901653          1.000000
Income composition of resources           -0.492280         -0.466868
Schooling                                 -0.537999         -0.512841
Life expectancy                           -0.503652         -0.500231

                                 Income composition of resources  Schooling  \
Adult Mortality                                        -0.467279  -0.439463
BMI                                                     0.595546   0.607242
Polio                                                   0.341070   0.380066
Diphtheria                                              0.369208   0.393962
HIV/AIDS                                               -0.319300  -0.229626
GDP                                                     0.500820   0.549445
Thinness age 10-19                                     -0.492280  -0.537999
Thinness age 5-9                                       -0.466868  -0.512841
Income composition of resources                         1.000000   0.822536
Schooling                                               0.822536   1.000000
Life expectancy                                         0.779404   0.754864
```

```
                                Life expectancy
Adult Mortality                       -0.685290
BMI                                    0.610687
Polio                                  0.365705
Diphtheria                             0.406798
HIV/AIDS                              -0.586578
GDP                                    0.486761
Thinness age 10-19                    -0.503652
Thinness age 5-9                      -0.500231
Income composition of resources        0.779404
Schooling                              0.754864
Life expectancy                        1.000000
```

- Sử dụng một số dòng code sau, ta có được hệ số tương quan giữa 10 đặc trưng với đặc trưng life expectancy.

```
correlations_2 = train_corr.corr()['Life expectancy'].drop('Life expectancy')
print(correlations_2)
```

```
Adult Mortality                 -0.685290
BMI                              0.610687
Polio                            0.365705
Diphtheria                       0.406798
HIV/AIDS                        -0.586578
GDP                              0.486761
Thinness age 10-19              -0.503652
Thinness age 5-9                -0.500231
Income composition of resources  0.779404
Schooling                        0.754864
Name: Life expectancy, dtype: float64
```

- Đến đây thì ta đã có được hệ số tương quan cụ thể, tuy nhiên còn một vấn đề cần xét đó chính là signification (hay còn gọi là p_value), chúng ta cần xem thử mức độ tin cậy của kết quả này có đủ cao hay không để có thể sử dụng chúng xây dựng mô hình.

- Tuy nhiên, ở đây chúng ta gặp phải một vấn đề, hàm *corr()* của pandas không trả ra signification. Điều này dẫn đến ta không biết được có nên tin tưởng vào hệ số tương quan này hay không. Rất tiếc pandas cũng không hỗ trợ hàm nào tính signification, tuy nhiên có một số hàm khác của numpy có thể trả ra signification, đó là *corrcoef()* hay hàm p*earsonr()*,… của scipy.
- Nhưng có vẻ như các hàm này chỉ nhận vào 2 đặc trưng riêng lẻ mà không nhận vào cả một mảng đặc trưng, sau một lúc thử thì vẫn không giải quyết được lỗi này. Vì thời gian có hạn nên, chúng ta sẽ không dùng hàm khác, mà sẽ đưa ra một số luận điểm để cho thấy rằng hệ số tương quan trên đáng tin cậy, và có thể phản ảnh tốt mối tương quan của các đặc trưng.

- Vậy thì nhìn vào kết quả tương quan, ta có một số luận điểm sau:

- Thứ nhất: bảng dữ liệu này trong thực tế còn 9 đặc trưng nữa, dựa theo file hướng dẫn thì cô Phan Thị Phương Uyên – giảng viên hướng dẫn đồ án có

đề cập, dựa trên độ tương quan cô đã loại bỏ đi 9 cột ít liên quan đến life expectancy nhất, chính vì vậy những đặc trưng còn lại là liên quan đến life expectancy nhất. Nhìn vào kết quả ta thấy được |r| > 0.3 với mọi r, tức là tất cả đặc trưng ở đây đều có độ tương quan từ trung bình trở lên. Chình vì vậy thuật toán cho ta thấy cả 10 đặc trưng đều có độ tương quan chấp nhận được.

- Thứ hai: Cũng nhìn vào bảng kết quả, ta thấy được đặc trưng Income composition of resources cho ta hệ số tương quan là lớn nhất, lại có câu trước đó 1b, ở câu 1b ta đã sử dụng thuật toán 5_Fold Cross Validation để tìm ra đặc trưng cho ra kết quả tốt nhất, và trùng hợp là cả 2 đều cho ra đặc trưng là Income composition of resources. Lại một lần nữa kết quả độ tương quan được khẳng định là đáng tin cậy
- Thứ ba: ta thấy được kết quả độ tương quan trả về là cả 10 đặc trưng đều có độ tương quan trung bình trở lên, điều này cho ta thấy, nếu độ tương quan này đáng tin thì sử dụng cả 10 đặc trưng này để xây dựng mô hình thì sẽ cho kết quả tốt hơn rất nhiều. Thật vậy, ở câu 1a, ta đã sử dụng cả 10 đặc trưng để xây dựng mô hình, và đến hiện tại, mô hình ấy vẫn cho kết quả tốt nhất.
- Chính vì những lí do trên, ta có thể khẳng định những độ tương quan này có thể tin cậy được.

- Tuy nhiên, mọi việc lại không đơn giản như vậy: như ta nhắm đến ban đầu, ta sẽ dùng thuật toán tìm độ tương quan ta có thể tìm ra những đặc trưng liên quan nhất và dùng đặc trưng đó để xây dựng mô hình hồi quy tuyến tính, với điều kiện tiên quyết là số đặc trưng tương quan đó phải ít hơn 10, nhưng vì dữ liệu đã được lọc nên giờ đây ta lại quay lại điểm xuất phát.

- Vậy thì giờ đây ta phải đi hướng khác, sau một số thử nghiệm, chúng ta quyết định vẫn sử dụng 10 đặc trưng này nhưng mô hình sẽ không giống với mô hình ở câu a vì ta sẽ chuẩn hóa đặc trưng để xem thử liệu ta có thể nào làm giảm RMSE xuống được hay không.

- Để biết hướng chuẩn hóa, chúng ta dựa vào hệ số tương quan. Giờ đây hệ số tương quan phát huy được tác dụng nhờ dấu âm dương của mình. Nếu như tương quan nghịch thì một đặc trưng giảm thì một đặc trưng còn lại sẽ tăng, nếu tương quan thuận thì 1 đặc trưng tăng thì đặc trưng còn lại cũng sẽ tăng. Ta có thể nhìn vào kết quả dự đoán khi sử dụng cả 10 đặc trưng để xem thử xu hướng lớn hơn hay nhỏ hơn của dữ liệu dự đoán từ đó có thể xác định nên tăng hay nên giảm. Nếu cần tăng ta cứ lấy mũ 2, nếu cần giảm ta cứ lấy căn bậc 2. Thật ra, chúng ta có thể viết một số dòng code thử, cứ thử thay phiên nhau lấy căn hoặc lấy mũ liên tục, ta cũng có thể thấy được xu hướng tăng hay giảm khi thay đổi đặc trưng. Ở dưới đây là bảng dự đoán tuổi thọ trung bình ở câu a. Ta thấy xu hướng dự đoán lớn hơn so với thực tế. Vì lẽ đó ta sẽ thử tìm cách giảm xuống.

```
    Actual  Predicted
0     65.0  62.123959
1     59.9  61.837164
2     77.8  76.307811
3     77.5  77.371350
4     75.4  76.482085
..     ...        ...
90    78.0  72.930919
91    76.8  77.806096
92    69.2  71.006486
93    61.1  63.981038
94    59.2  60.041279
```

- Ta chọn đặc trưng đầu tiên là Adult Mortality, nhìn vào bảng tương quan ta thấy được đặc trưng này tương quan nghịch với đặc trưng life expectancy do đó ta phải tìm cách nâng giá trị của đặc trưng này lên để life expectancy dự đoán giảm xuống. Sau một số dòng thử nghiệm ta có kết quả sau*: (phần code nằm trong cell test cho một số mô hình)*

```
Average RMSE (none norm):  3.742170233848056
Average RMSE (Adult Mortality power 1.1):  3.7319567825467876
Average RMSE (Adult Mortality power 1.3):  3.7226732462846557
Average RMSE (BMI power 2):  3.71563865690936
```

- Kết quả khá rõ ràng, ta nhận thấy xu cách ta chuẩn hóa mang đến kết quả tốt hơn, tiếp tục chuẩn hóa một số đặc trưng còn lại (một số đặc trưng không có độ tương quan cao ta sẽ không chuẩn hóa để tránh việc sau khi chuẩn hóa lại làm lệch kết quả dự đoán theo hướng tiêu cực) ta có được kết quả sau:

```
Average RMSE (none norm):  3.742170233848056
Average RMSE (Adult Mortality power 1.1):  3.7319567825467876
Average RMSE (Adult Mortality power 1.3):  3.7226732462846557
Average RMSE (BMI power 2):  3.71563865690936
Average RMSE (Diphtheria power 3):  3.6941871537430986
Average RMSE (HIV/AIDS power 0.5):  3.4691954225999893
Average RMSE (GDP power 0.5):  3.465415908775074
Average RMSE (Income.. * 10 power 0.5):  3.3827598455811567
Average RMSE (sSchooling power 0.5):  3.37113566337401
```

- Sau khi có được kết quả trên ta sẽ bắt đầu xây dựng đặc trưng thứ nhất, tuy nhiên trước khi xây dựng đặc trưng bằng phương pháp chuẩn hóa, thì ta cần xây dựng thêm một số mô hình để so sánh, các kết quả trên là test cho đơn lẻ đặc trưng thay đổi, khi nhiều đặc trưng cùng chuẩn hóa, không biết được liệu kết quả cho ra là tốt hay xấu, do đó ta phải test trước cho mô hình không chuẩn hóa nhằm lấy căn cứ để so sánh. Tức là nếu mô hình ta xây dựng là k đặc trưng trong đó k đặc trưng có một vài đặc trưng nào đó chuẩn hóa, thì trước đó ta test trước cho mô hình cũng với k đặc trưng nhưng không có đặc trưng nào được chuẩn hóa. Điều này giúp ta dễ so sánh hơn các kết quả *(phần code test này sẽ được để trong cell test cho một số mô hình)*

- **Mô hình 1:** sử dụng cả 10 đặc trưng và chuẩn hóa các đặc trưng sau: Adult Mortality, BMI, Diphtheria, HIV/AIDS, Income…, Schooling

```
train_3_1 = train_3.copy()
feature = ['Adult Mortality', 'BMI', 'Polio', 'Diphtheria', 'HIV/AI
DS', 'GDP', 'Thinness age 10-19', 'Thinness age 5-
9', 'Income composition of resources', 'Schooling']
X_train_1 = train_3_1[feature].copy()
y_train_1 = train_3_1.iloc[:, -1].copy()

X_train_1.iloc[:, 0:1] = (X_train_1.iloc[:, 0:1] ** 2).copy()
X_train_1.iloc[:, 1:2] = (X_train_1.iloc[:, 1:2] ** 2).copy()
X_train_1.iloc[:, 2:3] = (X_train_1.iloc[:, 2:3]).copy()
X_train_1.iloc[:, 3:4] = (X_train_1.iloc[:, 3:4] ** 3).copy()
X_train_1.iloc[:, 4:5] = (X_train_1.iloc[:, 4:5] ** 0.5).copy()
X_train_1.iloc[:, 5:6] = (X_train_1.iloc[:, 5:6]).copy()
X_train_1.iloc[:, 6:7] = (X_train_1.iloc[:, 6:7]).copy()
X_train_1.iloc[:, 7:8] = (X_train_1.iloc[:, 7:8]).copy()
X_train_1.iloc[:, 8:9] = ((X_train_1.iloc[:, 8:9] * 10) ** 2).copy(
)
X_train_1.iloc[:, 9:10] = (X_train_1.iloc[:, 9:10] ** 0.5).copy()

rg_3_1 = LinearRegression()
scores_3_1 = cross_val_score(rg_3_1, X_train_1, y_train_1, cv=5, sc
oring='neg_mean_squared_error')
print("Average RMSE: ", np.sqrt(-scores_3_1).mean())
ARMSE.append(np.sqrt(np.absolute(scores_3_1)).mean())
```

+ Phần code không có gì phức tạp, chỉ là trước khi đưa tập dữ liệu vào, ta phải chuẩn hóa trước thôi, còn lại thì không khác với câu 1b. Sau khi sử dụng thuật toán 5_Fold Cross Validation thì ta có kết quả RMSE sau:

```
Average RMSE:  3.408323208506707
```

```
Average RMSE(none norm):  3.742170233848056
```

+ So sánh với RMSE của mô hình không có chuẩn hóa: ta thấy được mô hình sau khi chuẩn hóa lại cho RMSE thấp hơn bất ngờ, chứng tỏ hướng chúng ta đang làm cho ra kết quả khả quan. Tuy nhiên mô hình này vẫn không phải là mô hình chúng ta nhắm đến, nhắc lại một chút mô hình ta nhắm đến ở phần này là mô hình sử dụng ít đặc trưng hơn nhưng lại thể hiện tốt hơn tức là cho kết quả tốt hơn. Do đó hãy thực hiện thêm 2 mô hình nữa rồi ta sẽ nhận xét hết cả 3 để xem liệu có mô hình nào mang kết quả tốt hơn hay không.

- Mô hình 2: Sau khi thấy được hiệu quả của mô hình 1, ta tìm cách bỏ đi những đặc trưng không chuẩn hóa, chỉ sử các đặc trưng chuẩn hóa thôi thì sẽ ra sao, tức là ở trong mô hình này ta sử dụng các đặc sau: Adult Mortality, BMI, Diphtheria, HIV/AIDS, Income…, Schooling, sử dụng thêm GDP nữa và ta sẽ chuẩn hóa y chang ở mô hình 1.

```
train_3_2 = train_3.copy()
feature = ['Adult Mortality', 'BMI', 'Diphtheria', 'HIV/AIDS', 'GDP
', 'Income composition of resources', 'Schooling']
X_train_2 = train_3_2[feature].copy()
y_train_2 = train_3_2.iloc[:, -1].copy()

X_train_2.iloc[:, 0:1] = (X_train_2.iloc[:, 0:1] ** 2).copy()
X_train_2.iloc[:, 1:2] = (X_train_2.iloc[:, 1:2] ** 2).copy()
X_train_2.iloc[:, 4:5] = (X_train_2.iloc[:, 4:5] ** 3).copy()
X_train_2.iloc[:, 3:4] = (X_train_2.iloc[:, 3:4] ** 0.5).copy()
X_train_2.iloc[:, 4:5] = (X_train_2.iloc[:, 4:5] ** 0.25).copy()
X_train_2.iloc[:, 5:6] = ((X_train_2.iloc[:, 5:6] * 10) ** 2).copy(
)
X_train_2.iloc[:, 6:7] = (X_train_2.iloc[:, 6:7] ** 0.5).copy()

rg_3_2 = LinearRegression()
scores_3_2 = cross_val_score(rg_3_2, X_train_2, y_train_2, cv=5, sc
oring='neg_mean_squared_error')
print("Average RMSE: ", np.sqrt(-scores_3_2).mean())
ARMSE.append(np.sqrt(np.absolute(scores_3_2)).mean())
```

+ Phần code tương tự ở mô hình 1, ta đi vào luôn kết quả:

```
Average RMSE:  3.419975208604298
```

```
Average RMSE (none norm):  3.7504471602130884
```

+ So sánh với mô hình không chuẩn hóa ta thấy được, kết quả cho ra vẫn tốt như mong đợi. Tuy nhiên nếu so với kết quả của mô hình 1 thì vẫn chưa thể gọi là tốt hơn được (vì RMSE vẫn còn cao hơn dù chỉ là một chút), điều này thể hiện rằng sử dụng cả 10 đặc trưng vẫn cho kết quả tốt hơn và việc lượt bỏ các đặc trưng này đi đã dẫn đến thất bại cho mô hình 2.

- Mô hình 3: Sau một lúc thử nghiệm, ta phát hiện ra rằng, việc chuẩn hóa vẫn mang đến cho ta kết quả tốt như mong đợi, tuy nhiên ở mô hình 2 ta thêm vào mô hình một số các đặc trưng có độ tương quan thấp so với các đặc trưng còn lại, có lẽ điều này dẫn đến việc xung đột gây là độ lệch vẫn còn cao. Do đó ở mô hình tiếp theo ta sẽ giữ lại những đặc trưng *có độ tương quan với life expectancy cao và độ tương quan lẫn nhau của chúng cũng cao* (dựa vào bảng độ tương quan). Sau một lúc lọc, chúng ta sẽ loại bỏ đi Diphtheria, GDP và schooling (2 đặc trưng này tương quan thấp với tương quan còn lại, mà khi 3 đặc trưng này biến thiên chỉ gây ra một số chênh lệch nhỏ không đáng kể). Đồng thời ta sẽ thay đổi một chút về hệ số chuẩn hóa. Có vẻ như hệ số chẵn như 2, 3 vẫn chưa đủ tốt. Vậy mô hình này ta sử dụng 4 đặc trưng: Adult Mortality, BMI, HIV/AIDS, Income…. đồng thời thay đổi hệ số chuẩn hóa.

```
train_3_3 = train_3.copy()
feature = ['Adult Mortality', 'BMI', 'HIV/AIDS', 'Income compositio
n of resources']
```

```
X_train_3 = train_3_3[feature].copy()
y_train_3 = train_3_3.iloc[:, -1].copy()

X_train_3.iloc[:, 0:1] = (X_train_3.iloc[:, 0:1] ** 1.5).copy()
X_train_3.iloc[:, 1:2] = (X_train_3.iloc[:, 1:2] ** 1.5).copy()
X_train_3.iloc[:, 2:3] = (X_train_3.iloc[:, 2:3] ** 0.25).copy()
X_train_3.iloc[:, 3:4] = ((X_train_3.iloc[:, 3:4] * 10) ** 2.5).cop
y()

rg_3_3 = LinearRegression()
scores_3_3 = cross_val_score(rg_3_3, X_train_3, y_train_3, cv=5, sc
oring='neg_mean_squared_error')
print("Average RMSE: ", np.sqrt(-scores_3_3).mean())
ARMSE.append(np.sqrt(np.absolute(scores_3_3)).mean())
```

+ Ta có kết quả sau khi test là:

```
Average RMSE:  3.3803649445182793
```

```
Average RMSE (none norm):  4.093320413096844
```

+ So sánh với RMSE của mô hình không chuẩn hóa, ta vẫn thấy được độ hiệu quả của việc chuẩn hóa các đặc trưng. Tiếp tục so sánh với mô hình 1 (hiện có kết quả tốt nhất trong 2 mô hình trước đó). Và thật bất ngờ khi RMSE của mô hình này nhỏ hơn 0.2 đơn vị so với mô hình 1. Tuy chỉ là số lẻ nhưng chứng tỏ hướng đi của ta hoàn toàn chính xác. Giờ thì xem lại kết quả của cả 3 mô hình:

```
              Model  Average RMSE
0  Use 10 features      3.408323
1   Use 7 features      3.419975
2   Use 4 features      3.380365
```

STT	Mô hình với một đặc trưng	RMSE
1	Sử dụng cả 10 đặc trưng	3.408323
2	Sử dụng 7 đặc trưng (Adult Mortality, BMI, Diphtheria, HIV/AIDS, Income composition of resources, GDP)	3.419975
3	Sử dụng 4 đặc trưng (Adult Mortality, BMI, HIV/AIDS, Income composition of resources)	3.380365

- Nhận xét:

- Đầu tiên ta phải nói đến mô hình sử dụng cả 10 đặc trưng, thật sự là mô hình sử dụng 10 đặc trưng này quá tốt, sau khi chuẩn hóa thì nó lại càng tốt hơn, thật khó để kiếm mô hình nào vượt qua nó.

- Lúc đầu bằng việc loại bỏ các đặc trưng không chuẩn hóa và chỉ sử dụng các đặc trưng chuẩn hóa để tạo ra mô hình 2 không được kết quả như mong muốn cho lắm. Nhưng sau khi tìm hiểu được mối tương quan giữa các cặp đặc trưng với nhau thì tiếp tục loại bỏ đi những đặc trưng không cần thiết để tạo nên mô hình 3.
- Thật sự là mạo hiểm khi chỉ dùng 4 đặc trưng để đánh giá mô hình, càng ít đặc trưng thì càng khó để phán ảnh tuổi thọ trung bình, mô hình 2 trước đó là sự thất bại của việc loại bỏ đặc trưng. Tuy nhiên mạo hiểm lại mang đến kết quả xứng đáng và thành công ngoài dự kiến ở mô hình 3 khi cho ta ra kết quả tốt nhất với việc sử dụng số đặc trưng ít nhất.

- Giờ thì huấn luyện lại mô hình tốt nhất này (mô hình 3) và sử dụng nó để dự đoán tuổi thọ trung bình trên tập test.

```
# Huấn luyện lại mô hình my_best_model trên toàn bộ tập huấn luyện
# Phần code cho yêu cầu 1c
train3 = pd.read_csv('train.csv')
test3 = pd.read_csv('test.csv')

feature = ['Adult Mortality', 'BMI', 'HIV/AIDS', 'Income compositio
n of resources']
X_train_best_model = train3[feature].copy()
X_test_best_model = test3[feature].copy()

X_train_best_model.iloc[:, 0:1] = (X_train_best_model.iloc[:, 0:1]
** 1.5).copy()
X_train_best_model.iloc[:, 1:2] = (X_train_best_model.iloc[:, 1:2]
** 1.5).copy()
X_train_best_model.iloc[:, 2:3] = (X_train_best_model.iloc[:, 2:3]
** 0.25).copy()
X_train_best_model.iloc[:, 3:4] = ((X_train_best_model.iloc[:, 3:4]
 * 10) ** 2.5).copy()

X_test_best_model.iloc[:, 0:1] = (X_test_best_model.iloc[:, 0:1] **
 1.5).copy()
X_test_best_model.iloc[:, 1:2] = (X_test_best_model.iloc[:, 1:2] **
 1.5).copy()
X_test_best_model.iloc[:, 2:3] = (X_test_best_model.iloc[:, 2:3] **
 0.25).copy()
X_test_best_model.iloc[:, 3:4] = ((X_test_best_model.iloc[:, 3:4] *
 10) ** 2.5).copy()

reg3 = LinearRegression().fit(X_train_best_model, train.iloc[:, -
1])

y_test_best_model_pred = reg3.predict(X_test_best_model)
prediction_best_model = pd.DataFrame({'Actual': test3.iloc[:, -
1], 'Predicted': y_test_best_model_pred})
```

```
print(prediction_best_model)

coef_best_model = pd.DataFrame({'Coefficient': X_test_best_model.co
lumns, 'Predicted': reg3.coef_})
print(coef_best_model)
```

- Sau khi huấn luyện lại và test trên tập test ta có được kết quả sau cũng như công thức cho mô hình hồi quy tốt nhất:

$$Life\ expectancy = -0.000660.Adult\ Mortality + 0.004581.BMI$$
$$+ -6.763862.HIV\ /\ AIDS + 0.070603.Income\ composition\ of\ \ resources$$

```
     Actual  Predicted
0      65.0  64.456895
1      59.9  64.257273
2      77.8  76.264851
3      77.5  76.591362
4      75.4  75.918359
..      ...        ...
90     78.0  73.363693
91     76.8  77.246326
92     69.2  71.859464
93     61.1  59.710435
94     59.2  56.424584

[95 rows x 2 columns]
                        Coefficient  Predicted
0                  Adult Mortality  -0.000660
1                              BMI   0.004581
2                          HIV/AIDS  -6.763862
3  Income composition of resources   0.070603
```

- Ta tính được kết quả RMSE là:

```
RMSE:  2.9920068304879517
```

- Nhận xét:

- Rõ ràng sau khi test trực tiếp ta thấy được kết quả tốt một cách bất ngờ, RMSE dưới 3. Điều này rõ ràng mô hình của chúng ta có hiệu quả
- Giải thích và đưa ra giả thuyết về mô hình:
  - ✓ Không có gì khó hiểu khi có 2 đặc trưng Adult Mortality và Income composition of resources trong mô hình, bởi ta đều biết 2 đặc trưng này có độ tương quan rất cao với life expectancy, không những vậy, Income composition of resources là một trong những đặc trưng đơn lẻ xây dựng mô hình tốt nhất, mỗi một đặc trưng này phản ảnh được tuổi thọ trung bình tốt hơn nhiều so với các đặc trưng khác.
  - ✓ Tiếp theo khi nhìn vào bảng độ tương quan của các đặc trưng đôi một thì ta nhận thấy BMI và HIV/AIDS lại có độ tương quan khá cao với 2 đặc trưng chính trên.
  - ✓ Giả thuyết có thể cho rằng HIV/AIDS tuy đã hạ thấp rất nhiều nhưng tỷ lệ HIV/AIDS di truyền vẫn còn, cũng như không thể hoàn toàn loại bỏ loại

bệnh này trên khắp thế giới, tuy giảm nhưng ở vẫn còn rải rác ở nhiều nơi nơi trên thế giới. Chính vì vậy việc mắc HIV lại ảnh hưởng đến tỷ lệ người trưởng thành chết (Adult Mortality). Đồng thời nếu thu nhập cao, cuộc sống tốt thì rất khó nhiễm HIV, nên đặc trưng này cũng ảnh hưởng qua lại với thu nhập

- ✓ BMI: tuy không thể hiện gì nhiều nhưng nó lại phản ảnh chất lượng sức khỏe của mỗi người, nếu chất lượng cuộc sống tốt, sức khỏe tốt thì khối lượng cơ thể cũng không quá thấp. Điều này ảnh hướng trực tiếp từ thu nhập. Rõ ràng thu nhập phản ảnh chất lượng cuộc sống.

6. Nhận xét chung cho 16 mô hình:

- Ở đây chỉ nhận xét chung, còn chi tiết tại sao lại theo tiêu chuẩn nêu dưới và hướng đi cũng như số liệu so sánh cụ thể thì vui lòng đọc lại phần nhận xét tại dưới mỗi mô hình, mỗi mô hình sau khi test xong đều có nhận xét và giải thích.

- Việc sử dụng cả 10 đặc trưng cho kết quả rất tốt, và gần như là tốt nhất trong nhóm các mô hình không chuẩn hóa. Vì mô hình này bao gồm hết tất cả các yếu tố ảnh hướng đến tuổi thọ.

- Việc sử dụng 1 đặc trưng riêng lẻ để xây dựng mô hình thì không phải lúc nào cũng tốt, có những đặc trưng ảnh hưởng rất ít đến cả dataset của chúng ta, việc sử dụng chỉ một mình nó khiến cho mô hình chúng ta không được tốt như mong muốn.

- Tuy nhiên cũng tồn tại một đặc trưng mà chỉ sử dụng nó cho ra kết quả không kém (Income composition of resources), tôi cho rằng điều đó đã là rất tốt, bởi chỉ sử dụng một đặc trưng nhưng kết quả cho ra cũng chênh lệch không quá lớn so với mô hình sử dụng 10 đặc trưng. Tuy nhiên như đã nói, life expectancy chịu ảnh hưởng bởi nhiều yếu tố. 1 đặc trưng chỉ ảnh hướng lớn nhất không có nghĩa chỉ có một mình nó ảnh hưởng. Do đó chỉ sử dụng 1 đặc trưng là *không đủ.*

- Tiếp theo là việc sử dụng mô hình chuẩn hóa, cùng là sử dụng bấy nhiêu đặc trưng nhưng có một số đặc trưng ảnh hưởng khá lớn thì ta phải làm mạnh đặc trưng đó, một số đặc trưng không tác động lớn thì ta phải làm giảm đi việc ảnh hưởng. Để làm điều đó thì ta sử dụng chuẩn hóa, việc chuẩn hóa mang đến kết quả tốt hơn nhiều so với việc xây dựng mô hình không chuẩn hóa với cùng số đặc trưng (tiêu điểm nhất là vẫn sử dụng 10 đặc trưng nhưng sau khi chuẩn hóa nó còn cho ra kết quả tốt hơn so với mô hình mà các đặc trưng không được chuẩn hóa)

- Việc lượt bỏ một số đặc trưng là một hướng đi tốt tuy nhiên nó lại mang nhiều rủi ro (điển hình là mô hình 2) đôi lúc lượt bỏ quá nhiều dẫn đến không đủ đặc trưng để đánh giá. Do đó phương pháp lượt bỏ thật sự là mạo hiểm, tuy nhiên nếu ta lượt bỏ theo một số tiêu chuẩn và giữ lại những đặc trưng theo một số tiêu chuẩn nhất định (tiêu chuẩn cụ thể có nêu ở phần nhận xét 1c) thì lại mang đến kết quả tốt bất ngờ.

7. Lời kết:

Tuy phần yêu cầu của báo cáo không có yêu cầu viết lời kết, tuy nhiên tôi vẫn muốn nếu một số cảm tưởng của mình về đồ án này.

Trước tiên, vì đồ án này được hoàn thành trong giai đoạn khá gấp gáp, thứ 7 mới bắt đầu làm và thứ 3 tôi vẫn còn ngồi viết báo cáo, thật sự deadline quá nhiều, nên nếu có sai sót gì đó *không đáng kể* mong rằng người đọc và cô cũng có thể cho qua.

Như đã nói trên vì thời gian không đủ nên thật sự cảm giác vẫn chưa thỏa mãn lắm khi hoàn thành đồ án này, bởi vì còn nhiều thứ có thể cải thiện và tìm hiểu (ví dụ như tính hệ số tương quan,…), nhưng vì quá gấp nên lại không có điều kiện để đi sâu vào để tìm kiếm các hướng đi tốt hơn.

Lúc đầu khi làm đồ án, bản thân cũng không hi vọng lắm việc tìm ra một hình tốt hơn so với mô hình sử dụng cả 10 đặc trưng nhưng sau khi thử nghiệm với dữ liệu và đi từ từ theo dòng suy nghĩ thì kết quả cho ra thật bất ngờ. Tuy chỉ giảm được sai số đi một phần nhỏ nhưng thật đáng mừng khi có thể hoàn thành mục đích ban đầu đặt ra khi làm đồ án. Đến đây tôi mới thấy được sự kì diệu của việc ứng dụng toán học (xác suất thống kê, đại số tuyến tính,…) đã học vào việc phân tích dữ liệu tìm lời giải cho các bài toán thực tế thế này.

Dù vậy cũng thật tốt khi có thể hoàn thành đồ án 100% đúng hạn.

Cảm ơn người đọc cũng như cô Uyên chấm điểm đã đọc báo cáo đến tận đây, xin cảm ơn rất nhiều!

8. Tài liệu tham khảo:

[1] https://pandas.pydata.org/pandas-docs/stable/reference/frame.html

[2] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[4] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection

[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

[6] https://scikit-learn.org/stable/modules/model_evaluation.html

[7] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

[8] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

[9] https://realpython.com/numpy-scipy-pandas-correlation-python/#pearson-correlation-coefficient