VIET NAM NATIONAL UNIVERSITY,HO CHI MINH CITY
UNIVERSITY OF SCIENCE
FACULTY OF: INFORMATIONTECHNOLOGY

Báo cáo đồ án 3

Toán ứng dụng và thống kê cho Công nghệ Thông tin

Họ và tên sinh viên: Lại Minh Thông
Mã số sinh viên: 20127635

Giáo viên hướng dẫn:

Thầy Nguyễn Văn Quang Huy

Thầy Vũ Quốc Hoàng

Cô Phan Thị Phương Uyên

Thầy Lê Thanh Tùng

Thành phố Hồ Chí Minh – 2022

Đồ án 3 – Linear Regression

I. Thông tin cá nhân.

Họ tên người làm đồ án: Lại Minh Thông

Mã số sinh viên: 20127635

II. Các vấn đề đã hoàn thành.

STT	Tên chức năng	Mức độ hoàn thành
1	Yêu cầu 1a - Sử dụng toàn bộ 10 đặc trưng đề bài cung cấp	100%
2	Yêu cầu 1b: Xây dựng model sử dụng duy nhất 1 đặc trưng, tìm model cho kết quả tốt nhất	100%
3	Yêu cầu 1c: Sinh viên tự xây dựng model, tìm model cho kết quả tốt nhất	100%

III. Cài đặt chi tiết

Ngôn ngữ lập trình được sử dụng để giải quyết bài toán của đồ án là Python và trên môi trường Jupyter Notebook.

Mã nguồn: https://github.com/ThongLai/HCMUS_LinearRegressionProject

Công thức tính toán *Linear Regression* và *RMSE* được lấy trong Lab 04.

Trong phần báo cáo này, các đường dẫn tới phần tham khảo sẽ được chú thích với các chỉ số sẽ dẫn đến đường dẫn. Tổng hợp các đường dẫn đến website, tài liệu tham khảo được đặt ở phần *Kết quả ước lượng* sai số tính theo *RMSE* của model trên là *3.87*. Sai số cho thấy model mới hoạt động hiệu quả hơn rất nhiều.

Tài liệu tham khảo

Các thư viện cần thiết:

+ numpy – Tính toán các phép toán ma trận.

+ pandas – Trình bày dữ liệu dễ hình dung dưới dạng DataFrame.

```
import pandas as pd
import numpy as np
```

1 - Khai báo thư viện

Các đặt:

1) *Yêu cầu 1a.*

Sử dụng toàn bộ 10 đặc trưng đề bài cung cấp.

- Công thức hồi quy:
  Life expectancy $= \theta_0 *$ Adult Mortality $+ \theta_1 *$ BMI $+ \theta_2 *$ Polio $+ \theta_3 *$ Diphtheria $+ \theta_4 *$ HIV/AIDS $+ \theta_5 *$ GDP $+ \theta_6 *$ Thinness age 10-19 $+ \theta_7 *$ Thinness age 5-9 $+ \theta_8 *$ Income composition of resources $+ \theta_9 *$ Schooling
- Sử dụng phương thức *fit()* của lớp *OLSLinearRegression()* đã được cài đặt để train model tương ứng với dữ liệu *X_train* và *y_train*.
- Kết quả được model như sau:

	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
0	0.015101	0.09022	0.042922	0.139289	-0.567333	-0.000101	0.740713	0.190936	24.505974	2.393517

1 – Kết quả các trọng số với 10 đặc trưng

- Sử dụng phương thức *predict()* cho model đưa kết quả dự đoán với dư liệu *X_test*, sau đó đánh giá bằng hàm *rmse()* với giá trị được phỏng đoán *y_hat* với giá trị thật *y_test*.

```
7.064046430584032
```

2 – Giá trị RMSE

- Kết quả ước lượng sai số tính theo *RMSE* của model trên là *7.06*

2) *Yêu cầu 1b.*

Xây dựng model sử dụng duy nhất 1 đặc trưng, tìm model cho kết quả tốt nhất.

- Xây dựng hàm *kFold()* để chi nhóm tập dữ liệu và hàm *aveRMSE_CV_best_feature_model()* để tìm ra đặc trưng tốt nhất. Tham khảo [1]
- Tham số đầu vào: *k* nhóm dữ liệu được gập, và tập dữ liệu train *X*,

y

❖ Kết quả trả về: Mảng giá trị RMSE trung bình của mỗi đặc trưng.

❖ Mô tả:

➢ Bước 1: Xáo trộn tập dữ liệu ngẫu nhiên.

```
def kFold(k, X, y):
    # Random data
    radIdx = np.arange(0, X.shape[0])
    np.random.shuffle(radIdx)
    X = (X.T[radIdx].T).reset_index(drop=True)
    y = (y.T[radIdx].T).reset_index(drop=True)
    ...
```

- Sử dụng hàm *np.random.shuffle()* của *numpy* để random theo các index sau đó mask với tập dữ liệu *X*, *y*.
- Đồng thời sử dụng *reset_index()* của *pandas* để các index của dataframe được khởi tạo lại với tham số *drop = True* để xóa cột index cũ.

➢ Bước 2: Tách tập dữ liệu thành *k* nhóm.

```
def kFold(k, X, y):
      ...

    # Split data into k groups
    splitIdx = (X.shape[0] / k * np.arange(1, k)).astype(int)
    X = np.array_split(X, splitIdx)
    y = np.array_split(y, splitIdx)

    X = np.repeat(np.expand_dims(X, 0), k, axis=0)
    y = np.repeat(np.expand_dims(y, 0), k, axis=0)

    Xtrain = X[~np.eye(k, dtype=bool)]
    Xtrain = np.concatenate(Xtrain, 0)
    Xtrain = np.array(np.array_split(Xtrain, k))

    ytrain = y[~np.eye(k, dtype=bool)]
    ytrain = np.concatenate(ytrain, 0)
    ytrain = np.array(np.array_split(ytrain, k))

    Xtest = X[np.eye(k, dtype=bool)]
    ytest = y[np.eye(k, dtype=bool)]


    return Xtrain, ytrain, Xtest, ytest
```

- Sử dụng hàm *np.array_split* để tách tập dữ liệu thành k nhóm theo 1 mảng index. Vì vậy cần định nghĩa mảng index cần tách *splitIdx*.
- Nhân bản các nhóm dữ liệu lên k lần tưng ướng với k lần train và test với hàm *np.repeat()*, đồng thời cần dùng *np.expand_dims()* mở rộng thêm 1 chiều không gian để lúc sau có thể lọc ra nhóm được train và test.
- Map các tập dữ liệu để train (*Xtrain*, *ytrain*) với ma trận đơn vị kiểu *bool* có đường chéo chính có chân trị *False* và phần còn lại *True*. Điều này sẽ tự động loại 1 nhóm test và giữa lại *k - 1* nhóm còn lại để train và với từng tập dữ liệu sẽ loại bỏ các nhóm test các khau.
- Nếu làm điều tương tự với ma trận đơn vị với các chân trị ngược lại thì ta được tập dữ liệu để test (*Xtest*, *ytest*).
- Sau khi lọc dữ liệu. Nhóm dữ liệu để test sẽ nằm rời rạc nhưng vẫn đúng thứ tự. Vì vậy dùng hàm *np. concatenate()* để hợp nhất các nhóm lại và phân thành k tập dữ liệu bằng hàm *np.array_split()*.

➢ Bước 3: Train Model

```
def aveRMSE_CV_best_feature_model(k, X, y):
    featuresNum = X.shape[1]
    Xtrain, ytrain, Xtest, ytest = kFold(k, X, y)

    #Evaluate the RMSE correspond to to each feature
    AveRMSE = np.zeros(featuresNum)
    for i in range(0, featuresNum):
        for j in range(0, k):
            lr = OLSLinearRegression().fit(Xtrain[j, :, i, None], ytrain[j])

            y_hat = lr.predict(Xtest[j, :, i, None])
            AveRMSE[i] += rmse(ytest[j], y_hat)
        AveRMSE[i] /= featuresNum

    return AveRMSE
```

- Lấy tập dữ liệu từ hàm *kFold()*.
- Với mỗi k nhóm, lấy 1 nhóm test và k-1 nhóm còn lại để train.
- Với mỗi đặc trưng i lấy ra từ tập dữ liệu, dùng phương thức *fit()* để khớp dữ liệu theo đúng đặc trưng i.
- Dùng phương thức *predict()* và hàm *rmse()* để lấy kết quả dự đoán và đánh giá RMSE tương ứng với của nhóm test. Sau khi có đủ các giá trị đánh giá RMSE của 1 đặc trưng, lấy trung bình cộng của RMSE của đặc trưng đó.

❖ Sau khi cài đặt hàm, có thể dùng để tìm được đặc trưng tốt nhất như sau:

```
Best Feature:  Schooling
RMSE Value:  5.891225289572967
```

STT	Mô hình với 1 đặc trưng	RMSE
0	Adult Mortality	23.120175
1	BMI	13.957293
2	Polio	9.017413
3	Diphtheria	8.012912
4	HIV/AIDS	33.533437
5	GDP	30.105327
6	Thinness age 10-19	25.93022
7	Thinness age 5-9	25.907665
8	Income composition of resources	6.594661
9	Schooling	5.891225

3 – RMSE với từng đặc trưng

❖ Nhận xét:
- ➢ Đặc trưng tốt nhất là *"Schooling"* với ước lượng sai số tính theo *RMSE* của model trên là *5.89*.
- ➢ Đặc trưng kém nhất là *"HIV/AIDS"* với ước lượng sai số tính theo *RMSE* của model trên là *33.52*.
- ➢ Theo kết quả trên ta có thể cho rằng được trình độ học vấn với mức tiêu chuẩn sống và phát triển (HDI) có thể ảnh hưởng và phản ánh rất nhiều đến tuổi thọ của 1 quốc gia.
- ➢ Ngược lại các chỉ số như GDP, HIV/AIDS thì ít ảnh hưởng đến tuổi thọ.
- ➢ Dựa vào bảng trên ta biết được các đặc trưng nào ảnh hưởng ít hoặc nhiều đến việc dự đoán tuổi thọ của 1 quốc gia. Từ đó ta có thể thay đổi model với để các đặc trưng quan trọng thì có ảnh hưởng nhiều hơn.

❖ Từ kết luận, huấn luyện lại model với duy nhất 1 đặc trưng tốt nhất.

❖ Công thức hồi quy:

$$\text{Life expectancy} = \theta * \text{Schooling}$$

❖ Trọng số θ sau khi được huấn luyện:

θ_0
5.557399

❖ Ước lượng sai số theo RMSE:

```
10.26095039165537
```

❖ Kết quả ước lượng sai số tính theo *RMSE* của model trên là *10.26*. Tuy được tạo ra từ đặc trưng tốt nhất trong tập dữ liệu, nhưng do số lượng đặc trưng chỉ có 1 so với model 10 đặc trưng gốc. Nên vẫn kém hiệu quả hơn so với model đủ 10 đặc trưng tuyến tính.

3) *Yêu cầu 1c.*

Tự xây dựng model, tìm model cho kết quả tốt nhất.

❖ Từ kết quả của yêu cầu 1b đã cho biết được đặc trưng *"Schooling"* là đặc trưng có ảnh hưởng nhiều nhất. Từ đặc trưng này sẽ tìm ra model khác hiệu quả hơn so với model thuần tuyến tính gốc.

❖ Để đánh giá nhận biết được model tốt nhất, sử dụng lại thủ thuật *5-Fold Cross Validation* để đánh giá. Xây dựng hàm *aveRMSE_CV()* để ước lượng giá trị RMSE trung bình cho 1 model như sau:

```
def aveRMSE_CV(k, X, y):
    Xtrain, ytrain, Xtest, ytest = kFold(k, X, y)

    #Evaluate the RMSE for the model
    X_pinv = np.linalg.inv(Xtrain.transpose(0, 2, 1) @ Xtrain) @ Xtrain.transpose(0, 2, 1)
    weights = X_pinv @ np.expand_dims(ytrain, 2)
    y_hat = np.sum(weights.transpose(0, 2, 1) * Xtest, axis=2)
    AveRMSE = rmse(ytest, y_hat)

    return AveRMSE
```

- Lấy tập dữ liệu từ hàm *kFold()*.
- Tính ma trận giả nghịch đảo cho toàn bộ tập *Xtrain*.
- Tìm giá trị các trọng số bằng cách nhân ma trận giả nghịch với tập *ytrain*. Do tập *ytrain* chỉ có đúng 1 đặc trưng *"Life expendency"*, để thực hiện phép nhân này cần chèn thêm 1 không gian bằng hàm *np.expand_dims()*.
- Sau khi có giá trị các trọng số, dùng tập dữ liệu test thay vào các trọng số để dự đoán các kết quả.
- Tính ước lượng trung bình bằng *rmse()*.

❖ Biến đổi model:

➢ Model 1: Một đặc trưng tốt nhất mũ 2

- Công thức hồi quy:

$$\text{Life expectancy} = \theta_0 * (\text{Adult Mortality}) + \theta_1 * (\text{BMI}) + \theta_2 * (\text{Polio}) + \theta_3 * (\text{Diphtheria}) + \theta_4 * (\text{HIV/AIDS}) + \theta_5 * (\text{GDP}) + \theta_6 * (\text{Thinness age 10-19}) + \theta_7 * (\text{Thinness age 5-9}) + \theta_8 * (\text{Income composition of resources}) + \theta_9 * (\text{Schooling})^2$$

- Nâng số mũ cho đặc trưng tốt nhất để thử nghiệm tầm ảnh hưởng có tỉ lệ với số mũ của nó.
- Ước lượng RMSE của model 1 như sau:

1	Một đặc trưng tốt nhất mũ 2	8.81337

- Nhận xét: ước lượng sai số của model này lớn so với model tuyến tính gốc, vì thế sẽ kém hiệu quả cho việc phỏng đoán.
  ⇨ Loại bỏ model này.

➢ Model 2: Một đặc trưng tốt nhất mũ ½

- Công thức hồi quy:

$$\text{Life expectancy} = \theta_0 * (\text{Adult Mortality}) + \theta_1 * (\text{BMI}) + \theta_2 * (\text{Polio}) + \theta_3 * (\text{Diphtheria}) + \theta_4 * (\text{HIV/AIDS}) + \theta_5 * (\text{GDP}) + \theta_6 * (\text{Thinness age 10-19}) + \theta_7 * (\text{Thinness age 5-9}) + \theta_8 * (\text{Income composition of resources}) + \theta_9 * (\text{Schooling})^{\frac{1}{2}}$$

- Sau khi biết rằng mức ảnh hưởng của đặc trưng tốt nhất không gia tăng khi tăng tham số mũ của nó. Vì vậy sẽ thử với phép lấy căn, tức mũ với 1 phân số.
- Ước lượng RMSE của model 2 như sau:

2	Một đặc trưng tốt nhất mũ 1/2	5.282443

- Nhận xét: Ước lượng sai số ở model 2 nhỏ hơn model gốc. Vì vậy model này hiện tại đang hiệu quả nhất.

➢ Model 3: Hai đặc trưng tốt nhất mũ ½

- Công thức hồi quy:

$$\text{Life expectancy} = \theta_0 * (\text{Adult Mortality}) + \theta_1 * (\text{BMI}) + \theta_2 * (\text{Polio}) + \theta_3 * (\text{Diphtheria}) + \theta_4 * (\text{HIV/AIDS}) + \theta_5 * (\text{GDP}) + \theta_6 * (\text{Thinness age 10-19}) + \theta_7 * (\text{Thinness age 5-9}) + \theta_8 * (\text{Income composition of resources})^{\frac{1}{2}} + \theta_9 * (\text{Schooling})^{\frac{1}{2}}$$

- Tương tự để tiếp tục kiểm tra các model, lấy căn đặc trưng tốt kế tiếp là *"Income composition of resources"* lên căn bậc 2 cùng với căn bậc 2 của đặc trưng tốt nhất là *"Schooling"*.

- Ước lượng RMSE của model 3 như sau:

3	Hai đặc trưng tốt nhất mũ 1/2	5.174356

- Nhận xét: Sau nhiều lần phấn phối và chạy ngẫu nhiên, ước lượng sai số ở model 3 nhỏ hơn so với model 2. Vì vậy model này hiện tại đang hiệu quả nhất.

➢ Model 4: Hai đặc trưng tốt nhất mũ ¼

- Công thức hồi quy:

$$\text{Life expectancy} = \theta_0 * (\text{Adult Mortality}) + \theta_1 * (\text{BMI}) + \theta_2 * (\text{Polio}) + \theta_3 * (\text{Diphtheria}) + \theta_4 * (\text{HIV/AIDS}) + \theta_5 * (\text{GDP}) + \theta_6 * (\text{Thinness age 10-19}) + \theta_7 * (\text{Thinness age 5-9}) + \theta_8 * (\text{Income composition of resources})^{\frac{1}{4}} + \theta_9 * (\text{Schooling})^{\frac{1}{4}}$$

- Thử với phép lấy căn lớn hơn là căn bậc 4 với 2 đặc đứng tốt nhất.
- Ước lượng RMSE của model 4 như sau:

4	Hai đặc trưng tốt nhất mũ 1/4	4.018675

- Nhận xét: Ước lượng sai số ở model 4 nhỏ hơn so với model 3. Vì vậy model này hiện tại đang hiệu quả nhất.

❖ Sau model 4, các model như 3 hay nhiều đặc trưng tốt nhất hoặc có căn bậc lơn hơn đều cho ra các model kém hiệu quả hơn.
❖ Vì vậy sẽ lựa chọn model 4 để thực hiện huấn luyện lại với tập dữ liệu gốc.
❖ Kết quả được model như sau:

θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
0.027189	-0.000074	0.044603	-0.015186	-0.052223	0.006812	0.000073	-0.564032	35.210323	3.838126

❖ Ước lượng sai số theo RMSE:

```
3.8706071903223678
```

❖ Kết quả ước lượng sai số tính theo *RMSE* của model trên là *3.87*. Sai số cho thấy model mới hoạt động hiệu quả hơn rất nhiều.

IV. Tài liệu tham khảo

Đồ án 3 – Linear Regression.

https://github.com/ThongLai/HCMUS_LinearRegressionProject

Tham khảo.

[1] https://machinelearningmastery.com/k-fold-cross-validation/