UNIVERSITY OF SCIENCE
FALCUTY OF INFORMATION TECHNOLOGY

SUBJECT: APPLIED MATH AND STATISTICS

PROJECT 3: LINEAR REGRESSION
CLASS 20CLC11

Student: Mai Quý Trung
ID: 20127370

Lecturers: **VŨ QUỐC HOÀNG, NGUYỄN VĂN QUANG HUY, TRẦN THỊ THẢO NHI, PHAN THỊ PHƯƠNG UYÊN**

Table of Contents

I) Bảng công việc

Công việc	Hoàn thành	Ghi chú
Yêu cầu 1a	100%	Độ khó thấp
Yêu cầu 1b	100%	Độ khó cao
Yêu cầu 1c	100%	Độ khó trung bình
Báo cáo kết quả	100%	

II) Tổng quan

1. Giới thiệu đồ án

- Đồ án là 1 bộ data chứ dữ liệu tuổi thọ trung bình được thu thập từ tổ chức WHO và trang web United Nations từ năm 2000 đến 2015 trên tất cả quốc gia.
- Sau quá trình tiền xử lý, dữ liệu có:
  + 1180 dòng dữ liệu
  + 11 cột dữ liệu: 1 giá trị mục tiêu **Life expectancy** và 10 đặc trưng giải thích (đặc trưng giúp tìm giá trị mục tiêu) gồm **Adult Mortality**, **BMI**, **Polio**, **Diphtheria**, **HIV/AIDS**, **GDP**, **Thinness age 10-19**, **Thinness age 5-9**, **Income composition of resources**, **Schooling**
- Sinh viên được cung cấp 2 tập tin:
  + **"train.csv"**: Chứa 1085 mẫu dùng để huấn luyện mô hình
- + **"test.csv"**: Chứa 95 mẫu dùng để kiểm tra mô hình

2. Yêu cầu đồ án

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

1. Xây dựng mô hình dự đoán tuổi thọ trung bình sử dụng hồi quy tuyến tính

- Yêu cầu 1a: Sử dụng toàn bộ 10 đặc trưng đề bài cung cấp
- Yêu cầu 1b: Xây dựng mô hình sử dụng duy nhất 1 đặc trưng, tìm mô hình cho kết quả tốt nhất (sử dụng phương pháp 5-fold Cross Validation)
- Yêu cầu 1c: Sinh viên tự xây mô hình, tìm mô hình cho kết quả tốt nhất (Sử dụng phương pháp 5-fold Cross Validation)

2. Báo cáo kết quả, đánh giá và nhận xét các mô hình đã xây dựng

III) Chi tiết đồ án

1. Môi trường làm việc

- Ngôn ngữ sử dụng: Python (version 3.10.4)
- Text editor và trình biên dịch: Jupyter Notebook
- Source code: 20127370.ipynb, train.csv, test.csv

2. Ý tưởng

- Ta sẽ xây dựng các hàm cần thiết cho toàn bộ project như hàm ***model_mse*** và ***model_rmse*** để tính toán độ sai lệch giữa 2 mô hình dự đoán và kiểm tra.

- Bên cạnh đó, ta còn phải xây dựng 1 class mô hình hồi quy tuyến tính (Linear Regression) và các phương thức đi kèm như ***fit*** và ***predict***.

3. Chi tiết các bước thực hiện

Bước 1: Import thư viện

Trong đồ án này, em sử dụng những thư viện sau:

- ***pandas***: Thư viện này được dùng để đọc dữ liệu từ file csv và chuyển nó thành các dòng và các cột có label riêng tương ứng.
- ***numpy***: Thư viện này giúp chúng ta trong việc xử lí các thuộc tính liên quan đến mảng và ma trận n chiều, ngoài ra còn có 1 số hàm hỗ trợ thao tác hoặc truy vấn các giá trị hoặc thứ tự trong mảng.
- ***sklearn***: Thư viện chuyên dụng cho machine learning, trong đó em sử dụng ***model_selection*** dùng để xáo trộn dữ liệu và chia chúng ra thành các phần bằng nhau theo phương pháp 5-fold Cross Validation.

Bước 2: Đọc dữ liệu

- Đầu tiên, ta sẽ đọc lần lượt 2 file "train.csv" và "test.csv" và lưu chúng vào 2 biến train và test bằng thư viện ***pandas*** với hàm ***pd.read_csv***.[6]
- Sau đó, chúng ta sẽ chia lần lượt train và test thành 4 phần nhỏ bằng hàm ***iloc*** (index location): X_train, y_train và X_test, y_test, trong đó X_train là dữ liệu của 10 đặc trưng trong train, y_train là dữ liệu chứa giá trị mục tiêu trong train, X_test là dữ liệu của 10 đặc trưng trong test và y_test là dữ liệu chứa giá trị mục tiêu trong test.[8]
- Bộ dữ liệu train và test sau khi đọc sẽ được trực quan hoá như sau:

	Adult Mortality	BMI	Polio	Diphtheria	HIV/AIDS	GDP	Thinness age 10-19	Thinness age 5-9	Income composition of resources	Schooling	Life expectancy
0	268.0	18.1	62.0	64.0	0.1	631.744976	17.7	17.7	0.470	9.9	59.9
1	272.0	17.6	67.0	67.0	0.1	669.959000	17.9	18.0	0.463	9.8	59.5
2	275.0	17.2	68.0	68.0	0.1	63.537231	18.2	18.2	0.454	9.5	59.2
3	279.0	16.7	66.0	66.0	0.1	553.328940	18.4	18.4	0.448	9.2	58.8
4	281.0	16.2	63.0	63.0	0.1	445.893298	18.6	18.7	0.434	8.9	58.6

(Hình 1: 5 dòng dữ liệu đầu tiên trong "train.csv")

	Adult Mortality	BMI	Polio	Diphtheria	HIV/AIDS	GDP	Thinness age 10-19	Thinness age 5-9	Income composition of resources	Schooling	Life expectancy
0	263.0	19.1	6.0	65.0	0.1	584.259210	17.2	17.3	0.479	10.1	65.0
1	271.0	18.6	58.0	62.0	0.1	612.696514	17.5	17.5	0.476	10.0	59.9
2	74.0	58.0	99.0	99.0	0.1	3954.227830	1.2	1.3	0.762	14.2	77.8
3	8.0	57.2	98.0	98.0	0.1	4575.763787	1.2	1.3	0.761	14.2	77.5
4	11.0	58.4	95.0	95.0	0.1	547.851700	6.0	5.8	0.741	14.4	75.4

(Hình 2: 5 dòng dữ liệu đầu tiên trong "test.csv")

Bước 3: Cài đặt các hàm cần thiết

Ta sẽ tiến hành xây dựng 1 class OLS Linear Regression model đã được học từ Lab 4 với 3 phương thức (hàm) sau: ***fit***, ***get_params*** và ***predict***. Bên cạnh đó, ta còn phải viết thêm 2 hàm dùng để tính sai số bình phương: ***model_mse*** và ***model_rmse***.

- ***fit***: Hàm này nhận thông số đầu vào là mô hình chứa đặc trưng A và mô hình chứa kết quả b, hàm có tác dụng tính giá trị ***x*** trong phương trình $\boldsymbol{Ax \approx b}$ hay nói cách khác sẽ lồng 2 mô hình A và b vào với nhau để trả về 1 bộ giá trị chung duy nhất. Công thức đó được tính như sau: $\boldsymbol{x = (A^TA)^{-1}.A^T.b}$ (trong đồ án này sẽ sử dụng w, X, y thay thế cho x, A, b).
- ***get_params***: Hàm sẽ trả về giá về giá trị ***w*** của mô hình.
- ***predict***: Thông số đầu vào của hàm là mô hình chứa đặc trưng dùng để kiểm tra với mô hình kết quả. Hàm sẽ thực hiện lấy bộ giá trị ***w*** chuyển đổi thành mảng phẳng liền với hàm ***np.ravel*** nhân với mô hình đặc trưng đó (nhân 2 ma trận) và trả về kết quả dự đoán.
- ***model_mse***: Nhận đầu vào là mô hình dự đoán và mô hình kết quả, hàm có tác dụng tính trung bình của các tổng bình phương của các giá trị độ lệch trong 2 mô hình, trong đó có dùng hàm ***np.ravel()*** và ***np.mean()*** để hỗ trợ tính toán.[1][2]
- ***model_rmse***: Chứa đầu vào tương tự như ***model_mse***, tuy nhiên hàm trả về giá trị lấy căn của hàm ***model_mse***. Do yêu cầu đồ án cần tính giá trị RMSE của các mô hình nên ta sẽ sử dụng chủ yếu hàm này.

Bước 4: Yêu cầu 1a

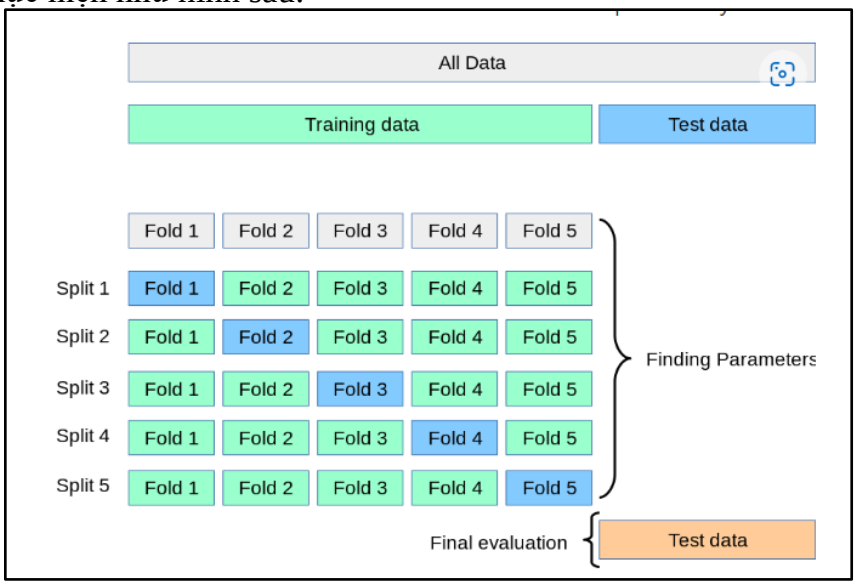
Trong yêu cầu này, ta xây dựng 2 hàm ***train_all_feature*** để huấn luyện toàn bộ 10 đặc trưng đề bài cung cấp và hàm ***task_1a*** để tính giá trị theo yêu cầu đề.

- ***train_all_feature***: Hàm nhận đầu vào là 3 mô hình gồm mô hình đặc trưng cho huấn luyện, mô hình kết quả cho huấn luyện và mô hình đặc trưng để kiểm tra. Hàm sẽ gọi hàm ***fit*** để trả ra bộ giá trị ***w*** và dùng nó để dự đoán mô hình bằng hàm ***predict***.
- ***task_1a***: Hàm này chứa 4 tham số đầu vào gồm mô hình đặc trưng huấn luyện, mô hình kết quả huấn luyện, mô hình đặc trưng kiểm thử và mô hình kết quả kiểm thử. Cả 4 tham số phải được chuyển về dạng mảng numpy bằng hàm ***to_numpy()***. Hàm sẽ sử dụng ***train_all_feature*** để lấy bộ giá trị ***w*** và mô hình dự đoán. Sau đó sẽ dùng hàm ***model_rmse*** để tính RMSE giữa mô hình dự đoán và mô hình kết quả kiểm thử.
- Sau khi xây dựng 2 hàm trên xong, ta gọi hàm và tìm ra được RMSE và công thức hồi quy của mô hình theo yêu cầu đề bài cung cấp (sẽ được nêu ở phần báo cáo kết quả).

<u>Bước 5</u>: Yêu cầu 1b

Tiếp theo, với yêu cầu 1b, ta sẽ xây dựng hàm ***task_1b*** để xây dựng mô hình sử dụng duy nhất 1 đặc trưng và tìm mô hình cho kết quả tốt nhất theo phương pháp 5-fold Cross Validation và dự đoán lại với mô hình kết quả kiểm thử.

- Đầu tiên, ta sẽ có biến category là 1 list chứa các tên đặc trưng tương ứng với các cột trong mô hình đề bài.
- ***model_selection.KFold()***: Ta sẽ dùng hàm này từ thư viện ***sklearn***, với tham số đầu vào là ***n_splits=5*** (theo yêu cầu đề) và ***shuffle=True***. Hàm sẽ thực hiện xáo trộn dataset một cách ngẫu nhiên (1 lần duy nhất), sau đó sẽ chia dữ liệu ra làm 5 phần bằng nhau với hàm ***split***. Sau đó, phương pháp 5-fold Cross Validation sẽ được thực hiện như hình sau:



(Hình 3: Phương pháp 5-fold Cross Validation)

- Đầu tiên, ta sẽ thực hiện phương pháp 5-fold Cross Validation trên mô hình huấn luyện, mô hình kiểm tra sẽ được sử dụng sau cùng. Mô hình huấn luyện được xáo trộn 1 lần duy nhất và chia làm 5 phần bằng nhau với hàm ***model_selection.KFold()*** và ***np.zeros*** để khởi tạo giá trị ban đầu.[3][5]
- Sau đó sẽ tiến hành thực hiện trong vòng lặp gồm 5 split, trong đó data huấn luyện chiếm 4 phần còn data kiểm thử chiếm 1 phần. Data kiểm thử chiếm 1/5 số dòng đầu tiên ở split 1, và cứ thế chiếm 1/5 số dòng trong các split sau cho đến hết.
- Mô hình huấn luyện lớn sẽ được chia ra thành 10 mô hình nhỏ tương ứng với 10 đặc trưng bằng các hàm ***iloc*** và ***loc***, mỗi đặc trưng sẽ đi chung với giá trị mục tiêu ***"Life expectancy"***.
- 10 mô hình nhỏ này sẽ được thực hiện qua 5 split, cứ mỗi split sẽ được huấn luyện và dự đoán data kiểm thử bằng hàm ***task_1a***.
- Sau cùng, mỗi mô hình sẽ có 5 giá trị ***RMSE***, tiến hành lấy trung bình ta có được ***RMSE*** của đặc trưng đó.

- Như vậy, chúng ta sẽ có 10 giá trị ***RMSE*** tương ứng với 10 đặc trưng trong 10 mô hình nhỏ. Từ đó ta sẽ chọn ra được đặc trưng có ***RMSE*** nhỏ nhất thông qua việc lấy vị trí trong mảng ***category*** chứa các tên đặc trưng bằng hàm ***np.argmin***, đó chính là đặc trưng tốt nhất của mô hình huấn luyện.[4]
- Có được đặc trưng tốt nhất, ta sẽ lấy mô hình chứa đặc trưng đó để huấn luyện lại trong toàn bộ mô hình huấn luyện và dự đoán kết quả mô hình kiểm thử bằng hàm ***task_1a***. Cuối cùng, ta sẽ thu về được giá trị ***RMSE*** của mô hình chứa đặc trưng tốt nhất cùng với công thức hồi quy tương ứng (sẽ được nêu ở phần báo cáo kết quả).

Bước 6: Yêu cầu 1c

- Với yêu cầu 1c, ta sẽ xây dựng hàm ***task_1c***, chức năng của hàm gần như tương tự câu 1b với việc sẽ chia mô hình huấn luyện thành m mô hình nhỏ (tối thiểu 3) bằng việc sử dụng các hàm ***loc*** và ***iloc***, mỗi mô hình nhỏ sẽ được xáo trộn 1 lần duy nhất và chia ra làm 5 phần bằng nhau theo phương pháp 5-fold Cross Validation với hàm ***model_selection.KFold***.
- Sau đó ta cũng huấn luyện và cho ra kết quả của m giá trị RMSE tương ứng với m mô hình nhờ vào hàm ***task_1a***. Từ đó ta tìm ra được mô hình tốt nhất huấn luyện được.
- Tương tự 1b, ta sẽ sử dụng mô hình tốt nhất đó cùng với hàm ***task_1a*** huấn luyện lại trên cả tập mô hình huấn luyện, dự đoán mô hình kiểm thử và cuối cùng cho ra giá trị RMSE được đúc kết từ mô hình tốt nhất trong m mô hình.
- Để cho ra kết quả trực quan và khách quan nhất, ta sẽ xây dựng tổng cộng 7 mô hình (m = 7). Đặc điểm của các mô hình đó như sau:
  + Mô hình 1: chứa 6 đặc trưng bao gồm *Thinness age 10-19, Polio, Diphtheria, Thinness age 5-9, Income composition of resources, Schooling.*
  + Mô hình 2: chứa 4 đặc trưng bao gồm *BMI, GDP, HIV/AIDS, Schooling*
  + Mô hình 3: chứa 3 đặc trưng bao gồm *BMI, GDP, Polio*
  + Mô hình 4: chứa 2 đặc trưng bao gồm *Schooling* và $Schooling^2$ (lý do ta chọn mô hình này để khảo sát tiềm năng mô hình có độ sai lệch nhỏ nhất và vì Schooling là đặc trưng tốt nhất từ câu 1b)
  + Mô hình 5: mô hình 10 đặc trưng nhưng các giá trị được bình phương
  + Mô hình 6: mô hình 10 đặc trưng nhưng các giá trị được lập phương
  + Mô hình 7: mô hình 1 đặc trưng $Schooling^2$ (để đối chiếu với mô hình Schooling ở câu 1b)
- Tóm lại, với mô hình tốt nhất từ 1 trong 7 mô hình trên, ta sẽ thu được RMSE và công thức hồi quy tương ứng (sẽ được nêu ở phần báo cáo kết quả). Các mô hình 4, 5, 6, 7 có sử dụng hàm pd.DataFrame để ép kiểu từ np array về DataFrame.[7]

IV) Phân tích và thống kê

1. Báo cáo kết quả

- Yêu cầu 1a:

STT	Mô hình	RMSE
1	Mô hình 10 đặc trưng	7.0640464305843516

Life expectancy *= 0.0151013627 * (Adult Mortality) + 0.0902199807 * (BMI) + 0.0429218175 * (Polio) + 0.139289117 * (Diphtheria) - 0.567332827 * (HIV/AIDS) - 0.000100765115 * (GDP) + 0.740713438 * (Thinness age 10-19) + 0.190935798 * (Thinness age 5-9) + 24.5059736 * (Income composition of resources) + 2.39351661 * (Schooling)*

- Yêu cầu 1b:

STT	Mô hình	RMSE
1	Adult Mortality	46.22218977
2	BMI	27.975445
3	Polio	17.9730464
4	Diphtheria	15.99293267
5	HIV/AIDS	67.11997174
6	GDP	60.19826139
7	Thinness age 10-19	51.79117489
8	Thinness age 5-9	51.68112705
9	Income composition of resources	13.22773781
10	Schooling	11.77709437

Life expectancy = 5.5573994 ** (Schooling)*

RMSE *= 10.26095039165537*

- Yêu cầu 1c:

STT	Mô hình	RMSE
1	Mô hình 6 đặc trưng *(Thinness age 10-19, Polio, Diphtheria, Thinness age 5-9, Income composition of resources, Schooling)*	8.35129891
2	Mô hình 4 đặc trưng *(BMI, GDP, HIV/AIDS, Schooling)*	11.1243026
3	Mô hình 3 đặc trưng *(BMI, GDP, Polio)*	15.58614572
4	Mô hình 2 đặc trưng *(School, School^2)*	6.88109152
5	Mô hình 10 đặc trưng bình phương	13.65583384
6	Mô hình 10 đặc trưng lập phương	19.10345167
7	Mô hình 1 đặc trưng bình phương *(Schooling)*	24.42916381

Life expectancy = 9.25853343 ** (Schooling) – 0.27667035 * (Schooling^2)*

***RMSE** = 5.9986721709342445*

2. Phân tích và nhận xét

- Ở câu 1a, do câu 1a yêu cầu đề được tinh chỉnh lược bỏ đi hệ số tự do, hay nói cách khác không sử dụng hàm ***preprocess*** có từ trong Lab 4, nên về mặt lý thuyết, chỉ số RMSE sẽ bị tăng cao hơn nhưng kết quả vẫn cho ra con số khoảng 7.06 cho độ lệch giữa mô hình dự đoán và kiểm tra. Kết quả này cho thấy được tuổi thọ (Life expectancy) được phân bố đều theo 1 tỉ lệ nhất định từ 10 đặc trưng đề bài cho, dẫn đến độ lệch không quá cao.
- Còn ở câu 1b, với kết quả Schooling có độ lệch RMSE thấp nhất, ta có thể dễ dàng rút ra được kết luận thống kê số năm trung bình 1 người đi học sẽ dự đoán tuổi thọ tốt hơn những đặc trưng còn lại (với chỉ số RMSE cuối cùng khoảng 11.78), xếp hạng thứ 2 và 3 sau đó là chỉ số phát triển con người tính theo thu nhập thành phần tài nguyên (Income composition of resources) và tỉ lệ tiêm ngừa uốn ván và ho gà ở trẻ 1 tuổi (Diphtheria) với RMSE lần lượt là 15.99 và 17.97 . Tỉ lệ dự đoán tuổi thọ thấp nhất thuộc về tỉ lệ tử vong nhiễm HIV/AIDS trên 1000 người. Chỉ số RMSE cuối cùng của mô hình tốt nhất khoảng 10,26.
- Cuối cùng, ở câu 1c, mô hình cho ra kết quả tốt nhất thuộc về mô hình 4 gồm 2 đặc trưng Schooling và Schooling^2 với RSME rất thấp khoảng 6.88 . Trong khi đó, mô hình 7 với đặc trưng Schooling^2 có RMSE cao nhất khoảng 24.43 . Điều đó cho thấy được cho dù đó là đặc trưng tốt nhất trong 10 đặc trưng đề bài cho thì khi bình phương giá trị chúng lên, độ sai lệch sẽ bị tăng đáng kể. Ngoài ra còn 1 chi tiết khá thú vị. Với 3 mô hình 1, 2 và 3, số lượng đặc trưng giảm đi thì chỉ số RMSE lại tăng lên. Cụ thể, với 6 đặc trưng RMSE ở mức 8.35, 4 đặc trưng RMSE ở mức 11.12 và 3 đặc trưng thì RMSE ở mức 15.58 . Từ đó ta biết được số lượng đặc trưng cũng ảnh hưởng đến độ sai lệch RMSE, càng nhiều đặc trưng RMSE càng ít sai lệch, dự đoán mô hình chính xác hơn và ngược lại. RMSE cuối cùng của mô hình tốt nhất là khoảng 6.

3. Giải thích và nêu giả thuyết

- Ở câu 1c, ta thấy được rằng càng nhiều đặc trưng thì RMSE càng giảm đi, có vẻ như sẽ gần chạm tới kết quả của câu 1a với đầy đủ 10 đặc trưng ở mức 7.06 RMSE. Tuy nhiên, với mô hình 4, ta thấy được RMSE của mô hình này còn thấp hơn kết quả câu 1a và còn cho ra RMSE cuối cùng là 5.99 thấp hơn nhiều so với 10 đặc trưng thông thường.
- Để giải thích được nguyên nhân này, ta phải xét đến đường tuyến tính đi qua các điểm trong mặt phẳng gồm đầy đủ 10 đặc trưng là đường thẳng (do không có hệ số tự do). Trong khi đó, với mô hình 4, với 2 đặc trưng Schooling thuần nhất và Schooling bậc 2, đồ thị đi qua các thông số trong mặt phẳng là đường cong, chính vì thế cho ra kết quả chính xác hơn dẫn đến RMSE thấp hơn. Một điều nữa làm

cho mô hình này có kết quả tốt hơn chính là do Schooling là đặc trưng tốt nhất mà ta phân tích được từ câu 1b.

- Dựa vào giả thuyết trên, ta có thể xây dựng 1 mô hình còn tốt hơn nữa bằng việc ghép các cột Schooling, Schooling^2, Schooling^3, … để cho ra RMSE ngày càng nhỏ hơn.

V) Kết luận

- Tóm gọn lại, đồ án 3 nghiên cứu về Data Fitting và Linear Regression cho ta có cái nhìn tổng quan hơn về việc huấn luyện các bộ dữ liệu và dự đoán từ các mô hình có tác động lớn trong thực tế: dựa vào các chỉ số đặc trưng chung của các quốc gia để dự đoán được tuổi thọ trung bình của quốc gia đó. Dù là huấn luyện mô hình hay dự đoán mô hình thì bài toán cũng quy về việc xử lí và thao tác trên các ma trận và tìm ra nghiệm từ phương trình hồi quy tuyến tính $\boldsymbol{Ax \approx b}$. Thông qua đó, chúng ta ngày càng am hiểu hơn về những ứng dụng cực kì phong phú và đa dạng của machine learning trong chuyên ngành Khoa học máy tính.

VI) Nguồn tham khảo

[1] **Numpy ravel:** numpy.org/numpy.ravel
[2] **Numpy mean:** numpy.org/numpy.mean
[3] **Numpy zeros:** numpy.org/numpy.zeros
[4] **Numpy argmin:** numpy.org/numpy.argmin
[5] **KFold from sklearn.model_selection:** scikit-learn.org/model-selection.KFold
[6] **Pandas read_csv:** pandas.org/pandas.read-csv
[7] **Pandas DataFrame:** pandas. org/pandas.DataFrame
[8] **Pandas iloc and loc:** statology.org/pandas-loc-vs-iloc

(Some other source code are referenced from Lab 4)