

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN 03

LINEAR REGRESSION

Giảng viên : Vũ Quốc Hoàng
Nguyễn Văn Quang Huy
Lê Thanh Tùng
Phan Thị Phương Uyên

Sinh viên : Bùi Quang Thành

MSSV : 20127329

TP. HỒ CHÍ MINH, THÁNG 7 NĂM 2022

MỤC LỤC

GIỚI THIỆU	3
GIẢI THÍCH HÀM.....	5
CÁC THƯ VIỆN ĐÃ THÊM.....	11
HÌNH ẢNH PHÂN BỐ CỦA TOÀN BỘ 10 ĐẶC TRƯNG.....	11
NHẬN XÉT KẾT QUẢ TỪ TOÀN BỘ CÁC MÔ HÌNH ĐƯỢC XÂY DỰNG.....	12
Câu a.....	12
Mô hình được xây dựng từ 10 đặc trưng	12
Câu b	13
Mô hình được xây dựng từ từng đặc trưng trong dữ liệu theo phương pháp Cross Validation	13
Báo cáo kết quả khi sử dụng đặc trưng tốt nhất (Schooling).....	18
Câu c.....	18
Báo cáo kết quả khi xây dựng 4 mô hình tự thiết kế	18
TÀI LIỆU THAM KHẢO	21

GIỚI THIỆU

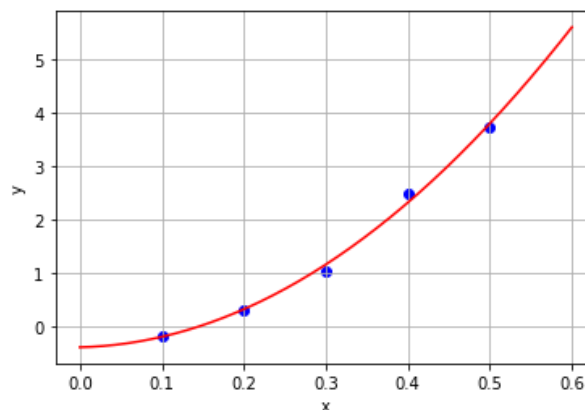
Hồi quy tuyến tính (Linear Regression) là bài toán cơ bản và đơn giản nhất của Machine Learning. Hồi quy tuyến tính thuộc nhóm Học có giám sát (Supervised Learning). Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Trong đề án lần này , chúng ta sẽ sử dụng Linear Regression để dự đoán tuổi thọ trung bình từ dữ liệu được thu thập từ tổ chức WHO trang web United Nations từ năm 2000 đến 2015 trên tất cả quốc gia.[0]

Môi trường thực hiện: Jupyter Notebook.

Ngôn ngữ: Python.

Đường hồi quy tuyến tính

Trong khi sử dụng hồi quy tuyến tính, mục tiêu của chúng ta là để làm sao một đường thẳng có thể tạo được sự phân bố gần nhất với hầu hết các điểm.



Hình ảnh: Đường hồi quy tuyến tính (màu đỏ)

Hình bên trên là đường hồi quy tuyến tính được xây dựng từ phương trình

$$s = s_0 + v_0 t + \frac{1}{2} g t^2$$

Người ta thực hiện thí nghiệm thu được kết quả như sau:

t (x)	0.1	0.2	0.3	0.4	0.5
s (y)	-0.18	0.31	1.03	2.48	3.73

Root Mean Square Error

Root Mean Square Error (RMSE) hoặc Root Mean Square Deviation (RMSD) là căn bậc hai của mức trung bình của các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán).[1]

Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ dàn trải của những phần dư này, nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh đường phù hợp nhất.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

\hat{x}_i = estimated time series

Ví dụ : Sau khi dự đoán tuổi bằng 65 và có sai số RMSE =1.8 → Hiểu đơn giản sai số RMSE là sai số cho phép tuổi dự đoán cộng trừ chúng → Tuổi thật sẽ dao động trong khoảng (63.2,66.8)

PHƯƠNG PHÁP 5-fold CROSS VALIDATION

Cross-validation là một phương pháp kiểm tra độ chính xác của 1 máy học dựa trên một tập dữ liệu học cho trước. Thay vì chỉ dùng một phần dữ liệu làm tập dữ liệu học thì cross-validation dùng toàn bộ dữ liệu để dạy cho máy.

Kỹ thuật này thường bao gồm các bước như sau[2]:

1. Xáo trộn dataset một cách ngẫu nhiên
2. Chia dataset thành k(5) nhóm
3. Với mỗi nhóm:
 1. Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình
 2. Các nhóm còn lại được sử dụng để huấn luyện mô hình
 3. Huấn luyện mô hình
 4. Đánh giá và sau đó hủy mô hình
4. Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá

GIẢI THÍCH HÀM

Tên hàm	def toNumpy(X_train,Y_train,X_test,Y_test):
Mục tiêu	Chuyển đổi các dữ liệu thành kiểu dữ liệu numpy để dễ dàng tính toán Ý tưởng : sử dụng thư viện Numpy
Tham số	<ul style="list-style-type: none"> X_train (Dataframe): dữ liệu huấn luyện 10 đặc trưng Y_train (Series): dữ liệu huấn luyện – chứa 1 giá trị mục tiêu kiểm tra X_test (Dataframe): dữ liệu kiểm tra Y_test (Series) : dữ liệu kiểm tra - chứa 1 giá trị mục tiêu kiểm tra
Trả về	<ul style="list-style-type: none"> X_train (numpy): dữ liệu huấn luyện 10 đặc trưng Y_train (numpy): dữ liệu huấn luyện – chứa 1 giá trị mục tiêu kiểm tra X_test (numpy): dữ liệu kiểm tra Y_test (numpy) : dữ liệu kiểm tra - chứa 1 giá trị mục tiêu kiểm tra

Tên hàm	def caculate_Coefficients_Matrix(X_train,Y_train):
Mục tiêu	Dựa vào dữ liệu huấn luyện để tính ra ma trận hệ số Ý tưởng : dựa vào công thức tính toán ma trận hệ số [6]
Tham số	<ul style="list-style-type: none"> X_train (numpy): dữ liệu huấn luyện 10 đặc trưng Y_train (numpy): dữ liệu huấn luyện – chứa 1 giá trị mục tiêu kiểm tra
Trả về	<ul style="list-style-type: none"> Coefficients (numpy) : 1 ma trận hệ số được tính toán từ dữ liệu X_Train và Y_Train

Tên hàm	def rmse(y, y_hat):
Mục tiêu	Tính toán độ lệch chuẩn của các phần dư (sai số dự đoán)
Tham số	<ul style="list-style-type: none"> y (numpy): dữ liệu kiểm tra – chứa 1 giá trị mục tiêu kiểm tra y_hat (numpy): dữ liệu dự đoán
Trả về	<ul style="list-style-type: none"> number (float) : sai số dự đoán của hai dữ liệu kiểm tra và dữ liệu dự đoán

Tên hàm	def predict(Coefficients,X_test):
Mục tiêu	Dựa vào ma trận hệ số và dữ liệu kiểm tra để dự đoán số tuổi tùy thuộc vào các đặc trưng.

Tham số	<ul style="list-style-type: none"> • Coefficients (numpy): ma trận hệ số • X_test(numpy): dữ liệu muốn kiểm tra
Trả về	<ul style="list-style-type: none"> • predicts (numpy) : là 1 ma trận (số tuổi) được dự đoán từ ma trận hệ số và dữ liệu muốn kiểm tra

Tên hàm	def takeNFeatureBestIndex(data,numberFeature):
Mục tiêu	<p>Dựa vào dữ liệu tìm ra index của các đặc trưng có chỉ số RMSE nhỏ nhất và sắp xếp theo thứ tự (từ rmse nhỏ nhất đến lớn nhất)</p> <p>Ý tưởng:</p> <ul style="list-style-type: none"> • Tạo ra 1 dữ liệu tạm thời bằng cách dùng np.copy() rồi sắp xếp tăng dần • Dùng 2 vòng lặp for để duyệt toàn bộ dữ liệu • Kiểm tra điều kiện rồi đưa các giá trị hợp lệ vào 1 danh sách • Trả về kết quả
Tham số	<ul style="list-style-type: none"> • data (numpy):dữ liệu huấn luyện • numberFeature (int): Số đặc trưng muốn lấy ra (ví dụ numberFeature=10 → lấy ra 10 đặc trưng tốt nhất)
Trả về	<ul style="list-style-type: none"> • indexs(list) :là 1 danh sách chứa vị trí của các đặc trưng tốt nhất (sắp xếp theo thứ tự tăng dần theo chỉ số RMSE)

Tên hàm	def fiveFoldCrossValidation(train_test_split,test_data,n_splits,message):
Mục tiêu	<p>Có nhiệm vụ xáo trộn dữ liệu 1 cách ngẫu nhiên , chia dữ liệu thành n_splits nhóm ,sử dụng các nhóm hiện tại để huấn luyện mô hình, các nhóm còn lại để đánh giá mô hình thông qua các chỉ số RMSE.</p> <p>Ý tưởng :</p> <ul style="list-style-type: none"> • Sử dụng 2 biến để lưu trữ các giá trị dữ liệu hiện tại. • Áp dụng phương pháp Cross Validation để chia dữ liệu thành 5 nhóm dữ liệu (Áp dụng các cú pháp của python (Slicing& Indexing)) • Dùng vòng lặp for duyệt từ 0 → 5 (do 5-fold) :Ứng với mỗi vòng lặp chia dữ liệu thành các tập huấn luyện và tập kiểm tra , với mỗi tập huấn luyện và kiểm tra tiến hành tính ma trận hệ số rồi dự đoán kết quả với mô hình vừa được huấn luyện → tương ứng với 5 RMSE • Tiếp tục hoán đổi để xáo trộn dữ liệu cho đến khi hết vòng lặp
Tham số	<ul style="list-style-type: none"> • train_test_split (numpy):dữ liệu huấn luyện muốn xáo trộn • test_data (numpy): dữ liệu kiểm tra muốn xáo trộn • n_splits (int): chia dữ liệu thành n_splits nhóm

	<ul style="list-style-type: none"> message(string): thông điệp mô hình có sử dụng độ lệch(bias) hay không (“No bias” và “Have bias”)
Trả về	<ul style="list-style-type: none"> mean(float) : là giá trị trung bình của RMSE sau n_splits lần. distance(list): là 1 danh sách chứa các giá trị RMSE sau khi n_splits lần xáo trộn dữ liệu để có thể tính ra giá trị trung bình RMSE ở trên .

Tên hàm	def bestNFeaturedIndex(data,numberFeature):
Mục tiêu	Tính ra các giá trị RMSE trung bình của từng đặc trưng (cụ thể 10 đặc trưng)
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện numberFeature(int) : số đặc trưng của dữ liệu
Trả về	<ul style="list-style-type: none"> nBestIndexs (list) : là 1 danh sách chứa vị trí của các đặc trưng tốt nhất (sắp xếp theo thứ tự tăng dần theo chỉ số RMSE) distances(list): là 1 danh sách chứa tất cả các giá trị RMSE sau khi n_splits lần xáo trộn của toàn bộ 10 đặc trưng.

Tên hàm	def model2FeatureBestData(data,bestIndexOrders):
Mục tiêu	Lấy 2 dữ liệu có chỉ số rmse bé nhất
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số RMSE từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4]
Trả về	<ul style="list-style-type: none"> data[:,[bestIndexOrders[0],bestIndexOrders[1]]] (numpy) : là tập dữ liệu gồm 2 đặc trưng tốt nhất .

Tên hàm	def rmse_Model2FeatureBest(data,bestIndexOrders,y_train):
Mục tiêu	Tính toán chỉ số RMSE trung bình sử dụng phương pháp Cross Validation trên tập dữ liệu huấn luyện
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số rmse từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4] y_train(numpy) : tập dữ liệu kiểm tra – chứa 1 giá trị mục tiêu

Trả về	<ul style="list-style-type: none"> mean(float): là chỉ số RMSE trung bình sau n_splits trên tập dữ liệu kết hợp từ 2 dữ liệu có chỉ số RMSE thấp nhất Coefficients(numpy): ma trận hệ số của tập dữ liệu mới gồm 2 đặc trưng
--------	--

Tên hàm	def model_3Feature_Best_Standardized_Data (data,bestIndexOrders):
Mục tiêu	Lấy 3 dữ liệu có chỉ số RMSE bé nhất và chuẩn hóa với số mũ 0.4 (giải thích bên dưới)
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số RMSE từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4]
Trả về	<ul style="list-style-type: none"> data[:,[bestIndexOrders[0],bestIndexOrders[1],bestIndexOrders[2]]]**0.4 (numpy) : là 1 tập dữ liệu gồm 3 đặc trưng đã được chuẩn hóa

Tên hàm	def rmse_Model3FeatureBestPower (data,bestIndexOrders,y_train):
Mục tiêu	Tính toán chỉ số rmse trung bình sử dụng phương pháp Cross Validation trên tập dữ liệu huấn luyện
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số RMSE từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4] y_train(numpy) : tập dữ liệu kiểm tra – chứa 1 giá trị mục tiêu
Trả về	<ul style="list-style-type: none"> mean(float): là chỉ số RMSE trung bình sau n_splits trên tập dữ liệu kết hợp từ 3 đặc trưng có chỉ số RMSE thấp nhất và được chuẩn hóa Coefficients(numpy): ma trận hệ số của tập dữ liệu mới gồm 3 đặc trưng mới

Tên hàm	def model_2Feature_Best_Plus_Data (data,bestIndexOrders):
Mục tiêu	Lấy ra 1 tập dữ liệu là sự kết hợp của “dữ liệu có chỉ số RMSE thấp nhất” + “dữ liệu có chỉ số RMSE thấp thứ hai”
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện

	<ul style="list-style-type: none"> bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số rmse từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4]
Trả về	<ul style="list-style-type: none"> (data[:,bestIndexOrders[0]]+data[:,bestIndexOrders[1]]) (numpy) là 1 tập dữ liệu gồm 1 đặc trưng mới được cộng từ 2 đặc trưng có chỉ số rmse thấp nhất

Tên hàm	def model_Matrix_Algebra_HaveBias(data):
Mục tiêu	Tính ma trận hệ số và độ lệch (bias)
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện
Trả về	<ul style="list-style-type: none"> coefficients(numpy): là 1 ma trận hệ số bias(float) : độ lệch

Tên hàm	def rmse_Model2FeatureBestPlus(data,bestIndexOrders,y_train):
Mục tiêu	Tính toán chỉ số rmse trung bình sử dụng phương pháp Cross Validation trên tập dữ liệu huấn luyện
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số rmse từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4] y_train(numpy) : tập dữ liệu kiểm tra – chứa 1 giá trị mục tiêu
Trả về	<ul style="list-style-type: none"> mean(float): là chỉ số RMSE trung bình sau n_splits trên tập dữ liệu kết hợp từ 2 đặc trưng có chỉ số RMSE thấp nhất cộng với nhau Coefficients(numpy): ma trận hệ số của tập dữ liệu mới gồm 1 đặc trưng mới

Tên hàm	def rmse_Model_Matrix_Algebra_HaveBias(X_train,y_train):
Mục tiêu	Tính toán chỉ số RMSE trung bình sử dụng phương pháp Cross Validation trên tập dữ liệu huấn luyện Ý tưởng:

	<ul style="list-style-type: none"> Do chúng ta có thêm bias nên thêm 1 cột toàn là số 1 vào cuối phương trình hồi quy tuyến tính → Dữ liệu đã có 11 đặc trưng gán chúng vào 1 biến dữ liệu mới và thao tác trên chúng Tiếp tục tiến hành sử dụng phương pháp Cross Validation trên tập dữ liệu mới để tính RMSE
Tham số	<ul style="list-style-type: none"> X_train (numpy): dữ liệu huấn luyện (gồm 10 đặc trưng) y_train(numpy) : tập dữ liệu kiểm tra – chứa 1 giá trị mục tiêu
Trả về	<ul style="list-style-type: none"> mean(float): là chỉ số rmse trung bình sau n_splits trên tập dữ liệu kết hợp từ 11 đặc trưng. Coefficients(numpy): ma trận hệ số của tập dữ liệu 11 đặc trưng (do có thêm bias) bias(float): độ lệch

Tên hàm	def my_best_model(data,bestIndexOrders,y_train):
Mục tiêu	Tìm mô hình có rmse trung bình bé nhất (sử dụng phương pháp Cross Validation) để tìm mô hình tốt nhất
Tham số	<ul style="list-style-type: none"> data (numpy): dữ liệu huấn luyện bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số rmse từ thấp cho đến cao Ví dụ danh sách index ứng với các đặc trưng tốt nhất [9, 8, 3, 2, 1, 0, 7, 6, 5, 4] y_train(numpy) : tập dữ liệu kiểm tra – chứa 1 giá trị mục tiêu
Trả về	<ul style="list-style-type: none"> mean(float): là chỉ số rmse trung bình sau n_splits trên tập dữ liệu kết hợp từ 11 đặc trưng. Coefficients(numpy): ma trận hệ số của tập dữ liệu mới gồm 11 đặc trưng (do có thêm bias) bias(float): độ lệch

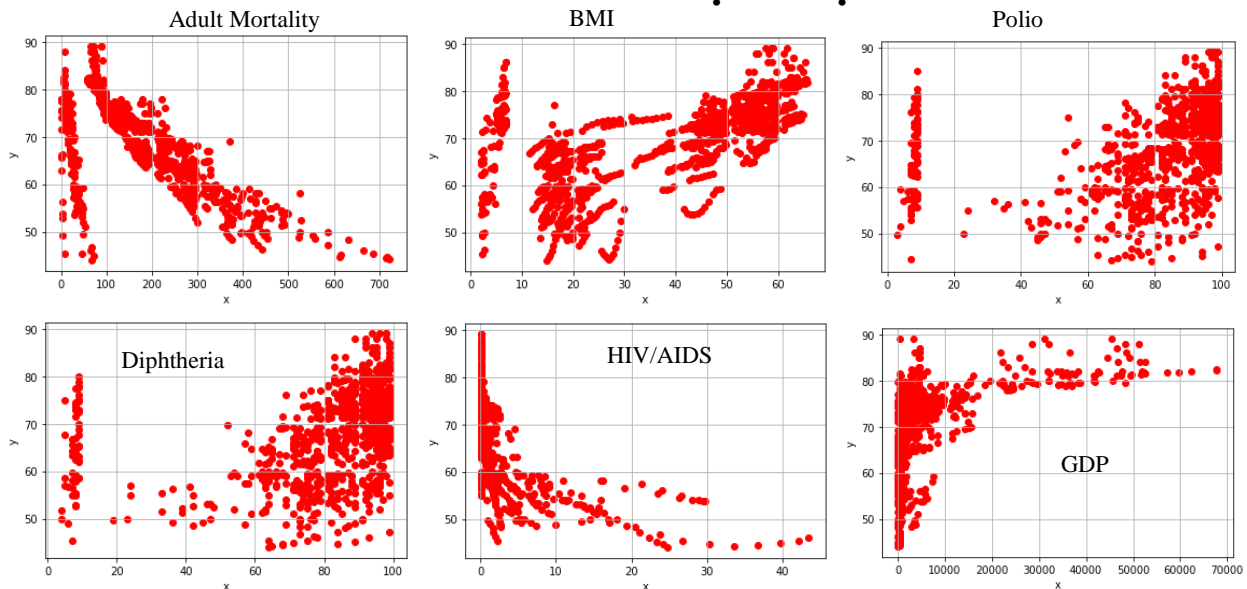
Tên hàm	def rankingDesignModel(rmses):
Mục tiêu	Vẽ 1 bảng chứa thông tin của các mô hình tự thiết kế
Tham số	<ul style="list-style-type: none"> rmses (list): danh sách chứa các giá trị RMSE trung bình sau khi sử dụng phương pháp Cross Validation.
Trả về	<ul style="list-style-type: none"> info (Dataframe) : Chứa các thông tin cần thiết trong các mô hình.

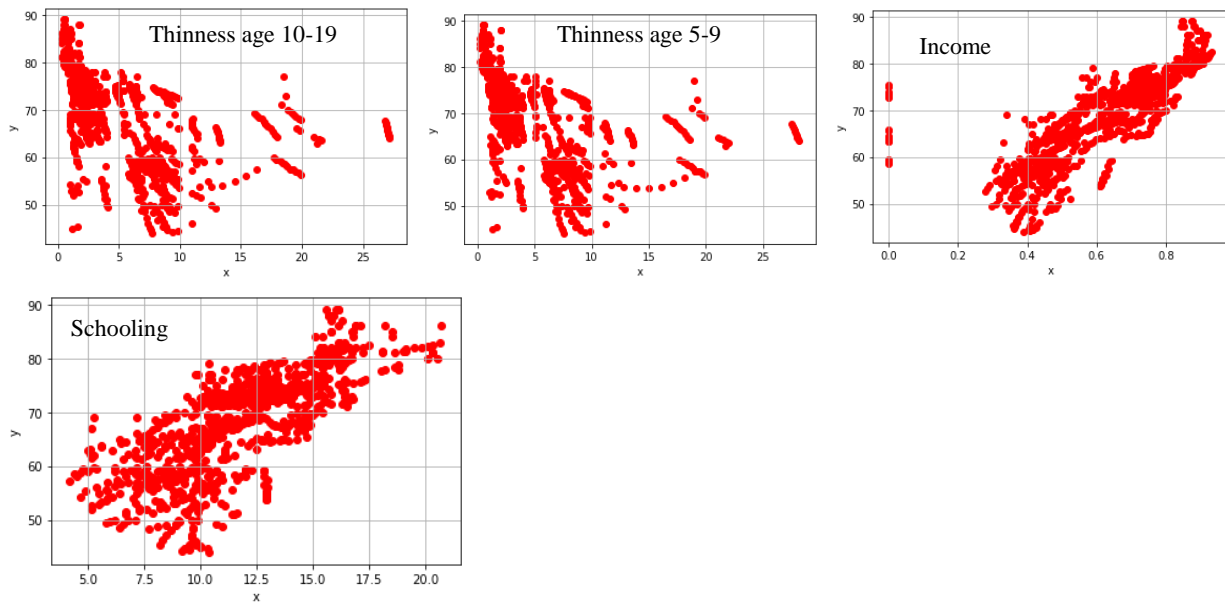
Tên hàm	def rankingFeature(distances,headers,bestIndexOrders):
Mục tiêu	Vẽ 1 bảng chứa thông tin xếp hạng của 10 đặc trưng
Tham số	<ul style="list-style-type: none"> distances (list): danh sách chứa các danh sách giá trị RMSE sau khi sử dụng phương pháp Cross Validation. headers(list): chứa các tên của các đặc trưng bestIndexOrders (list): là 1 danh sách chứa các index của các đặc trưng có chỉ số rmse từ thấp cho đến cao
Trả về	<ul style="list-style-type: none"> info (Dataframe) : Chứa các thông tin xếp hạng cần thiết của 10 đặc trưng

CÁC THƯ VIỆN ĐÃ THÊM

Tên thư viện	Lý do
import pandas as pd	Có sẵn (đọc dữ liệu)
import numpy as np	(chuyển về numpy để dễ tính toán)
import matplotlib.pyplot as plt	Thêm để vẽ các biểu đồ tượng trưng cho 10 đặc trưng → dễ dàng biết được đặc trưng nào tốt nhất
import math	Thêm để tính toán các biểu thức cơ bản

HÌNH ẢNH PHÂN BỐ CỦA TOÀN BỘ 10 ĐẶC TRƯNG





Hình ảnh toàn bộ các đặc trưng được hiện thị bằng biểu đồ

NHẬN XÉT KẾT QUẢ TỪ TOÀN BỘ CÁC MÔ HÌNH ĐƯỢC XÂY DỰNG

Câu a

Mô hình được xây dựng từ 10 đặc trưng

RMSE : 7.064046430584037

Đây có thể coi là 1 kết quả chấp nhận được do có sự kết hợp cả 10 đặc trưng. Tuy nhiên để dự đoán tuổi có độ chính xác cao cần phải xây dựng mô hình tốt hơn. Đối với mô hình này tuổi dự đoán sẽ có sai số làm tròn ± 7 . Giả sử: kiểm tra trên 1 tập dữ liệu cụ thể, mô hình cho kết quả dự đoán là 65 tuổi. Khi áp dụng mô hình này, tuổi thật sẽ nằm trong khoảng từ 58-72.

Câu b

Mô hình được xây dựng từ từng đặc trưng trong dữ liệu theo phương pháp Cross Validation

	x	Tính chất	RMSE	Xếp hạng
0	x1	Schooling	11.820071	1
1	x2	Income composition of resources	13.299791	2
2	x3	Diphtheria	16.019288	3
3	x4	Polio	17.912636	4
4	x5	BMI	27.963793	5
5	x6	Adult Mortality	46.767300	6
6	x7	Thinness age 5-9	51.775059	7
7	x8	Thinness age 10-19	51.899815	8
8	x9	GDP	60.450393	9
9	x10	HIV/AIDS	69.081327	10

Đánh giá theo chỉ số RMSE ta thấy đặc trưng Schooling có chỉ số RMSE thấp nhất trong 10 đặc trưng điều này chứng tỏ mô hình được xây dựng từ đặc trưng này cũng khá hiệu quả trong việc dự đoán (chỉ số RMSE càng thấp, mô hình học càng tốt)

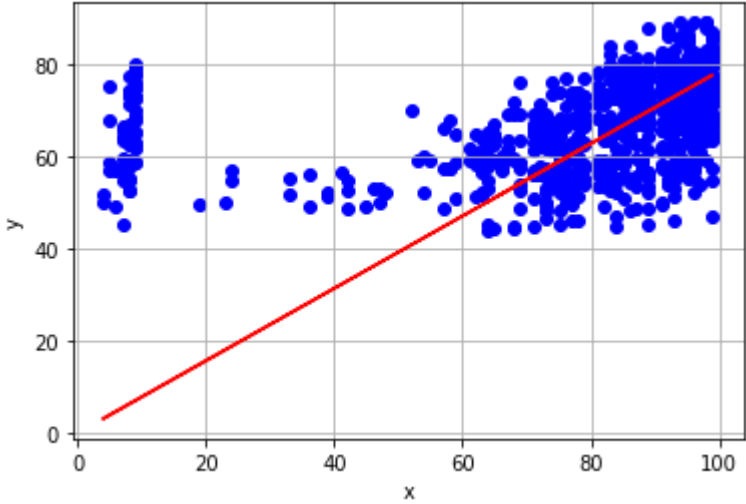
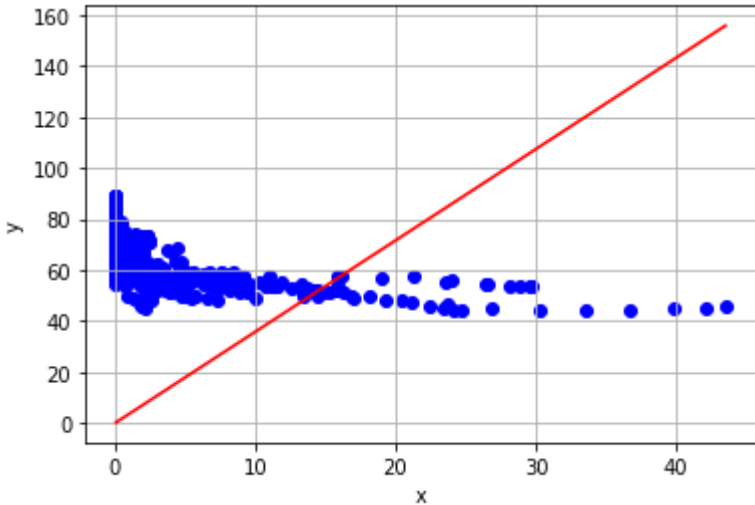
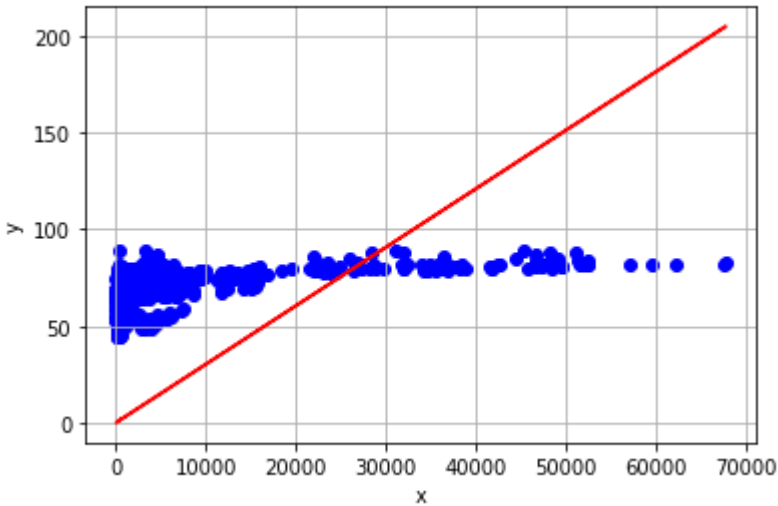
Ví dụ: Vẽ đường hồi quy tuyến tính của 10 đặc trưng tương ứng.

Đối với các đặc trưng có RMSE cao thì kết quả dự đoán có thể lệch khá nhiều so với kết quả kiểm tra nên các biểu đồ dữ liệu có thể không giống với các hình biểu đồ dữ liệu ban đầu.

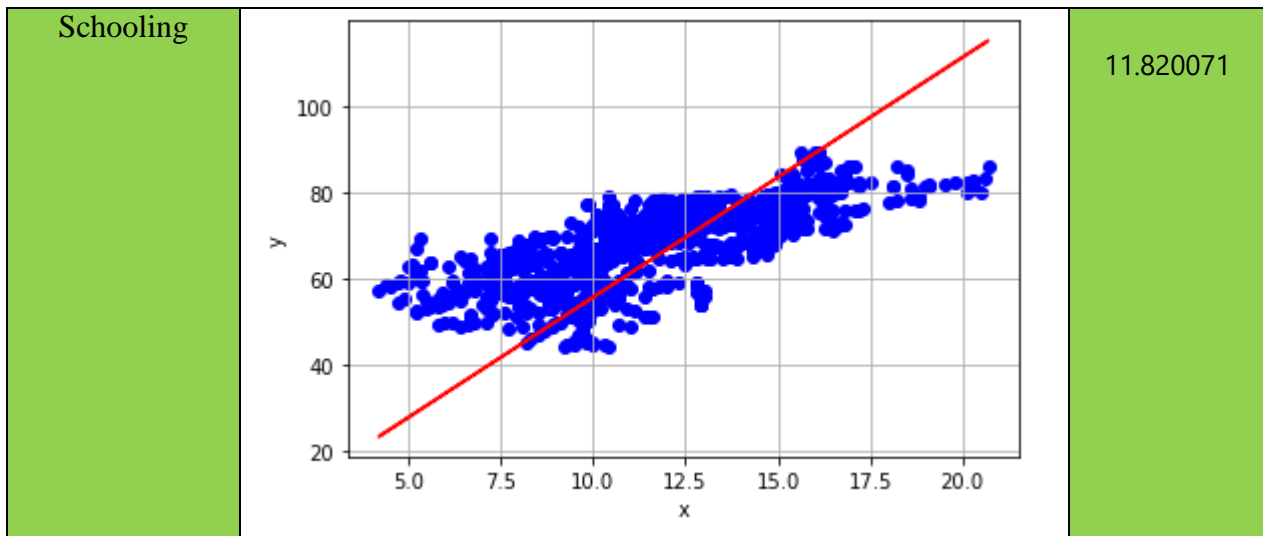
Một số quy định về màu sắc để dễ dàng đánh giá

Màu sắc	Mức độ
Xanh lá	RMSE thấp
Màu be	RMSE trung bình
Màu xám	RMSE cao

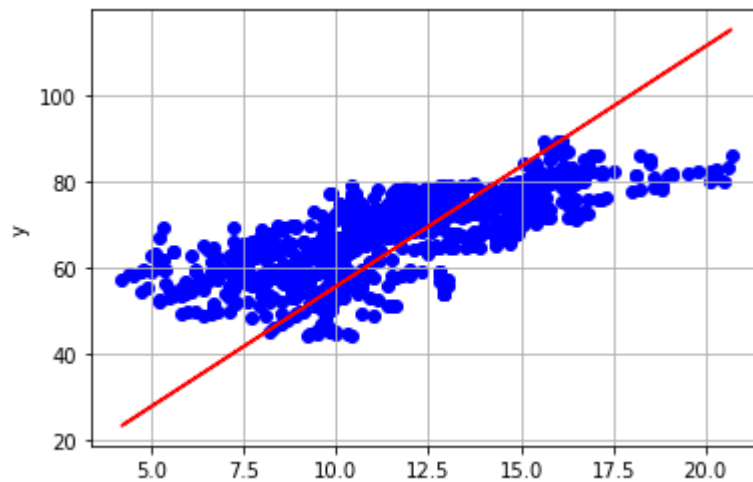
Đặc trưng	Hình ảnh	RMSE trung bình
Adult Mortality		46.767300
BMI		27.963793
Polio		17.912636

Diphtheria		16.019288
HIV/AIDS		69.081327
GDP		60.450393

Thinness age 10-19		51.899815
Thinness age 5-9		51.775059
Income composition of resources		13.299791



Schooling



Hình ảnh đường hồi quy tuyến tính (màu đỏ)

Từ hình ảnh, chúng ta có thể dễ dàng thấy bằng mắt biểu đồ cuối cùng các điểm dữ liệu tập trung gần đường hồi quy tuyến tính hơn so với các biểu đồ khác sau khi thực hiện bằng phương pháp Cross Validation .

Đặc điểm chung : các đường hồi quy tuyến tính đều đi qua gốc tọa độ (0,0). Đây là 1 trong những cơ sở để xây dựng 1 mô hình mới có chỉ số RMSE thấp hơn.

Qua các bảng xếp hạng và hình ảnh, chúng ta thấy rằng đặc trưng Schooling là đặc trưng có RMSE tốt nhất , bởi vì nếu chúng ta so sánh với các đặc trưng khác thì các điểm dữ liệu ở đặc trưng này gần với đường hồi quy tuyến tính hơn.

Báo cáo kết quả khi sử dụng đặc trưng tốt nhất (Schooling)

RMSE 10.26095039165537

Kết quả này có chỉ số RMSE cao hơn khi sử dụng mô hình kết hợp toàn bộ 10 đặc trưng. ($10.26 > 7.06$). Dễ thấy bởi khi dự đoán tuổi ta phải dựa vào nhiều cơ sở (nhiều đặc trưng) thì tỉ lệ dự đoán sẽ cao. Tuy nhiên để lựa chọn 1 mô hình chỉ sử dụng 1 đặc trưng thì ta có thể chọn ngay đặc trưng Schooling.

Câu c

Báo cáo kết quả khi xây dựng 4 mô hình tự thiết kế

- **Mô hình kết hợp 2 đặc trưng tốt nhất**

Giải thích: Sử dụng phương pháp Cross Validation để tìm ra các RMSE trung bình trên tập dữ liệu huấn luyện → Tìm ra 2 đặc trưng có chỉ số RMSE thấp nhất là đặc trưng “Schooling” và “Income composition of resources” nên quyết định xây dựng mô hình từ 2 đặc trưng này. Bởi vì 2 đặc trưng có RMSE thấp nhất sẽ dự đoán được tuổi gần đúng với kết quả kiểm tra hơn.

- **Mô hình kết hợp 3 đặc trưng và được chuẩn hóa (mũ 0.4)**

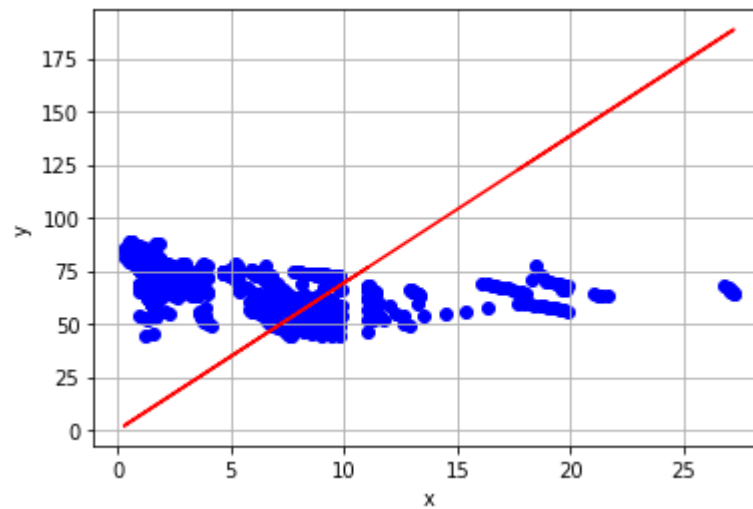
Giải thích: 3 đặc trưng tốt nhất là đặc trưng “Schooling”, “Income composition of resources” và “Diphtheria” và được chuẩn hóa bằng số mũ. Nhận thấy nếu a^k trong đó a là số lớn hơn 0 và $k \in (0,1)$ thì thực hiện sẽ trả về 1 con số luôn bé hơn hoặc bằng a . Dựa vào tính chất này, cho phát sinh ngẫu nhiên 1 số nằm trong khoảng $(0,1)$ với vòng lặp 1000 lần → tìm được số mũ bằng 0.4

- **Mô hình kết hợp đặc trưng tốt nhất + tốt nhì**

Giải thích: Xây dựng 1 đặc trưng mới là sự kết hợp của 2 đặc trưng tốt nhất cộng lại. Bởi vì kết quả dự đoán sẽ cho thấy tính gần đúng với dữ liệu kiểm tra là cao nhất.

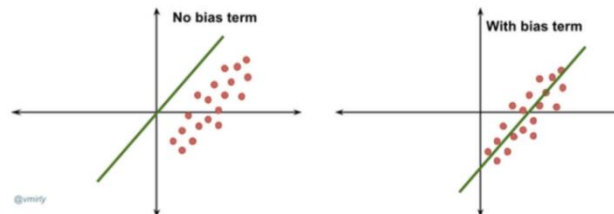
- **Mô hình sử dụng 10 đặc trưng + bias**

Giải thích: Khi sử dụng mô hình mà không có độ lệch (bias) thì đường hồi quy tuyến tính sẽ luôn đi gốc tọa độ $(0,0)$ nên bị hạn chế do có nhiều điểm dữ liệu ngoại lai nằm xa đường hồi quy tuyến tính → bị hạn chế và không linh hoạt. Ví dụ minh họa



Đặc trưng Thinness age 5-9

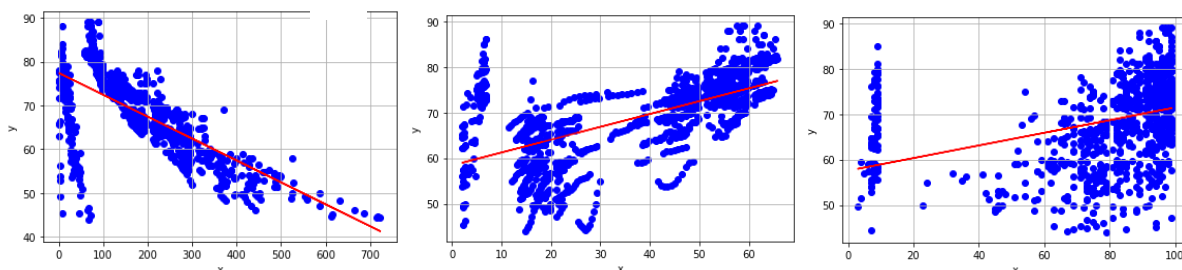
Còn nếu có bias (độ lệch), đường hồi quy tuyến tính sẽ không cần phải đi gốc tọa độ (0,0) mà có thể nằm ngang, dọc tùy thuộc vào dữ liệu và cách cài đặt [3]. Điều này có thể nói mô hình sẽ có chỉ số RMSE thấp nhất. Nên mô hình này là ưu tiên hàng đầu.

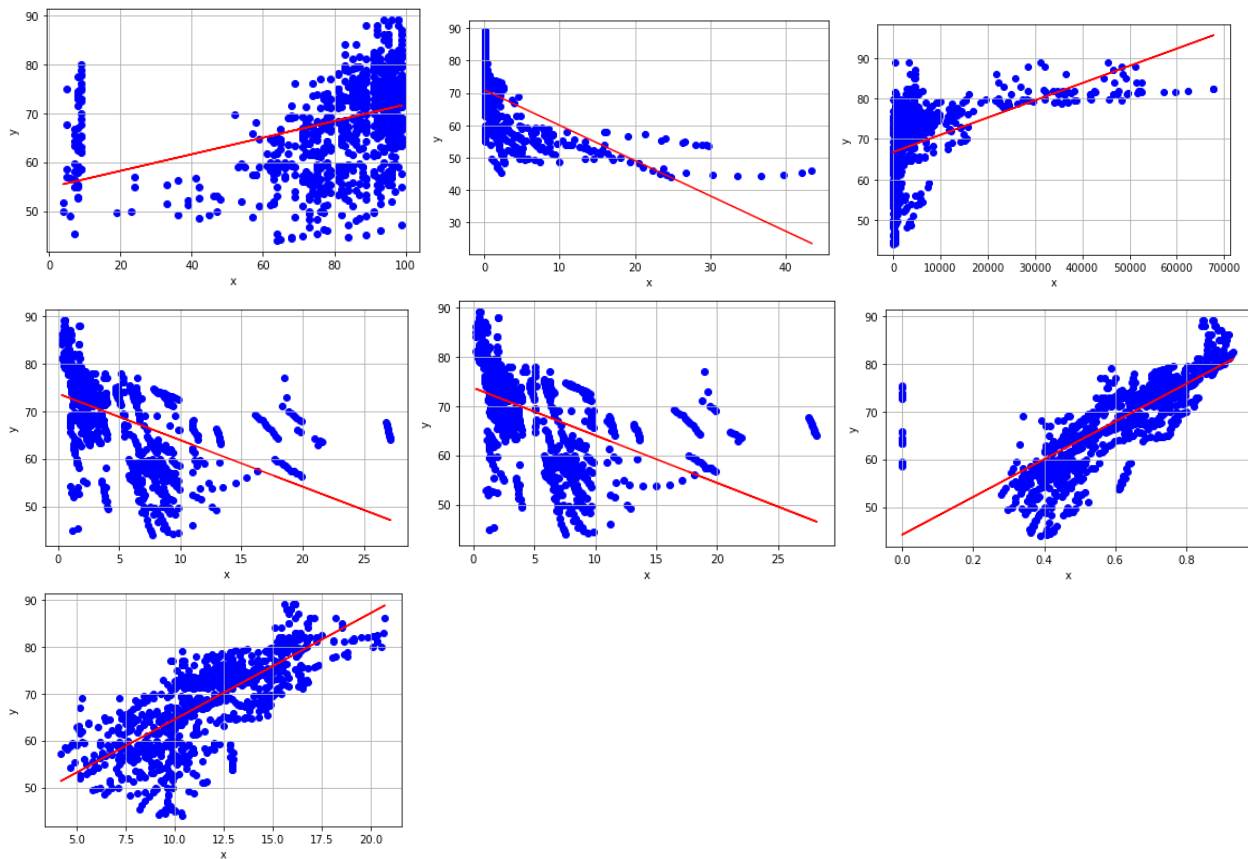


Hình ảnh minh họa [4]

Minh chứng:

Hình ảnh đường hồi quy tuyến tính có kết hợp của 10 đặc trưng + độ lệch (bias) (theo thứ tự của từng đặc trưng từ trái sang phải)





Ở mô hình có thể thấy các đường hồi quy tuyến tính khá linh hoạt, khá khớp với từng điểm dữ liệu nên do đó đây là mô hình tốt nhất trong 4 mô hình tự xây dựng.

Kết quả của 4 mô hình tự thiết kế

RMSE trong bảng dưới là RMSE trung bình của từng mô hình bằng các sử dụng phương pháp Cross Validation ở câu b

STT	Mô hình	RMSE
0	1 Sử dụng 2 đặc trưng tốt nhất	11.431486
1	2 Sử dụng đặc trưng được chuẩn hóa(mũ 0.4)	7.129564
2	3 Đặc trưng tốt nhất + tốt nhì	11.663742
3	4 Sử dụng 10 đặc trưng + bias	4.043032

TÀI LIỆU THAM KHẢO

- [0] <https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/>
- [0] <https://github.com/vaasha/Machine-learning-in-examples/blob/master/sklearn/cross-validation/Cross%20Validation.ipynb>
- [1] <https://viblo.asia/p/danh-gia-model-trong-machine-learning-RnB5pAq7KPG>
- [2] <https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/>
- [3] <https://ai.stackexchange.com/questions/23774/is-there-a-connection-between-the-bias-term-in-a-linear-regression-model-and-the>
- [4] <https://medium.com/@shivangisareen/linear-regression-least-squares-bc2ac1e6a3aa>
- [6] <https://stattrek.com/multiple-regression/regression-coefficients?tutorial=reg>