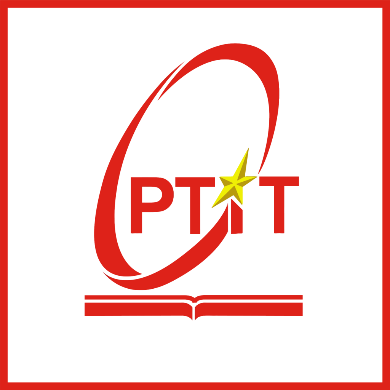
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIÊN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**Tiểu luận**

**Phát triển các hệ thông thông minh**

**Môn học:** Phát triển các hệ thống thông minh

**Giảng viên:** Trần Đình Quế

**Sinh viên:** Nguyễn Đình Đạt

**Mã SV:** B19DCCN173

Hà Nội – 2022

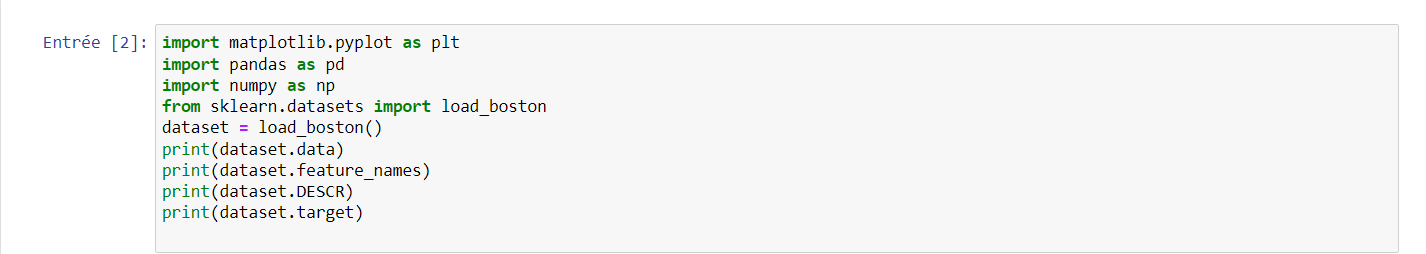
Phần 1: Machine Learning cơ bản.

1. Xử lý dữ liệu, các thư viện cơ bản.
2. Các thư viện cơ bản.
   1. Thư viện NumPy.

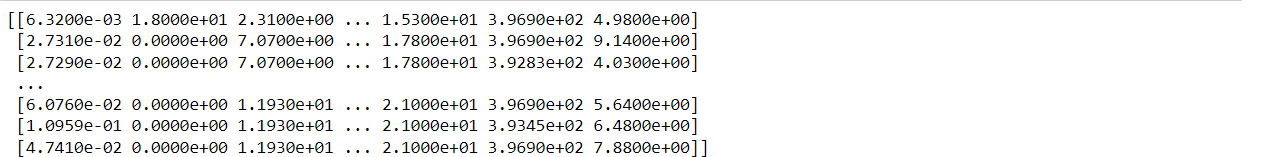
* Numpy(Numeric Python): Là một thư viện toán học phổ biến và mạnh mẽ của Python. NumPy được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng Python đơn thuần.
* Ứng dụng của NumPy:
* Các phép toán học và logic trên mảng.
* Các biến đổi Fourier và quy trình để thao tác shape.
* Các phép toán liên quan đến đại số tuyến tính. NumPy tích hợp sẵn các hàm cho đại số tuyến tính và tạo số ngẫu nhiên.
* Một số hàm trong NumPy:
* np.arange(): Tạo ra các giá trị có khoảng cách đều trong một khoảng nhất định trong Numpy. Các giá trị được sinh ra nằm trong khoảng [start,stop) (sẽ bao hàm giá trị start nhưng không chứa stop).
* np.zeros(): Tạo ra một mảng toàn số ) với kiểu dữ liệu cho trước.
* np.shape(): Hiển thị kích thước của mảng.
* np.full(): Điền toàn bộ mảng với một giá trị cho sẵn.
* np.eye(): Điền toàn bộ mảng với giá trị cho sẵ trên đường chéo và còn lại là số 0.
* np.random(row,cl): Điền mảng với kích thước cho trước với các giá trị ngẫu nhiên trong khoảng(0,1).
* np.array(): Khởi tạo một mảng.
* np.add(): Cộng hai mảng với nhau
* np.asmatrix(): Tạo một bản sao với đầu vào là một ma trận.
* np.sort(): Trả về một bản sao đã được sắp xếp.
* np.argsort(): Sắp xếp theo cột được chọn.
  1. Thư viện Pandas.
* Pandas là một thư viện Python cung cấp các cấu trúc dữ liệu nhanh, mạnh mẽ, linh hoạt. Pandas được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc (dạng bảng, đa chiều, không đồng nhất) và dữ liệu chuỗi thời gian.
* Cách sử dụng của Pandas.
* Xử lý dữ liệu mất mát, được biểu thị dưới dạng NaN, trong dữ liệu dấu phẩy động cũng như dấu phẩy tĩnh theo ý người dung mong muốn: bỏ qua hoặc chuyển sang 0.
* Khả năng thay đổi kích thước: các cột có thể được chèn và xóa khỏi DataFarme và các đối tượng chiều cao hơn.
* Căn chỉnh dữ liệu tư động và rõ rang: Các đối tượng có thể được căn chỉnh rõ rang với nột bộ nhãn hoặc người dung chỉ cần bỏ qua các nhãn và Series, DataFarme,…
* Tải dữ liệu từ các tệp phẳng (flat file) như CSV và delimited, tệp Excel, cơ sở dữ liệu và lưu/ tải dữ liệu từ định dạng HDF5 cực nhanh.
* Gộp (merging) và nối (joining) các tập dữ liệu trực quan.
* Một số hàm trong Pandas.
* pd.Series(): Khởi tạo mảng 1 chiều với nhãn và dữ liệu
* pd.date\_range(): Tạo ra một mảng bao gồm các ngày nằm trong khoảng ngày bắt đầu và ngày kết thúc với khoảng cách thời gian nhất định.
* pd.DataFrame(): Tạo ra một mảng 2 chiều có thể thay đổi kích thước, có khả năng không đồng nhất.
* pd.read\_csv(): Đọc file dữ liệu CSV
* pd.describe(): Tạo số liệu thống kê.
* .head():Lấy số lượng dòng đầu tiên (mặc định là 5 dòng).
* .tail(): Lấy số lượng dòng cuối cùng (mặc định là 5 dòng).
* .sort\_index():Sắp xếp theo nhãn(sắp xếp thứ tự các cột)
* .sort\_values(): Sắp xếp các giá trị dọc theo từng cột(Sắp xếp thứ tự trong hàng).
* .add(): Áp dụng một hàm học theo cột.
* .drop(): Loại bỏ dữ liệu trong DataFrame.
* pd.crosstab(): Thống kê số lượng theo bảng chéo.
  1. Thư viện matplotlib
* Matplotlib là một thư viện toàn diện để tạo các hình ảnh trực quan tĩnh, hoạt hình và tương tác bằng Python.
* Ứng dụng của matplotlib
* Vẽ biểu đồ.
* Vẽ đồ thị
* Một số hàm trong matplotlib
* plt.plot(): Vẽ đồ thị y so với x với các đường kẻ và điểm.
* plt.title(): Đặt tiêu đề cho một đồ thị.
* plt..xlabel(): Đặt tên cho trục x.
* plt.ylabel(): Đặt tên cho trục y.
* style: sử dụng để chọn kiểu đồ thị được hiển thị.
* plt.bar(): Được sử dụng để vẽ biểu đồ cột.
* plt.legend(): Đặt chú giải.
* plt.grid(): Hiển thị các đường lưới.
* plt.show(): Hiển thị tất cả các thứ đã được đặt.
* plt.xticks(): Sắp xếp hiển thị của trục x.
* plt.pie(): Được sử dụng để vẽ biểu đồ hình tròn.
* plt.axis(): Thiết lập thuộc tính.
* plt.savefig(): Lưu số liệu.
  1. Thư viện Scikit-learn.
* Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling.
* Ứng dụng của Scikit-learn:
* Tiền xử lý giảm chiều dữ liệu.
* Phân loại dữ liệu.
* Phân cụm dữ liệu.
* Một số hàm phổ biến của Scikit-learn:
* Scikit-learn datasets:
  + datasets.load\_iris(): Tải vè trả về tập dữ liệu thông nhất( Tự phân loại).
  + .DESCR: lấy mô tả về tập dữ liệu
  + .data: lấy thông số của tập dữ liệu.
  + .feature\_names: lấy tên của tập dữ liệu.
  + .target: phân loại
  + .target\_names: các tên nhãn của các dữ liệu đã được phân loại.

1. Học có giám sát
2. Linear Regression(Hồi quy tuyến tính)

* "Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. 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. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...
* Code ví dụ:
* Lấy tập dữ liệu Boston trực tiếp từ thư viện StatLib và ta có thể tải thông qua sklearn.datasets. Sau đây là một số thông tin của tập dữ liệu Boston.



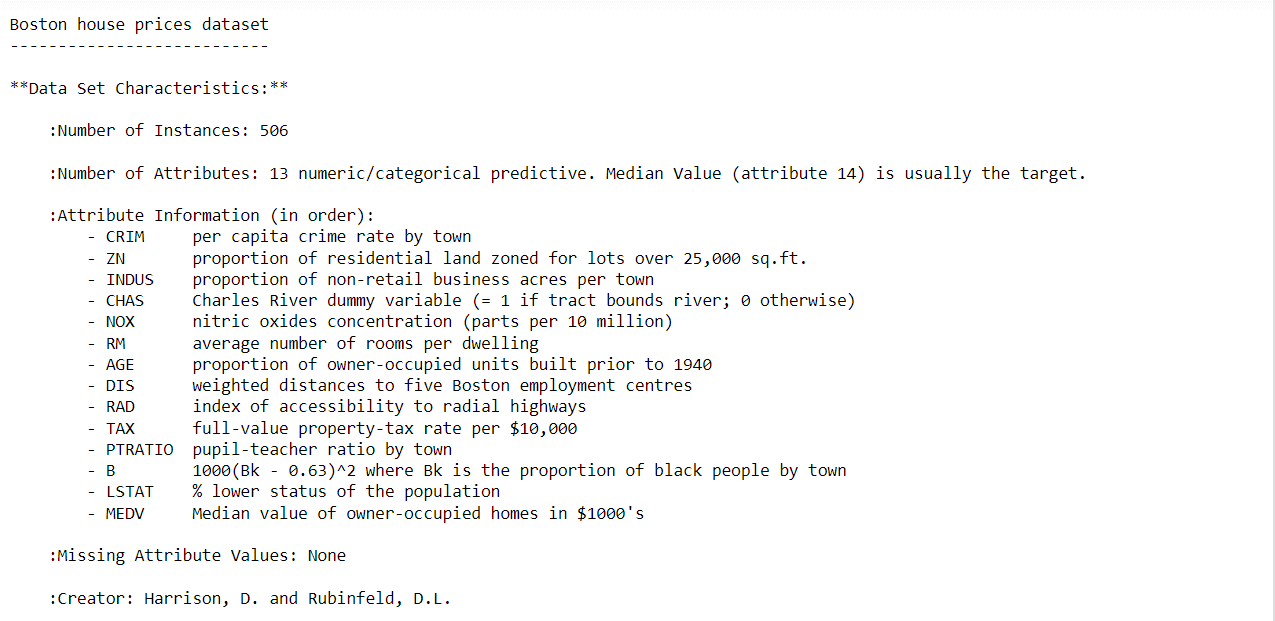
Dữ liệu:



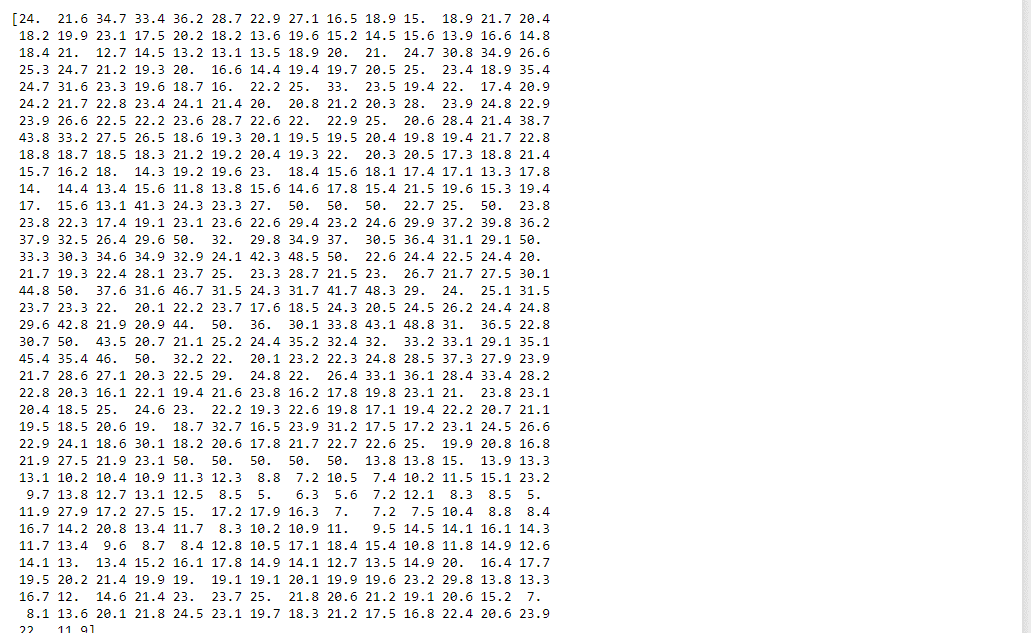
Tên:



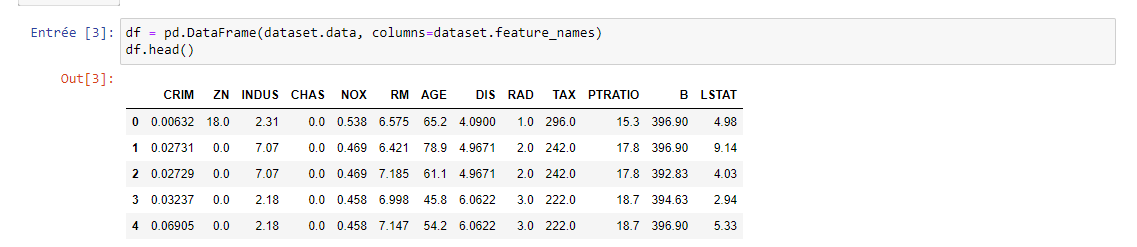
Mô tả chi tiết:



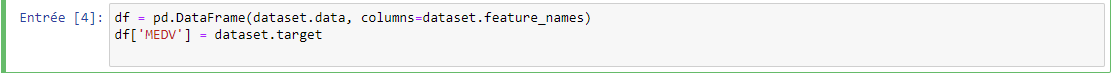
Tập dữ liệu đích.



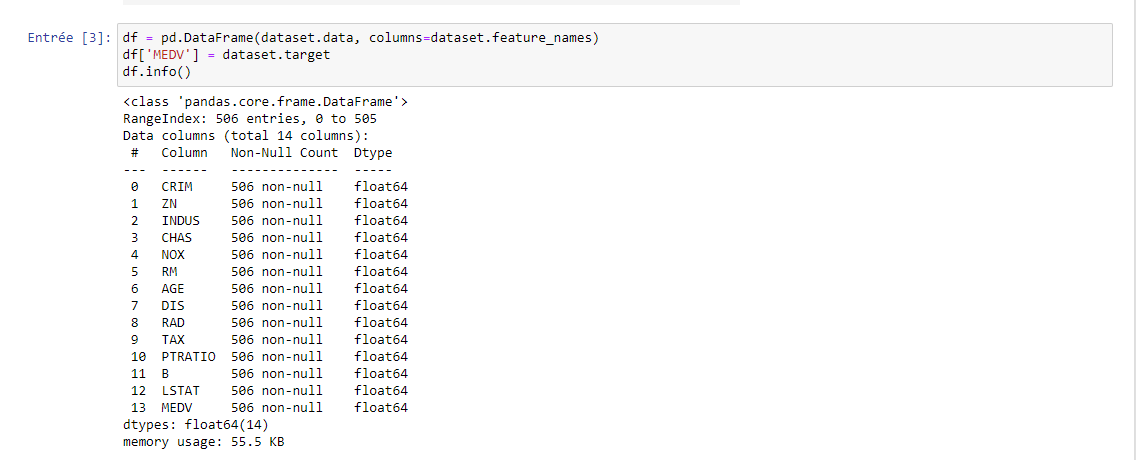
* Load tập dữ liệu vào datafarme



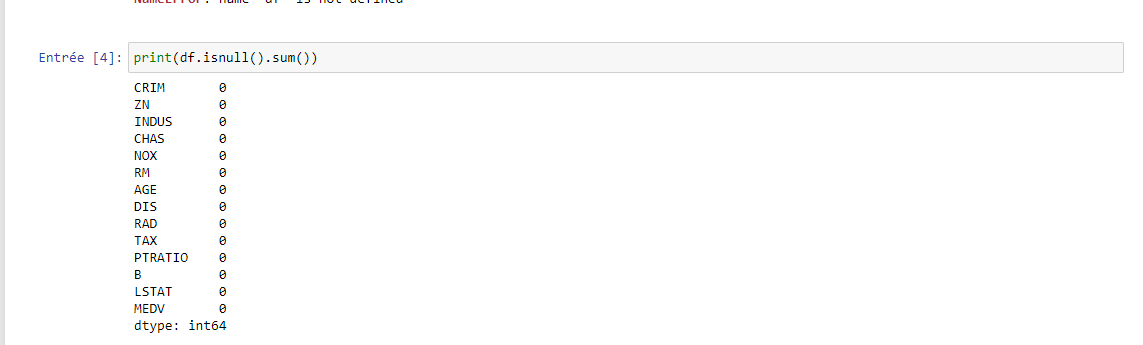
* Thêm cột giá của ngôi nhà( dữ liệu mục tiêu) vào trong bảng dữ liệu:



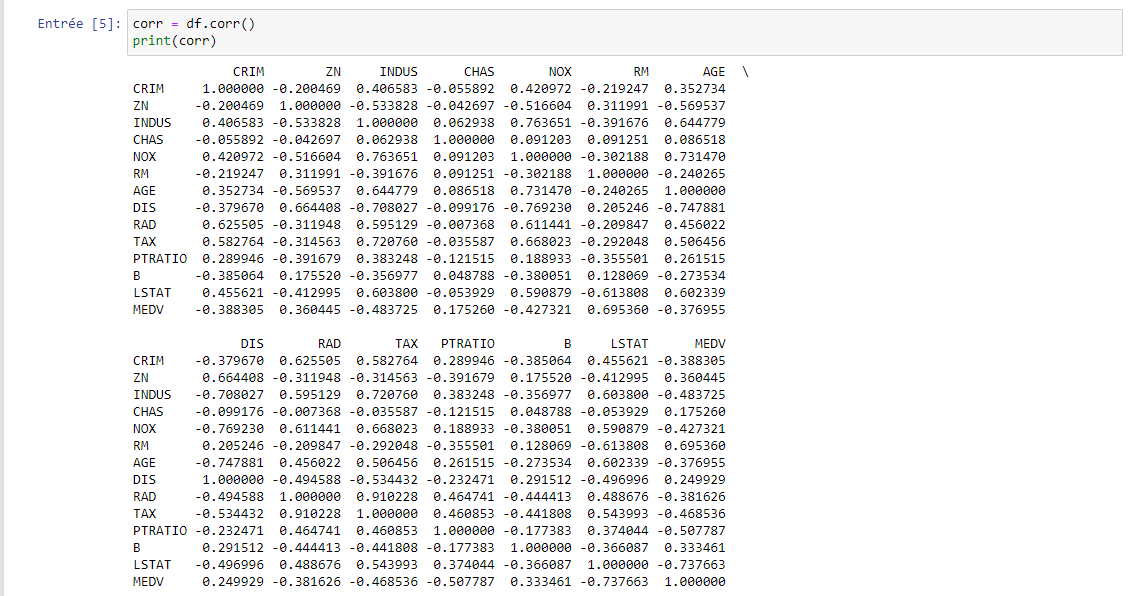
* Thông tin dữ liệu của các cột.



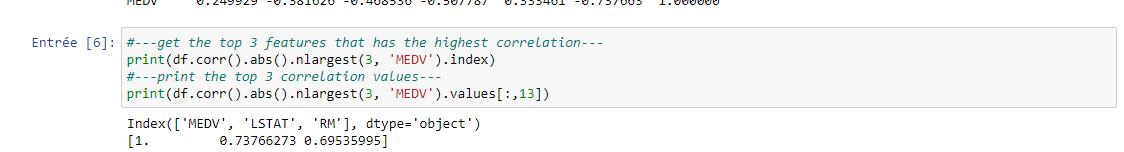
* Kiểm tra xem có cột nào bị thiếu dữ liệu không



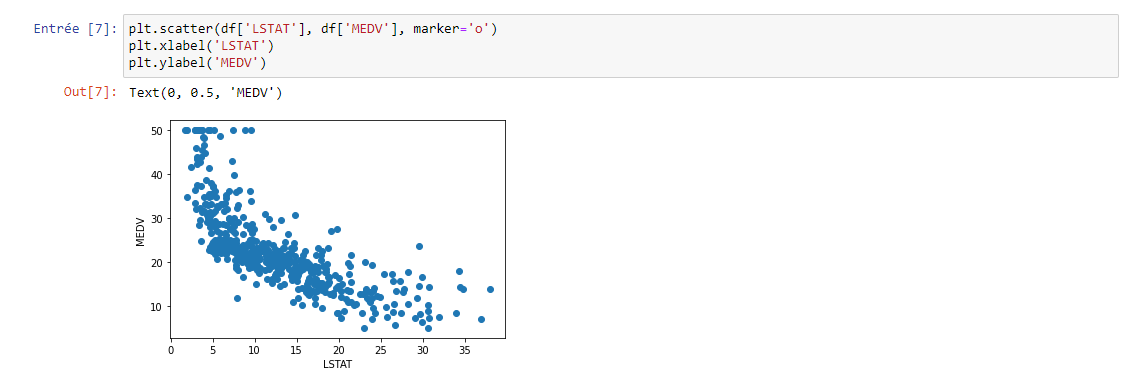
* Tính toán sự tương quan của các cột.



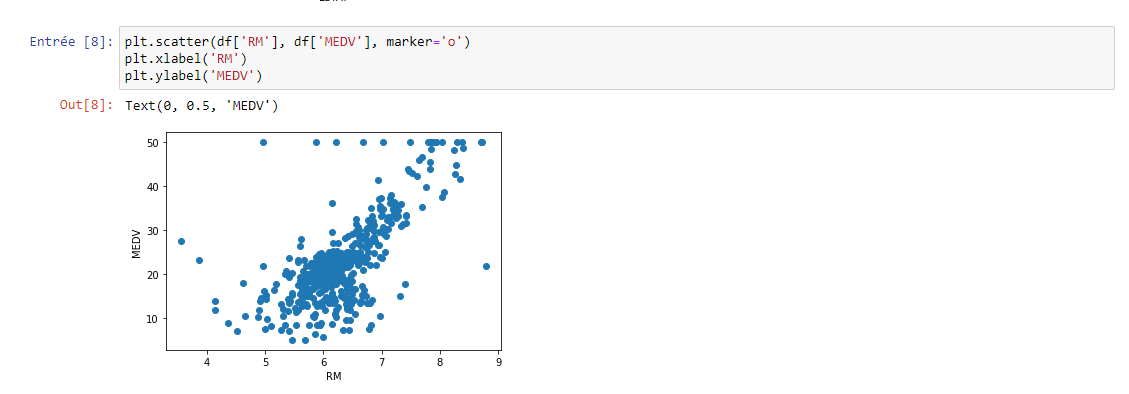
* Lấy giá trị và tên của các cột có giá trị tương quan lớn nhất.



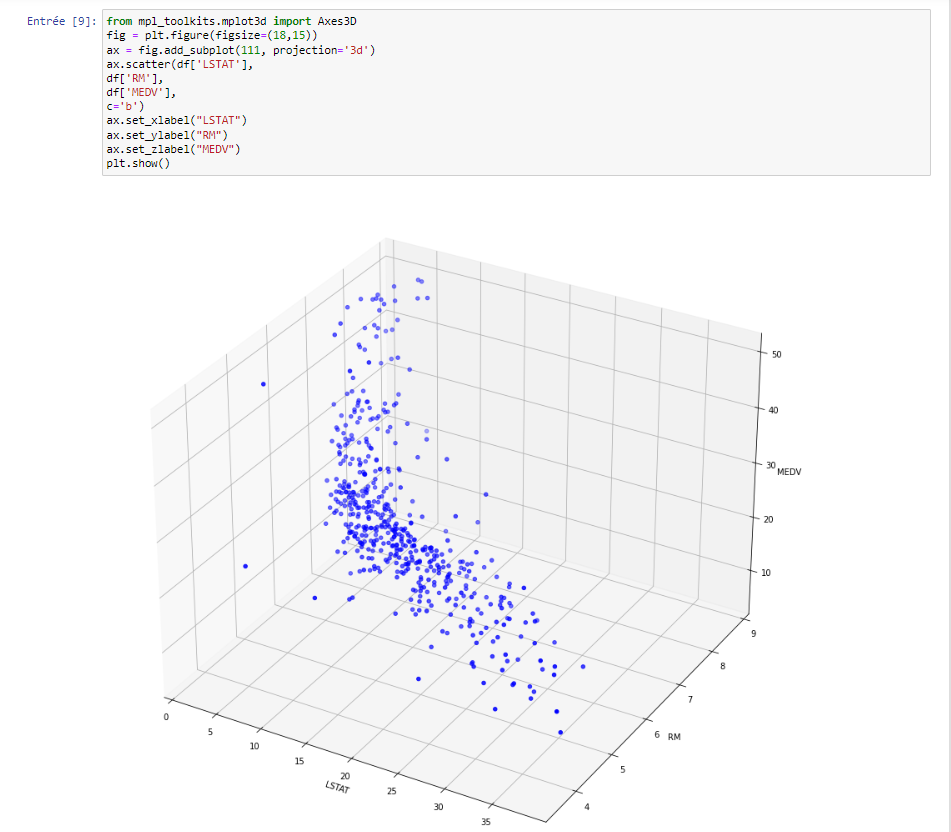
* Vẽ biểu đồi mối quan hệ của LSAT và MEDV

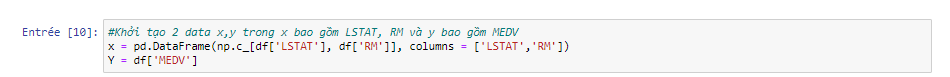
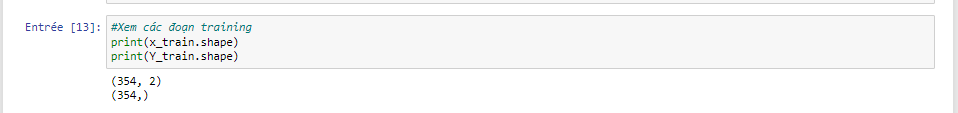
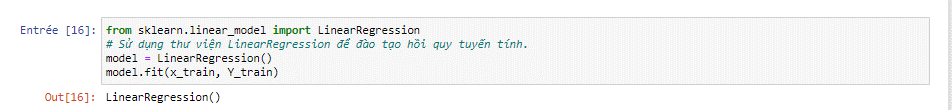
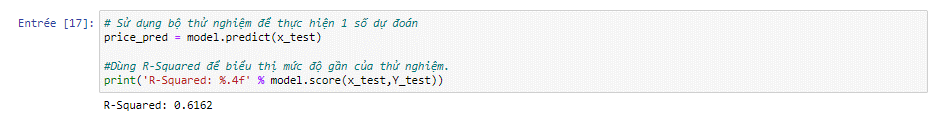
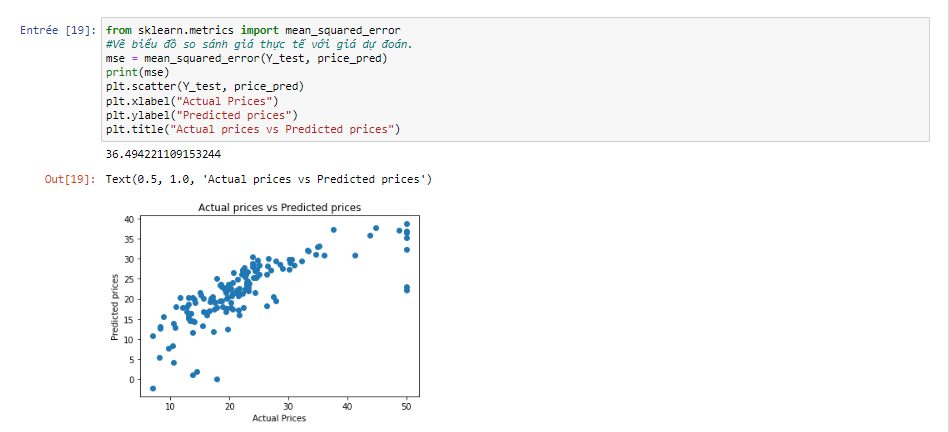
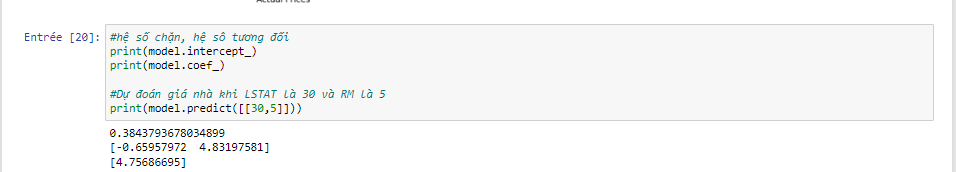


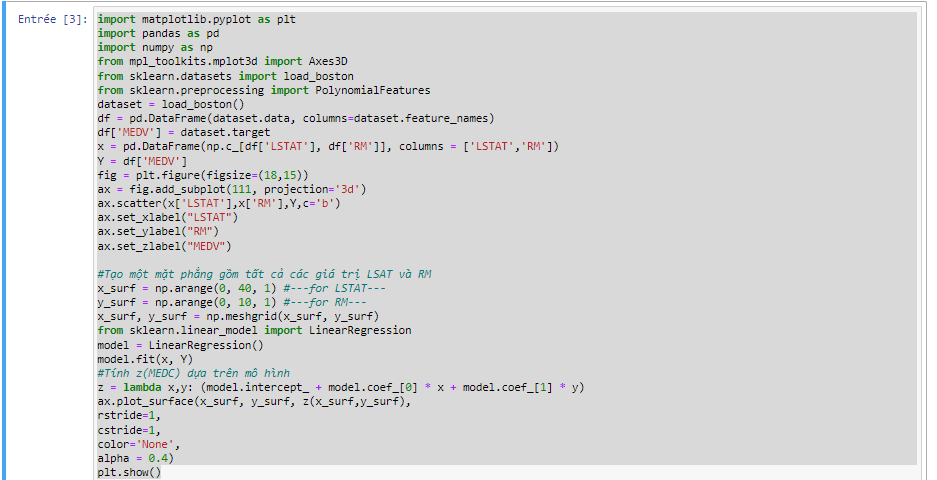
* Vẽ đồ thị của mối quan hệ của RM và MEDV.

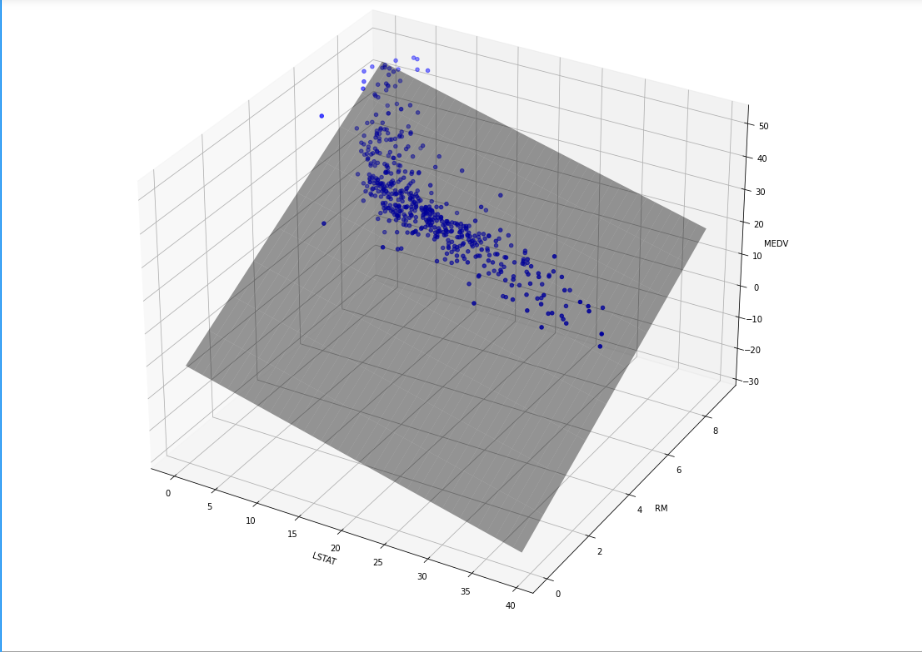


* Vẽ đồ thị 3 chiều của RM, MEDV, LSAT.

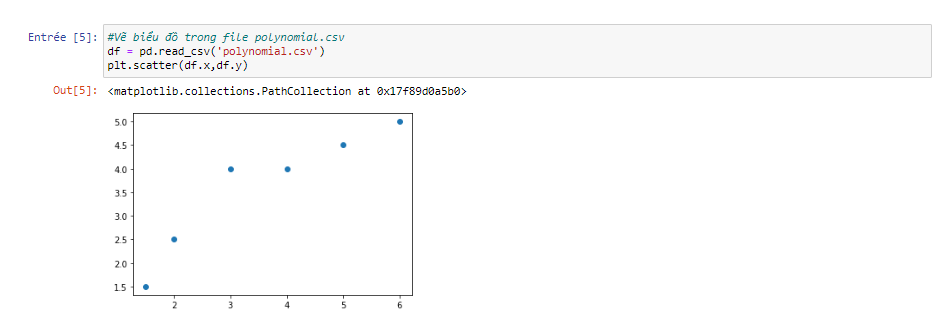


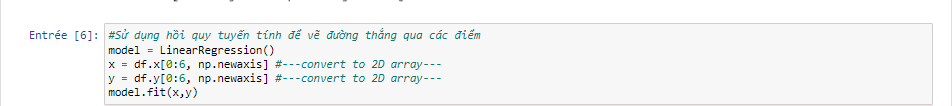
* 
* 
* 
* 
* 
* 
* 
* 
* Vẽ một mặt phẳng 3D biểu thị các dự đoán đựa trên hệ số chặn và hệ số của mô hình.

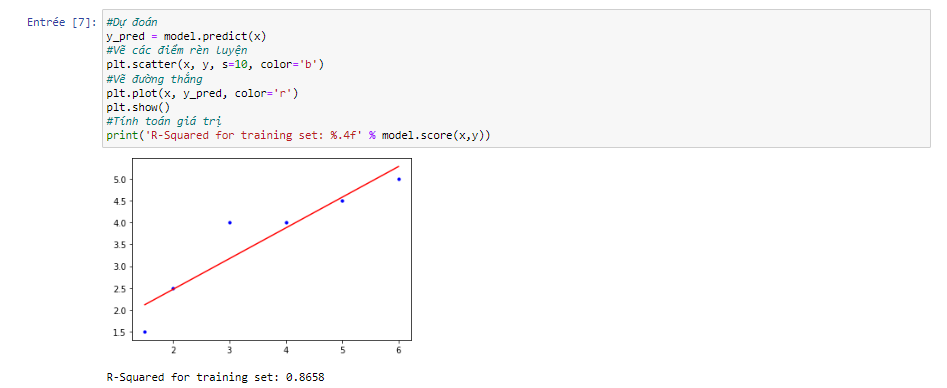


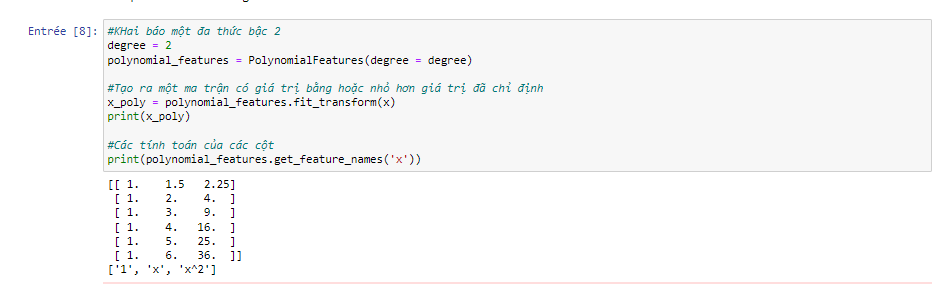


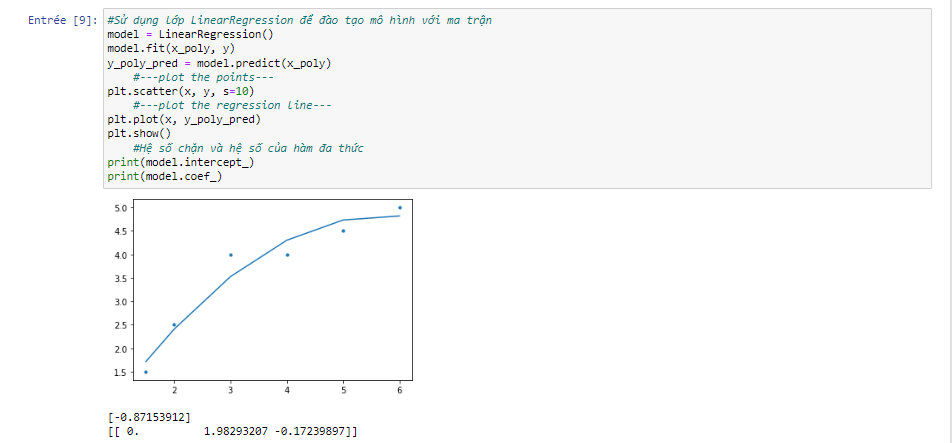
* Hồi quy đa thức (Đôi lúc 1 đường cong sẽ làm việc tốt hơn 1 đường thẳng). Ở đây chúng ta sẽ xét trên tập dữ liệu có tên là polynomial.csv gồm 2 giá trị x,y.

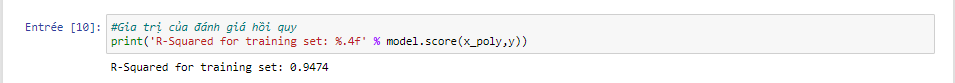




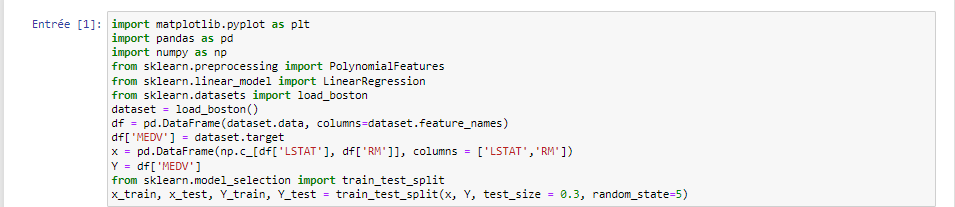


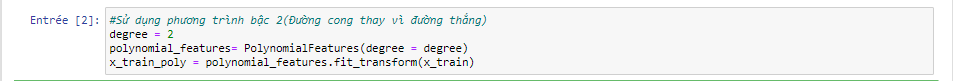


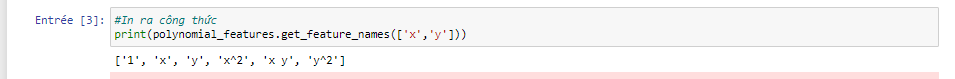


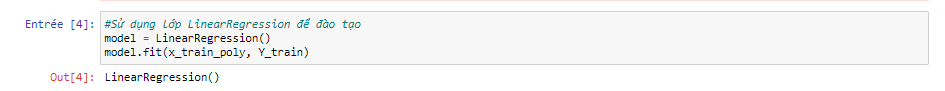


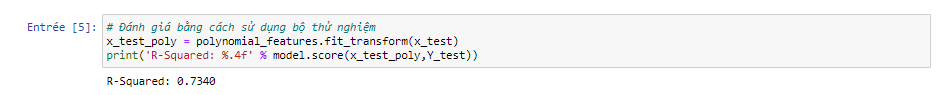
* Bây giờ chúng ta sẽ áp dụng đường cong vào việc đánh giá và dự đoán giá nhà xem có khả thi hơn việc đánh giá theo một đường thẳng hay không.

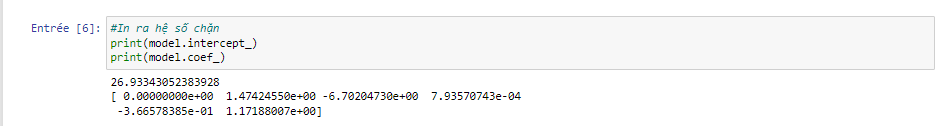












1. Phân loại sử dụng hồi quy logistic.

Phân tích hồi qui logistic là một kỹ thuật thống kê để xem xét mối liên hệ giữa

biến độc lập (biến số hoặc biến phân loại) với biến phụ thuộc là biến nhị phân.

Trong hồi qui tuyến tính đơn, biến độc lập x và phụ thuộc y là biến số liên tục liên hệ

qua phương trình:

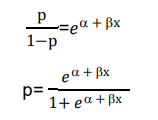


Trong hồi qui logistic, biến phụ thuộc y chỉ có 2 trạng thái 1 ( ví dụ tử vong) và 0 (ví dụ sống). Muốn đổi ra biến số liên tục người ta tính xác suất của 2 trạng thái này.

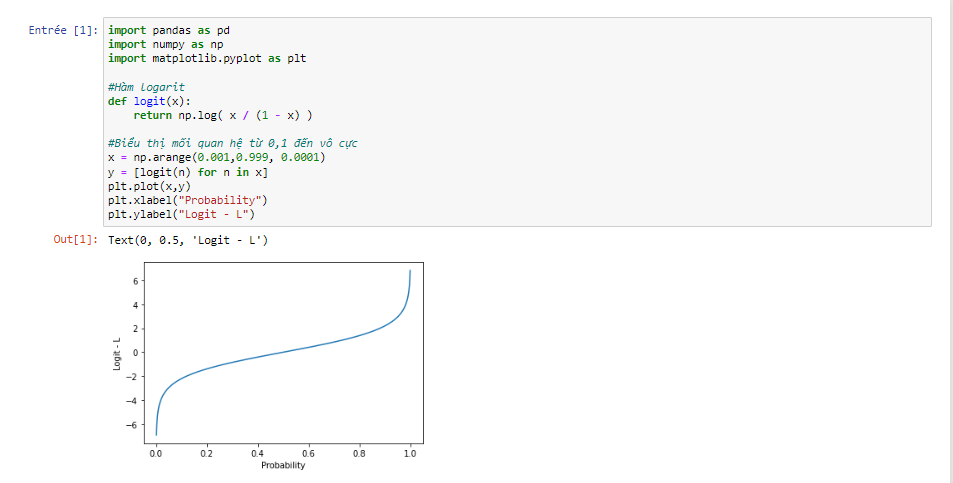
Nếu gọi p là xác suất để một biến cố xảy ra (ví dụ: tử vong), thì 1-p là xác suất để biến cố không xảy ra (ví dụ: sống). Phương trình hồi qui logistic phát biểu:



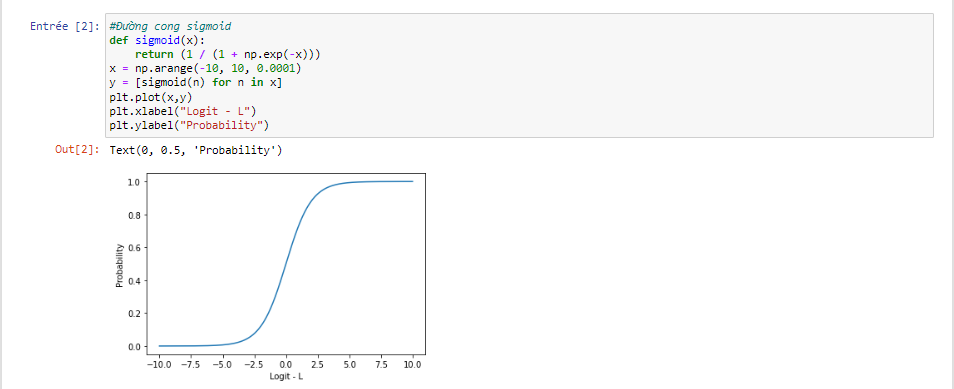
Từ phương trình này, ta có thể tính xác suất tiên đoán tử vong theo trị số của x.



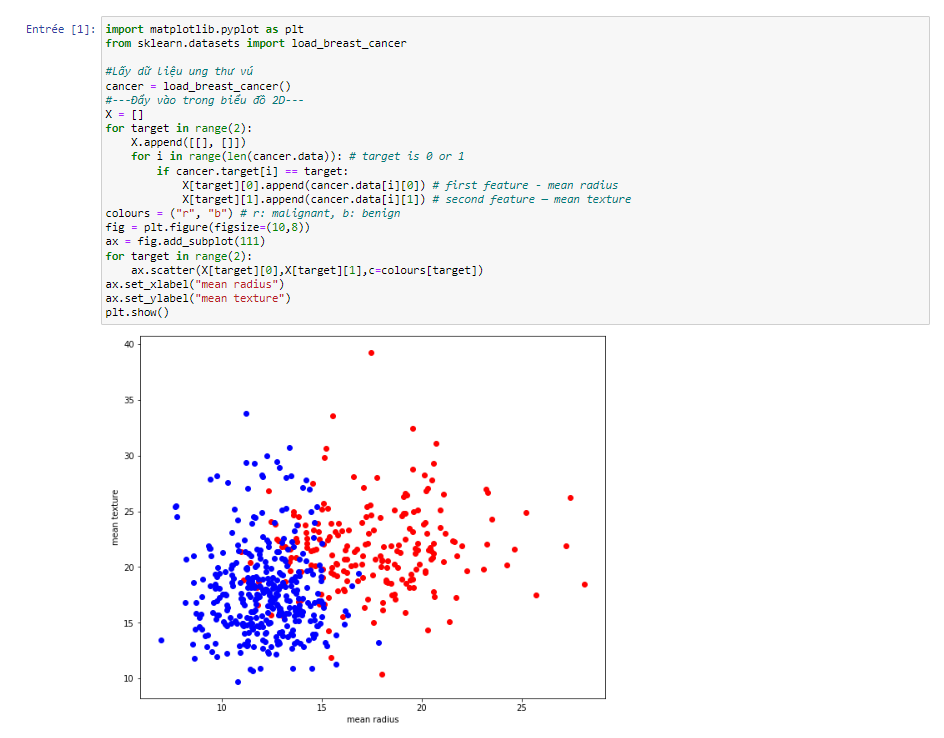
* Code ví dụ cụ thể
* Tạo hàm logarit



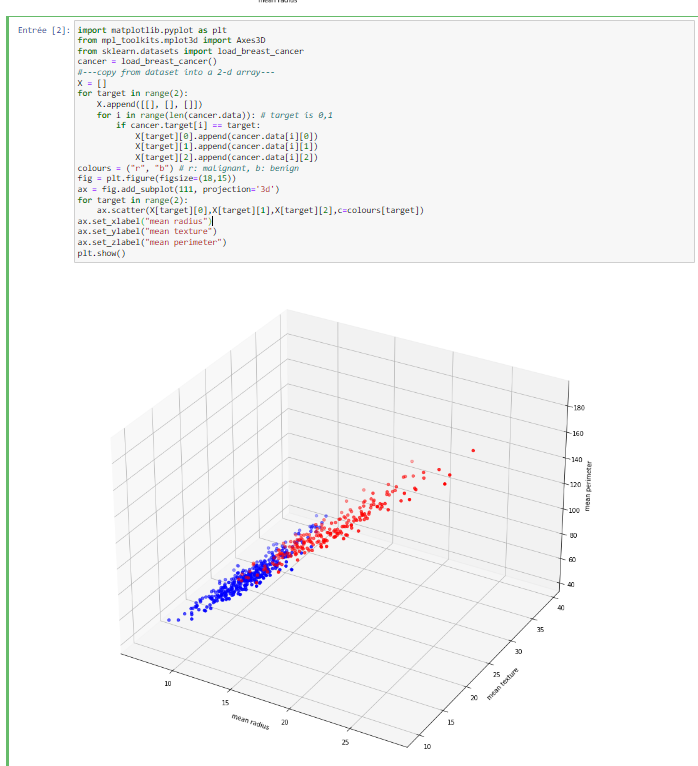
* Tạo hàm đường cong sigmoid



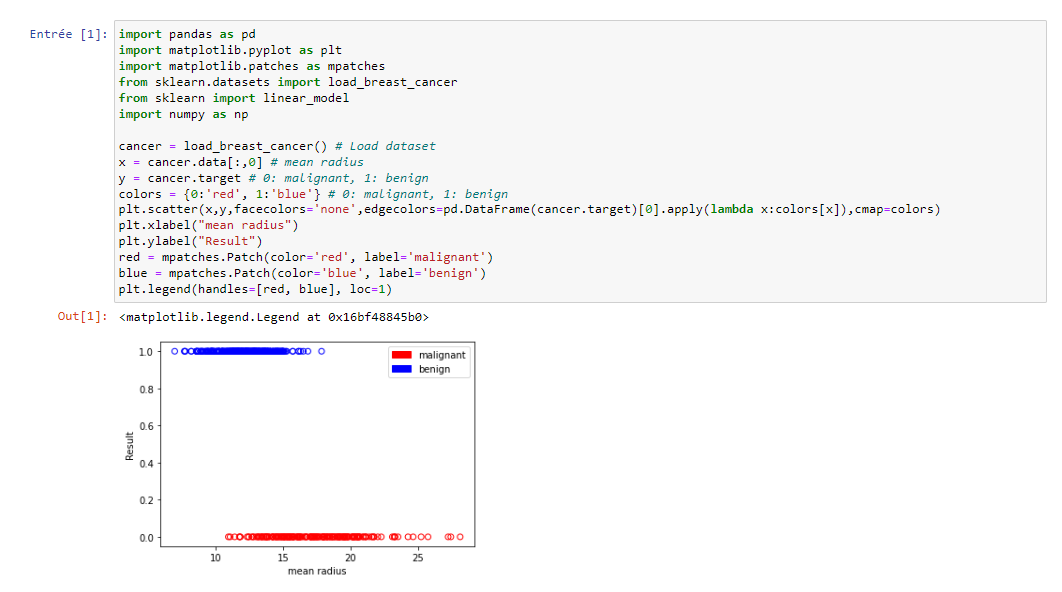
* Vẽ đồ thị của tập dữ liệu ung thư vú với 2 dữ liệu là bán kính trung bình và xác suất bị ung thư.

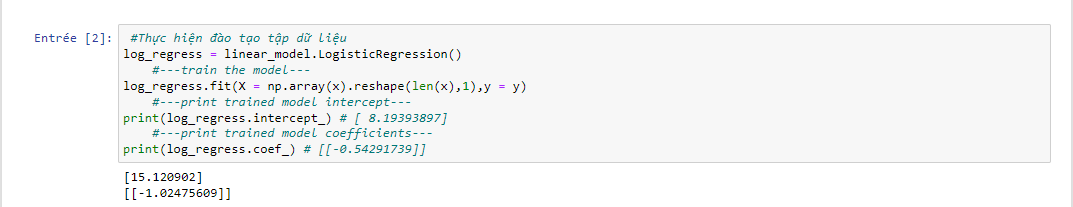


* Vẽ đồ thị của tập dữ liệu ung thư vú với 3 dữ liệu là bán kính trung bình, chu vi trung bình và xác suất bị ung thư.

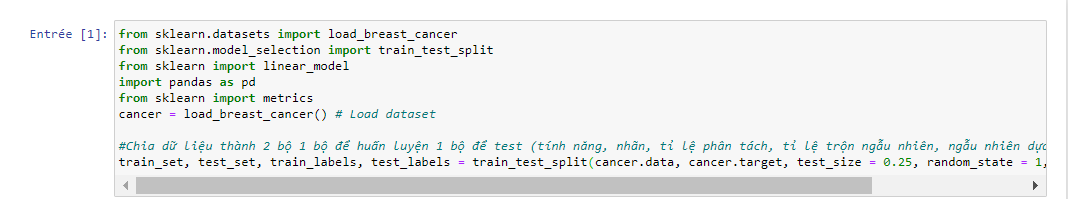


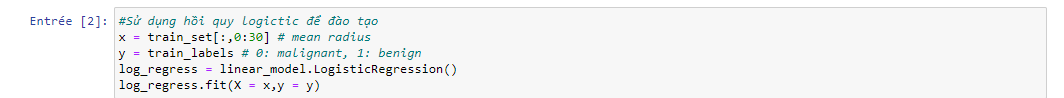
* Sử dụng hồi quy Logistic để đào tạo tập dữ liệu và dự đoán xem đó có phải ung thư vú hay không.

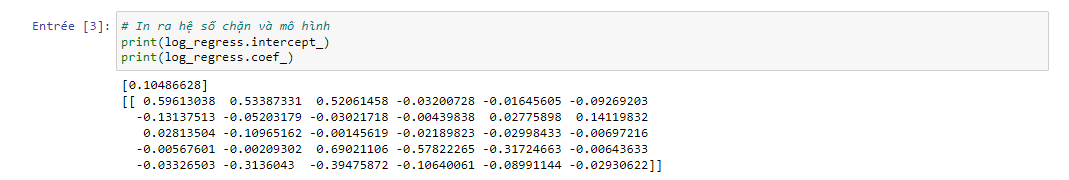


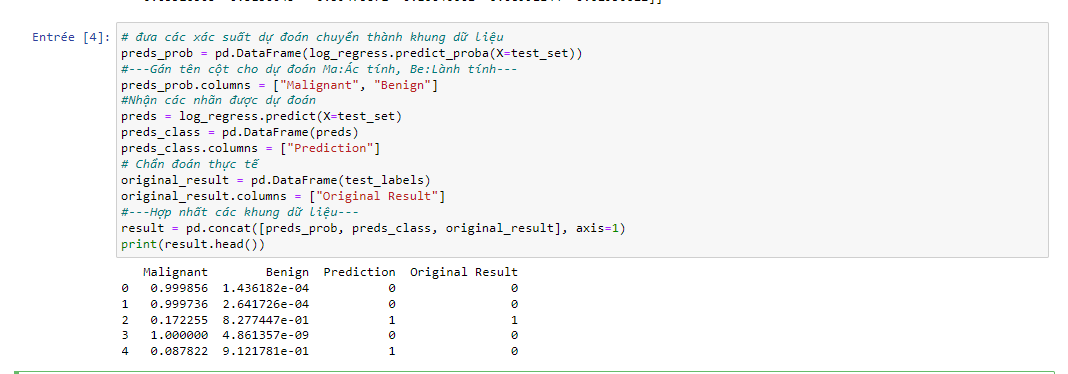


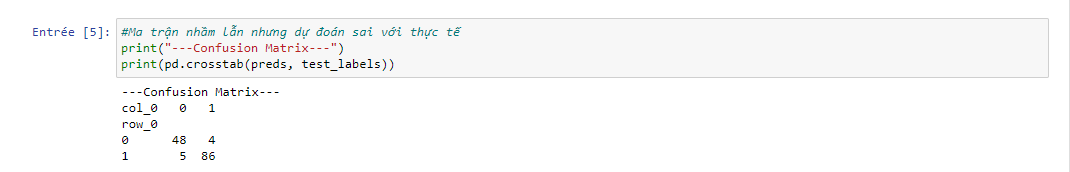
* Đào tạo mô hình sử dụng tất cả các tính năng.

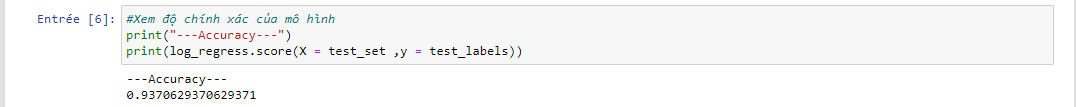


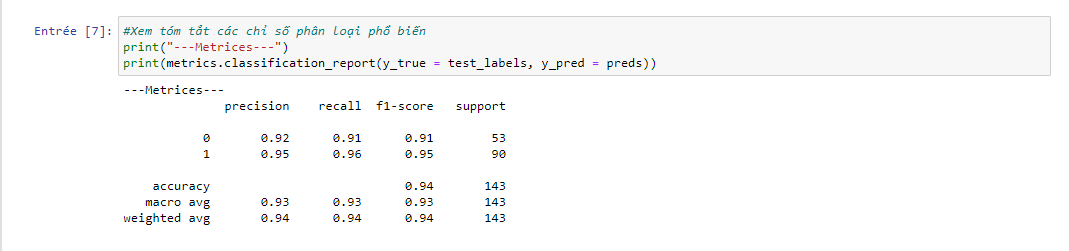












1. Phân loại sử dụng máy vector hỗ trợ.

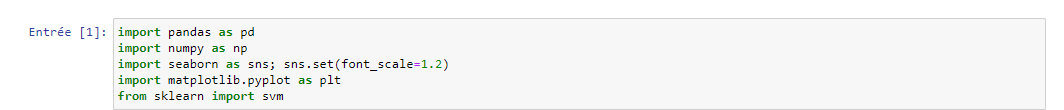
Máy vector hỗ trợ trong tiếng Anh là Support Vector Machine, viết tắt: SVM.

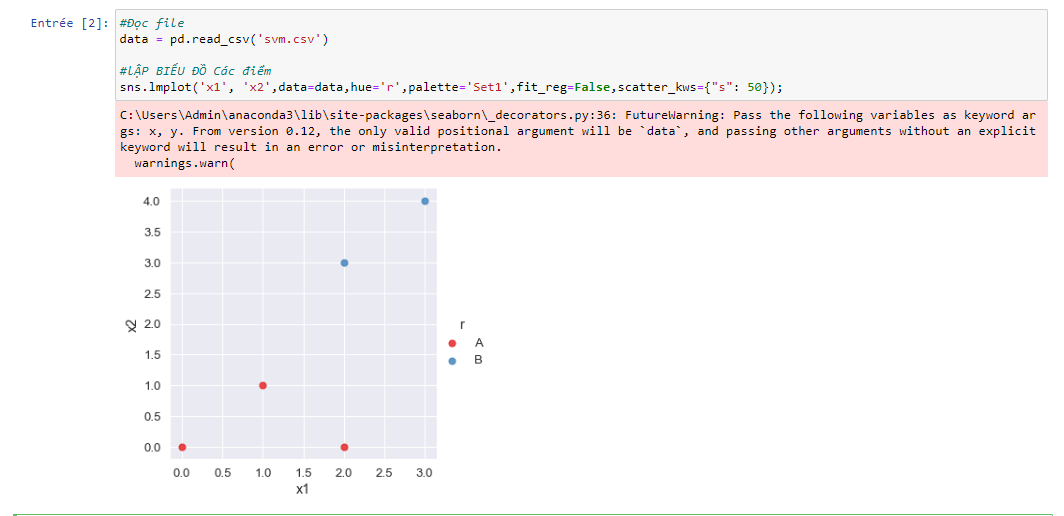
Máy vector hỗ trợ là một thuật toán giúp tìm ra một siêu phẳng phân cách tối ưu để có thể phân chia dữ liệu tuyến tính ra làm hai lớp khác nhau.

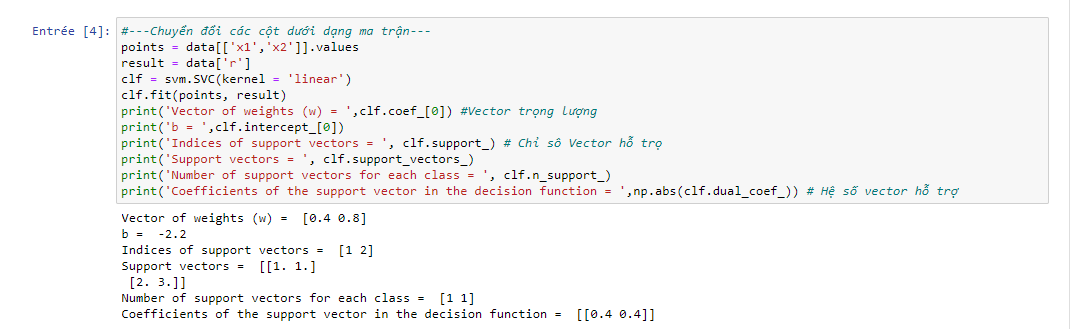
Máy vectơ hỗ trợ (SVM) là một trong những thuật toán phổ biến nhất trong học máy, được sử dụng để phân loại, hồi qui và phát hiện điểm dữ liệu bất thường.

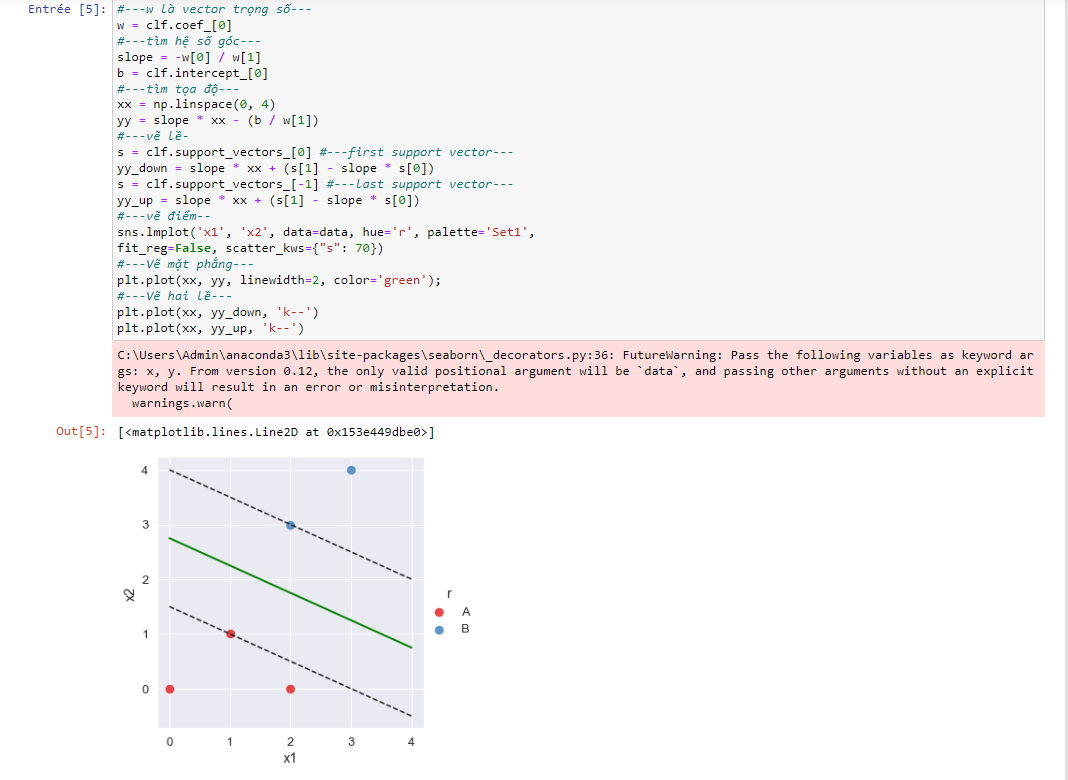
* Code ví dụ

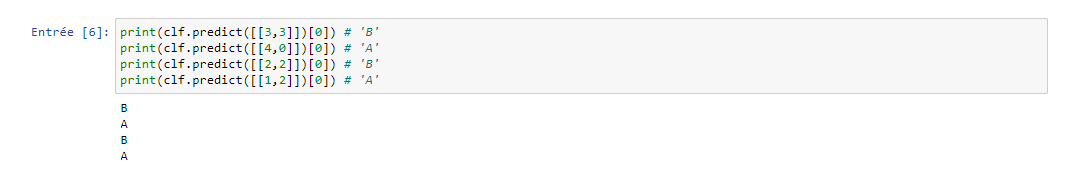
Nhập các thư viện cần thiết.



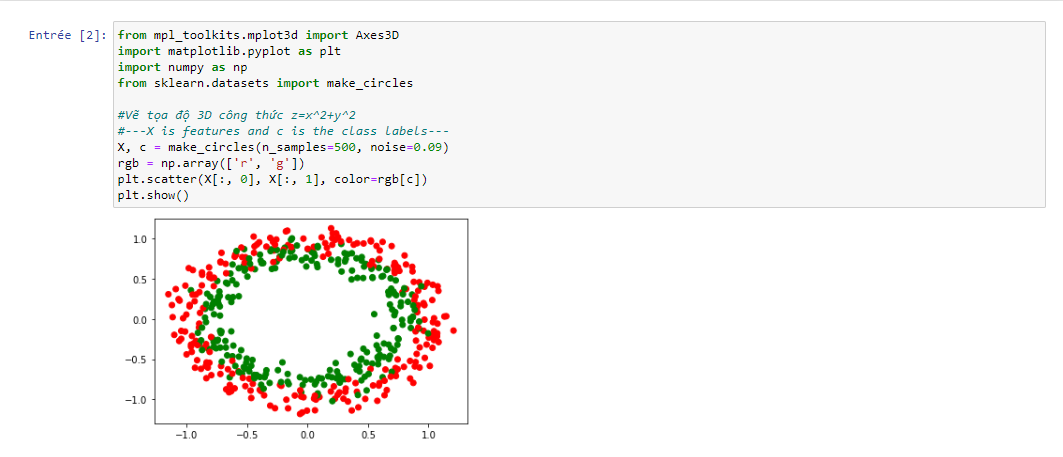


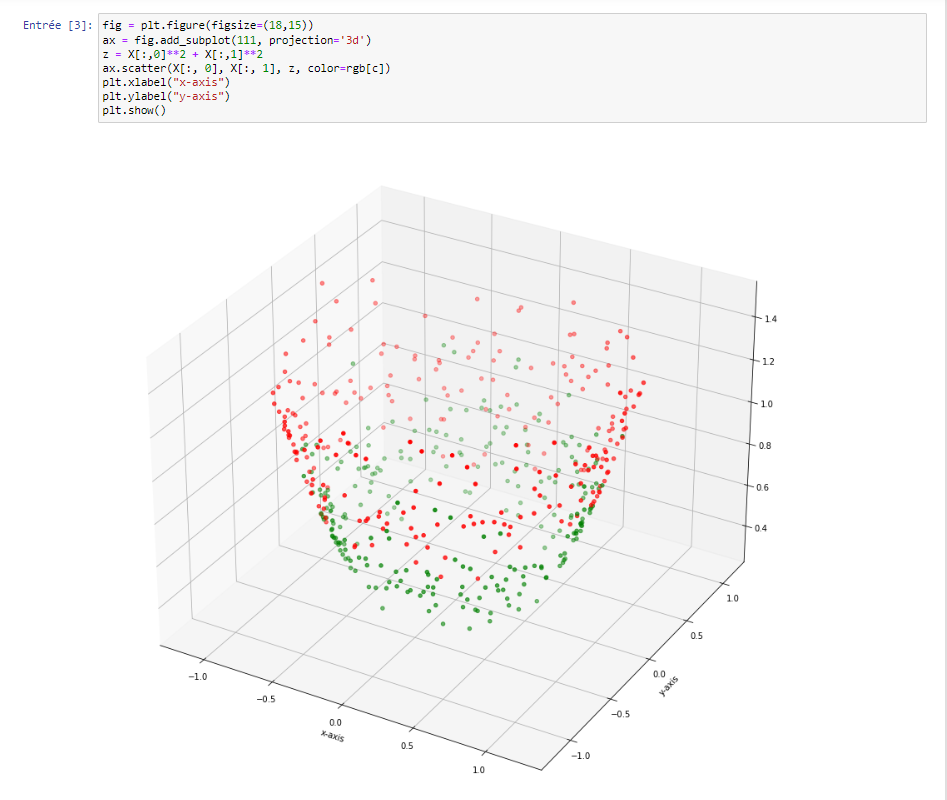


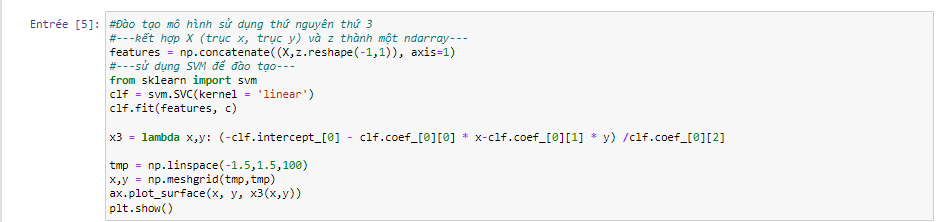




* Vẽ biểu đồ sử dụng thêm thứ nguyên thứ 3 ở đây ta sử dụng thêm z với công thức z=x^2+y^2.

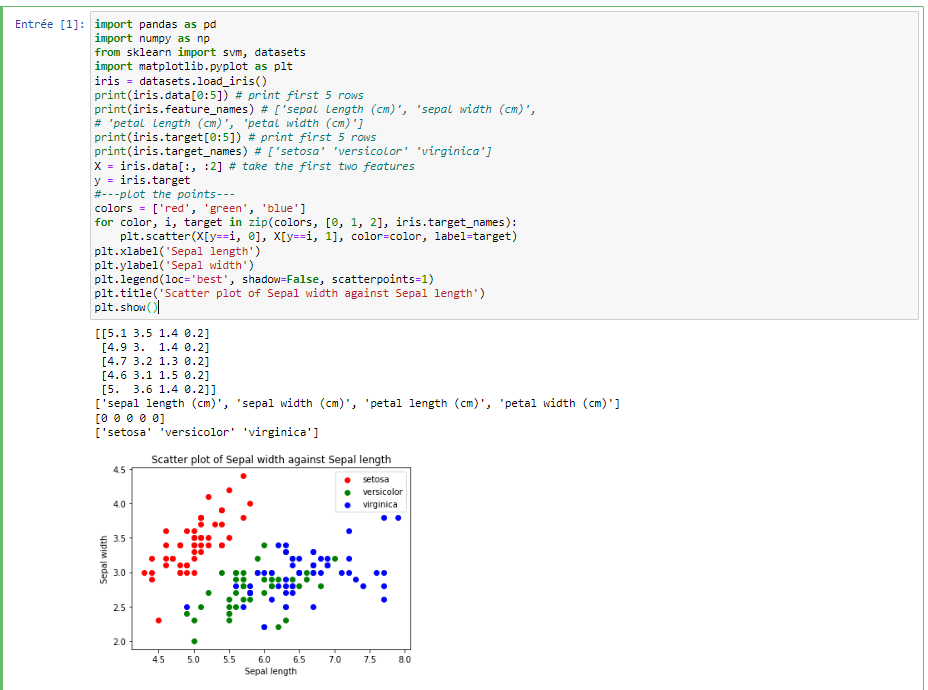


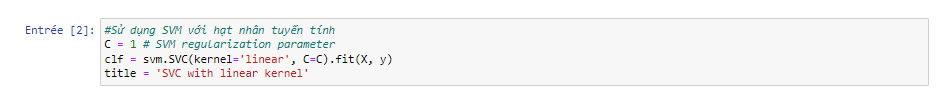


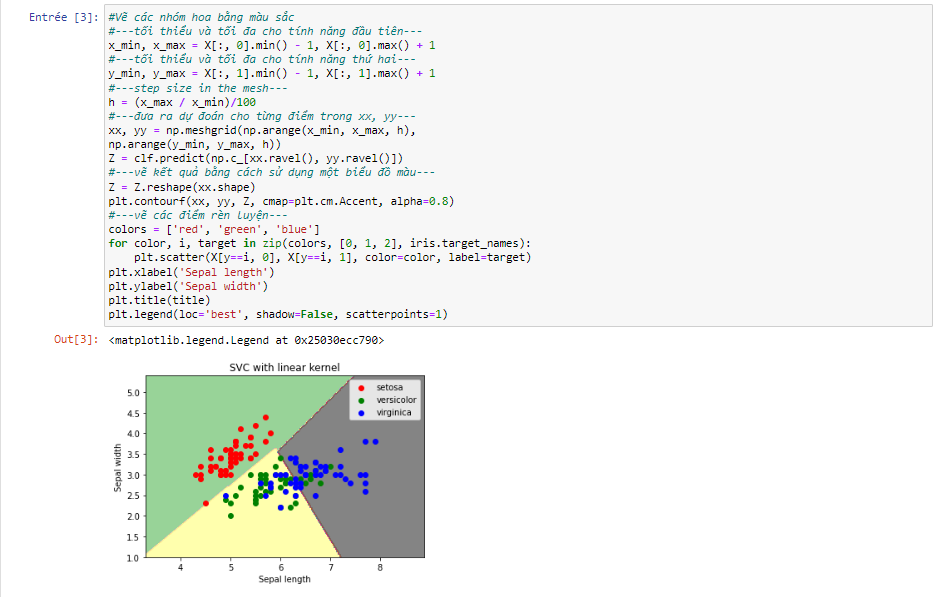


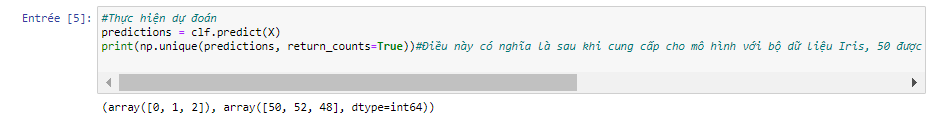
* Đôi lúc chúng ta sử dụng các đường thẳng để có thể coi đó như là một vector nhưng đôi lúc cnos sẽ không phù hợp với tập dữ liệu nên bây giwof ta sẽ thử nghiệm với các đường cong.

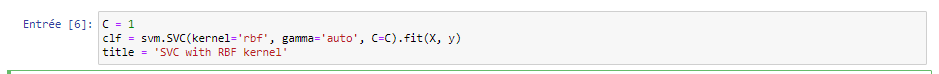
Thêm các thư viện cần thiết lấy tập dữ liệu iris bao gồm tập hợp các loài hoa. Vẽ các điểm trên đồ thị với các màu khác nhau mỗi màu đại diện cho một loài hoa.

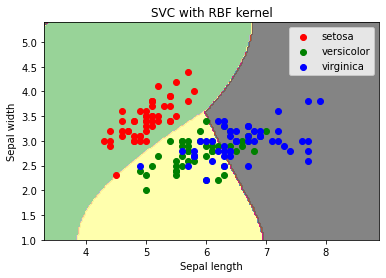










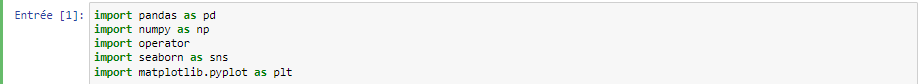


1. Phân loại sử dụng K-Nearest Neighbors (KNN).

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

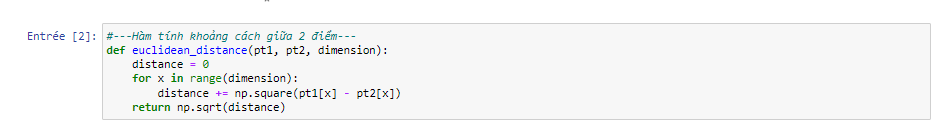
Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

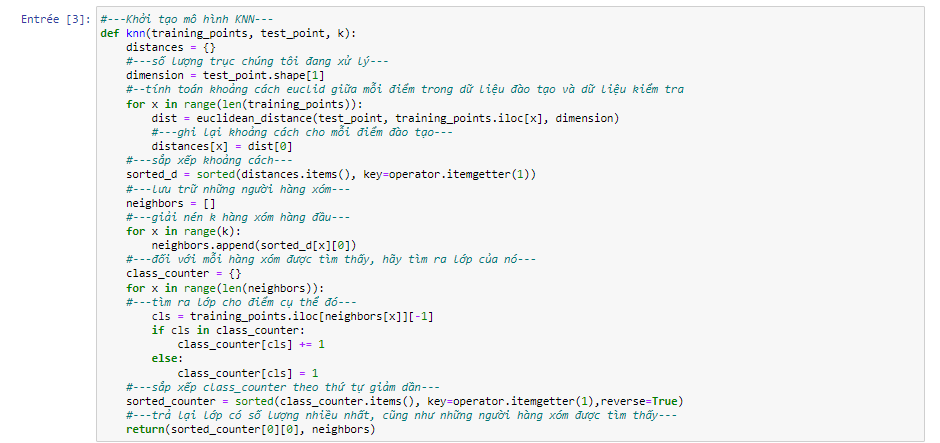
* Code ví dụ
* Bây giờ chúng ta sẽ sử dụng KNN lên một tập dữ liệu có tên là knn.csv.
* Đầu tiên thêm các thư viện cần thiết

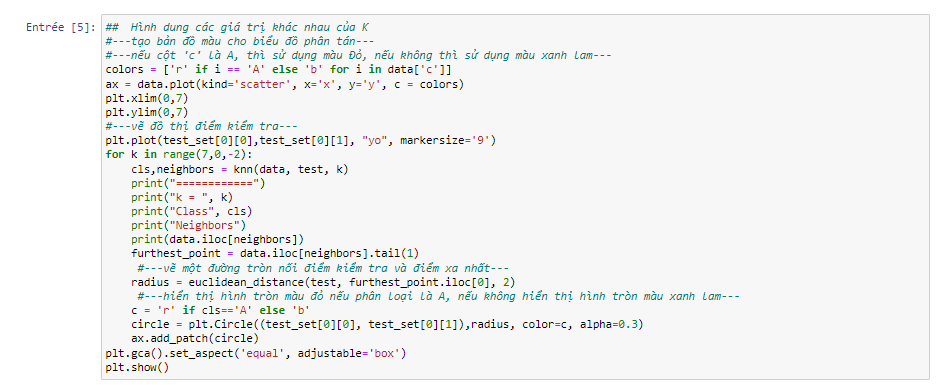


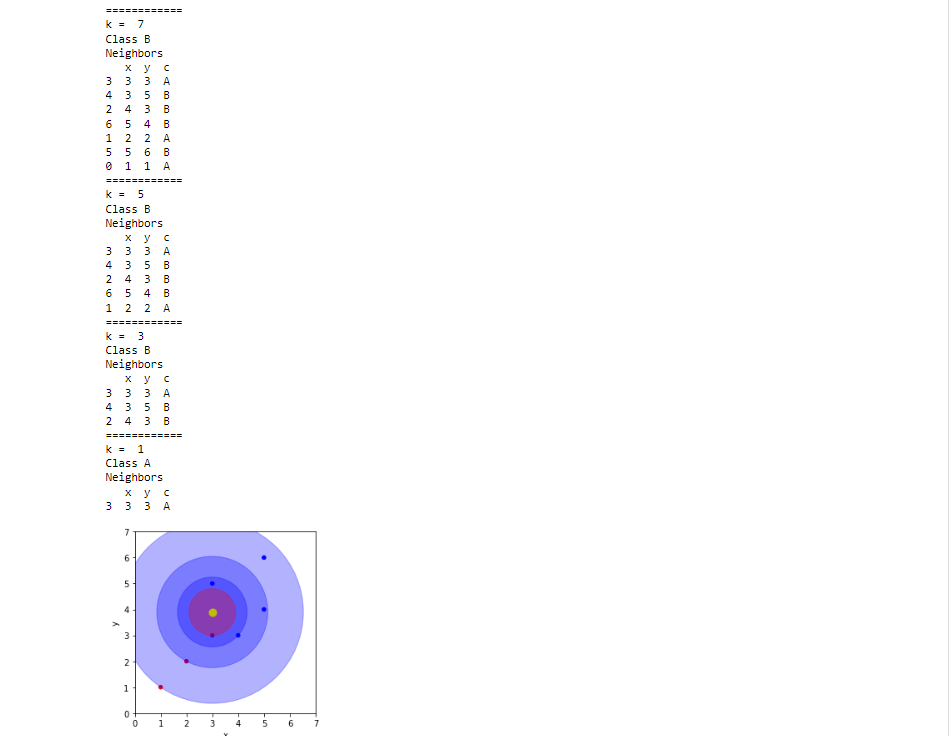
* sau đó chúng ta bắt đầu từng bước áp dụng KNN.

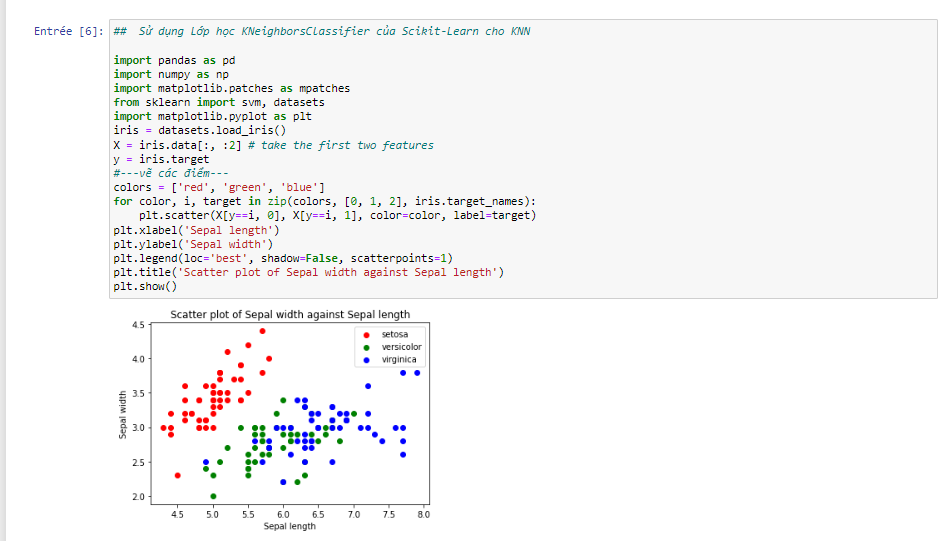


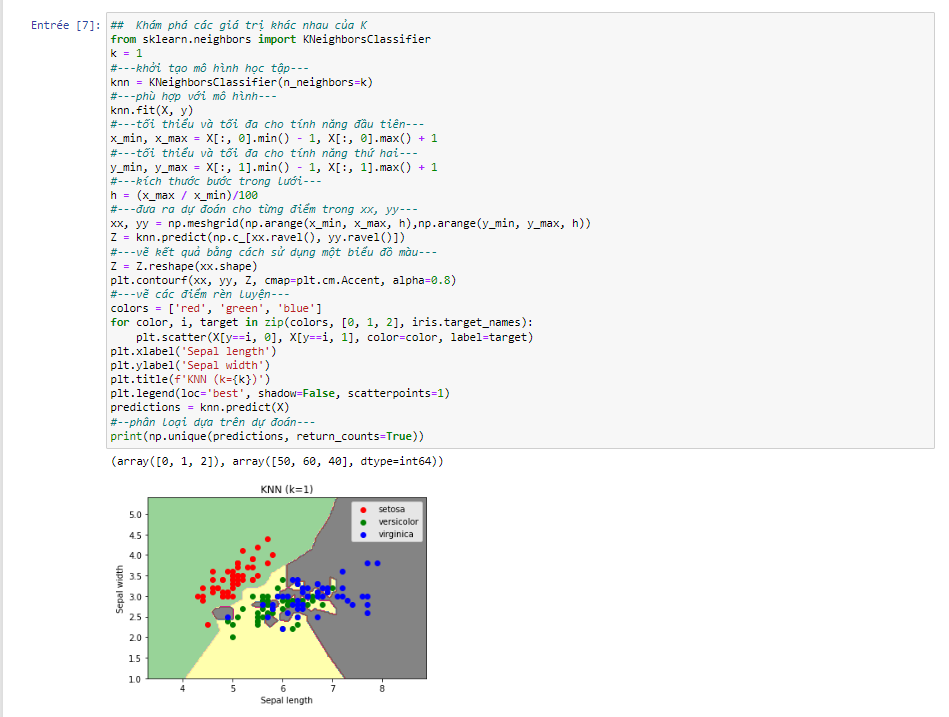


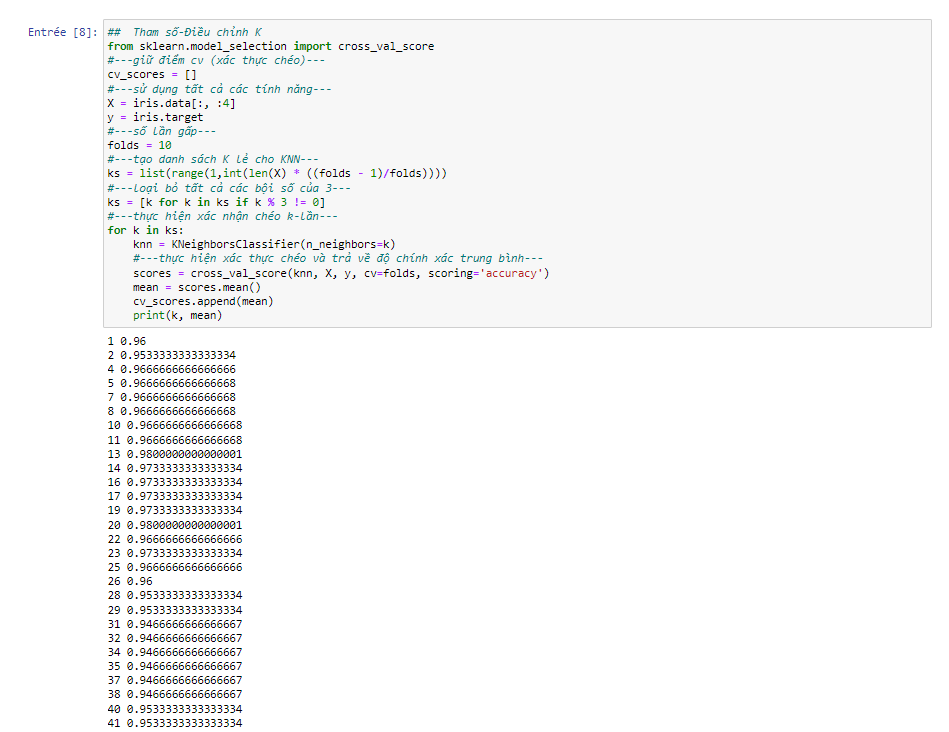


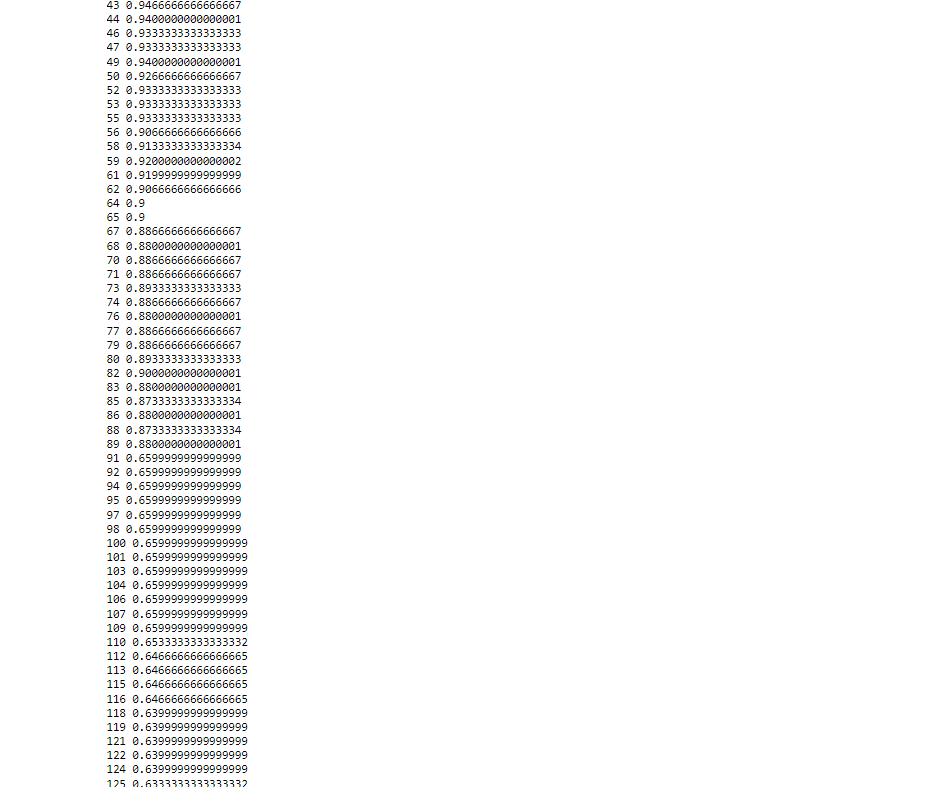


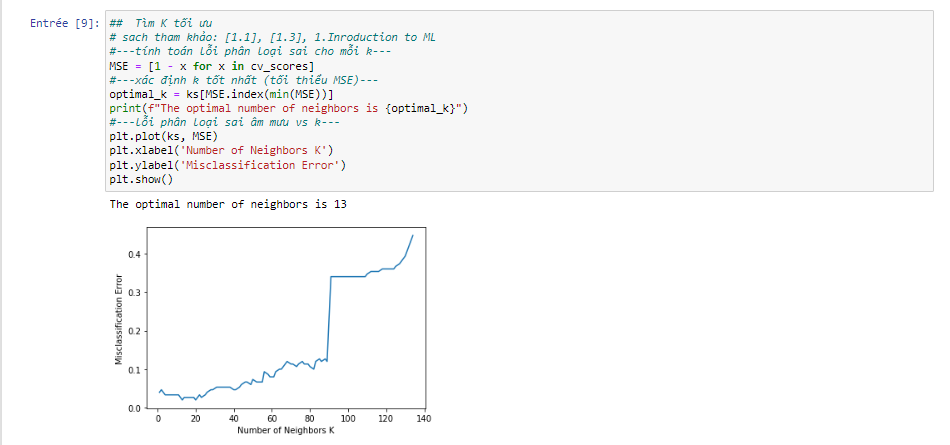






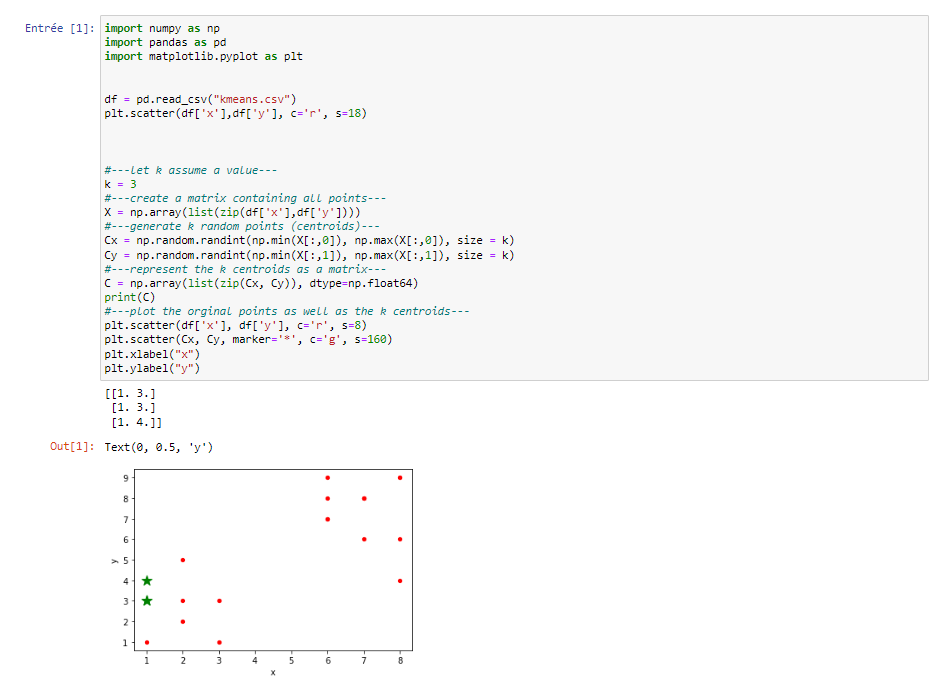


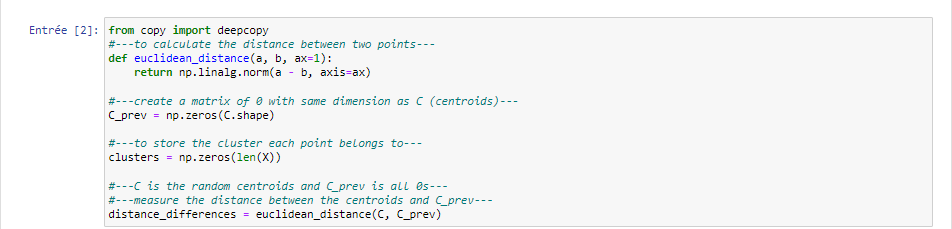


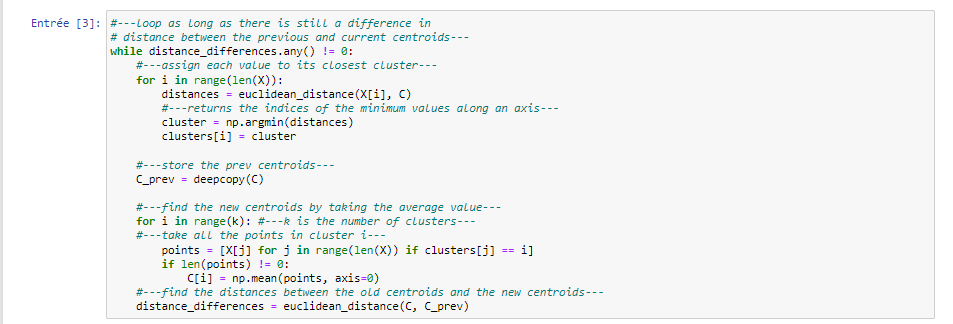


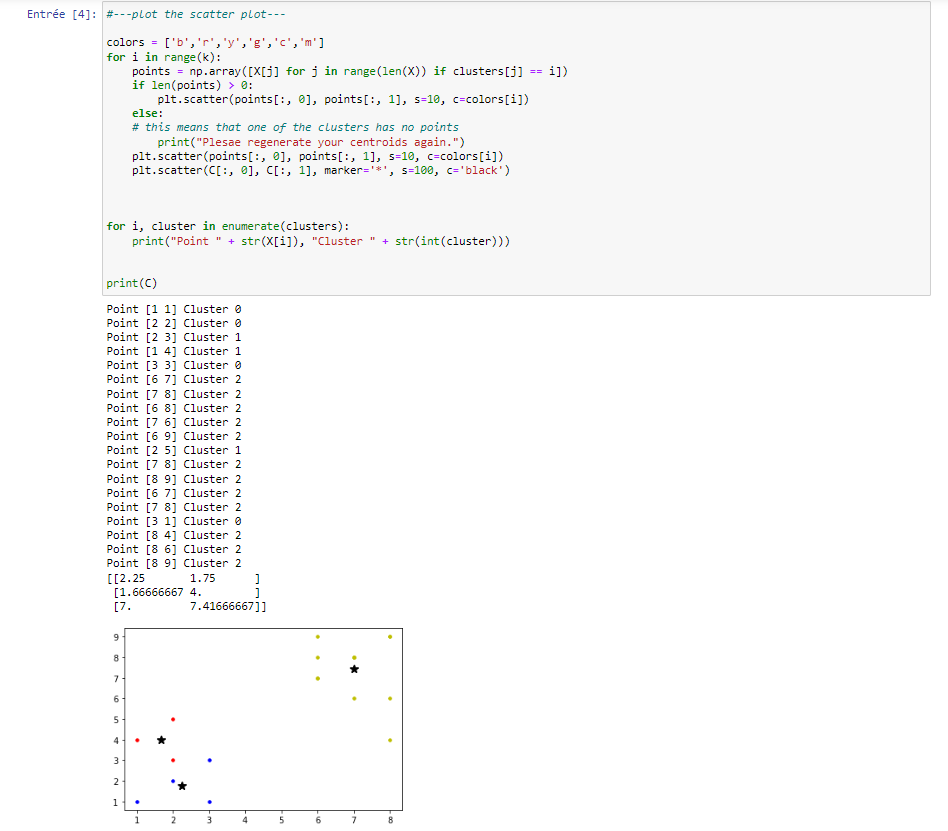
1. Học không giám sát.
2. Phân cụm bằng K-Means

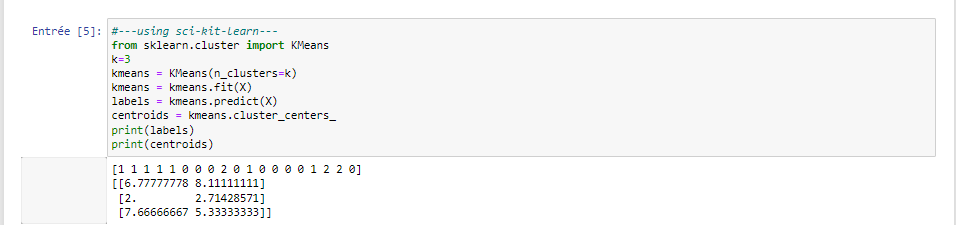
* K-means là một thuật toán phân cụm đơn giản thuộc loại học không giám sát(tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm k-means là phân chia 1 bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k. Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng 1 cụm thì phải có cùng 1 số tính chất nhất định. Tức là giữa các điểm trong cùng 1 cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong 1 cụm đó sẽ là các điểm dữ liệu gần nhau.
* Thuật toán phân cụm k-means thường được sử dụng trong các ứng dụng cỗ máy tìm kiếm, phân đoạn khách hàng, thống kê dữ liệu,…
* Triển khai kmean trên tập kmeans.csv

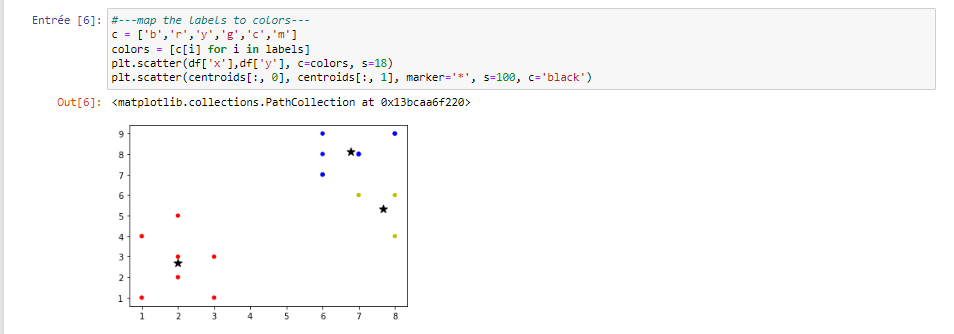




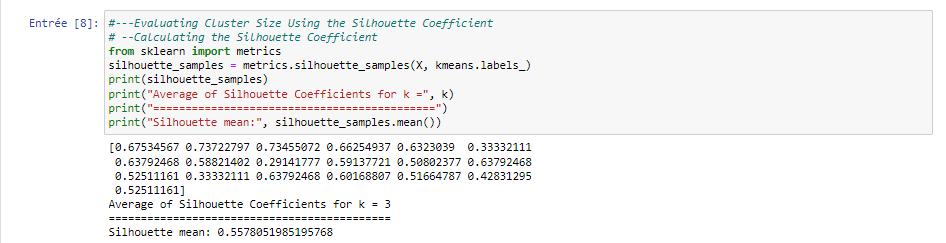


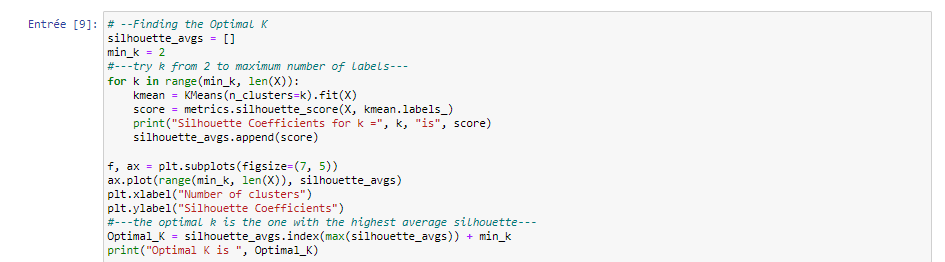


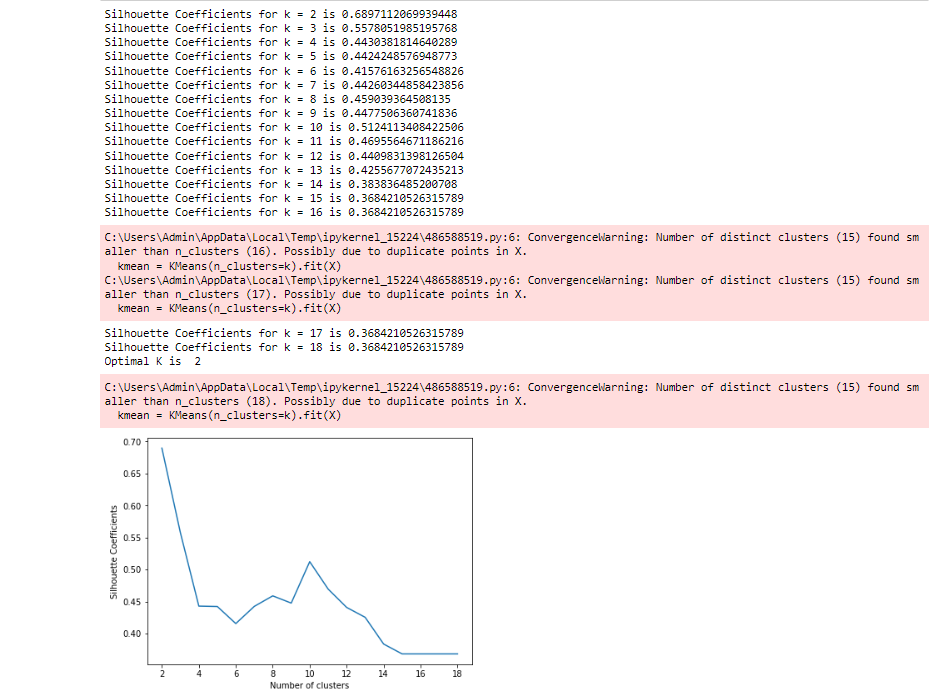












1. CASE STUDY VÀ DEPLOY
2. **Nghiên cứu điển hình**

* Nghiên cứu data về y tế và các tính năng bao gồm:
* Mang thai: Số lần mang thai
* Glucose: Nồng độ glucose trong huyết tương sau 2 giờ trong xét nghiệm dung nạp glucose qua đường uống
* Áp lực máu: Huyết áp tâm trương (mm Hg)
* Độ dày da: Độ dày nếp gấp da cơ tam đầu (mm)
* Insulin: Insulin huyết thanh trong 2 giờ (mu U / ml)
* BMI: Chỉ số khối cơ thể (cân nặng tính bằng kg / (chiều cao tính bằng m) ^ 2)
* DiabetesPedigreeFunction: Chức năng phả hệ của bệnh tiểu đường
* Tuổi: Tuổi (năm)
* Kết quả: 0 (không tiểu đường) hoặc 1 (tiểu đường)
* Bộ dữ liệu có 768 bản ghi và tất cả bệnh nhân đều là nữ từ 21 tuổi trở lên tuổi và là người gốc Ấn Độ Pima.

1. Kiểm tra mối tương quan giữa các tính năng

Text

Description automatically generated

1. Lập biểu đồ tương quan giữa các tính năng

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

%matplotlib inline

df = pd.read\_csv('diabetes.csv')

corr = df.corr()

*# khởi tạo plot*

fig, ax = plt.subplots(figsize=(10, 10))  *# size hình vẽ*

*# tạo plot với giá trị từ -1 tới 1*

cax = ax.matshow(corr, cmap='coolwarm', vmin=-1, vmax=1)

*# hiển thị chú thích về giải màu*

fig.colorbar(cax)

*# tạo 1 mảng thứ tự tương ứng với số cột của data*

ticks = np.arange(0, len(df.columns), 1)

ax.set\_xticks(ticks)

ax.set\_xticklabels(df.columns) *# gán tên*

plt.xticks(rotation=90)

ax.set\_yticklabels(df.columns)

ax.set\_yticks(ticks)

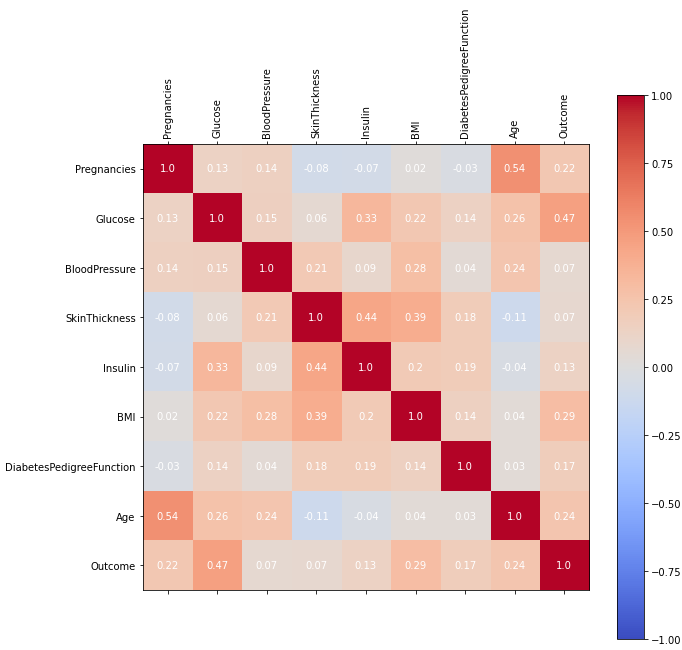
for i in range(df.shape[1]): *# duyệt các hàng trong data*

    for j in range(9): *# duyệt các cột*

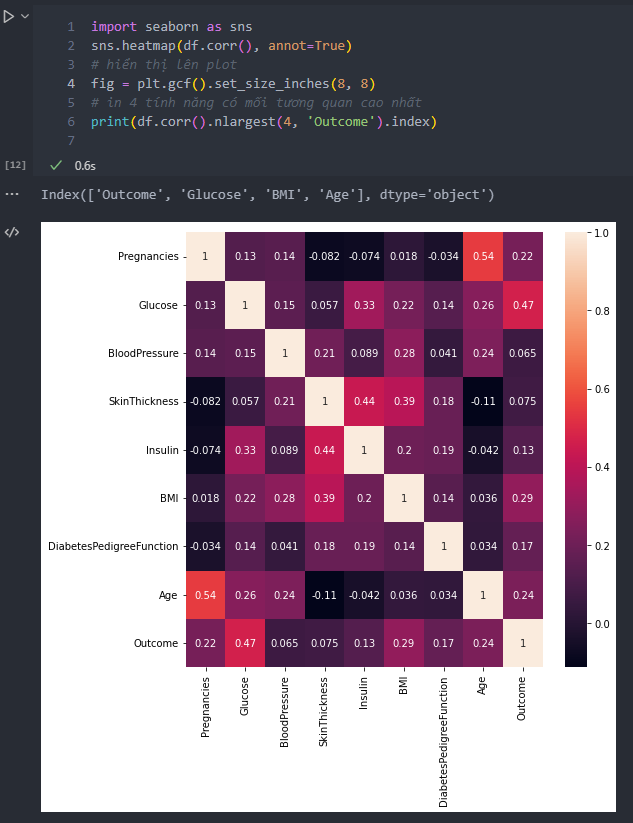
        text = ax.text(j, i, round(

            corr.iloc[i][j], 2), ha="center", va="center", color="w")

plt.show()



* Vẽ biểu đồ tương quan bằng bản đồ nhiệt của seaborn thông qua hàm heatmap():



1. **Triển khai mô hình**

* Triển khai mô hình dưới dạng REST API

**Text

Description automatically generated**

* Tạo 1 route /diabetes/v1/predict sử dụng route
* Tuyến đường có thể truy cập thông qua POST
* Để đưa ra dự đoán, người dùng thực hiện cuộc gọi đến tuyến đường này và chuyển các tính năng khác nhau bằng chuỗi JSON.
* Kết quả của dự đoán được trả về dưới dạng chuỗi JSON

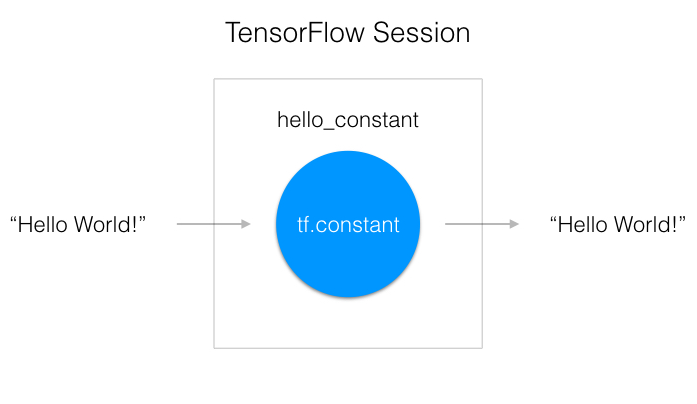
**PHẦN II: HỌC SÂU**

**CHƯƠNG 5: BIỂU DIỄN DỮ LIỆU VÀ TENSORFLOW**

**5.1. Giới thiệu về TENSOFLOW**

* Đây là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Nó hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn.
* Mội số điều cần biết về TensorFlow.

+ Dữ liệu được đóng gói trong một đối tượng dữ liệu gọi là tensor.



+ Như biểu đò ở trên ta có thể thấy việc hoạt động của TensorFlow nó sẽ khởi tạo một môi trường để chạy biểu đồ. Nó sẽ phụ trách phân bổ các hoạt động cho (các) GPU hoặc (các) CPU, bao gồm cả các máy từ xa.

* Một sô hàm trong TensorFlow.

+ tf.constant(): khởi tạo một tensor không đổi bởi vì giá trị của tensor đó không bao giờ thay đổi.

+ tf.placeholder(): khởi tạo ra một tensor có định dạng nhưng chư có dữ liệu. Ta có thể truyền dữ liệu vào đó sau bằng một số hàm.

+ tf.add(): Phép cộng

+ tf.subtract(): Phép trừ.

+ tf.multiply(): Phép nhân.

+tf.Variable(): tạo ra một tensor với giá trị ban đầu có thể được sửa đổi, giống như một biến Python bình thường. Tensor này lưu trữ trạng thái của nó trong phiên, vì vậy bạn phải khởi tạo trạng thái của tensor theo cách thủ công. Bạn sẽ sử dụng tf.global\_variables\_initializer() hàm để khởi tạo trạng thái của tất cả các tensors biến.

+ tf.truncated\_normal(): trả về một tensor có giá trị ngẫu nhiên từ phân phối chuẩn có độ lớn không quá 2 độ lệch chuẩn so với giá trị trung bình

+ tf.zeros(): trả về một tensor với tất cả các số không.

+ tf.nn.softmax(): triển khai chức năng softmax. Là chức năng tính toán xác suất sảy ra của một sự kiện.

**CHƯƠNG 6: HỌC SÂU VÀ KERAS**

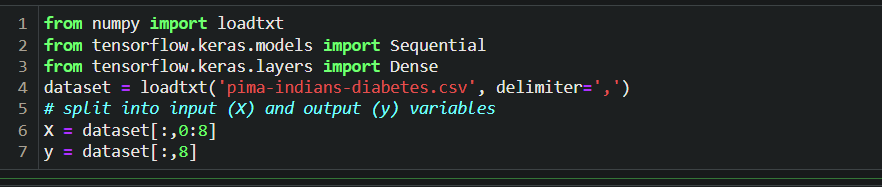
* 1. Các bước xây dựng mô hình Deep Learning với Tensorflow và Keras

Bước 1: Load data:

Sử dụng [thư viện NumPy](https://www.numpy.org/) để tải tập dữ liệu và hai lớp từ [thư viện Keras](https://www.tensorflow.org/api_docs/python/tf/keras) để xác định mô hình.

tải tệp dưới dạng ma trận số bằng cách sử dụng hàm NumPy [loadtxt ()](https://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html) .

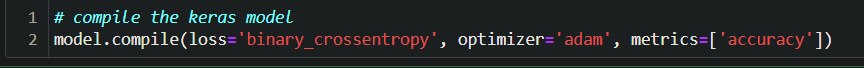
Có tám biến đầu vào (X) và một biến đầu ra (cột cuối cùng)(y)



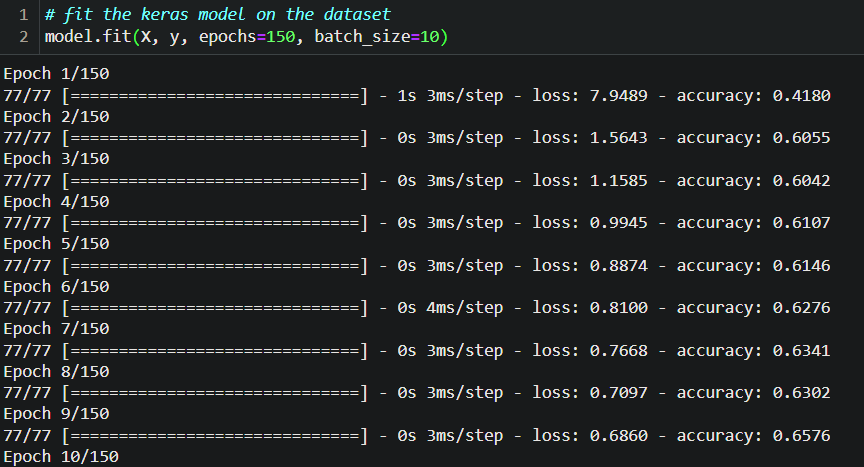
## Bước 2: Xác định mô hình Keras

## 

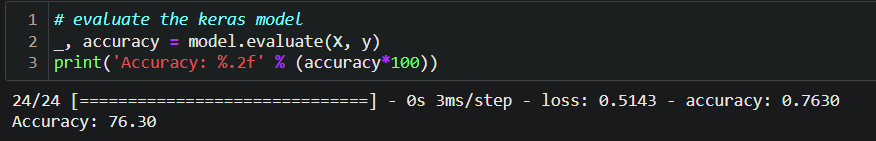
Bước 3: Biên dịch mô hình Keras



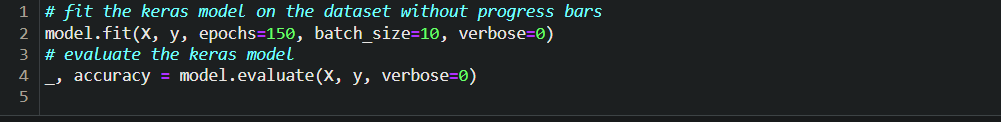
Bước 4: Fit Keras Model



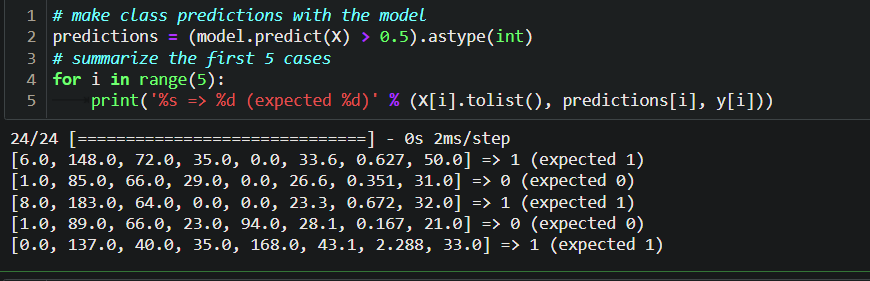
Bước 5: Đánh giá Mô hình Keras



Bước 6: Buộc tất cả lại với nhau



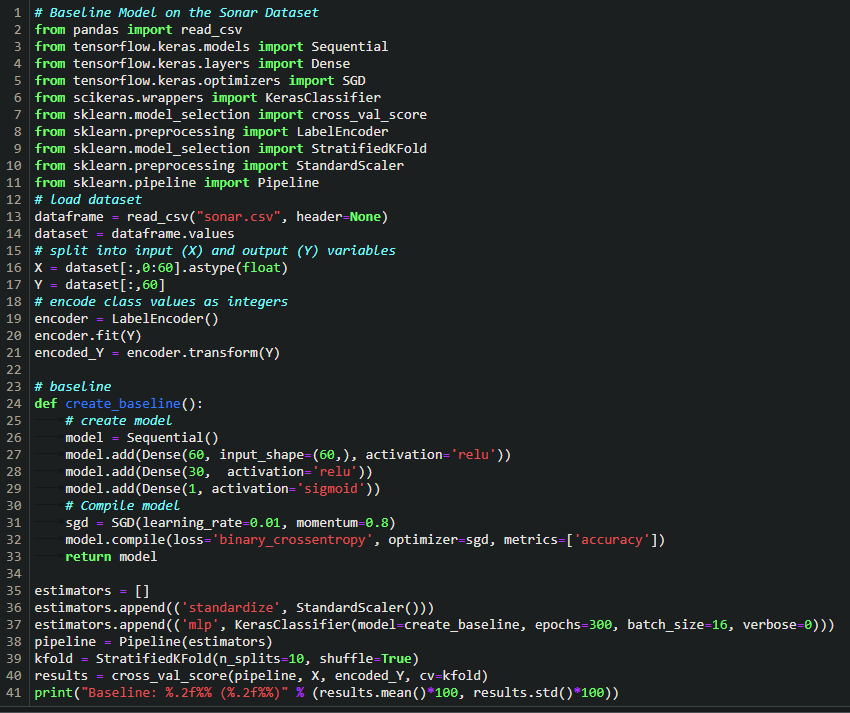
Bước 7: Đưa ra dự đoán



* 1. Dropout

Dropout là một kỹ thuật điều chỉnh đơn giản và mạnh mẽ cho mạng nơ-ron và mô hình học sâu.

Dropout là một kỹ thuật mà các tế bào thần kinh được lựa chọn ngẫu nhiên bị bỏ qua trong quá trình đào tạo. Việc bỏ cuộc được thực hiện dễ dàng bằng cách chọn ngẫu nhiên các nút bị loại bỏ với xác suất cho trước (ví dụ: 20%) trong mỗi chu kỳ cập nhật trọng số.





* 1. Các khái niệm Keras:

1. Epoch

Epoch sẽ đề cập đến một lần chuyển hoàn chỉnh của tập dữ liệu đào tạo thông qua thuật toán.

Epoch được định nghĩa là tổng số lần lặp lại để đào tạo mô hình học máy với tất cả dữ liệu đào tạo trong một chu kỳ. Trong Epoch, tất cả dữ liệu đào tạo được sử dụng chính xác một lần. Nói cách khác, Epoch còn có thể được hiểu là tổng số lần vượt qua mà một thuật toán đã hoàn thành xung quanh tập dữ liệu huấn luyện. Một đường chuyền tiến và một đường chuyền lùi cùng nhau được tính là một đường chuyền trong huấn luyện.

1. Bath(Lô)

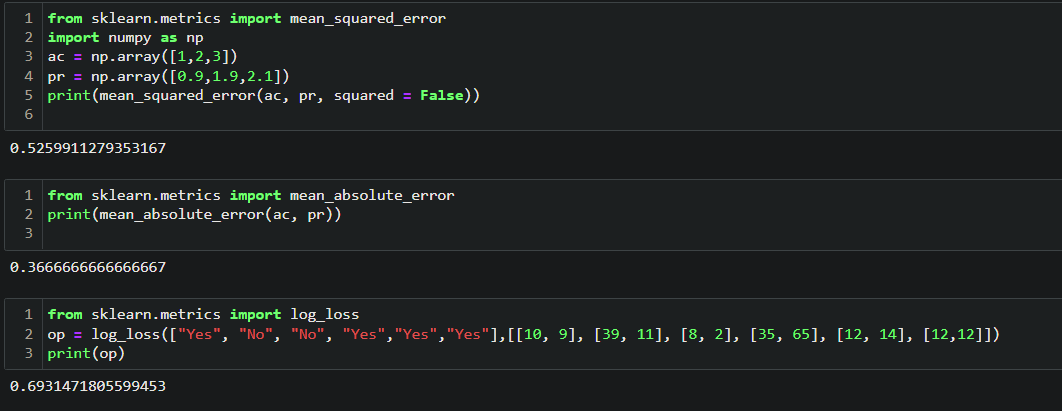
Kích thước lô và lô là hai thực thể riêng biệt trong học máy.

Kích thước lô được định nghĩa là tổng số ví dụ đào tạo tồn tại trong một lô duy nhất. Bạn cũng có thể hiểu theo lô với ví dụ được đề cập ở trên, trong đó chúng tôi đã chia toàn bộ tập dữ liệu / ví dụ đào tạo thành các lô hoặc bộ hoặc phần khác nhau.

1. Loss

Mạng nơ-ron được đào tạo bằng cách sử dụng dốc nghiêng ngẫu nhiên và yêu cầu bạn chọn một hàm mất mát khi thiết kế và cấu hình mô hình của mình.

Có rất nhiều hàm mất mát để lựa chọn và có thể rất khó khăn khi biết phải chọn cái gì, hoặc thậm chí hàm mất mát là gì và vai trò của nó khi đào tạo một mạng nơ-ron.



1. Optimizer

**CHƯƠNG 7: MỘT SỐ KỸ THUẬT VÀ ỨNG DỤNG**

//sinh viên tự chọn một case study để thể hiện