**Giảng viên hướng dẫn:**

**Thầy: Huỳnh Ngọc Tín**

**Sinh viên thực hiện:**

**Ngô Trung Hiếu- 16520396**

**Bùi Đặng Hồng Chung- 16520122**

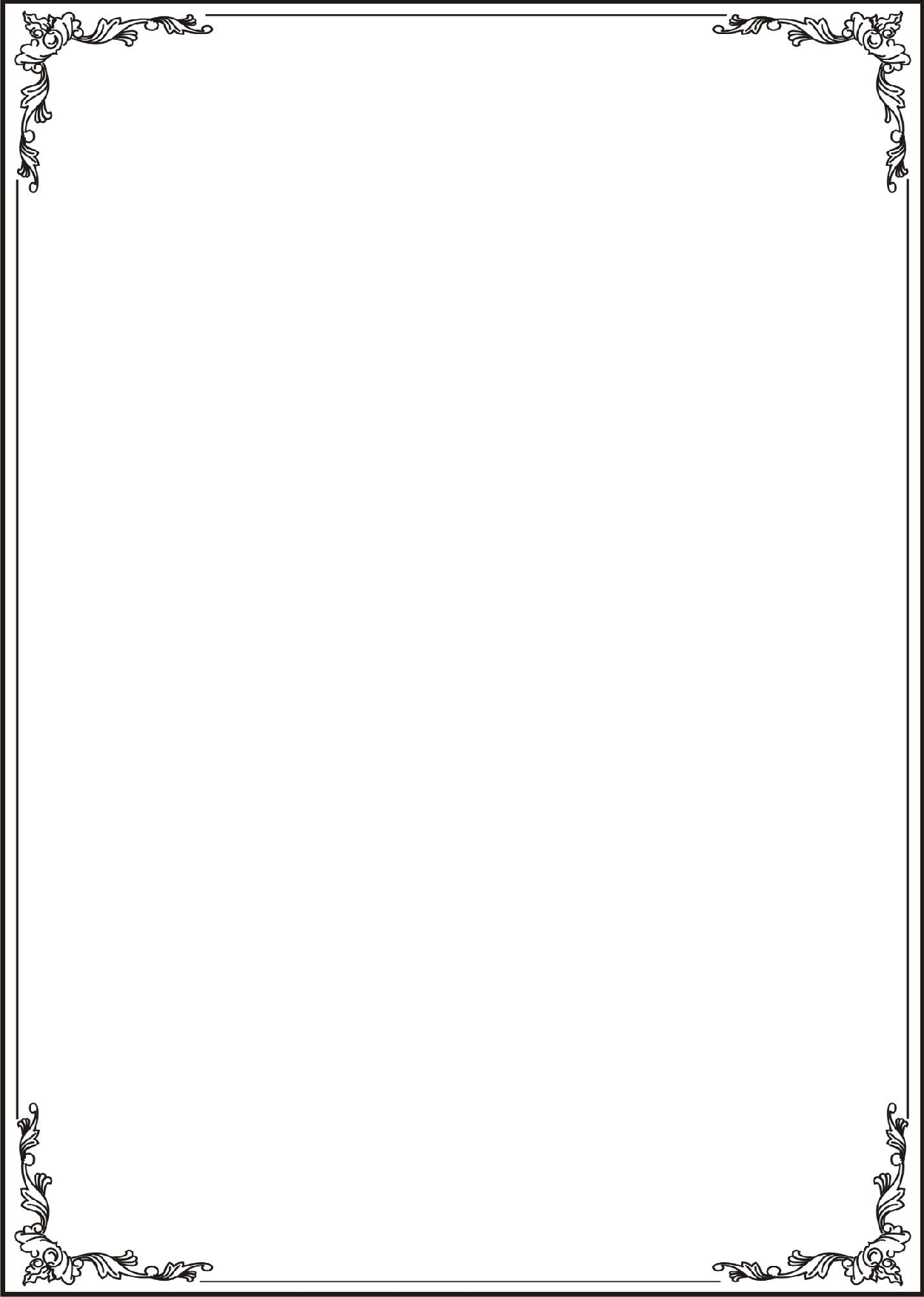
**ĐỒ ÁN MÃ NGUỒN MỞ**

**BÀI TOÁN DỰ ĐOÁN NGƯỜI SỐNG SÓT TRÊN TÀU TITANIC**

**ĐẠI HỌC QUỐC GIA TPHCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**MỤC LỤC**

[**I.** **CÁC MÃ NGUỒN MỞ ÁP DỤNG VÀ KIẾN THỨC YÊU CẦU:** 3](#_Toc28511399)

[**1.** **Thư viện Pandas:** 3](#_Toc28511400)

[**a)** **Thư viện Pandas Python là gì?** 3](#_Toc28511401)

[**b)** **Đọc file csv sử dụng thư viện Pandas:** 3](#_Toc28511402)

[**c)** **Thao tác với dataframe trong Pandas:** 3](#_Toc28511403)

[**d)** **Tạo mới dataframe:** 5](#_Toc28511404)

[**e)** **Lưu dataframe về file csv:** 5](#_Toc28511405)

[**2.** **Thư viện matplotlib.pyplot:** 5](#_Toc28511406)

[**a)** **Biểu đồ thanh:** 7](#_Toc28511407)

[**b)** **Biểu đồ tròn:** 8](#_Toc28511408)

[**c)** **Histogram:** 9](#_Toc28511409)

[**d)** **Sơ đồ phân tán và 3 chiều:** 9](#_Toc28511410)

[**3.** **Thư viện Seaborn:** 11](#_Toc28511411)

[**4.** **Thư viện sklearn:** 11](#_Toc28511412)

[**II.** **BÀI TOÁN DỰ ĐOÁN NGƯỜI SỐNG SÓT TRÊN TÀU TITANIC:** 12](#_Toc28511413)

[**1.** **Bài toán:** 12](#_Toc28511414)

[**2.** **Hiện thực bài toán:** 14](#_Toc28511415)

[**3.** **Xử lí data:** 14](#_Toc28511416)

[**3.1. Xử lí missing value:** 14](#_Toc28511417)

[**Feature Age:** 14](#_Toc28511418)

[**Feature Embarked:** 15](#_Toc28511419)

[**Feature Fare:** 16](#_Toc28511420)

[**3.2. Feature Engineering:** 16](#_Toc28511421)

[**Feature Family:** 16](#_Toc28511422)

[**Feature Title:** 16](#_Toc28511423)

[**3.3. OneHotEncoder cho data:** 17](#_Toc28511424)

[**4.** **Parameter tuning using gridCV:** 17](#_Toc28511425)

[**5.** **Xây dựng model bằng pipeline:** 18](#_Toc28511426)

[**6.** **Training model, predict và kết quả đạt được:** 19](#_Toc28511427)

[**7.** **Tài liệu tham khảo:** 20](#_Toc28511428)

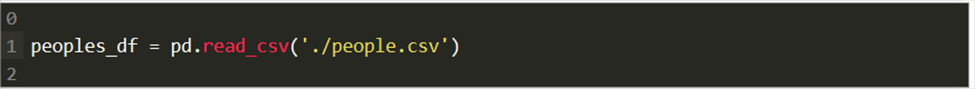
1. **CÁC MÃ NGUỒN MỞ ÁP DỤNG VÀ KIẾN THỨC YÊU CẦU:**
2. **Thư viện Pandas:**
3. **Thư viện Pandas Python là gì?**

Thư viện Pandas trong Python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình Python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Pandas sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho Pandas được sử dụng rộng rãi.

1. **Đọc file csv sử dụng thư viện Pandas:**

Đọc csv file vào dataframe:

Bạn có thể dễ dàng đọc vào một file .csv bằng cách sử dụng hàm **read\_csv** và được trả về 1 dataframe. Mặc định, hàm này sẽ phân biệt các trường của file csv theo dấu phẩy. Cách đọc hết sức đơn giản như sau:



1. **Thao tác với dataframe trong Pandas:**

Khi bạn đọc file csv như phía trên, peoples\_df sẽ là một đối tượng có hàng, cột gọi là DataFrame.

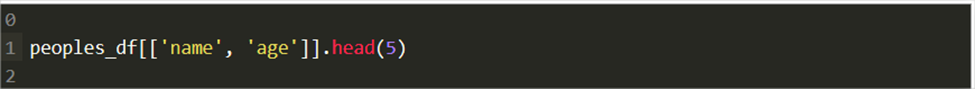
Bạn có thể xem thông tin của dataframe vừa đọc vào bằng cách sử dụng hàm .info() hoặc xem kích thước của dataframe này với thuộc tính shape.

*Truy xuất dữ liệu trên dataframe:*

Lấy 1 cột theo tên cột:



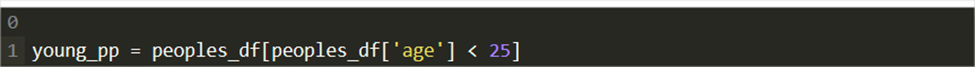
Lấy theo nhiều cột:

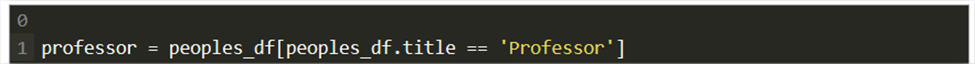


Lấy bản ghi theo chỉ số:



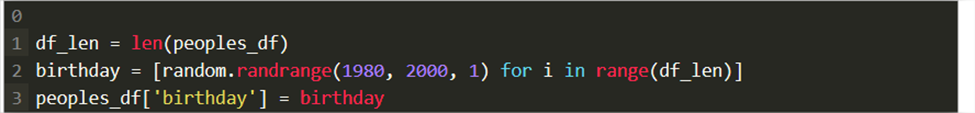
Lấy các bản ghi theo điều kiện:



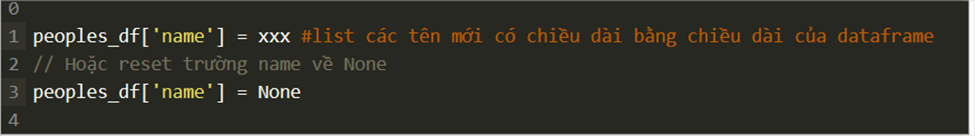


*Thêm, sửa, xóa trong dataframe:*

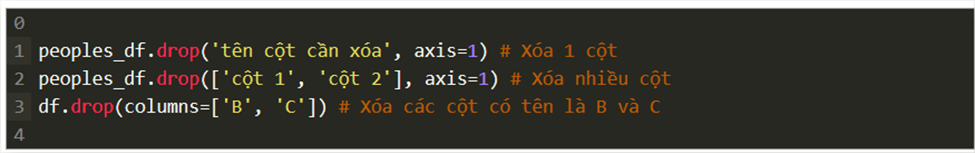
Thêm cột từ dữ liệu mới:



Sửa giá trị của cột:

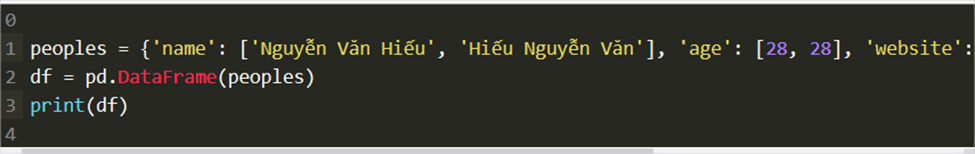


Xóa cột trong dataframe:



1. **Tạo mới dataframe:**

Tạo mới dataframe từ python dictionary:

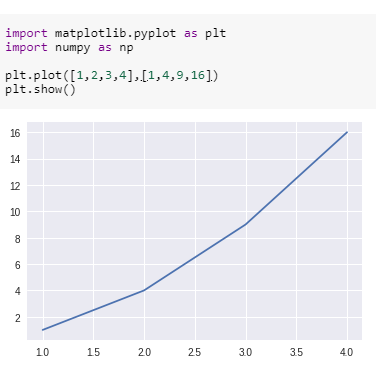


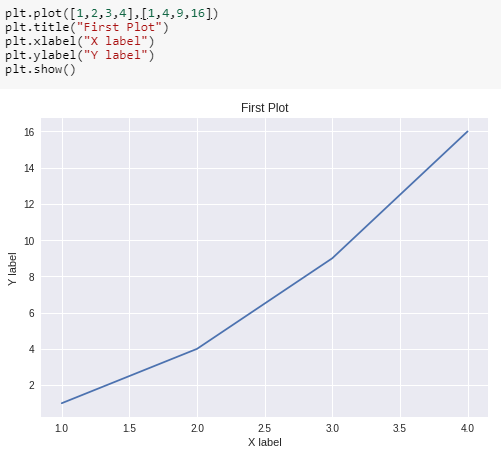
1. **Lưu dataframe về file csv:**

****

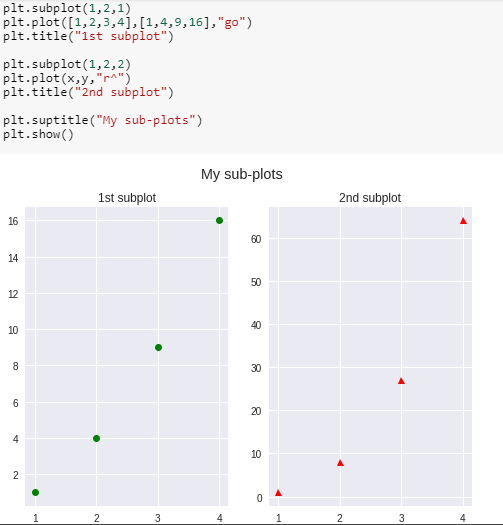
1. **Thư viện matplotlib.pyplot:**

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.



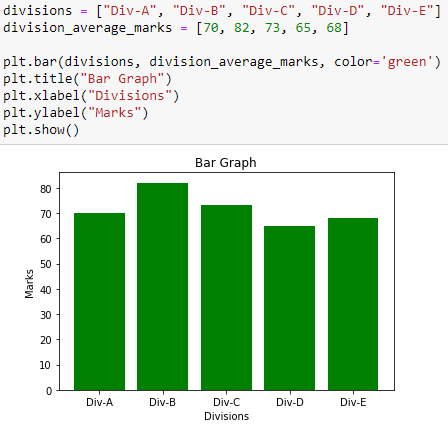


Nhiều biểu đồ trong 1 figure:

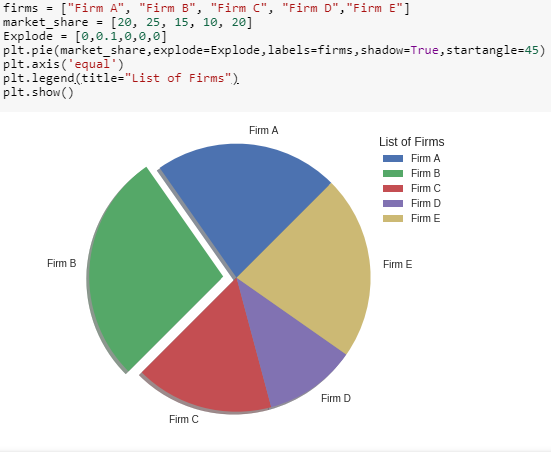


Tạo các loại biểu đồ khác nhau với Pyplot:

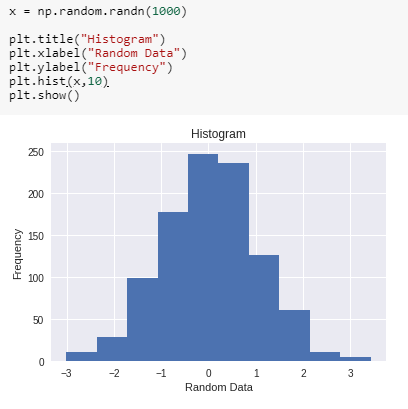
1. **Biểu đồ thanh:**



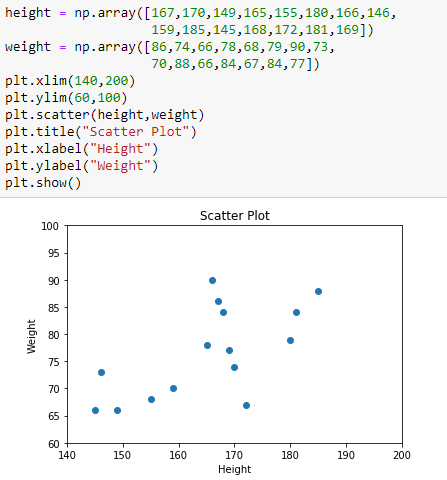
1. **Biểu đồ tròn:**



1. **Histogram:**



1. **Sơ đồ phân tán và 3 chiều:**



Kết luận: các phương thức cần thiết

* plot(x-axis values, y-axis values) —đồ thị đường đơn giản với các giá trị trục x so với giá trị trục y.
* show() - hiển thị biểu đồ.
* title(“string”) - đặt tiêu đề.
* xlabel(“string”) - đặt nhãn cho trục x..
* ylabel(“string”) - đặt nhãn cho trục y.
* figure() - dùng để control các thuộc tính của mức hình.
* subplot(nrows, ncols, index) - thêm một subplot vào figure hiện tại.
* suptitle(“string”) - thêm một tiêu đề chung vào hình.
* subplots(nrows, ncols, figsize) - một cách thuận tiện để tạo các subplots. Nó trả về một tuple của figure và số lượng axes.
* set\_title(“string”) - dùng để đặt tiêu đề cho các ô con.
* bar(categorical variables, values, color) - được sử dụng để tạo đồ thị thanh dọc.
* barh(categorical variables, values, color) - dùng để tạo biểu đồ thanh ngang
* legend(loc) - dùng để tạo chú thích của đồ thị.
* xticks(index, categorical variables) - get hoặc set vị trí đánh dấu hiện tại và nhãn của trục x.
* pie(value, categorical variables) - dùng để tạo biểu đồ hình tròn.
* hist(values, number of bins) - được sử dụng để tạo histogram.
* xlim(start value, end value) - được sử dụng để đặt giới hạn giá trị của trục x.
* ylim(start value, end value) - được sử dụng để đặt giới hạn giá trị của trục y.
* scatter(x-axis values, y-axis values) - vẽ sơ đồ phân tán với các giá trị trục x so với giá trị trục y.
* axes() - thêm một axes vào hình hiện tại.
* set\_xlabel(“string”) - được sử dụng đễ set nhãn x cho plot được chỉ định.
* set\_ylabel(“string”) - được sử dụng đễ set nhãn y cho plot được chỉ định.
* scatter3D(x-axis values, y-axis values) - vẽ sơ đồ phân tán ba chiều với các giá trị trục x so với giá trị trục y.
* plot3D(x-axis values, y-axis values) - vẽ đồ thị đường ba chiều với các giá trị trục x so với giá trị trục y.

1. **Thư viện Seaborn:**

Seaborn cung cấp các APIs xây dựng trên thư viện Matplotlib, cung cấp các lựa chọn cho các kiểu mặc định và màu sắc, định nghĩa các chức năng cấp cao đơn giản cho các kiểu thống kê chung và tích hợp với các chức năng do Pandas DataFrames cung cấp.

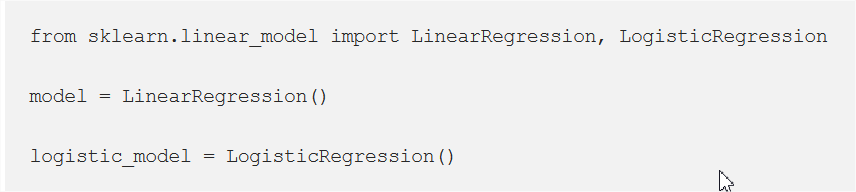
Seaborn nên được coi là một tool bổ sung cho matplotlib chứ không phải là một sự thay thế cho nó. Khi sử dụng seaborn, rất có thể bạn sẽ thường gọi các hàm matplotlib trực tiếp để vẽ các đồ thị đơn giản đã có sẵn qua không gian tên pyplot. Hơn nữa, các chức năng trên Seaborn nhằm mục đích tạo ra các plots có mẫu sẵn, hợp lý (bao gồm khai thác thông tin ngữ nghĩa từ các đối tượng Pandas để thêm các nhãn thông tin), nhưng tuỳ biến đầy đủ sẽ yêu cầu thay đổi trực tiếp các thuộc tính trên các đối tượng matplotlib. Sự kết hợp giữa giao diện cấp cao của seaborn và khả năng tuỳ biến của matplotlib và các phụ trợ dễ dàng tạo ra các figures chất lượng cao.

Seaborn là thư viện giúp cho người sử dụng dễ dàng visulize dữ liệu chỉ qua một vài bước đơn giản. Sự ra đời của Seaborn cũng đã giúp cho python có thể cạch tranh với R về khả năng visualization nhanh dữ liệu.

1. **Thư viện sklearn:**

Sklearn là thư viện về machine learning phổ biến nhất của Python. Nó viết sẵn các thuật toán lằng nhằng phức tạp, bạn chỉ việc nhét dữ liệu vào, chờ nó tính toán rồi lấy kết quả.

Cách implement 1 model cực kì đơn giản trong sklearn:



Ngoài ra còn các lớp mà sklearn cung cấp chúng ta sẽ tìm hiểu trong quá trình thực hiện đồ án.

1. **BÀI TOÁN DỰ ĐOÁN NGƯỜI SỐNG SÓT TRÊN TÀU TITANIC:**
2. **Bài toán:**

***Các thách thức:***

Vụ đắm tàu Titanic là một trong những vụ đắm tàu khét tiếng nhất trong lịch sử.

Vào ngày 15 tháng 4 năm 1912, trong chuyến hành trình đầu tiên của mình, tàu Titan RMS Titanic được coi là không thể tưởng tượng được đã bị chìm sau khi va chạm với một tảng băng trôi. Thật không may, không có đủ xuồng cứu sinh cho mọi người trên tàu, dẫn đến cái chết của 1502 trong số 2224 hành khách và phi hành đoàn.

Mặc dù có một số yếu tố may mắn liên quan đến việc sống sót, nhưng có vẻ như một số nhóm người có khả năng sống sót cao hơn những người khác.

Trong thử thách này, chúng tôi yêu cầu bạn xây dựng một mô hình dự đoán để trả lời câu hỏi: Những loại người nào có khả năng sống sót cao hơn? Sử dụng dữ liệu hành khách (ví dụ như tên, tuổi, giới tính, tầng lớp kinh tế xã hội, v.v.).

***Mục tiêu:***

Nhiệm vụ của bạn là dự đoán liệu một hành khách có sống sót sau vụ đắm tàu Titanic hay không.

Đối với mỗi trong bộ kiểm tra, bạn phải dự đoán giá trị 0 hoặc 1 cho biến.

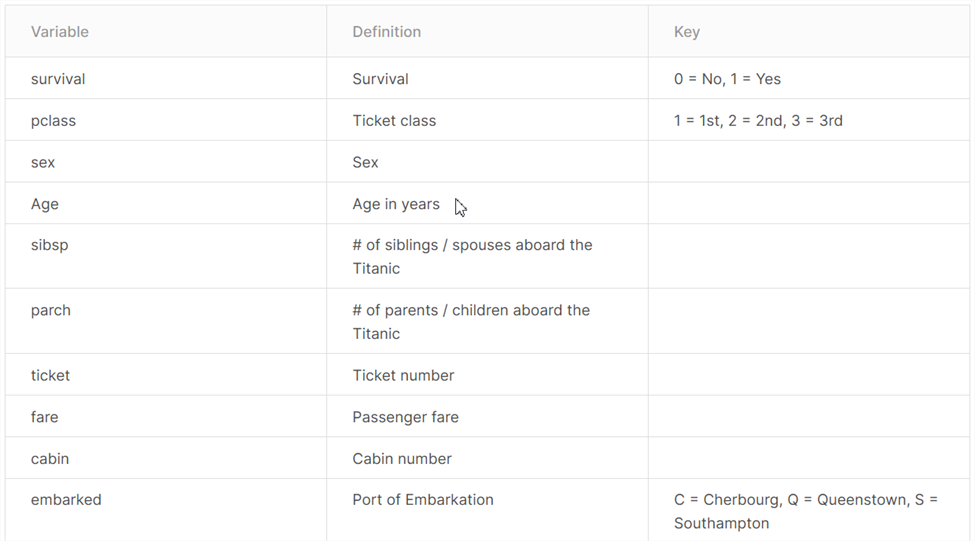
***Metric:***

Điểm của bạn là tỷ lệ phần trăm hành khách bạn dự đoán chính xác. Điều này được gọi là chính xác.

***Mô tả dữ liệu:***

Dữ liệu đã được chia thành hai nhóm:

* training set (train.csv)
* test set (test.csv)

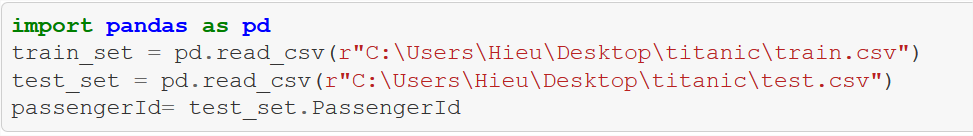


Ghi chú biến:

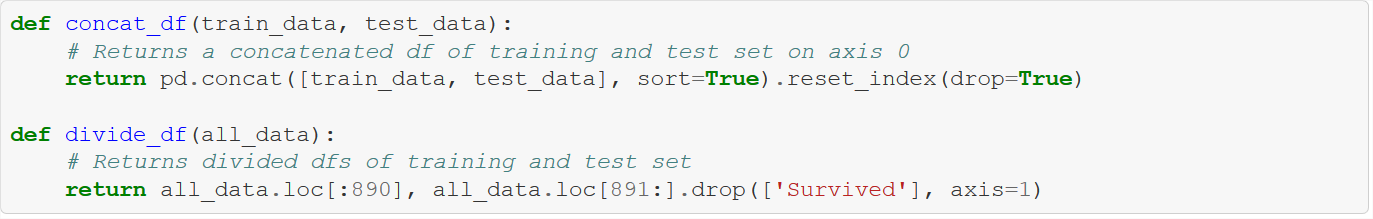
**pclass**: A proxy for socio-economic status (SES)  
1st = Upper  
2nd = Middle  
3rd = Lower  
  
**age**: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5  
  
**sibsp**: The dataset defines family relations in this way...  
Sibling = brother, sister, stepbrother, stepsister  
Spouse = husband, wife (mistresses and fiancés were ignored)  
  
**parch**: The dataset defines family relations in this way...  
Parent = mother, father  
Child = daughter, son, stepdaughter, stepson  
Some children travelled only with a nanny, therefore parch=0 for them.

1. **Hiện thực bài toán:**

Import thư viện Pandas và data



Một số function cần thiết manipulate DF:



Concat: kết hợp train\_data và test\_data để xử lí data 1 lần duy nhất.

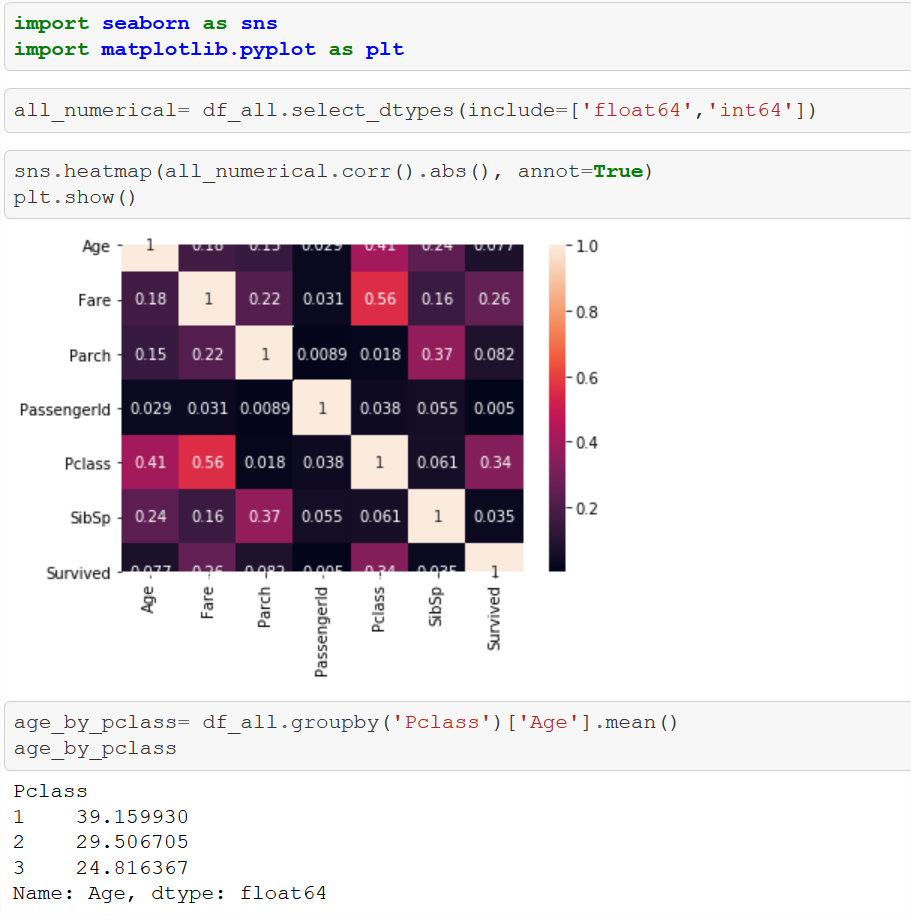
Devide: Sau khi xong bước trên, tách lại ra như ban đầu.



1. **Xử lí data:**

### **3.1. Xử lí missing value:**

**Feature Age:**



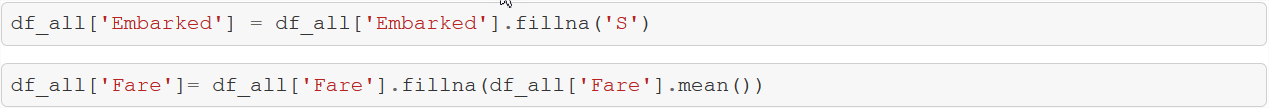
Import thư viện plt và seaborn.

Plot heatmap để xem sự tương quan giữa đặc tính tuổi tác so với các đặc tính còn lại -> Pclass.



**Feature Embarked:**

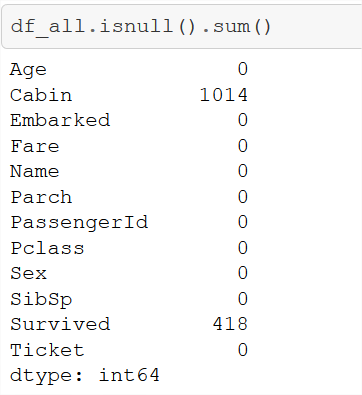
Tiến hành apply các row missing value theo feature Pclass.



Đối với feature Embarked, dùng một chút google search chúng ta có thể thấy hành khách missing bắt đầu ở ‘S’.

**Feature Fare:**

Đối với feature Fare, do chỉ thiếu 1 row, chúng ta fillna bằng giá trị mean().



🡺 Chỉ còn sót lại feature Cabin (chúng ta sẽ drop vì missing values khá nhiều) và survived missing ở test data (không quan trọng).

### **3.2. Feature Engineering:**

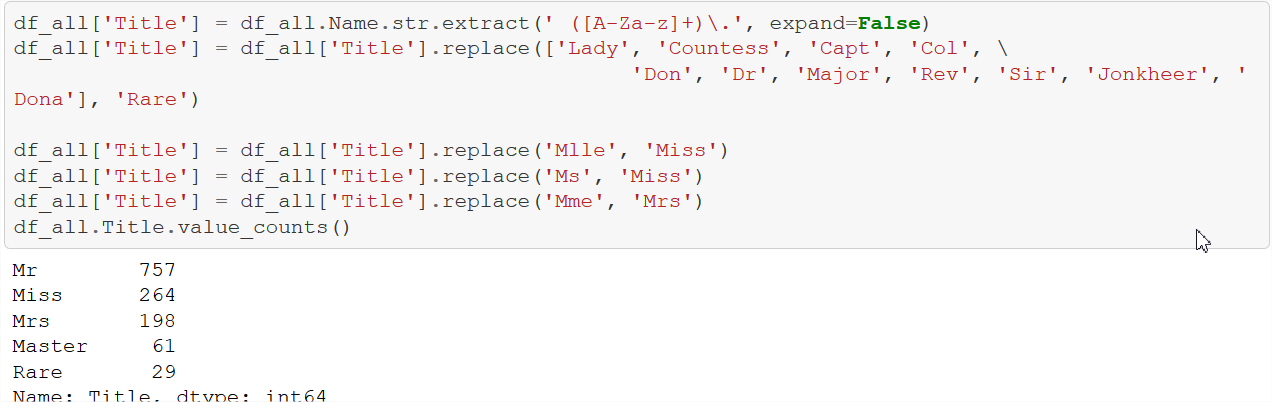
### **Feature Family:**

Tạo feature family bằng 2 feature Sibsp, Parch:



**Feature Title:**

Tạo feature Tittle bằng feature Name:

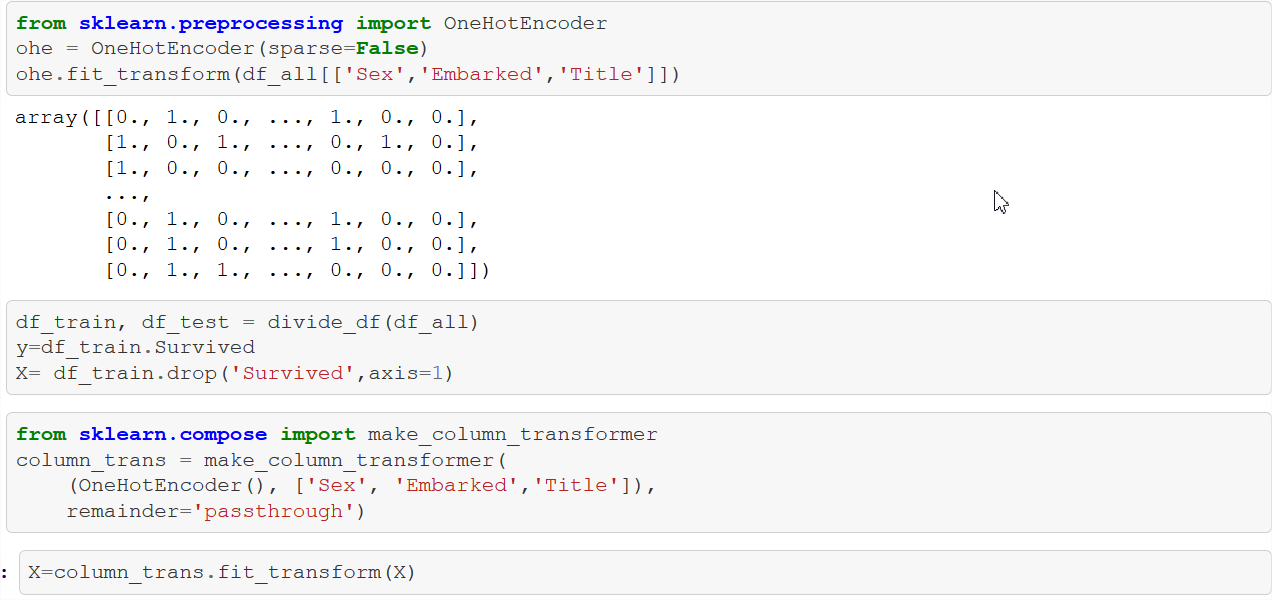


Sau khi đã đầy đủ, drop tất cả các column không cần thiết.



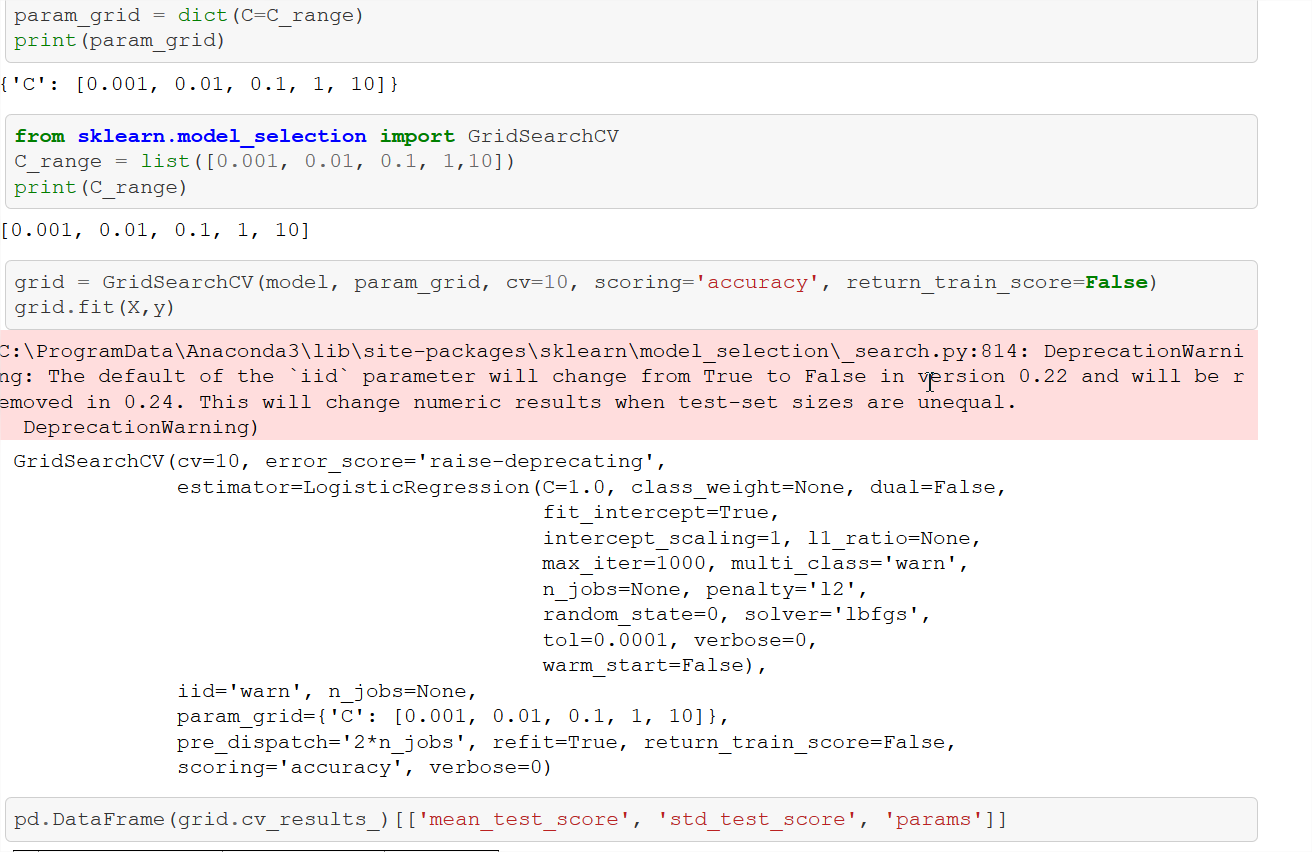
**3.3. OneHotEncoder cho data:**

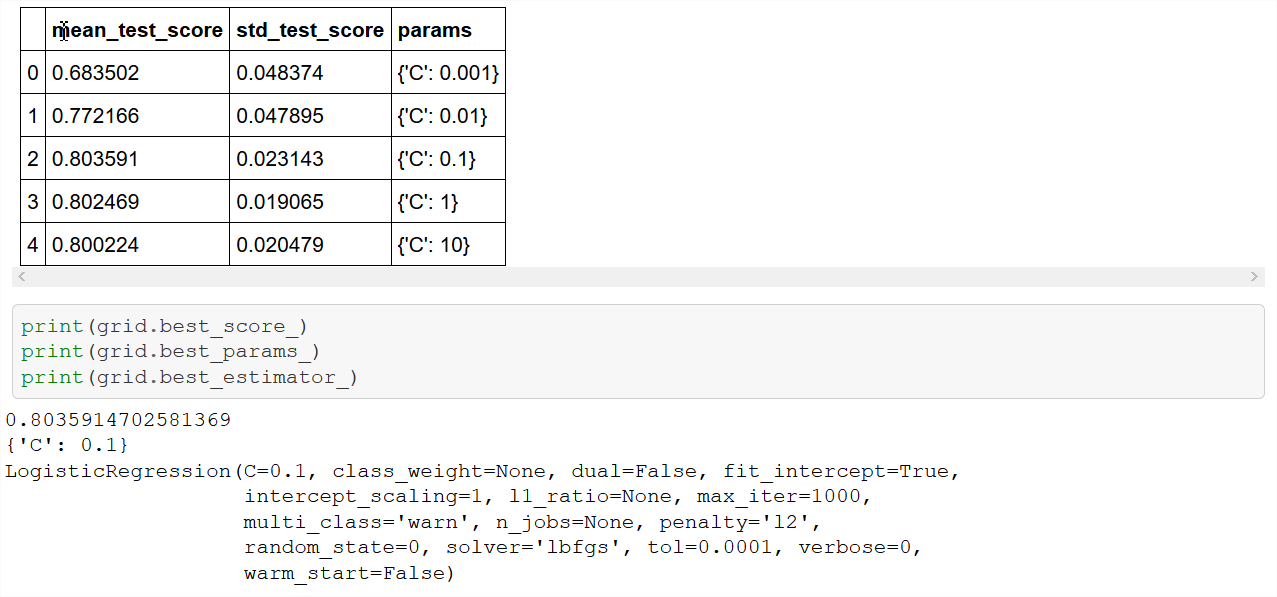
Tiến hành OneHotEncoder các feature Sex, Embarked, Tittle:



1. **Parameter tuning using gridCV:**

Tiến hành GridSearchCV để tuning parameter cụ thể ở đây là C.





Từ đó chọn được parameter C là 0.1.

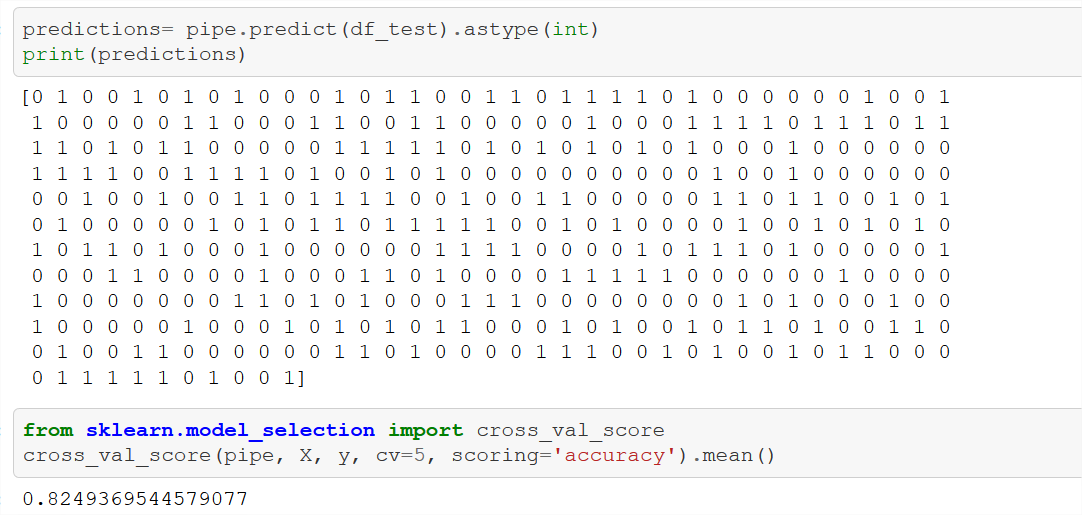
1. **Xây dựng model bằng pipeline:**

Thực hiện tạo pipeline.

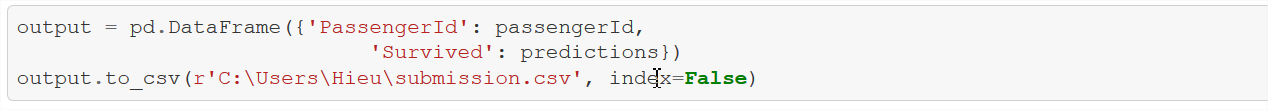


1. **Training model, predict và kết quả đạt được:**

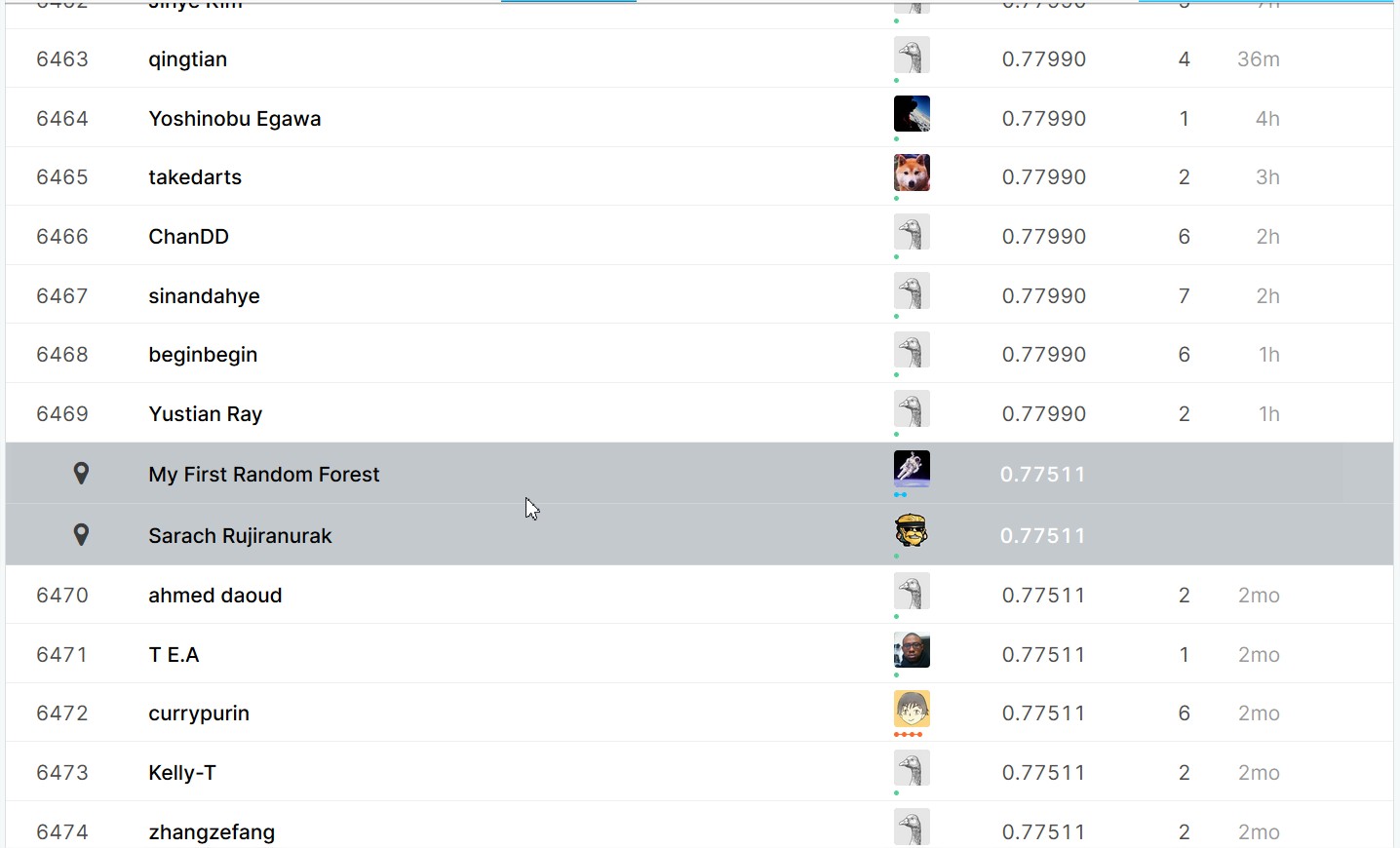
Tiến hành train data bằng method fit()



Dự đoán kết quả trên test set và thử nghiêm trên test validation score



Xuất file:



Chúng ta được 0.77511% chính xác, tương đương vị trí ~6500 trên leaderboard. Một kết quả tạm khả quan.

1. **Tài liệu tham khảo:**

* Channel Data School: <https://www.youtube.com/user/dataschool>
* Machine learning by Andrew Ng: <https://www.coursera.org/learn/machine-learning>
* https://scikit-learn.org/stable/