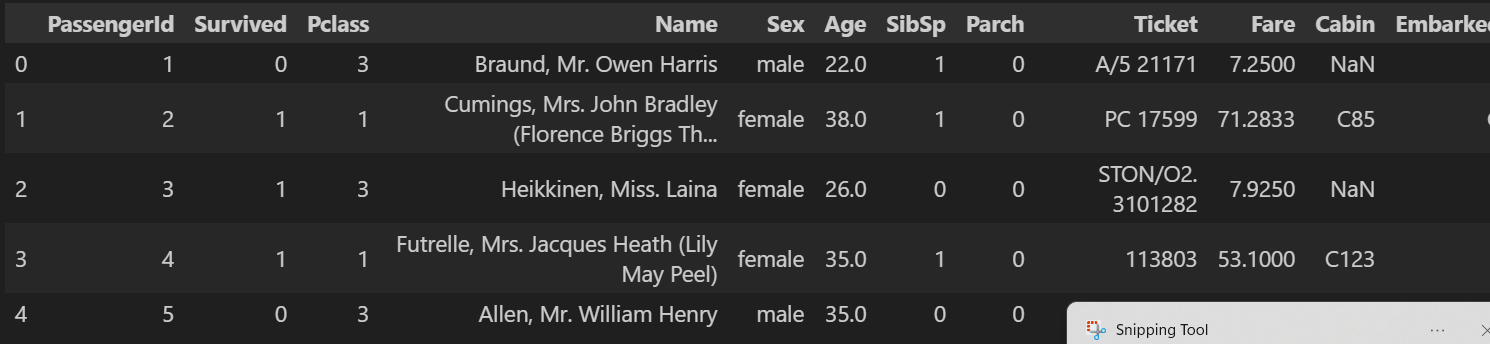
**BÁO CÁO KẾT QUẢ TRAIN MODEL**

Họ và tên: Trương Văn Thiện

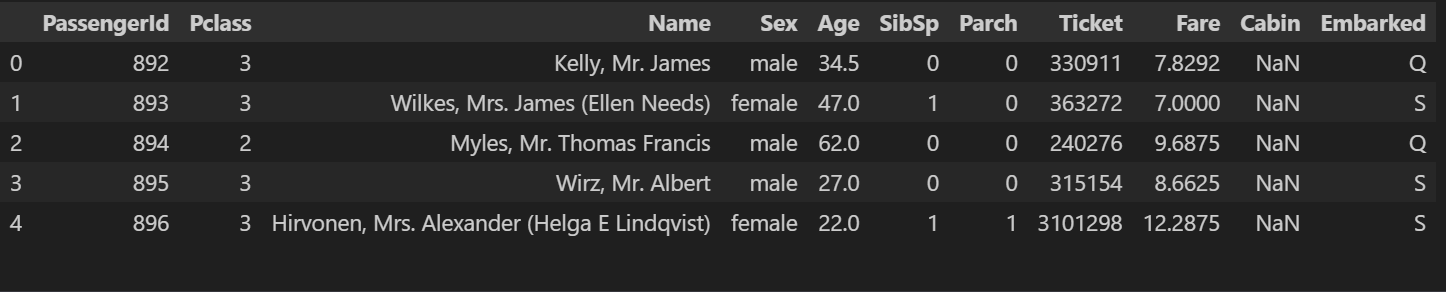
Mssv: 3123410356

Bước 1: load dữ liệu huấn luyện và dữ liệu test modle titanic sử dụng hàm read\_csv(file\_name). Sau khi load in ra 5 dòng dữ liệu để xem cấu trúc của file dữ liệu sẽ có dạng.

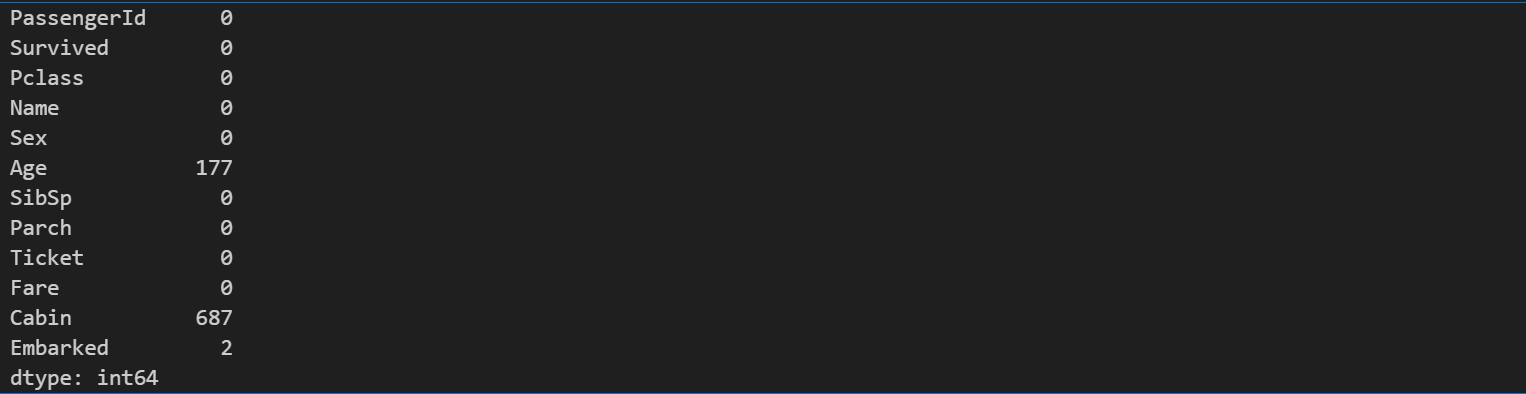
Dữ liệu train:



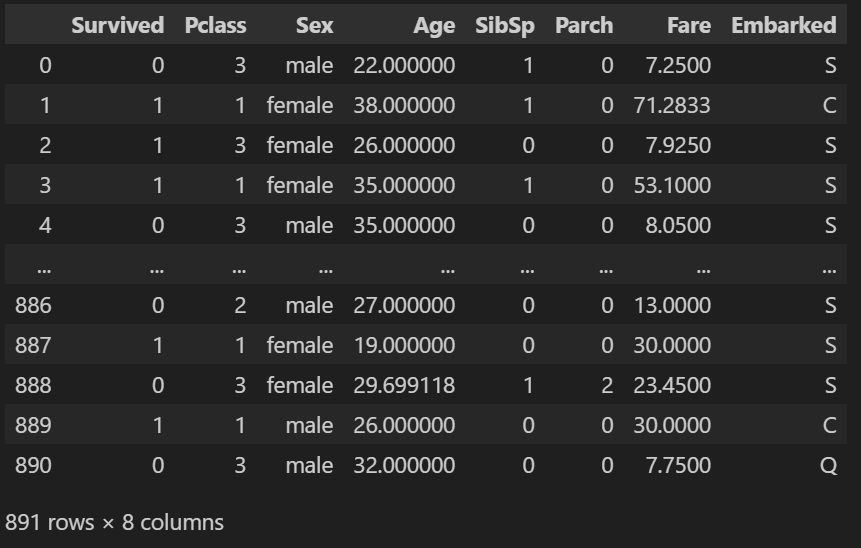
Dữ liệu test:



Bước 2: Kiểm tra có bao nhiêu dòng dữ liệu bị thiếu (NaN) để xử lí, sử dụng hàm isnull().sum().



Bước 3. Tiến hành tiền xử lí dữ liệu bổ sung dữ liệu vào các dòng bị thiếu và loại bỏ các cột không cần thiết để train mô hình như (passengerId,name,..).Sau khi chạy dữ liệu sẽ có dạng:

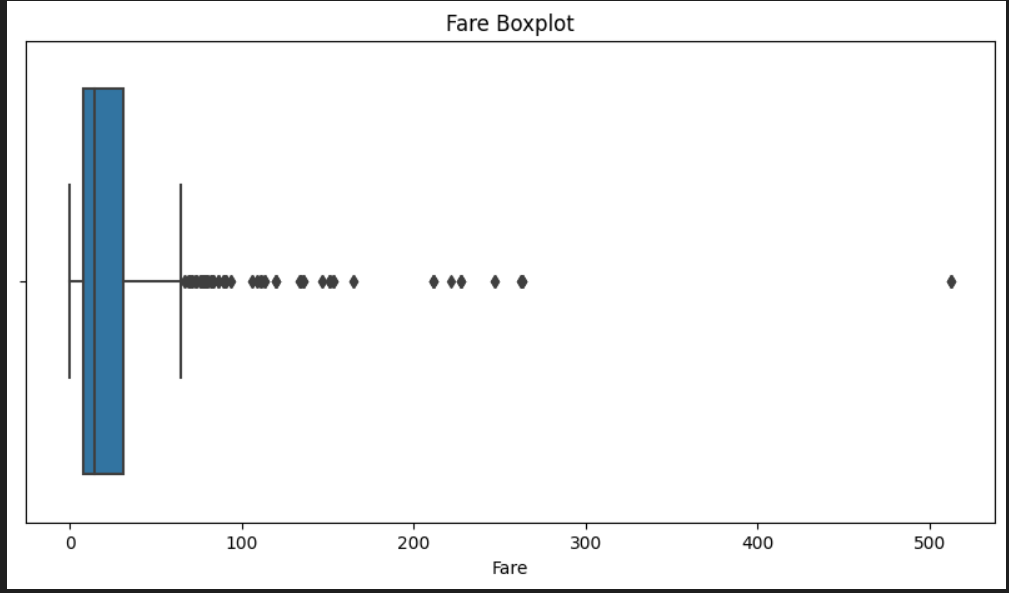


dữ liệu test làm tương tự.

Bước 4: kiểm tra các cột có kiểu là object để tiến hành mã hóa dữ liệu những cột này thành dạng dữ liệu nhị phân 0/1(binary). Sau khi thực hiện song bước này thì sẽ không còn cột nào có kiểu là object trong dữ liệu train nữa.

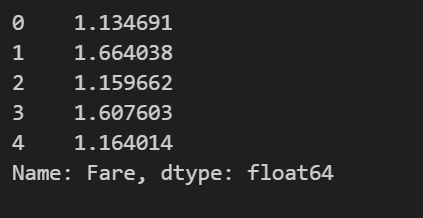


Bước 5: vẽ biểu đồ để xem mức phân bổ của giá vé đối với từng khách hàng sau khi chạy sẽ có kết quả như sau:

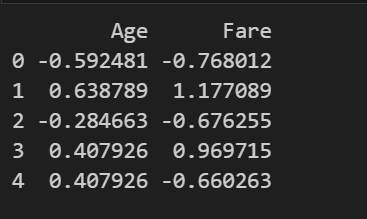


Nhìn vào biểu đồ ta thấy. Đa số giá vé của mỗi khách hàng là thấp chỉ có một vài vé cao

Bước 6: Sử dụng hàm logarit để nén các giá trị lớn của cột Fare, phân phối dữ liệu trở nên gần với phân phối chuẩn (ít lệch hơn). Sau khi chạy các giá trị của cột fare :



Bước 7: chuẩn hóa dữ liệu các cột Age, Fare về phân phối có mean = 0, std = 1. Sau khi chạy dữ liệu cột Age và Fare sẽ có dạng:



Bước 8: Tách tập dữ liệu thành các tập huấn luyện train, val. Trong đó 20% dữ liệu nằm trong tập val, 80% còn lại nằm trong tập train.Sau khi thực hiện bước này ta sẽ có các tập dữ liệu x\_train , x\_val , y\_train , y\_val.

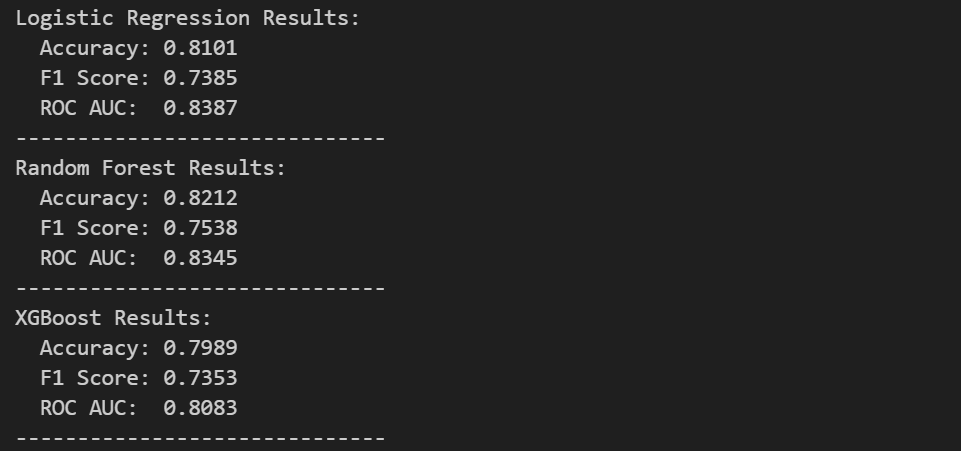
Bước 9: Tiến hành train modle với 3 loại mô hình phổ biến

* LogisticRegression: mô hình tuyến tính baseLine cơ bản
* RanDomForestClassifier: mô hình esemble dựa trên nhiêu cây quyết định
* XGBClassifier: mô hình boosting mạnh

Sau khi thực hiện in ra 3 thông số để đánh giá các mô hình:

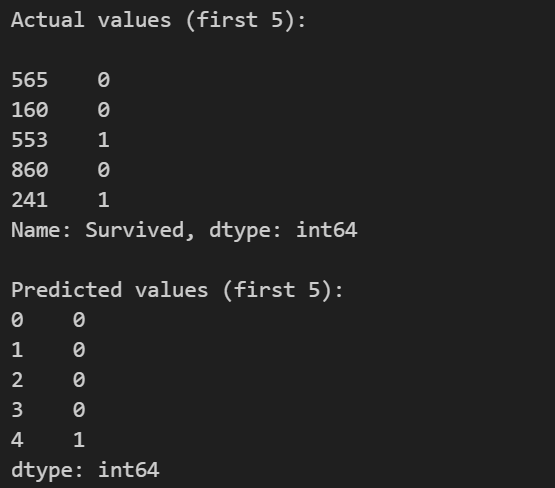
* Accuracy: tỷ lệ dự đoán đúng
* F1 score: mức độ trung hòa precision & recall
* ROC AUC: đo mức độ phân biệt class(0/1) càng gần 1 càng tốt

Kết quả sau khi train 3 mô hình:



Nhìn vào kết quả ta thấy các model không chênh nhau là mấy.

Bước 10: chọn một mô hình trong 3 mô hình ở trên để tiến hành dự đoán tập y\_val so sánh với tập giá trị ban đầu của chúng. Kết quả sau khi thực hiện :



Nhìn vào kết quả ta thấy kết quả dự đoán của model so với dữ liệu cũ gần giống nhau hoàn toàn

Bước 11: Dự đoán tập dữ liệu test với mô hình trên và lưu vào file submission.csv chúng ta chỉ lưu 2 cột giá trị đó là PassengerId và Survived. Kết quả cuối cùng của quá trình train là chúng ta sẽ có một file submission.csv dự đoán các hành khách trong file test là sống hay chết.

Bước 12: nộp file submission đã dự đoán lên trang kaggle.com để chấm điểm kết quả

