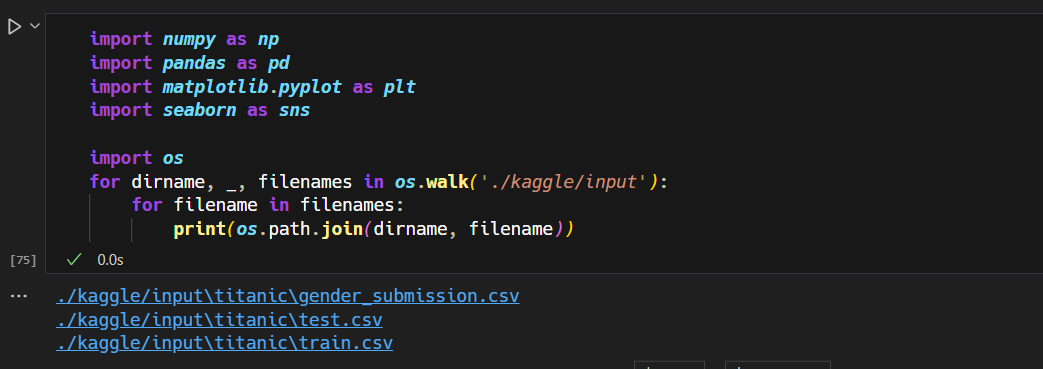
Tên: Đỗ Quốc Việt

MSSV: 3123410426

**BÁO CÁO TUẦN NHẬP MÔN MÁY HỌC**

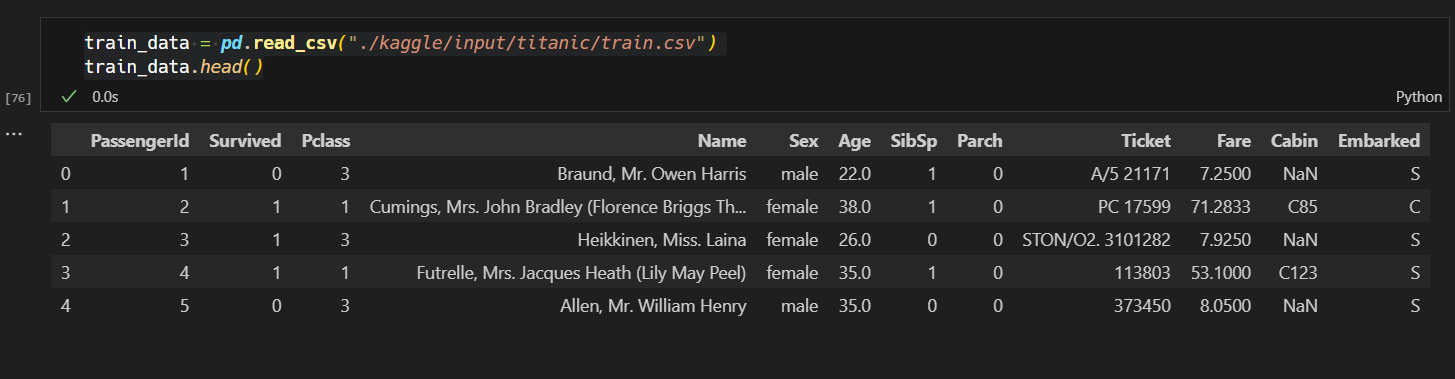
**CHỦ ĐỀ: Titanic - Machine Learning from Disaster**

1. **Tìm các file dataset trong thư mục kaggle/input.**

****

Hàm walk của thư viện os liệt kê tất cả các thư mục con bên trong ./kaggle/input, trả về đường dẫn hiện tại (dirname), danh sách thư mục con (\_) và danh sách file trong thư mục đó (filenames). Tiếp theo, vòng lặp for liệt kê từng file trong danh sách file. Cuối cùng, in ra đường dẫn đến file bằng cách nối dirname và filename.

1. **Đọc file dataset huấn luyện, test và lấy ra 5 dòng đầu.**

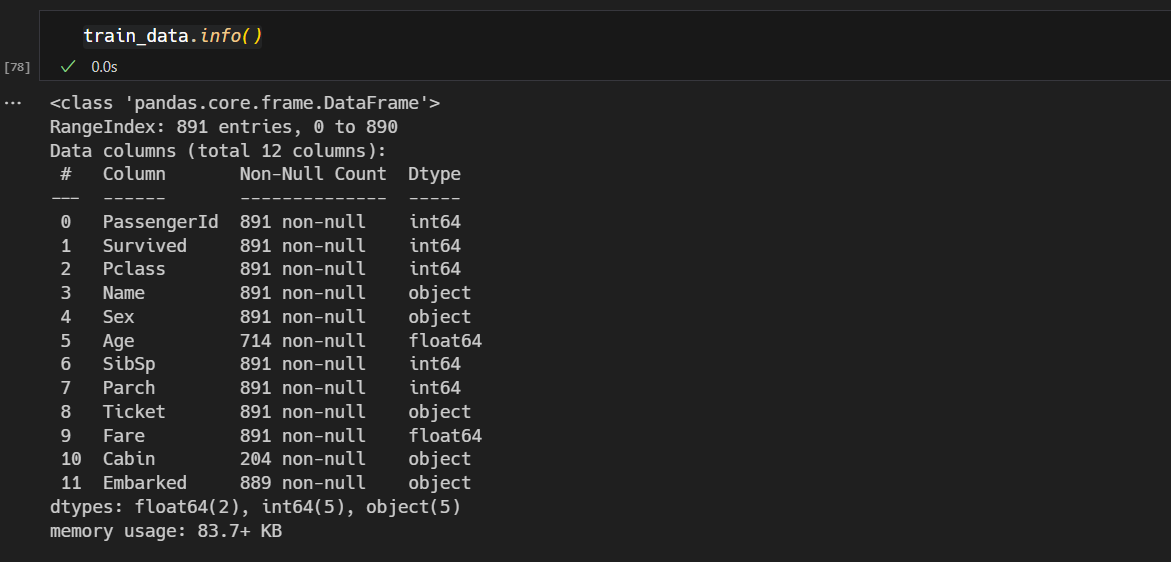
****

**A screenshot of a computer

AI-generated content may be incorrect.**

Dùng read\_csv để đọc file dữ liệu csv của các dataset và truyền vào từng đối tượng. Phương thức head có tác dụng lấy ra n dòng đầu tiên của dataset, phương thức lấy mặc định 5 dòng đầu tiên nếu không truyền tham số n.

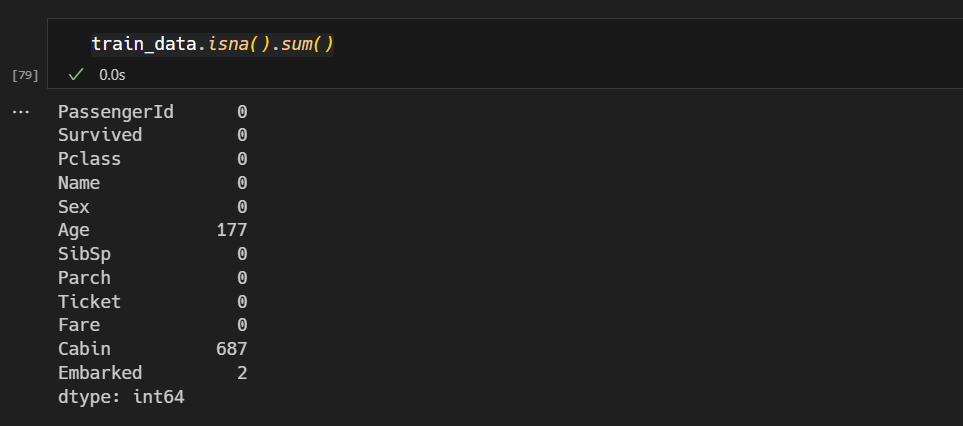
1. **Liệt kê các thông tin dữ liệu bên trong dataset huấn luyện.**

****

Phương thức info in ra các thông tin tổng quan của dataset huấn luyện bao gồm:

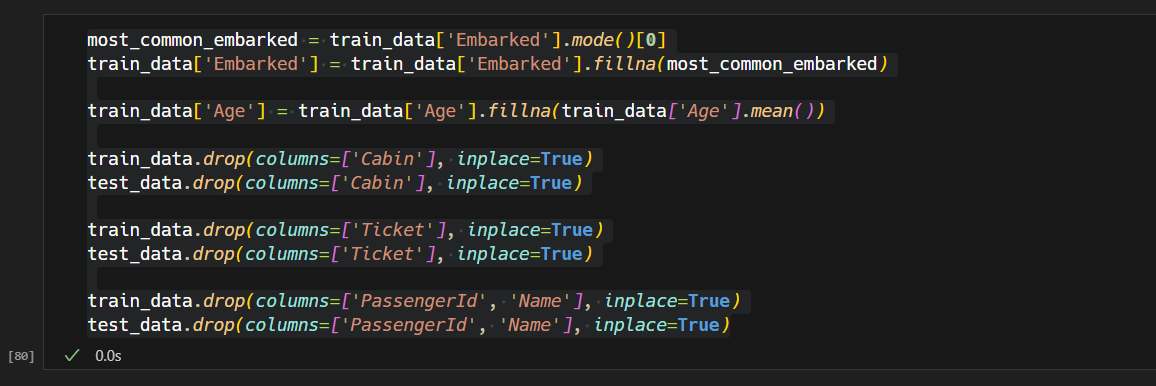
* Tổng số hàng (891), số cột (12).
* Tên cột và kiểu dữ liệu cột (int64, float64, object).
* Số lượng giá trị không trống (non-null) của từng cột.
* Dung lượng bộ nhớ mà dataset chiếm (83.7+ KB).

1. **Liệt kê tổng số ô bị trống trên từng cột của dataset huấn luyện.**

****

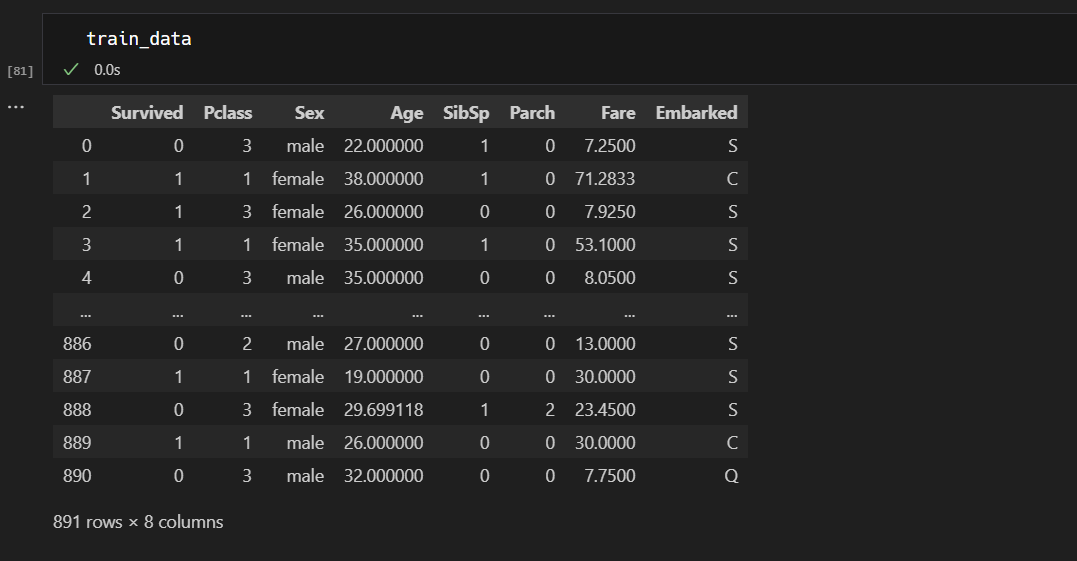
Phương thức isna() kiểm tra trên mỗi cột nếu dữ liệu của ô đó bị thiếu dữ liệu thì trả về True. Phương thức sum() được áp dụng trên giá trị trả về của isna() và cộng thêm vào khi trả về True. Kết quả trả về là tổng số ô bị trống trên từng cột của dataset.

1. **Xử lý các ô bị trống trên dataset.**

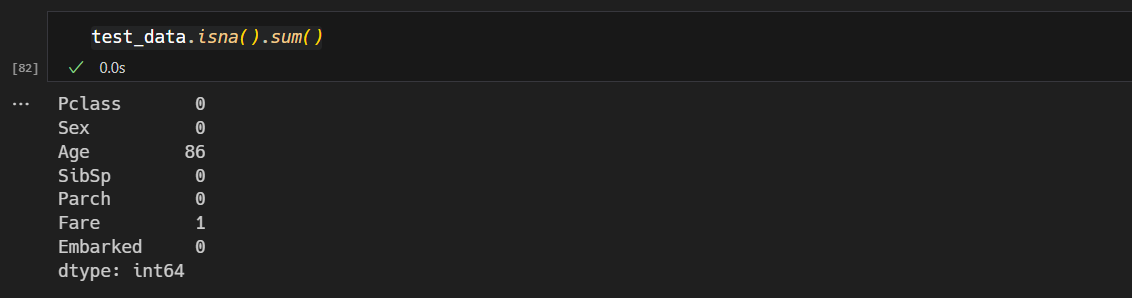
****

Ở cột Embarked, phương thức mode lấy giá trị xuất hiện nhiều nhất trong cột, vị trí 0 chọn ra phần tử đầu tiên của danh sách mode, lấy giá trị này cập nhật vào những ô bị trống. Giá trị trung bình của cột Age được tính và thêm vào những ô còn trống trong cột Age. Sau đó, loại bỏ các cột Cabin, Ticket và PassengerID do các cột này không hữu ích, khó xử lý.

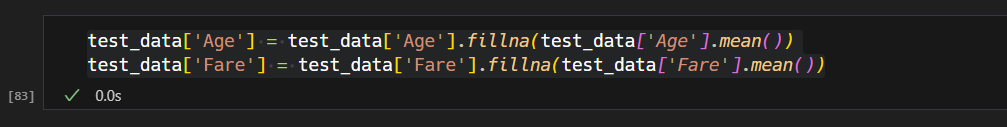
1. **In ra dataset huấn luyện sau khi xử lý các ô bị trống.**

****

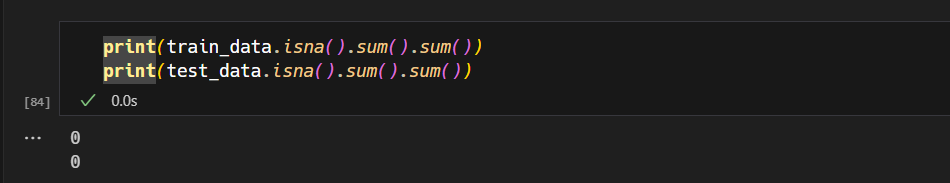
1. **Liệt kê tổng các giá trị bị trống trên từng cột của dataset test.**



1. **Xử lý các giá trị bị trống trên dataset thử nghiệm.**



1. **Kiểm tra dữ liệu không còn ô trống trên các dataset.**



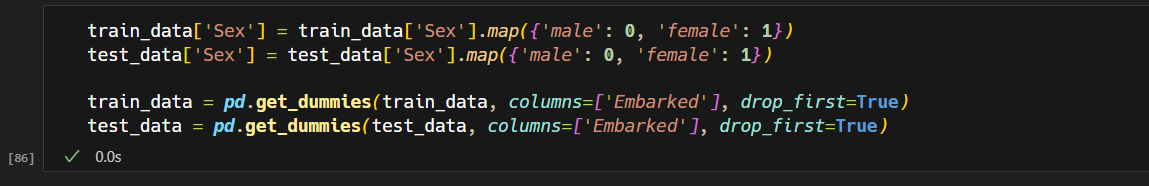
Sử dụng phương thức sum().sum() để đếm theo từng cột và theo từng hàng của các dataset. Điều này kiểm tra từng ô trong dataset để bảo đảm mỗi ô đều có dữ liệu.

1. **Lấy ra những cột có dạng object trong dataset huấn luyện.**



Dùng phương thức select\_dtypes(include=[‘object’].columns) để lấy ra những cột có dạng object (ký tự, chuỗi,…). Dạng object không thể được xử lý trực tiếp nên cần phải được chuyển về dạng số.

1. **Đưa dữ liệu những cột có dạng object về dạng số nhị phân.**



Đầu tiên, Các giá trị ‘male’ và ‘female’ ở cột Sex được lần lượt đưa về dạng nhị phân là 0 và 1.

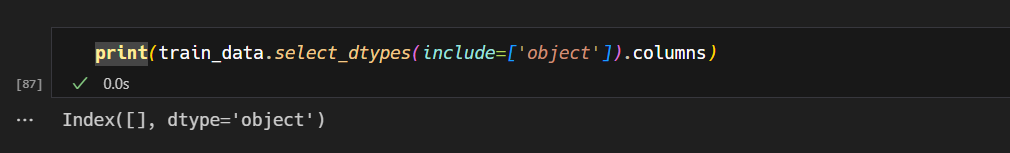
Tiếp theo, cột Embarked được biến đổi thành các biến giả (dummy variables), giả sử cột Embarked chứa 3 trạng thái S, C, Q thì 3 cột mới (Embarked\_C, Embarked\_Q, Embarked\_S) chứa giá trị nhị phân có dạng:

|  |  |  |  |
| --- | --- | --- | --- |
| Embarked | Embarked\_C | Embarked\_Q | Embarked\_S |
| S | 0 | 0 | 1 |
| C | 1 | 0 | 0 |
| Q | 0 | 1 | 0 |
| S | 0 | 0 | 1 |

Tuy nhiên khi cộng lại mỗi cột thì giá trị luôn bằng 1, gây ra hiện tượng đa cộng tuyến (dummy variable trap). Điều này làm cho ma trận thiết kế trong các mô hình hồi quy tuyến tính trở nên không khả nghịch, gây khó khan cho việc ước lượng. Để tránh điều này, dùng cờ drop\_first=True để loại bỏ đi 1 cột, giữ lại 2 cột (giả sử cột C bị loại bỏ):

|  |  |  |
| --- | --- | --- |
| Embarked | Embarked\_Q | Embarked\_S |
| S | 0 | 1 |
| C | 0 | 0 |
| Q | 1 | 0 |
| S | 0 | 1 |

1. **Kiểm tra dataset huấn luyện không còn dữ liệu dạng object.**



1. **Vẽ Boxplot cho cột Fare của dataset huấn luyện.**

A screenshot of a computer

AI-generated content may be incorrect.

Thông qua đồ thị Fare Boxplot, dữ liệu cột Fare có xu hướng tập trung ở mức thấp (0 – 100), có nhiều giá trị ngoại lai nằm rải rác ở phía bên phải với khoảng cách xa. Điều này cho thấy dữ liệu có xu hướng lệch phải.

1. **Biến đổi dữ liệu cột Fare bằng cách sử dụng log transformation.**

A screen shot of a computer code

AI-generated content may be incorrect.

Do đồ thị có xu hướng lệch phải, dùng np.log1p để đưa về dạng gần chuẩn để giúp mô hình máy học hoạt động hiệu quả hơn. np.log1p an toàn hơn so với np.logp với dữ liệu nhỏ (tránh lỗi log(0)).

1. **Chuẩn hóa dữ liệu cột Age và Fare.**

A screen shot of a computer code

AI-generated content may be incorrect.

StandardScaler() chuẩn hóa dữ liệu về phân phối có trung bình = 0 và độ lệch chuẩn = 1 theo công thức:

Với x là giá trị gốc, µ là giá trị trung bình của cột, σ là độ lệch chuẩn.

fit\_transform (dùng cho huấn luyện): tính toán μ và σ từ tập huấn luyện, áp dụng công thức chuẩn hóa cho dataset huấn luyện.

transform (dùng cho test): Chỉ áp dụng μ và σ đã học từ dataset huấn luyện, không tính lại từ test để tránh rò rỉ dữ liệu (data leakage).

1. **Chia dữ liệu huấn luyện/validation.**

A screen shot of a computer code

AI-generated content may be incorrect.

Trong hàm train\_test\_split của scikit-learn, test\_size = 0.2 dùng để tách 20% dữ liệu để validate, 80% dữ liệu để huấn luyện. Dùng random\_state = 42 để đảm bảo lần nào chạy cũng ra kết quả chia giống nhau (tái thiết lập kết quả). Cờ hiệu stratify=y đảm bảo tỷ lệ class (0 và 1 trong Survived) ở dữ liệu huấn luyện/validation giữ nguyên như ban đầu, tránh tình huống lệch nhãn.

1. **Huấn luyện và so sánh trên nhiều mô hình máy học.**

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Các model được sử dụng cho việc huấn luyện:

* LogisticRegression: mô hình tuyến tính cơ bản.
* RandomForestClassifier: mô hình cây ngẫu nhiên.
* XGBClassifier: mô hình gradient boosting.

Các tham số đánh giá:

* accuracy\_score: độ chính xác.
* f1\_score: trung bình điều hòa giữa độ chính xác và tái thiết lập.
* roc\_auc\_score: đánh giá khả năng phân biệt giữa 2 lớp.

Thông qua các tham số đánh giá trên, model RandomForestClassifier cho kết quả tốt nhất trên hầu hết các tham số đánh giá độ chính xác.

1. **Kiểm chứng dự đoán của mô hình máy học.**

A screen shot of a computer program

AI-generated content may be incorrect.

Dùng mô hình RandomForestClassifier cho việc dự đoán và so sánh 5 giá trị đầu trên tập validation, mô hình dự đoán đúng 80% (4/5) số mẫu.

1. **Tạo ra file submission.**

A screen shot of a computer program

AI-generated content may be incorrect.

Tải lại file test gốc từ Kaggle do trong quá trình xử lý dữ liệu các cột Cabin, Ticket và PassengerID đã bị xóa nên cần phải tải lại file này để lưu đúng định dạng với Kaggle. Sử dụng best\_model để dự đoán trên test data gốc của Kaggle, lưu kết quả vào test\_preds. Tiếp đến, tạo DataFrame submission với đúng format của Kaggle yêu cầu (gồm 2 cột PassengerId và Survived). Cuối cùng, gọi phương thức to\_csv để xuất ra file submission.csv để Kaggle chấm điểm.

1. **Submit bài làm lên Kaggle**

**A screenshot of a computer

AI-generated content may be incorrect.**

Submit lên Kaggle để chấm điểm, xem kết quả và vị trí xếp hạng ở trong Leaderboard.