**Họ và tên:** Nguyễn Đăng Khoa

**MSSV :** 3123410169

**Báo cáo Lab\_3: Bài toán Titanic**

# Giới thiệu (Introduction)

## Bối cảnh và cơ sở dữ liệu Titanic

Vụ đắm tàu Titanic năm 1912 đã trở thành một trong những thảm họa hàng hải lớn nhất vào đầu thế kỉ XX cũng như trong lịch sử, gây ra cái chết của hơn 1500 hành khách và thủy thủ trên tàu. Đáng chú ý, khả năng sống sót của các hành khách không phải ngẫu nhiên mà chịu ảnh hưởng từ nhiều yếu tố như độ tuổi, giới tính, hạng vé, ... Bộ dữ liệu về Titanic được công bố trên Kaggle đã trở thành một trong những bộ dữ liệu kinh điển, thường được sử dụng thực hành các kỹ thuật học máy và phân tích dữ liệu.

## Mục tiêu nghiên cứu và ý nghĩa thực tiễn

Mục tiêu của nghiên cứu này là có thể xây dựng được một mô hình máy học có khả năng dự đoán xác xuất sống sót của một hành khách dựa vào các đặc trưng được cung cấp. Ý nghĩa bài toán này không chỉ dừng lại ở việc minh họa khả năng máy học trong phân loại nhị phân, mà còn giúp người học nắm vững quy trình chuẩn: từ tiền xử lý dữ liệu, lựa chọn đặc trưng, huấn luyện mô hình, đánh giá hiệu năng, cho tới triển khai dự đoán trên dữ liệu mới.

## Phát biểu bài toán học máy

Bài toán Titanic có thể được phát biểu như bài toán phân loại nhị phân trong máy học. Với đầu vào là 1 vector X = {x1, x2, x3, ..., xn} mô tả các thông tin cơ bản của một hành khách (tuổi, giới tính, hạng vé, ...), mục tiêu cần mô hình thực hiện là dự đoán đầu ra , trong đó:

* y = 0: hành khách không sống sót
* y = 1: hành khách sống sót.

Nhiệm vụ của bài toán máy học là có thể tìm ra một hàm có khả năng dự đoán chính xác y cho những hành khách chưa từng được quan sát.

# Tập dữ liệu và mô tả (Dataset Description)

## Nguồn dữ liệu

Bộ dữ liệu sử dụng cho bài toán này được cung cấp bởi Kaggle trong cuộc thi “Titanic: Machine Learning from Disaster”. Dữ liệu gồm hai tệp chính:

* **train.csv** : dữ liệu dùng cho huấn luyện, cung cấp các thông tin của khách hàng và cột đánh giá Survied (sống sót = 1, chết = 0).
* **test.csv** : dữ liệu dùng để kiểm tra, tập dữ liệu này chỉ bao gồm thông tin của khách hàng mà không có nhãn Survied. Nhiệm vụ của chúng ta là dự đoán nhãn này.

Ngoài ra, còn có tệp **gender\_submission.csv**, đây là một ví dụ về format kết quả cần nộp.

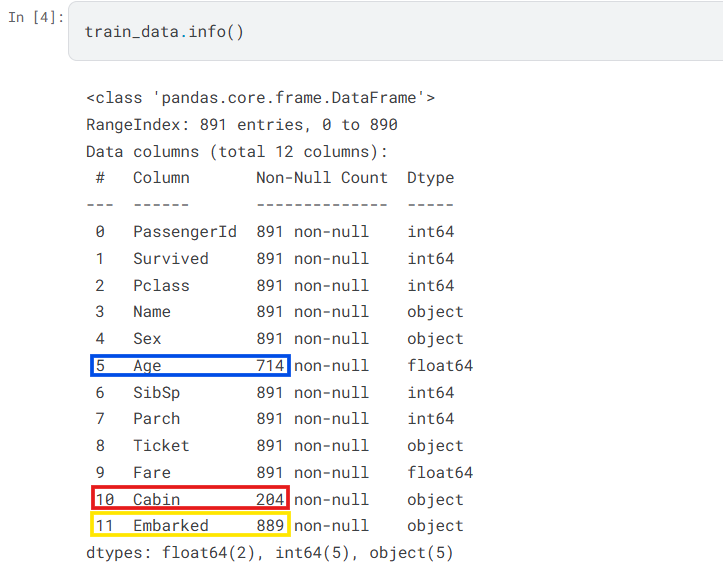
## Các thuộc tính trong bộ dữ liệu

Các cột trong dữ liệu huấn luyện gồm:

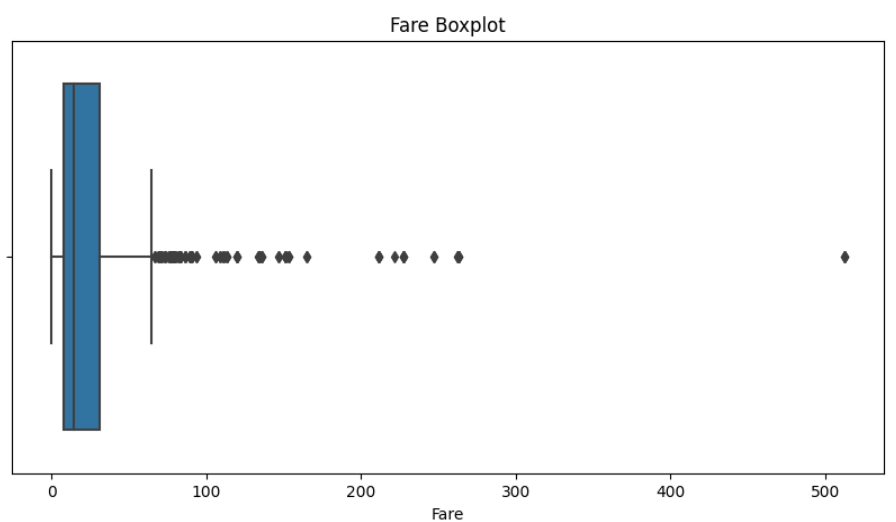
* *PassengerId*: số thứ tự hành khách (ID).
* *Survived*: biến mục tiêu (0 = chết, 1 = sống sót).
* *Pclass*: hạng vé (1, 2 hoặc 3).
* *Name*: tên hành khách.
* *Sex*: giới tính.
* *Age*: tuổi.
* *SibSp*: số lượng anh/chị/em hoặc vợ/chồng đi cùng.
* *Parch*: số lượng cha/mẹ hoặc con đi cùng.
* *Ticket*: số vé.
* *Fare*: giá vé.
* *Cabin:* số phòng.
* *Embarked***:** cảng lên tàu (C = Cherbourg, Q = Queenstown, S = Southampton).

## Phân tích sơ bộ dữ liệu

Khi xem qua dữ liệu sơ bộ ban đầu bằng ***train\_data.info()*** và ***test\_data.info()***, ta có thể nhận thấy ở một số cột vẫn còn thiếu thông tin, cụ thể là cột *Age*, *Cabin* và *Embarked.*



Ngoài ra, nếu vẽ biểu đồ *boxplot* cho cột *Fare*, ta thấy dữ liệu có nhiều điểm ngoại lai (outliers). Điều này cho thấy cần biến đổi (ví dụ lấy log) để dữ liệu *“cân đối”* hơn cho mô hình.



# Tiền xử lý dữ liệu (Data Preprocessing)

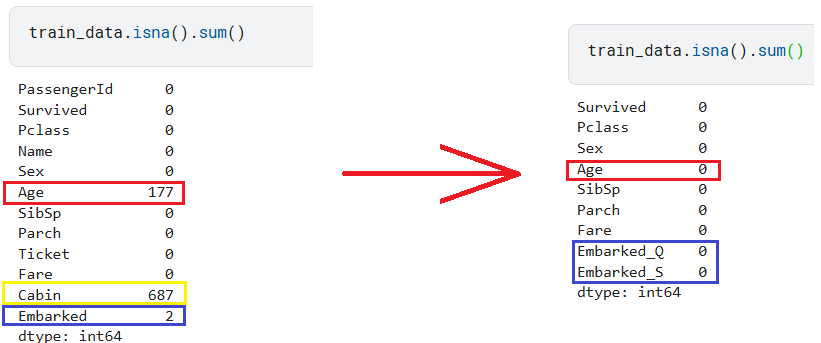
Trước khi bắt đầu huấn luyện mô hình, ta cần phải làm sạch và biến đổi dữ liệu để dễ dàng cho việc huấn luyện:

* Xử lý dữ liệu đang bị thiếu:

- Điền giá trị phổ biến nhất cho các vị trí trống trong cột *Embarked.*

*-* Điền giá trị trung bình của cột *Age* vào các ô còn thiếu.

- Cột *Cabin* còn thiếu quá nhiều dữ liệu nên bị loại bỏ.



- Nhận thấy cột *Fare* trong *test.csv* còn thiếu dữ liệu nên điền giá trị trung bình vào các ô này.

* Loại bỏ những cột không cần thiết:

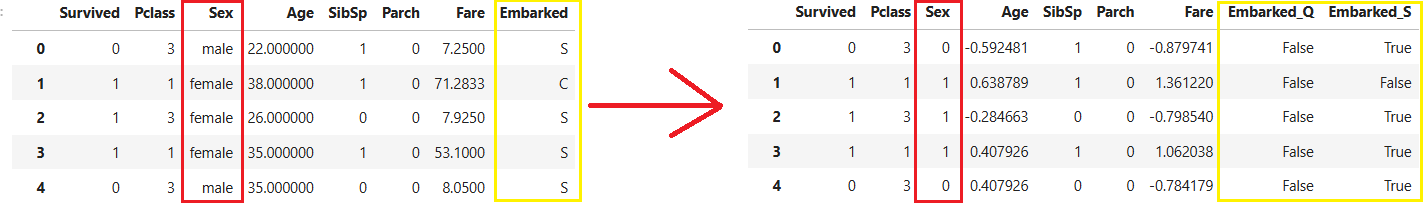
*PassengerId, Name, Ticket* không mang lại quá nhiều ý nghĩa cho mô hình nên bị loại bỏ.



* Mã hóa các biến phân loại:

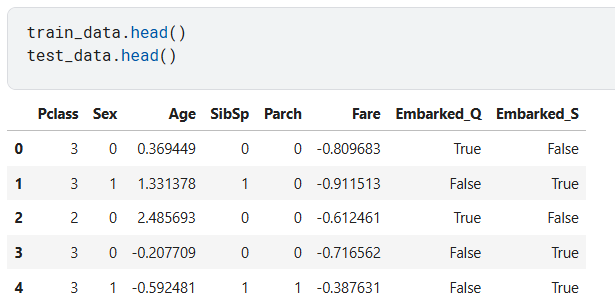
- Thay đổi giá trị trong cột *Sex* thành: 0 là *male* và 1 là *female.*

- Cột *Emberked* sẽ dùng *one-hot encoding.*

**

* Chuẩn hóa dữ liệu số:

- Các giá trị của *Age* và *Fare* có sự chênh lệch về khoảng khá lớn sẽ làm cho việc huấn luyện bị chi phối không còn chính xác nên cần dùng *StandardScaler* để chuẩn hóa đưa về cùng thang đo.



# Phương pháp và mô hình hóa (Methodology and Models)

Trong bài toán này, ta sẽ thử với ba mô hình phổ biến để so sánh hiệu quả huấn luyện:

* ***Logistic Regression:***

Mô hình cơ bản cho phân loại nhị phân. Dễ triển khai, chạy nhanh, thường được dùng làm baseline.

* ***Random Forest:***

Mô hình tập hợp nhiều cây quyết định (decision trees). Khả năng dự đoán khá tốt, ít bị overfitting, phù hợp với dữ liệu có nhiều loại đặc trưng.

* ***XGBoost:***

Mô hình boosting mạnh mẽ, thường đạt kết quả cao trong nhiều bài toán Kaggle. Tuy nhiên cần tinh chỉnh nhiều siêu tham số hơn.

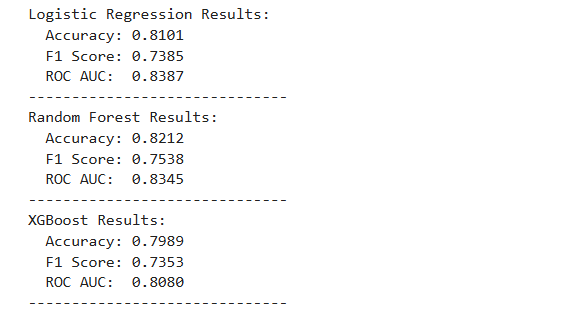
Các mô hình được huấn luyện trên tập train (80%) và kiểm tra trên tập validation (20%). Các thước đo để so sánh gồm *Accuracy*, *F1 Score*, *ROC-AUC*.

# Thực nghiệm và kết quả (Experiments and Results)

***Cách thực hiện:***

* Dữ liệu được chia thành 80% huấn luyện và 20% kiểm tra (*validation*).
* Mỗi mô hình được huấn luyện trên tập train và dự đoán trên tập validation.
* Kết quả được đánh giá bằng 3 chỉ số: *Accuracy*, *F1 Score*, *ROC-AUC*.

***Kết quả:***



***Đánh giá:***

* *Logictis*: *Accuracy* khá cao, *ROC AUC* có thể phân biệt sống/ chết khá tốt, *F1\_Score* cân bằng tốt giữa precision và recall.
* *Random Forest*: *Accuracy* và *F1\_Score* cao nhất trong 3 mô hình, *ROC AUC* thấp hơn *Logictis* một chút nhưng vẫn khá cao.
* *XGBoost*: *Accuracy* và *ROC AUC* thấp nhất trong 3 mô hình, tức là khả năng phân biệt chưa bằng 2 mô hình trên, *F1\_Score* cao hơn *Logictis* 1 chút nhưng vẫn kém *Random Forest*.

***Random Forest*** là mô hình tốt nhất trong 3 mô hình trên.

# Dự đoán và nộp kết quả (Prediction and Submission)

Sau khi so sánh ta chọn mô hình *Random Forest* làm mô hình cuối cùng vì đạt được kết quả cao nhất.

Sử dụng mô hình *Random Forest* để thực hiện dự đoán dữ liệu trong file *test.csv.*

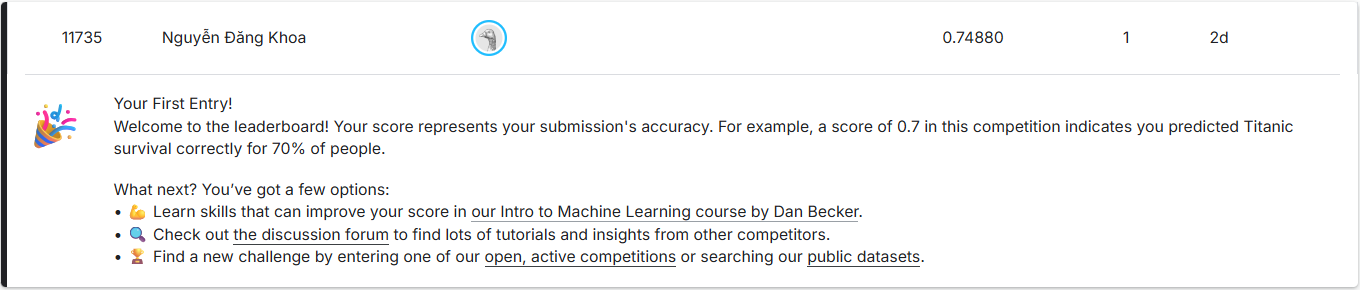
Kết quả được lưu thành file s*ubmission.csv* gồm:

- PassengerId.

- Survived (giá trị dự đoán).



Sau khi nộp bài lên *Kaggle* đã được hệ thống chấm điểm trực tiếp với điểm số và thứ hạng tương ứng.



# Kết luận và hướng phát triển (Conclusion and Future Work).

Trong bài toán Titanic, em đã áp dụng các bước cơ bản của học máy: tiền xử lý dữ liệu, chọn đặc trưng, huấn luyện và so sánh mô hình. Kết quả cho thấy *Random Fores*t cho độ chính xác cao nhất, được chọn làm mô hình cuối cùng để dự đoán.

Tuy nhiên, bài toán vẫn còn có thể cải thiện:

* Tạo thêm đặc trưng mới (ví dụ: nhóm gia đình, độ dài tên).
* Tối ưu siêu tham số (*Hyperparameter Tuning*).
* Kết hợp nhiều mô hình (*Ensemble/Stacking*).