Phân tích khả năng sống sót Titanic bằng các phương pháp học máy

GVHD: Bùi Tiến Đức

Hoa Toàn Hạc – 2201917

Mai Huy Hiệp – 2211045

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

Giới thiệu đề tài

- Mục tiêu

Dự đoán khả năng sống sót của hành khách dựa trên dữ liệu thu thập được từ Kaggle.

- Phương pháp nghiên cứu

Sử dụng các mô hình học máy phổ biến như Logistic Regression, Random Forest, XGBoost, LightGBM, SVC, MLP.

- Phạm vi nghiên cứu về khả năng sống sót hành khách Titanic.

Dữ liệu Titanic (891 mẫu train, 418 mẫu test), tập trung vào biến số chính như Pclass, Sex, Age, Fare, Embarked, SibSp, Parch,…

1.Giới thiệu về biến

1.1 Tổng quan về dữ liệu

- Dữ liệu Titanic từ Kaggle dùng để dự đoán khả năng sống sót của hành khách.
- Titanic là tàu chở khách sang trọng, bị chìm vào năm 1912 trong chuyến đầu tiên từ Southampton đến New York.
- Tai nạn làm hơn 1.500 người thiệt mạng trong khoảng 2.224 người trên tàu.
- Dữ liệu cung cấp các yếu tố ảnh hưởng đến khả năng sống sót.

1.1 Tổng quan về dữ liệu

Thông tin về bộ dữ liệu Titanic

Thuộc tính	Thông tin
Nguồn dữ liệu	train.csv, test.csv
Đối tượng quan sát	Hành khách trên tàu Titanic
Số lượng quan sát (train.csv)	891
Số lượng quan sát (test.csv)	418
Số lượng biến	12

1.2 Tổng quan về biến

Tên biến	Phân loại biến	Đơn vị	Mô tả
PassengerId	Định lượng (Nominal)	-	ID của hành khách
Survived	Định tính (Nominal)	-	Biến mục tiêu: 0 = Không sống sót 1 = Sống sót
Pclass	Định tính (Ordinal)	-	Hạng vé: 1 = Hạng nhất 2 = Hạng nhì 3 = Hạng ba
Name	Định tính (Nominal)	-	Tên của hành khách
Sex	Định tính (Nominal)	-	Giới tính của hành khách
Age	Định lượng (Continuous)	Năm	Tuổi của hành khách

1.2 Tổng quan về biến

Tên biến	Phân loại biến	Đơn vị	Mô tả
SibSp	Định lượng (Discrete)	-	Số lượng anh chị em/vợ chồng đi cùng trên tàu.
Parch	Định lượng (Discrete)	-	Số lượng cha mẹ/con cái đi cùng trên tàu.
Ticket	Định tính (Nominal)	-	Số vé của hành khách.
Fare	Định lượng (Continuous)	Đơn vị tiền tệ	Giá vé của hành khách.
Cabin	Định tính (Nominal)	-	Số cabin của hành khách.
Embarked	Định tính (Nominal)	-	Cảng lên tàu C = Cherbourg Q = Queenstown S = Southampton)

2. Khám phá dữ liệu (EDA)

2.1 Tổng quan về tập dữ liệu

- Bảng dữ liệu được phân tích từ train_df.info()

Cột	Kiểu dữ liệu	Số giá trị không thiếu
PassengerId	int64	891
Survived	int64	891
Pclass	int64	891
Name	object	891
Sex	object	891
Age	float64	714

2.1 Tổng quan về tập dữ liệu

Cột	Kiểu dữ liệu	Số giá trị không thiếu
SibSp	int64	891
Parch	int64	891
Ticket	object	891
Fare	float64	891
Cabin	object	204
Embarked	object	889

• Cột Age, Cabin, và Embarked có giá trị thiếu
Cabin có tỷ lệ thiếu cao nhất (chỉ 204/891 giá trị không thiếu).

• Cột Name, Sex, Ticket, Cabin, và Embarked là kiểu phân loại (object), trong khi các cột còn lại là kiểu số (int64 hoặc float64).

2.2 Phân tích thống kê

- Ta sử dụng phương pháp describe() để tính toán các thống kê mô tả

Thống kê	Passenger d	Survived	Pclass	Age	SibSp	Parch	Fare
Count	891	891	891	714	891	891	891
Mean	446.00	0.38	2.31	29.70	0.52	0.38	32.20
Std	257.35	0.49	0.84	14.53	1.10	0.81	49.69
Min	1	0	1	0.42	0	0	0.00
25%	223.50	0	2	20.12	0	0	7.91
50% (Median)	446.00	0	3	28.00	0	0	14.45
75%	668.50	1	3	38.00	1	0	31.00
Max	891	1	3	80.00	8	6	512.33

2.2 Phân tích thống kê

- **Survived:** Tỷ lệ sống sót chỉ 38%, dữ liệu mất cân bằng.
- **Pclass:** Chủ yếu là hạng 3 – tầng lớp kinh tế thấp.
- **Age:** Tuổi trung bình ~30, độ tuổi đa dạng từ trẻ sơ sinh đến người cao tuổi.
- **SibSp & Parch:** Phần lớn đi một mình hoặc ít người thân; có trường hợp đi cùng đông người.
- **Fare:** Giá vé lệch mạnh, phản ánh sự chênh lệch kinh tế rõ rệt giữa các hành khách.

2.3 Phân tích giá trị thiếu

- Cabin: Thiếu 77.1% → cần loại bỏ hoặc xử lý đặc biệt do khó khai thác trực tiếp.
- Age: Thiếu 19.87% → có thể điền bằng trung bình, trung vị hoặc dựa trên đặc trưng khác.
- Embarked: Thiếu 0.22% → điền giá trị phổ biến nhất (thường là "S").
- Các cột còn lại: Không có giá trị thiếu.

2.4 Nhận xét ban đầu

- Dữ liệu mất cân bằng: Tỷ lệ sống sót chỉ ~38%, cần áp dụng oversampling hoặc đánh giá bằng F1-score, AUC.
- Biến quan trọng: Sex, Pclass, Embarked có ảnh hưởng lớn đến khả năng sống sót.
- Giá trị thiếu:
- Cabin: Thiếu nhiều → nên loại hoặc dùng để tạo đặc trưng nhị phân.
- Age: Xử lý bằng trung bình, trung vị hoặc theo danh xưng (Title).

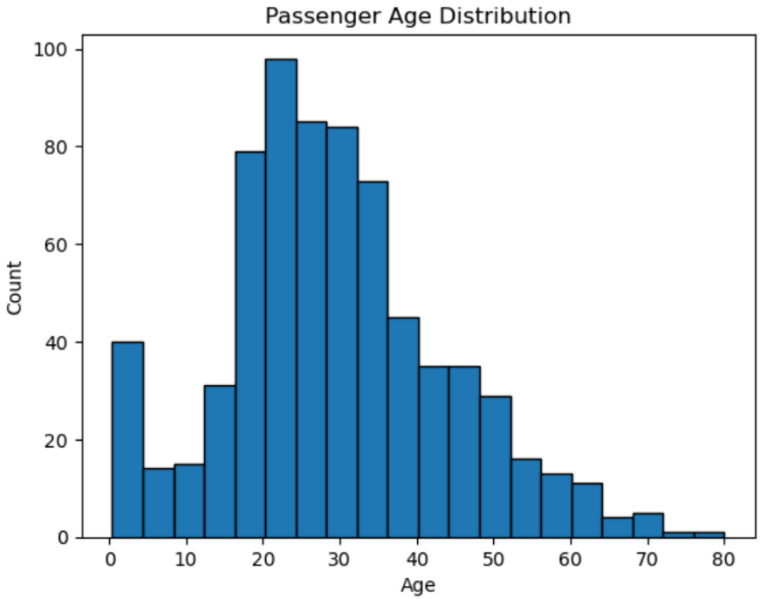
2.4 Nhận xét ban đầu

- Embarked: Có thể điền bằng giá trị phổ biến (S).
- Feature Engineering:
- Trích xuất danh xưng (Mr, Miss, Master…) từ Name.
- Nhóm Age, Fare thành nhóm tuổi, nhóm giá vé.
- Kết hợp SibSp + Parch → tạo đặc trưng mới FamilySize.
- Fare bị lệch: Cần chuẩn hóa (log) hoặc nhóm để mô hình học tốt hơn.

3. Data Visualization

3.1 Phân phối của Age

- Để phân tích phân phối độ tuổi của hành khách, ta sử dụng biểu đồ histogram



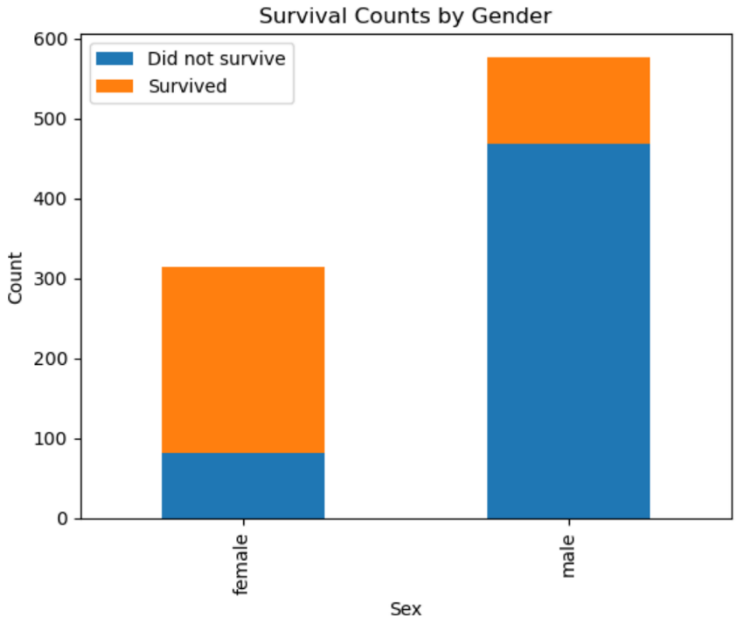
Hình 1: Histogram thể hiện phân phối của Age trong tập huấn luyện.

3.1 Phân phối của Age

- Tuổi phân bố lệch phải, tập trung chủ yếu ở nhóm 20–40 tuổi.
- Có một số trẻ em (<10 tuổi) và ít người cao tuổi (>60 tuổi), tuổi cao nhất khoảng 80.
- Gợi ý nên nhóm tuổi theo các giai đoạn (trẻ, thanh niên, người lớn, già) để tăng hiệu quả mô hình.
- Với 19.87% giá trị thiếu, Age cần được xử lý bằng trung bình hoặc dựa theo danh xưng (Title).

3.2 Tỷ lệ sống sót theo Sex

- Để đánh giá mối quan hệ giữa giới tính và khả năng sống sót, ta sử dụng biểu đồ cột phân loại



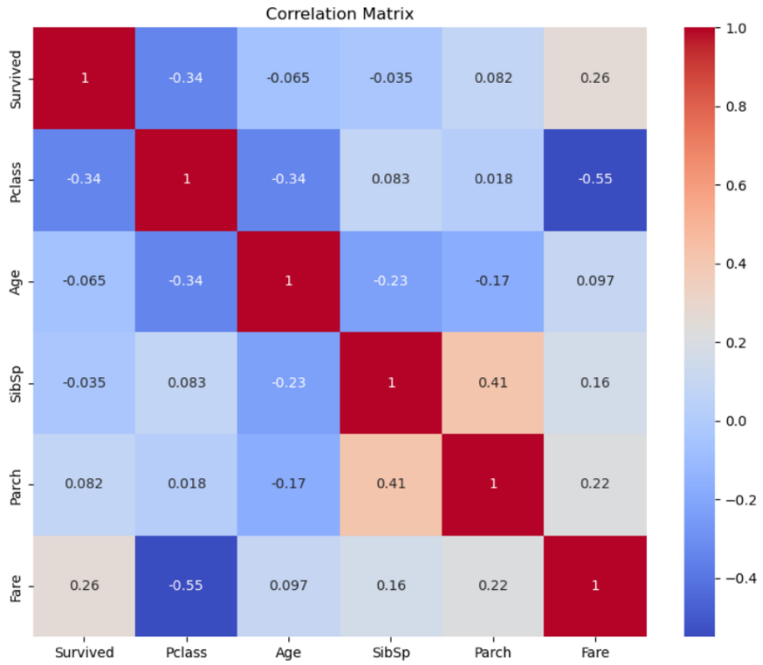
Hình 2: Biểu đồ cột thể hiện số lượng sống sót theo Sex trong tập huấn luyện.

3.2 Tỷ lệ sống sót theo Sex

- Phụ nữ có tỷ lệ sống sót cao hơn đáng kể so với nam giới.
- Phù hợp với lịch sử tàu Titanic – ưu tiên phụ nữ và trẻ em trong cứu hộ.
- Biến giới tính là yếu tố quan trọng trong dự đoán và sẽ được mã hóa bằng One-Hot Encoding trong tiền xử lý.

3.3 Ma trận tương quan

- Để đánh giá mối quan hệ tuyến tính giữa các biến số, ta sử dụng heatmap



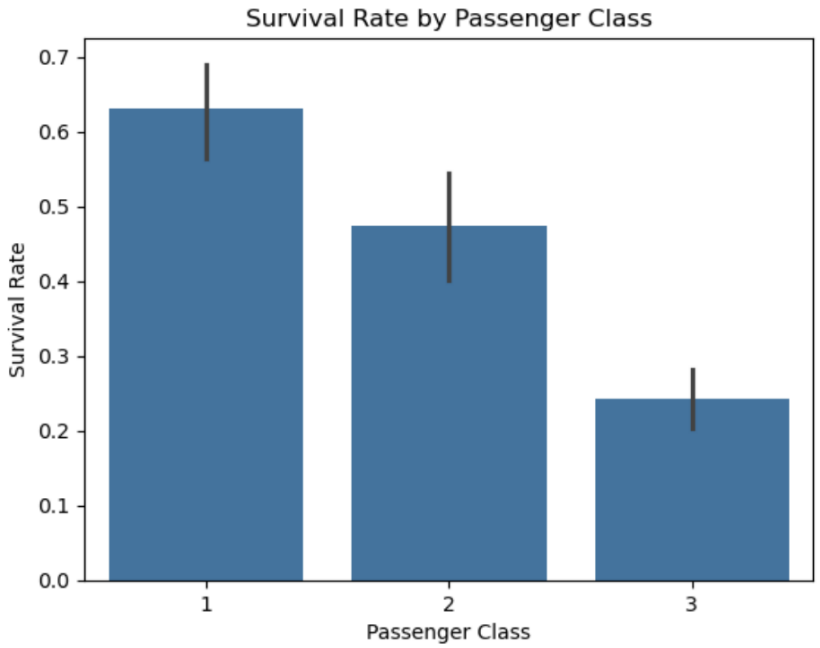
Hình 3: Heatmap thể hiện ma trận tương quan giữa các biến số trong tập huấn luyện.

3.3 Ma trận tương quan

- Survived & Pclass: Tương quan âm (-0.34) → hành khách ở hạng vé cao sống sót nhiều hơn.
- Survived & Fare: Tương quan dương (0.26) → vé càng đắt, tỷ lệ sống sót càng cao.
- Pclass & Fare: Tương quan âm mạnh (-0.55) → hạng vé cao đi kèm giá vé cao.
- SibSp & Parch: Tương quan dương (0.41) → thường đi cùng gia đình (family size).
- Age & Survived: Tương quan yếu (-0.065) → nên nhóm tuổi để mô hình hiểu rõ hơn.

3.4 Tỷ lệ sống sót theo Pclass

- Để phân tích tỷ lệ sống sót theo hạng vé, ta sử dụng biểu đồ cột



Hình 4: Biểu đồ cột thể hiện tỷ lệ sống sót theo Pclass trong tập huấn luyện.

3.4 Tỷ lệ sống sót theo Pclass

- Tỷ lệ sống sót giảm dần theo hạng:

  Hạng 1: ~63%

  Hạng 2: ~47%

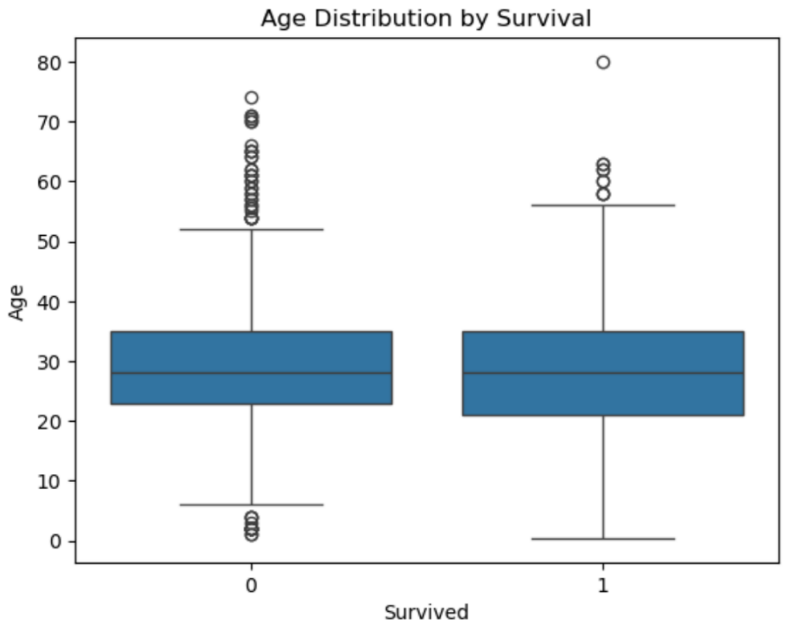
  Hạng 3: ~24%

- Phản ánh rõ mối liên hệ giữa điều kiện kinh tế và khả năng tiếp cận phương tiện cứu hộ.

- Củng cố kết quả từ phân tích tương quan rằng Pclass là đặc trưng quan trọng trong dự đoán.

3.5 Phân phối Age theo Survived

- Để đánh giá sự khác biệt trong phân phối độ tuổi giữa hai nhóm sống sót và không sống sót, ta sử dụng biểu đồ box plot



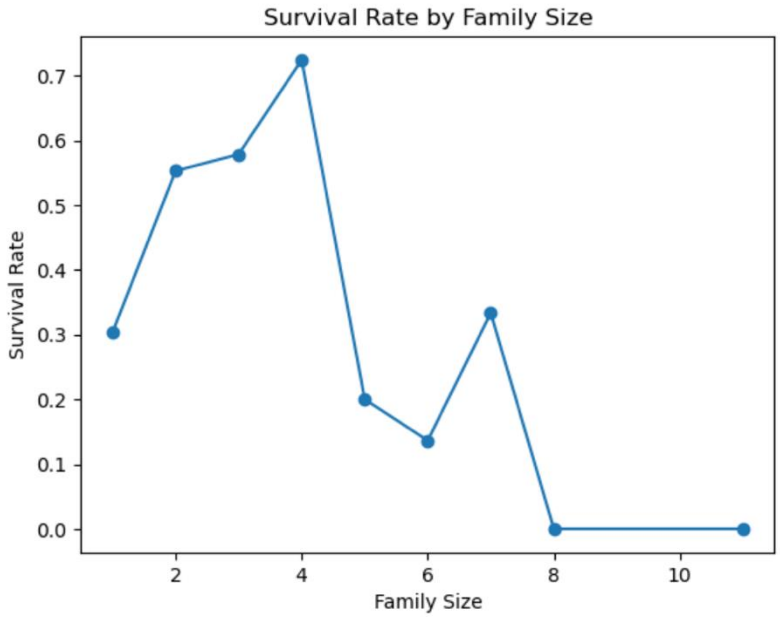
Hình 5: Box plot thể hiện phân phối của Age theo Survived trong tập huấn luyện.

3.5 Phân phối Age theo Survived

- Tuổi trung vị của cả hai nhóm (sống và không sống sót) gần bằng nhau (~28–30 tuổi).
- Nhóm không sống sót có nhiều ngoại lệ ở độ tuổi cao (70–80 tuổi).
- Nhóm sống sót có ít outliers hơn, cho thấy có thể trẻ em được ưu tiên cứu hộ.
- Gợi ý tạo đặc trưng mới như AgeGroup để mô hình học tốt hơn sự khác biệt theo độ tuổi.

3.6 Tỷ lệ sống sót theo Family Size

- Để phân tích tác động của kích thước gia đình đến khả năng sống sót, ta sử dụng biểu đồ đường



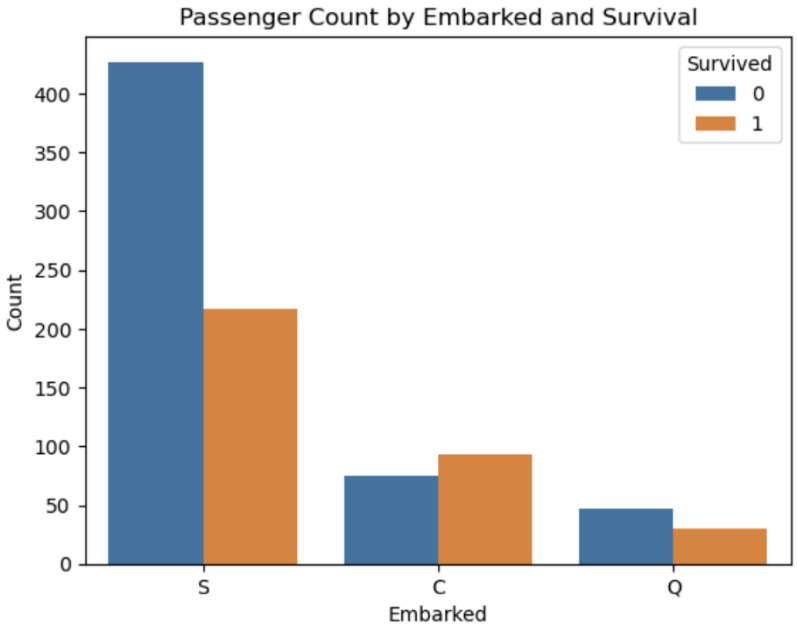
Hình 6: Biểu đồ đường thể hiện tỷ lệ sống sót theo Family Size trong tập huấn luyện.

3.6 Tỷ lệ sống sót theo Family Size

- Tỷ lệ sống sót cao nhất ở nhóm gia đình 3–4 người (~0.55–0.72).
- Gia đình lớn (>5 người) hoặc đi một mình có tỷ lệ sống sót thấp hơn, gần 0 với 8–11 người.
- Kích thước gia đình vừa phải có thể tăng khả năng sống sót.
- Đặc trưng FamilySize là lựa chọn hợp lý để đưa vào bước feature engineering.

3.7 Tỷ lệ sống sót theo Embarked

- Để đánh giá tác động của cảng lên tàu đến khả năng sống sót, ta sử dụng biểu đồ cột



Hình 7: Biểu đồ cột thể hiện số lượng sống sót theo Embarked trong tập huấn luyện.

3.7 Tỷ lệ sống sót theo Embarked

- Phần lớn hành khách lên tàu tại Southampton (S) – tỷ lệ không sống sót cao.
- Hành khách từ Cherbourg (C) có tỷ lệ sống sót cao nhất (~50%).
- Hành khách từ Queenstown (Q) có tỷ lệ sống sót thấp nhất.
- Gợi ý có thể liên quan đến phân bố Pclass tại từng cảng (C có nhiều khách hạng 1).
- Cần phân tích thêm để xác định rõ vai trò của Embarked với Survived.

3.8 Nhận xét từ trực quan hóa

- Phân phối độ tuổi (Age): phân phối lệch phải, phần nằm trong độ tuổi 20–40.
- Tác động của Sex và Pclass: Phụ nữ và hành khách hạng nhất có tỷ lệ sống sót cao hơn, phù hợp với chính sách ưu tiên cứu hộ
- Tương quan giữa các biến: Survived có tương quan âm với Pclass và dương với Fare. Age có tương quan yếu với Survived, cần xử lý lại bằng cách phân nhóm.

4. Xử lý dữ liệu

4.1 Tiền xử lý dữ liệu

- Tiền xử lý là bước nền tảng giúp làm sạch dữ liệu, xử lý giá trị thiếu và tạo đặc trưng mới, nhằm tối ưu hóa hiệu quả mô hình.

- Dựa trên phân tích ban đầu, nhóm thực hiện các bước sau:

  Xử lý giá trị thiếu (Age, Cabin, Embarked)

  Mã hóa biến phân loại (Sex, Embarked, Cabin)

  Tạo đặc trưng mới (AgeGroup, FamilySize, Title)

  Đồng bộ dữ liệu giữa tập train và test

4.1.1 Kiểm tra và xử lý giá trị thiếu

- Cabin: Thiếu ~77%, không thể khai thác ⇒ loại bỏ.
- Age: Thiếu ~19.9%, dùng SimpleImputer (mean) để giữ phân phối hợp lý.
- Fare: Thiếu trong tập test, dùng SimpleImputer (mean) để điền giá trị.
- Embarked: Thiếu 0.22%, gán nhãn giá trị thiếu (Embarked_nan), sau đó mã hóa One-Hot.

4.1.2 Mã hoá biến phân loại

- Các biến phân loại như Sex, Embarked, Title, AgeGroup, FareGroup cần chuyển thành dạng số.
- Sử dụng OneHotEncoder (drop="first") để tránh đa cộng tuyến.
- Tạo cột nhị phân như: Sex_male, Embarked_S, Embarked_C, v.v.
- Sau mã hóa, loại bỏ các cột gốc để tránh dư thừa thông tin.

4.2 Feature Engineering

- Mục tiêu: Tạo ra các đặc trưng mới giúp mô hình hiểu rõ hơn các yếu tố tiềm ẩn trong dữ liệu.
- Tập trung vào: yếu tố xã hội, kinh tế, và nhân khẩu học có ảnh hưởng đến khả năng sống sót.
- Một số đặc trưng được tạo:

  Danh xưng (Title) từ tên.

  Nhóm tuổi (AgeGroup).

  Kích thước gia đình (Family Size).

  Nhóm giá vé (FareGroup)

4.2.1 Trích xuất danh xưng (Title)

- Danh xưng như Mr, Mrs, Miss, Master được tách từ cột Name, cung cấp thông tin về giới tính, tuổi, tình trạng hôn nhân, và địa vị xã hội.

- Title giúp bổ sung ngữ cảnh cho mô hình, ví dụ:

  Master → thường là trẻ em nam.

  Dr, Rev → thể hiện địa vị xã hội cao.

  Sau khi trích xuất, danh xưng được mã hóa One-Hot Encoding.

4.2.2 Phân nhóm tuổi (AgeGroup)

- Thay vì dùng Age liên tục, dữ liệu được chia thành 4 nhóm:

  Trẻ em: $0 \leq \text{Age} < 16$

  Thanh niên: $16 \leq \text{Age} < 30$

  Người lớn: $30 \leq \text{Age} < 60$

  Người già: $\text{Age} \geq 60$

- Giúp giảm nhiễu và làm nổi bật xu hướng sống sót theo độ tuổi (ví dụ: trẻ em thường được ưu tiên cứu hộ).

- Cột AgeGroup được mã hóa bằng One-Hot Encoding.

4.2.3 Phân nhóm giá vé (FareGroup)

- Giá vé Fare được chia làm 3 nhóm phản ánh tình trạng kinh tế:

  Thấp: $0 \leq \text{Fare} < 10$

  Trung bình: $10 \leq \text{Fare} < 30$

  Cao: $\text{Fare} \geq 30$

- Hành khách trả vé cao thường thuộc hạng nhất → tỷ lệ sống sót cao hơn.

- Cột FareGroup được mã hóa bằng One-Hot Encoding.

4.2.4 Loại bỏ các đặc trưng không cần thiết

- Loại bỏ các cột không mang giá trị dự đoán hoặc có tỷ lệ thiếu quá cao:

  PassengerId: Chỉ là mã định danh, không có ý nghĩa.

  Name: Đã trích xuất danh xưng (Title), nên không còn hữu ích.

  Ticket: Giá trị rời rạc, khó khai thác.

  Cabin: Thiếu tới 77% → loại bỏ do không đủ thông tin.

- → Giảm chiều dữ liệu, tăng hiệu quả tính toán, tập trung vào đặc trưng quan trọng.

4.2.5 Kết quả của Feature Engineering

- Sau quá trình xử lý, các đặc trưng mới được tạo ra và mã hóa bao gồm:

Đặc trưng	Mô tả
Title	Danh xưng (Mr, Mrs, Miss, Master,...)
AgeGroup	Nhóm tuổi (Trẻ em, Thanh niên, Người lớn, Người già)
FareGroup	Nhóm giá vé (Thấp, Trung bình, Cao)
Sex	Giới tính (male, female)
Embarked	Cảng lên tàu (C, Q, S)

- → Các đặc trưng này làm nổi bật yếu tố xã hội, tuổi tác, và kinh tế, giúp mô hình dễ nhận diện mẫu liên quan đến khả năng sống sót.

5. Giới thiệu các mô hình dùng để dự đoán người sống sót

5.1 Logistic Regression

- Là mô hình phân loại nhị phân đơn giản, dựa trên hàm sigmoid để dự đoán xác suất. Dễ huấn luyện, dễ diễn giải và thường được dùng làm baseline cho các mô hình phức tạp hơn.

5.2 Random Forest

- Là mô hình ensemble sử dụng nhiều cây quyết định kết hợp bằng kỹ thuật bagging. Mỗi cây huấn luyện trên tập con dữ liệu và đưa ra dự đoán, sau đó lấy kết quả theo voting. Mô hình này giúp giảm overfitting và tăng tính ổn định.

5.3 Gradient Boosting

- Xây dựng mô hình bằng cách cộng dồn nhiều mô hình yếu (thường là cây quyết định) một cách tuần tự, mỗi mô hình mới học từ phần sai của mô hình trước. Hiệu quả cao nhưng dễ overfitting nếu không kiểm soát tốt.

5.4 Support Vector Classifier (SVC)

- Tìm siêu phẳng phân tách tối ưu giữa các lớp bằng cách tối đa hóa khoảng cách (margin). Với dữ liệu không tuyến tính, SVC dùng các kernel như RBF để ánh xạ vào không gian cao hơn.

5.5 Multi-layer Perceptron (MLP)

- MLP là mạng nơ-ron nhiều lớp có khả năng học các quan hệ phi tuyến phức tạp. Mô hình mạnh mẽ nhưng dễ overfit nếu không được tinh chỉnh tốt.

5.6 Extreme Gradient Boosting (XGBoost)

- Là phiên bản cải tiến của Gradient Boosting, bổ sung cơ chế regularization để chống overfitting và tối ưu hiệu suất. XGBoost có tốc độ huấn luyện nhanh và được sử dụng rộng rãi trong thực tế.

5.7 LightGBM

- LightGBM là thuật toán boosting tối ưu hóa cho tốc độ huấn luyện và xử lý dữ liệu lớn. Khác với XGBoost, LightGBM sử dụng chiến lược chia cây theo lá, giúp đạt độ chính xác cao hơn và giảm thời gian tính toán đáng kể.

5.8 Voting Classifier (Ensemble)

- Là mô hình tổng hợp kết quả từ nhiều mô hình con như Logistic Regression, Random Forest, XGBoost, LightGBM, v.v.

- Có hai kiểu voting:

  Hard Voting: chọn nhãn được dự đoán nhiều nhất.

  Soft Voting: tính trung bình xác suất và chọn nhãn có xác suất cao nhất.

Tổng thể, việc kết hợp các mô hình giúp hệ thống dự đoán ổn định hơn, tận dụng được thế mạnh của từng mô hình.

6. Kết quả từ mô hình Logistic Regression

6.1 Quy trình đánh giá (Logistic Regression)

- Để đánh giá hiệu quả mô hình Logistic Regression trong bài toán Titanic, phương pháp 5-fold Cross-Validation được sử dụng. Cụ thể:

  Dữ liệu được chia thành 5 phần gần bằng nhau.

  Trong mỗi vòng lặp, huấn luyện mô hình trên 4 phần, kiểm tra trên 1 phần còn lại.

  Quá trình lặp lại 5 lần, mỗi phần đều được làm tập kiểm tra một lần.

  Độ chính xác được tính trung bình từ 5 lần kiểm tra để đánh giá độ ổn định và tổng quát của mô hình.

6.2 Kết quả đạt được

- Các độ chính xác thu được từ 5 lần cross-validation là:
  - Fold 1: 81%
  - Fold 2: 78%
  - Fold 3: 82%
  - Fold 4: 79%
  - Fold 5: 80%

- Tính trung bình, mô hình Logistic Regression đạt độ chính xác khoảng:

- $\text{Accuracy}_{\text{mean}} = 0.80$ tức là khoảng 80%.

6.3 Phân tích kết quả

- Logistic Regression cho thấy hiệu quả ổn định, không quá phụ thuộc vào từng phần dữ liệu.
- 80% accuracy là một kết quả tốt với mô hình tuyến tính đơn giản.
- Tuy nhiên, mô hình hạn chế trong việc khai thác quan hệ phi tuyến giữa các đặc trưng.

Vì vậy, Logistic Regression là một baseline tốt, và nhóm đã tiếp tục thử nghiệm các mô hình mạnh hơn như Random Forest, SVC, XGBoost, LightGBM, v.v.

7. Hyperparameter Tuning

7.1 Phương pháp tối ưu

- Áp dụng Bayesian Optimization (qua BayesSearchCV của scikit-optimize) để tìm bộ siêu tham số tốt nhất.
- Phương pháp này khai thác thông tin từ các lần thử trước → hiệu quả hơn so với Grid Search hay Random Search.

7.2 Không gian siêu tham số

- Logistic Regression:
  C ~ Log-Uniform(1e-3, 1e3)

- Random Forest:
  max_depth ~ Integer(2, 50)

- Gradient Boosting:
  n_estimators ~ Integer(50, 500)
  learning_rate ~ Log-Uniform(0.01, 0.5)
  max_depth ~ Integer(2, 20)

7.3 Kết quả sau quá trình Hyperparameter Tuning

- Sau khi tối ưu siêu tham số bằng Bayesian Optimization, độ chính xác (accuracy) của các mô hình như sau:

Mô hình	Accuracy (%)
Logistic Regression	82.15
Random Forest	83.39
Gradient Boosting	83.84
Support Vector Classifier (SVC)	82.83
MLP Classifier	81.71
XGBoost	**84.40**
LightGBM	84.29

- **XGBoost** đạt hiệu suất cao nhất (84.40%), theo sát là **LightGBM** (84.29%).
- **MLP Classifier** có kết quả thấp nhất (81.71%).

7.4 Giải thích kết quả

- XGBoost và LightGBM vượt trội vì có khả năng tổng hợp nhiều cây nhỏ để giảm bias–variance và xử lý tốt các quan hệ phi tuyến. Chúng còn hỗ trợ các kỹ thuật regularization mạnh giúp tránh overfitting.

- Ngược lại, MLP Classifier hoạt động kém do:

  Dữ liệu dạng bảng không phải thế mạnh của mạng nơ-ron, vốn thích hợp với dữ liệu lớn hoặc phức tạp như ảnh/văn bản.

  Dễ bị overfit trên dữ liệu nhỏ nếu regularization không đủ tốt.

7.5 Mô hình tổng hợp: Voting Classifier

- Nhóm áp dụng kỹ thuật Voting Classifier để kết hợp sức mạnh của các mô hình như Logistic Regression, Random Forest, Gradient Boosting, SVC, MLP, XGBoost, và LightGBM nhằm nâng cao độ chính xác tổng thể.

  Hard voting: lấy nhãn được dự đoán nhiều nhất.

  Soft voting: lấy trung bình có trọng số của xác suất dự đoán.

- Trong bài toán này, chọn soft voting vì các mô hình mạnh như XGBoost và LightGBM cho xác suất tốt. Kết quả cuối cùng nhờ đó linh hoạt và chính xác hơn.

- Dù độ chính xác không vượt qua XGBoost hoặc LightGBM đơn lẻ, Voting Classifier vẫn mang lại hiệu suất ổn định, giảm sai số và tổng quát hóa tốt đủ tốt.

8. Tổng kết

Dự án áp dụng quy trình từ xử lý dữ liệu, tạo đặc trưng, đến huấn luyện và tối ưu mô hình để dự đoán khả năng sống sót trên tàu Titanic. Các đặc trưng quan trọng như Title, AgeGroup, FareGroup,... được tạo ra giúp cải thiện hiệu quả mô hình.

Nhiều mô hình đã được thử nghiệm, trong đó XGBoost và LightGBM cho kết quả tốt nhất (84.40% và 84.29%). Logistic Regression là baseline ổn định (82.15%), trong khi MLP hoạt động kém nhất (81.71%) do dễ overfit và dữ liệu không đủ lớn.

Voting Classifier giúp tăng độ chính xác và ổn định bằng cách kết hợp nhiều mô hình mạnh.

8. Tổng kết

Hướng phát triển:

Dùng SHAP/LIME để giải thích mô hình.

Bổ sung dữ liệu bằng SMOTE.

Khám phá AutoML và mô hình attention.

Thank you