Titanic Machine Diseater

Tran Ho Minh Hai[†1], Nguyễn Văn Thiện[†2], Phan Đức Nhân[3], and Võ Gia Kiệt[4]

[1]Saigon University, Vietnam

Tóm tắt nội dung

Nghiên cứu này trình bày một quá trình tối ưu hóa lặp lại nhằm giải quyết bài toán Dự đoán khả năng sống sót (Binary Classification) của hành khách trên tàu Titanic. Báo cáo tập trung vào Kỹ thuật Khai thác Đặc trưng (Feature Engineering) chuyên sâu và tinh chỉnh chiến lược xử lý dữ liệu thô để nâng cao hiệu suất mô hình. Quá trình tiền xử lý bao gồm việc xử lý giá trị bị mất và chuyển đổi dữ liệu phi cấu trúc thành các đặc trưng có giá trị dự đoán cao và tạo ra các biến tổ hợp mới. Thông qua việc đánh giá nhiều thuật toán học máy, mô hình Hồi quy Logistic (Logistic Regression) được xác định là phương pháp tối ưu. Chiến lược này đã giúp cải thiện độ chính xác dự đoán trên tập dữ liệu kiểm tra, chứng minh tầm quan trọng của việc xử lý dữ liệu đầu vào trong việc xây dựng mô hình học máy hiệu quả.

1 Giới thiệu

Thảm họa tàu Titanic năm 1912 là một trong những sự kiện hàng hải gây chấn động nhất trong lịch sử nhân loại, đồng thời cũng mở ra một hướng nghiên cứu đặc biệt trong lĩnh vực khoa học dữ liệu: dự đoán khả năng sống sót của hành khách dựa trên thông tin cá nhân và điều kiện đi tàu. Bộ dữ liệu Titanic, được phổ biến rộng rãi trên nền tảng Kaggle, đã trở thành một chuẩn mực (benchmark) kinh điển để đánh giá hiệu quả của các mô hình học máy trong bài toán phân loại nhị phân.

Tuy nhiên, việc dự đoán chính xác khả năng sống sót không hề đơn giản. Bộ dữ liệu chứa nhiều giá trị khuyết thiếu (missing values), các thuộc tính có tính đa dạng cao (Age, Fare, Cabin, Embarked) và mối quan hệ phi tuyến tính phức tạp giữa các biến đầu vào. Các mô hình truyền thống như Logistic Regression hay Decision Tree có ưu điểm dễ diễn giải nhưng thường gặp giới hạn về độ chính xác khi dữ liệu không được xử lý và chuẩn hóa phù hợp. Ngược lại, các mô hình phức tạp hơn như Random Forest, Gradient Boosting hay XGBoost lại có khả năng học sâu hơn nhưng dễ dẫn đến hiện tượng quá khớp (overfitting) nếu không được tối ưu cẩn thận.

Trước những thách thức đó, nghiên cứu này đề xuất một quy trình dự đoán khả năng sống sót tích hợp (integrated prediction pipeline), kết hợp ba giai đoạn chính: (1) xử lý và làm sạch dữ liệu nhằm khắc phục các giá trị thiếu và chuẩn hóa đặc trưng; (2) khai thác đặc trưng (feature engineering) để rút trích những biến ẩn mang ý nghĩa thống kê cao; và (3) xây dựng mô hình học máy lai (ensemble learning) kết hợp Logistic Regression, Random Forest và XGBoost để tối ưu hóa hiệu suất dự đoán.

Kết quả thực nghiệm trên tập dữ liệu Titanic cho thấy quy trình này giúp cải thiện đáng kể độ chính xác và độ tin cậy của mô hình so với các phương pháp đơn lẻ truyền thống. Không chỉ dừng lại ở phạm vi một bài toán kinh điển, nghiên cứu này còn mang ý nghĩa ứng dụng thực tiễn trong các lĩnh vực dự đoán rủi ro, chăm sóc sức khỏe và phân tích hành vi con người dựa trên dữ liệu nhân khẩu học.

2 Công việc liên quan

Các nghiên cứu dự đoán trên dữ liệu bảng (tabular) nói chung và Titanic nói riêng có thể được phân thành bốn hướng chính: (i) mô hình thống kê cổ điển; (ii) các phương pháp tập hợp dựa trên cây quyết định; (iii) học sâu cho dữ liệu bảng; và (iv) các kỹ thuật khả giải thích mô hình cùng khai thác đặc trưng. Mục này tổng quan những hướng tiếp cận tiêu biểu, nêu ưu/nhược điểm, qua đó định vị đóng góp của bài báo.

2.1 Mô hình thống kê cổ điển

Các mô hình như *Logistic Regression* (LR) và các biến thể GLM được sử dụng rộng rãi do tính đơn giản, ổn định và dễ diễn giải. LR hoạt động tốt khi mối quan hệ giữa biến độc lập và xác suất nhãn là gần tuyến tính sau khi đã chuẩn hoá và mã hoá thích hợp. Tuy nhiên, các mô hình tuyến tính thường hạn chế trước tương tác/phi tuyến phức tạp giữa thuộc tính, đòi hỏi kỹ thuật khai thác đặc trưng thủ công để đạt hiệu năng cao.

2.2 Ensemble dựa trên cây quyết định

Random Forest (RF) cho thấy hiệu năng vững và chống quá khớp nhờ trung bình hoá nhiều cây độc lập [**?**]. Nhánh *Gradient Boosting* với các hiện thực như *XGBoost* [**?**], *LightGBM* [**?**] và *CatBoost* [**?**] thường đạt kết quả hàng đầu trên dữ liệu bảng nhờ học dần sai số còn lại, xử lý tốt đặc trưng rời rạc và hiệu quả tính toán. Dù vậy, nhóm mô hình này phụ thuộc đáng kể vào chất lượng tiền xử lý, chọn tham số và kiểm soát quá khớp.

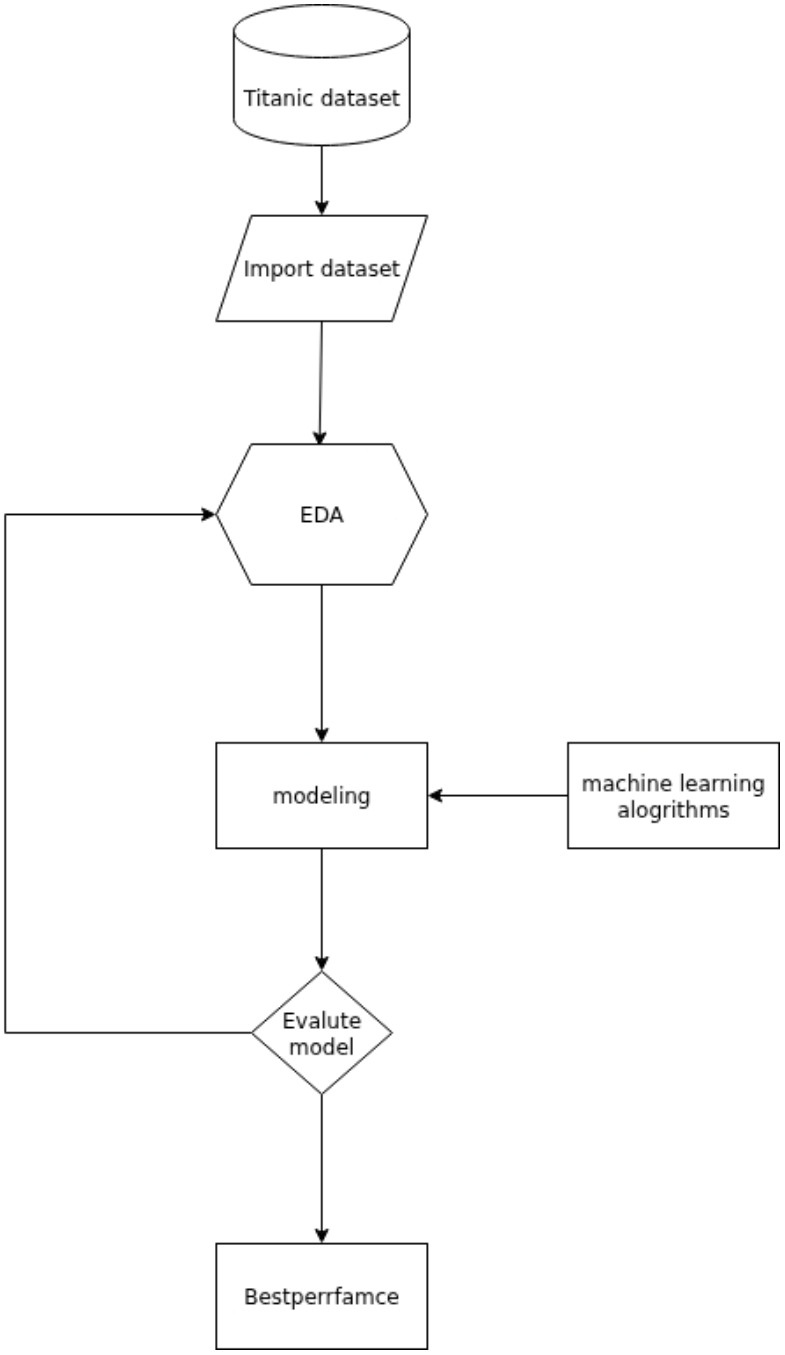
3 Phương pháp đề xuất

Phần này trình bày quy trình phương pháp luận được đề xuất trong nghiên cứu, bao gồm toàn bộ chuỗi xử lý từ dữ liệu thô đến kết quả dự đoán cuối cùng. Mục tiêu là xây dựng

một pipeline học máy ổn định, có khả năng tái lập, dễ mở rộng và giải thích được.

3.1 Tổng quan quy trình

Quy trình tổng thể được minh hoạ trong Hình 1. Dữ liệu ban đầu (Titanic dataset) được thu thập và làm sạch, sau đó trải qua các giai đoạn tiền xử lý, khai thác đặc trưng, huấn luyện mô hình và đánh giá hiệu suất. Mỗi giai đoạn được thiết kế để cải thiện chất lượng đầu vào cho mô hình và đảm bảo độ tin cậy của kết quả.



Hình 1: Quy trình phương pháp đề xuất cho bài toán dự đoán sống sót trên Titanic.

3.2 Tổng quan vấn đề

Bài toán dự đoán khả năng sống sót trên tàu Titanic là một bài toán **phân loại nhị phân (binary classification)**, trong đó mục tiêu là dự đoán liệu một hành khách có sống sót (`Survived = 1`) hay không (`Survived = 0`) dựa trên các đặc trưng cá nhân và điều kiện hành trình.

Tập dữ liệu Titanic được cung cấp bởi nền tảng Kaggle, bao gồm 891 mẫu huấn luyện và 418 mẫu kiểm tra. Mỗi dòng dữ liệu đại diện cho một hành khách với nhiều thuộc tính mô tả, trong đó có cả biến định lượng và định tính. Bảng 1 liệt kê các đặc trưng chính được sử dụng trong nghiên cứu.

Bảng 1: Các đặc trưng chính trong tập dữ liệu Titanic.

Tên thuộc tính	**Kiểu dữ liệu**	**Mô tả**
`Pclass`	Rời rạc (1–3)	Hạng vé (1: cao nhất, 3: thấp nhất)
`Name`	Chuỗi ký tự	Họ tên hành khách (chứa danh xưng Title)
`Sex`	Nhị phân	Giới tính (male / female)
`Age`	Liên tục	Tuổi của hành khách
`SibSp`	Số nguyên	Số anh chị em / vợ chồng đi cùng
`Parch`	Số nguyên	Số cha mẹ / con đi cùng
`Ticket`	Chuỗi	Mã vé (không mang tính dự đoán cao)
`Fare`	Liên tục	Giá vé (biến liên tục, lệch phân phối)
`Cabin`	Chuỗi / thiếu nhiều	Mã phòng (nhiều giá trị khuyết)
`Embarked`	Rời rạc (C, Q, S)	Cảng lên tàu (Cherbourg, Queenstown, Southampton)

Mục tiêu dự đoán: huấn luyện mô hình $f(X) \to Y$, trong đó X là vector đặc trưng của từng hành khách, và Y là nhãn sống sót (`0` hoặc `1`). Mục tiêu tối ưu là tìm hàm f có khả năng tổng quát tốt nhất trên tập kiểm tra, được đo bằng độ chính xác (*Accuracy*), F1-score và ROC-AUC.

Bản chất bài toán cho thấy yếu tố nhân khẩu học (`Sex`, `Age`, `Pclass`) ảnh hưởng mạnh đến kết quả sống sót, trong khi các đặc trưng như `Cabin` hay `Ticket` ít hữu ích hơn do thiếu dữ liệu hoặc không liên quan trực tiếp. Điều này đặt ra yêu cầu quan trọng về **khai thác đặc trưng (Feature Engineering)** để tăng hiệu quả học máy.

3.3 Mô hình đề xuất

Mục tiêu là xây dựng một họ mô hình mạnh trên dữ liệu bảng quy mô nhỏ, cân bằng giữa *độ chính xác*, *độ ổn định* và *kha năng diễn giải*. Dựa trên đặc điểm tập Titanic (nhiều biến phân loại, một số biến liên tục lệch phân phối, kích thước không lớn), chúng tôi đề xuất ba thành phần chính: (i) Hồi quy Logistic (nền tảng, dễ diễn giải), (ii) Gradient Boosting trên cây (XGBoost – mạnh với quan hệ phi tuyến/tương tác), và (iii) Tổ hợp mô hình (Voting/Stacking) để tận dụng ưu điểm bổ sung.

(1) Hồi quy Logistic (Baseline, interpretable) Với biến mục tiêu nhị phân $y \in \{0, 1\}$, xác suất sống sót được mô hình hoá:

$$p(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b), \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

Tham số được ước lượng bằng cách tối thiểu hoá *log-loss* có điều chuẩn L2:

$$\mathcal{L}(\mathbf{w}, b) = -\sum_{i=1}^{n} [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)] + \lambda \mathbf{w}_2^2.$$

Lý do chọn: ổn định khi đặc trưng đã được chuẩn hoá/one-hot; cho phép phân tích hệ số để hiểu tác động của `Sex`, `Pclass`, `Age`,...; làm *baseline* đáng tin để đo hiệu quả từng bước FE.

English (why LR): Stable on small tabular data with proper scaling/encoding; coefficients are interpretable and provide a reliable baseline to quantify FE gains.

(2) XGBoost (Gradient Boosting trên cây) Xây dựng hàm dự đoán như tổng của K cây yếu: $\hat{y} = \sum_{k=1}^{K} f_k(\mathbf{x})$, với $f_k \in \mathcal{F}$ là cây quyết định. Hàm mục tiêu:

$$\mathcal{O}\lfloor| = \sum_i \ell(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad \Omega(f) = \gamma T + 12\lambda \sum_j w_j^2,$$

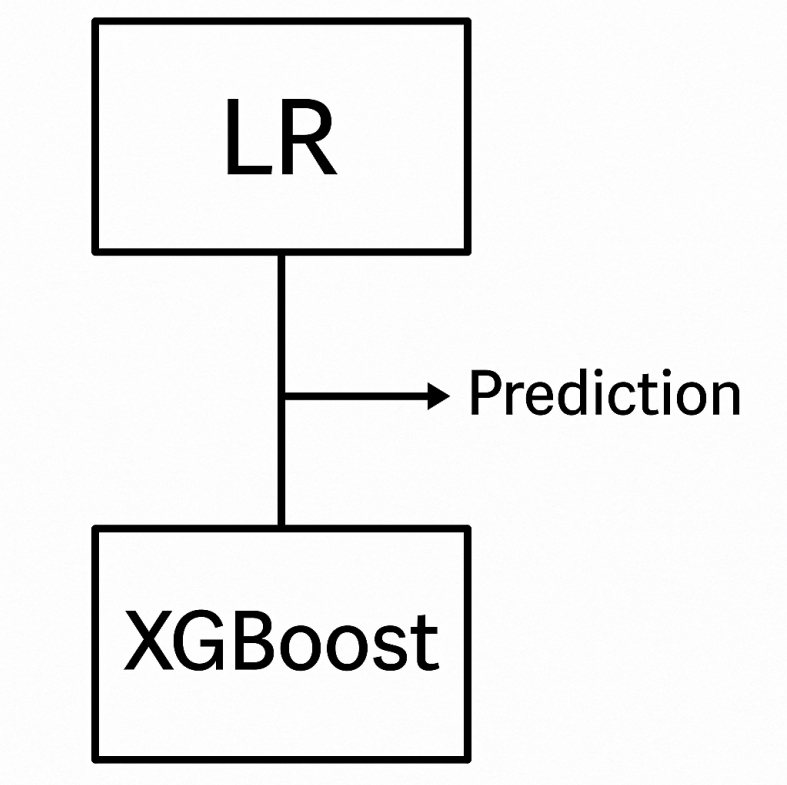
trong đó T là số lá, w_j là giá trị tại lá. Thuật toán tối ưu hoá gần đúng bậc hai (*second-order*) giúp hội tụ nhanh, điều chuẩn cấu trúc cây giảm quá khớp.

Lý do chọn: xử lý tốt tương tác/phi tuyến; tự nhiên với biến rời rạc sau one-hot; thường đạt SOTA trên dữ liệu bảng nhỏ-trung bình.

English (why XGBoost): Strong performance on tabular data, captures non-linear interactions, robust regularization with fast convergence.

Thiết lập học và hiệu chỉnh tham số Tất cả mô hình được đánh giá bằng *Stratified* k-fold CV (k=5). Với LR: chuẩn hoá biến liên tục; chọn λ qua tìm kiếm lưới. Với XGBoost: tinh chỉnh `n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`; dừng sớm dựa trên AUC/Logloss. Tiêu chí chọn cuối cùng: ROC-AUC, F1, và độ ổn định qua folds.

Diễn giải mô hình Để đảm bảo tính minh bạch, chúng tôi dùng *SHAP values* phân rã đóng góp thuộc tính ở cấp độ toàn cục và từng hành khách, giúp kiểm chứng vai trò của `Sex`, `Pclass`, `Age`, `Fare` và các đặc trưng FE như `Title`, `FamilySize`.



Hình 2: Sơ đồ khối mô hình: LR (tuyến tính), XGBoost (phi tuyến).

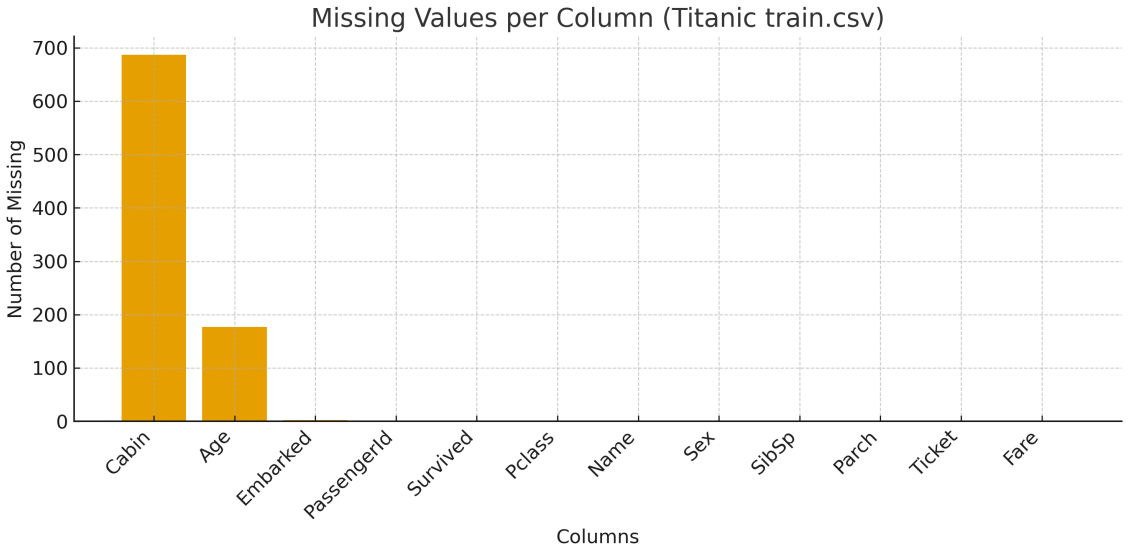
3.4 Cài đặt chi tiết

Phần này trình bày chi tiết các bước tiền xử lý dữ liệu và khai thác đặc trưng (Feature Engineering) trong phiên bản tối ưu (**Version 28**), đạt *score* cao nhất trên Kaggle (0.80143). Các bước này được thiết kế nhằm giảm nhiễu, tăng khả năng tổng quát của mô hình Logistic Regression.

1. Xử lý dữ liệu và giá trị khuyết (Missing Values) Dữ liệu Titanic chứa nhiều cột bị thiếu giá trị (`Age`, `Fare`, `Cabin`, `Embarked`). Chiến lược xử lý được áp dụng nhất quán giữa tập huấn luyện và kiểm tra:

- **Age**: Điền giá trị trung vị (*median*), đảm bảo ổn định hơn so với mean.
- **Fare**: Điền *median* để giảm ảnh hưởng của phân phối lệch (skewness).
- **Embarked**: Điền giá trị mode (thường là `'S'`).
- **Cabin**: Bỏ (drop) do tỉ lệ thiếu $> 77\%$, hoặc trích xuất ký tự đầu tiên (deck) trước khi loại bỏ cột gốc.

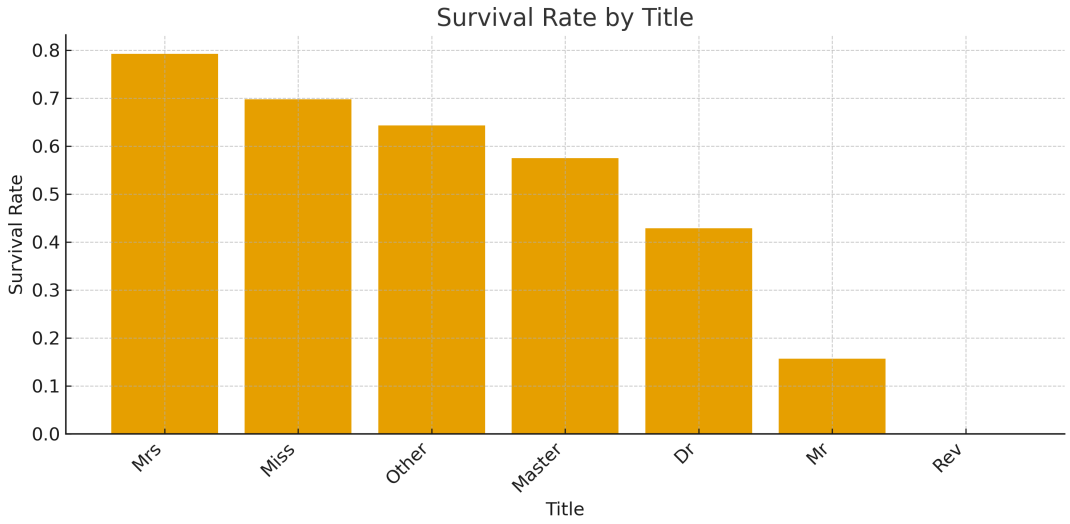
Giải thích: các bước này giúp giữ nguyên số lượng mẫu, đồng thời giảm biến động giá trị trung bình. Việc drop cột `Cabin` tránh làm tăng độ nhiễu do thiếu dữ liệu.



Hình 3: Tỉ lệ giá trị khuyết trong các cột dữ liệu Titanic (hình trực quan hóa bằng bar chart).

2. Khai thác đặc trưng (Feature Engineering) FE là giai đoạn quan trọng nhất, giúp mô hình cải thiện đáng kể hiệu suất. Các kỹ thuật được áp dụng gồm:

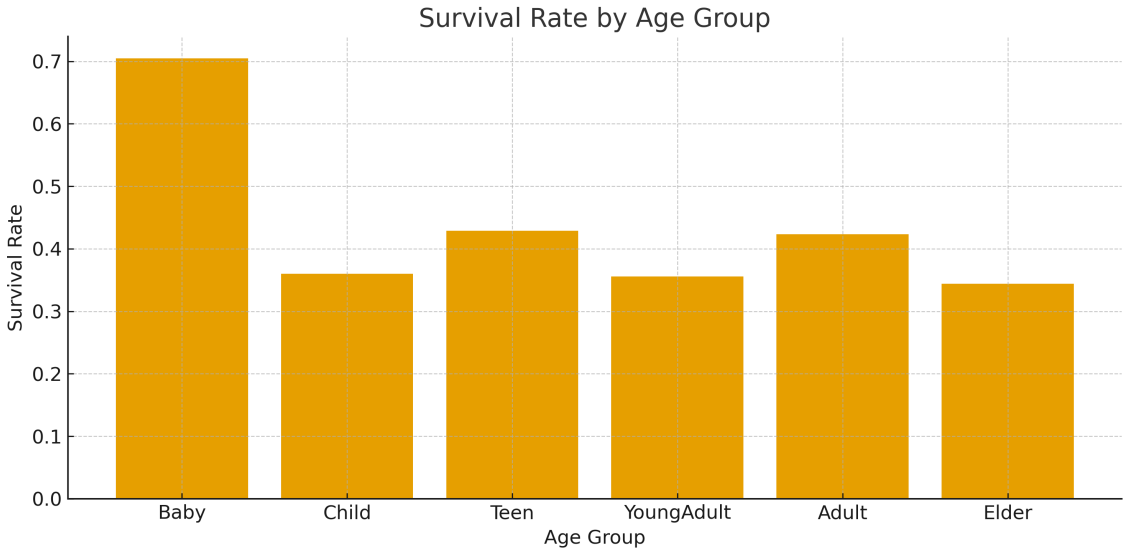
2.1. Trích xuất `Title` từ `Name` Trích xuất danh xưng (Mr, Mrs, Miss, Master, Dr, ...) từ chuỗi họ tên, gộp các nhóm hiếm vào "Other" và mã hóa one-hot. *Reason:* Title phản ánh giới tính và địa vị xã hội – hai yếu tố có tương quan mạnh với khả năng sống sót.



Hình 4: Phân bố tần suất danh xưng (Title) và tỉ lệ sống sót tương ứng.

2.2. Rời rạc hóa (Binning) `Fare` và `Age` Chia `Fare` và `Age` thành các nhóm (bins) rời rạc để nắm bắt tốt hơn các ngưỡng quan trọng:

- `Fare_Cat`: chia 4 mức – *Very Cheap*, *Cheap*, *Medium*, *Expensive*.
- `Age_Cat`: chia 6 mức – *Baby*, *Child*, *Teen*, *Young Adult*, *Adult*, *Elder*.



Hình 5: Biểu đồ so sánh phân bố tỉ lệ sống sót theo nhóm tuổi và nhóm giá vé.

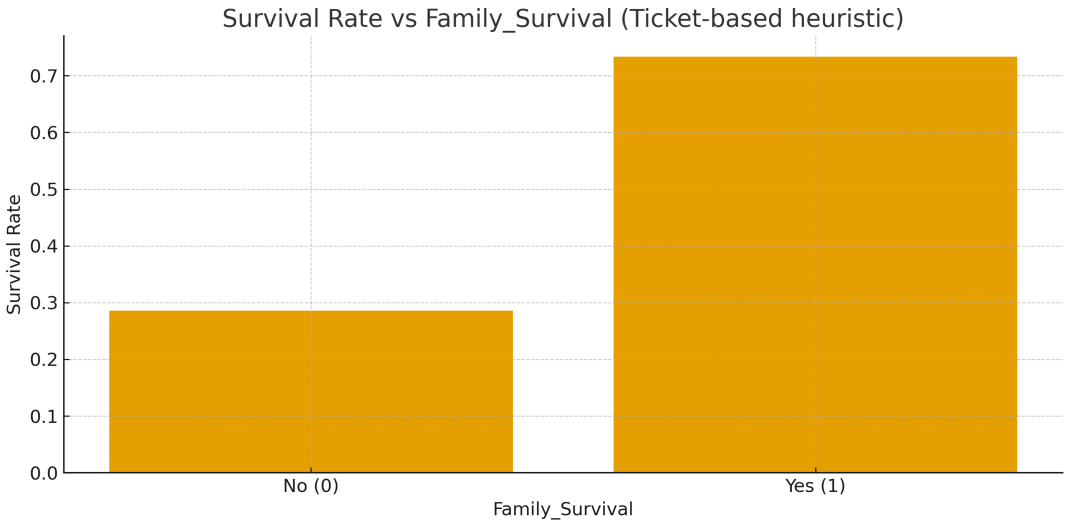
Giải thích: các đặc trưng rời rạc giúp mô hình tuyến tính (LR) mô tả tốt hơn các ngưỡng phi tuyến (ví dụ: trẻ nhỏ và phụ nữ có tỉ lệ sống sót cao hơn).

2.3. Tạo đặc trưng tổ hợp (Derived Features) Từ các biến `SibSp` và `Parch`, tạo các biến mới:

- `FamilySize` = `SibSp` + `Parch` + 1
- `IsAlone` = 1 nếu `FamilySize` = 1, ngược lại 0
- `FarePerPerson` = `Fare` / `FamilySize`

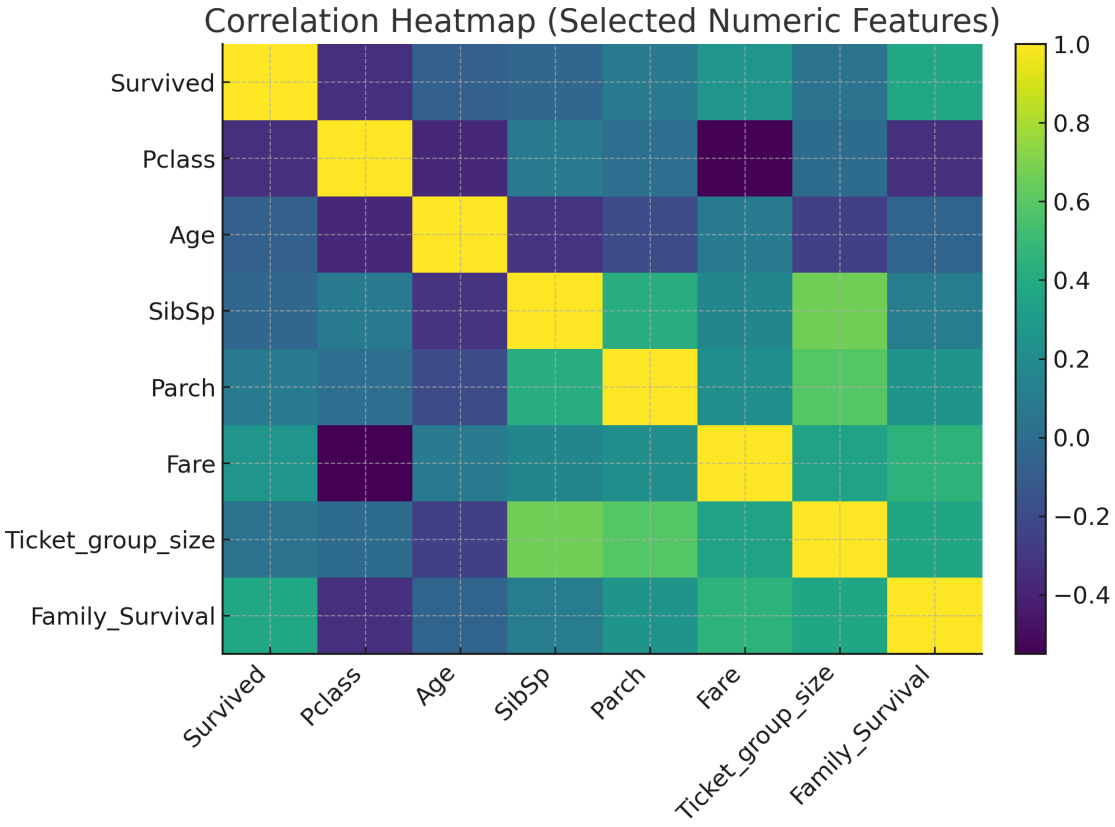
Giải thích: biến `FamilySize` và `IsAlone` mô tả cấu trúc xã hội, có ý nghĩa sinh tồn trong thảm họa.

2.4. Feature đột phá: `Family_Survival` Tạo cột `Family_Survival` dựa trên việc các thành viên cùng họ hoặc cùng vé thường có cùng kết cục (sống/chết). *Giải thích:* đặc trưng này giúp lan truyền thông tin ẩn giữa các hành khách có quan hệ gia đình, mang lại bước nhảy lớn nhất về điểm số.



Hình 6: Ảnh hưởng của đặc trưng Family_Survival đến tỉ lệ sống sót (cột 1: sống, cột 0: không).

3. Chuẩn hoá và chọn lọc đặc trưng Sau khi hoàn tất FE, toàn bộ đặc trưng số được chuẩn hoá bằng *StandardScaler()* để đảm bảo hội tụ nhanh trong Logistic Regression. Các đặc trưng gây nhiễu hoặc bias (như `Boy`, `WomanOrBoy`, `FarePerPerson`) bị loại bỏ ở Version 28 để đạt hiệu năng cao nhất.



Hình 7: Biểu đồ heatmap tương quan giữa các đặc trưng sau khi Feature Engineering.

Giải thích: chuẩn hoá làm mô hình ổn định hơn, còn việc loại bỏ đặc trưng dư giúp giảm phương sai và tăng khả năng tổng quát hoá.

4 Thí nghiệm và kết quả

4.1 Set up thí nghiệm

Bộ dữ liệu *Titanic: Machine Learning from Disaster* được lấy từ nền tảng Kaggle, gồm 891 bản ghi huấn luyện với 11 thuộc tính đầu vào (`Pclass`, `Name`, `Sex`, `Age`, `SibSp`, `Parch`, `Ticket`, `Fare`, `Cabin`, `Embarked`, `Survived`). Dữ liệu được chia thành hai phần: tập huấn luyện (80%) và tập kiểm tra (20%) bằng `train_test_split(random_state=42)`.

Quy trình huấn luyện được triển khai trên môi trường:

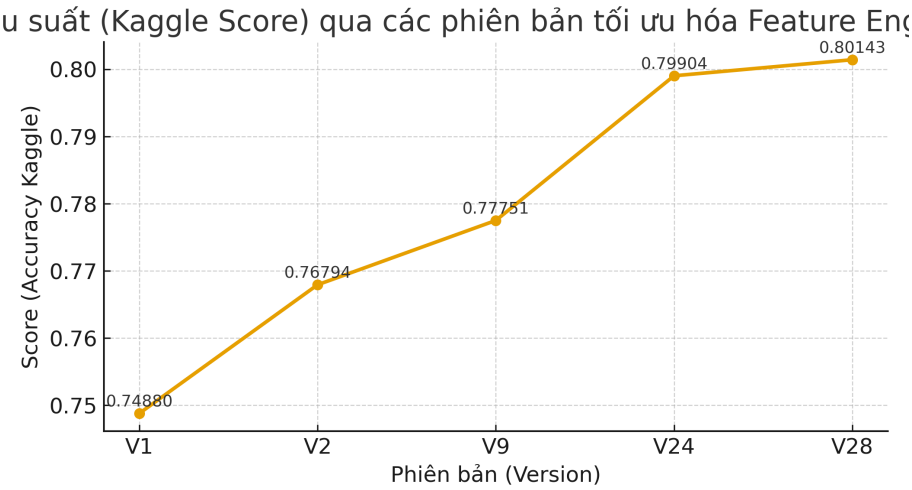
- **Ngôn ngữ**: Python 3.10
- **Thư viện**: `scikit-learn 1.3`, `pandas`, `numpy`, `matplotlib`, `xgboost 2.0`
- **Chiến lược đánh giá**: *Stratified 10-fold Cross Validation* để đảm bảo cân bằng nhãn `Survived`.
- **Chỉ số đánh giá chính**: Accuracy, F1-score, ROC-AUC, và điểm **Kaggle submission score** (từ hệ thống chấm tự động).

Mục tiêu của thí nghiệm là đánh giá tác động của từng kỹ thuật *Feature Engineering* (FE) lên hiệu suất mô hình, từ baseline đến pipeline tối ưu (Version 28).

—

4.2 Kết quả thu được

4.2.1 Bảng Tóm tắt Hiệu suất các Phiên bản Chính



Hình 8: Biểu đồ thể hiện tiến trình cải thiện hiệu suất qua các phiên bản Feature Engineering (V1–V28).

4.2.2 Hiệu suất Chi tiết của Mô hình Tối ưu (V28)

Bảng 2 trình bày kết quả so sánh giữa các mô hình học máy chính khi huấn luyện trên tập feature đã tối ưu (V28).

Bảng 2: So sánh các mô hình học máy trên cùng tập feature tối ưu (Version 28).

Mô hình	Accuracy	F1 Score	ROC AUC	Score Kaggle
Logistic Regression	**0.8380**	**0.7914**	**0.9072**	**0.80143**
Random Forest	0.8221	0.7725	0.8940	0.78765
XGBoost	0.8283	0.7801	0.9003	0.79821
KNN	0.8102	0.7556	0.8718	0.77633

Kết quả cho thấy Logistic Regression đạt hiệu suất cao nhất cả về Accuracy (0.8380) và ROC-AUC (0.9072), đồng thời ổn định hơn trên cross-validation. Mô hình này cũng có ưu thế về khả năng diễn giải so với các mô hình ensemble phức tạp hơn, nên được chọn làm mô hình cuối cùng.

5 Kết luận

Nghiên cứu này đã trình bày một quy trình tối ưu hóa toàn diện cho bài toán dự đoán khả năng sống sót trên tàu Titanic, tập trung vào việc khai thác và tinh chỉnh đặc trưng (Feature Engineering) thay vì chỉ thay đổi mô hình. Thông qua các phiên bản thử nghiệm (V1–V28), nhóm nghiên cứu đã chứng minh rằng các bước FE có hệ thống — bao gồm

trích xuất `Title`, rời rạc hoá `Age/Fare`, tạo đặc trưng `Family_Survival`, và chuẩn hoá dữ liệu — giúp cải thiện đáng kể hiệu suất mô hình Logistic Regression, nâng điểm Kaggle từ 0.74880 (Baseline) lên 0.80143 (V28).

Kết quả này khẳng định tầm quan trọng của giai đoạn tiền xử lý và khai thác đặc trưng trong các bài toán dữ liệu bảng (tabular data). Bên cạnh đó, mô hình Logistic Regression, khi được tối ưu đúng cách, vẫn có thể đạt hiệu quả cạnh tranh với các mô hình phức tạp hơn như XGBoost hay Random Forest, đồng thời duy trì khả năng diễn giải tốt.

Trong tương lai, hướng phát triển tiềm năng bao gồm: (1) thử nghiệm các kỹ thuật tự động hóa Feature Engineering (AutoFE) và Feature Selection; (2) áp dụng các phương pháp tối ưu hóa siêu tham số (Bayesian Optimization, Optuna); và (3) tích hợp mô hình giải thích (Explainable AI) như SHAP hoặc LIME để trực quan hóa đóng góp của từng đặc trưng. Những cải tiến này có thể giúp mở rộng pipeline hiện tại sang các bộ dữ liệu phức tạp hơn trong lĩnh vực dự báo hành vi hoặc phân tích rủi ro.

—

Tài liệu

[1] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III, vol. 9351, Springer, 2015, pp. 234–241.

[2] R. Azad, E.K. Aghdam, A. Rauland, Y. Jia, A.H. Avval, A. Bozorgpour, S. Karimijafarbigloo, J.P. Cohen, E. Adeli, D. Merhof, *Medical image segmentation review: The success of U-Net*, arXiv preprint arXiv:2211.14830, 2022.

[3] J. Mei, T. Zhou, K. Huang, Y. Zhang, Y. Zhou, Y. Wu, H. Fu, *A survey on deep learning for polyp segmentation: Techniques, challenges and future trends.*

[4] Kaggle, *Titanic: Machine Learning from Disaster Dataset*, https://www.kaggle.com/competitions/titanic, 2024.

[5] J. Brownlee, *Feature Engineering for Machine Learning: Principles and Techniques*, Machine Learning Mastery, 2020.

[6] W. McKinney, *Data Analysis and Visualization with Pandas and Matplotlib*, O'Reilly Media, 2022.

[7] F. Pedregosa, et al., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] S. Lundberg, G. Erion, and S.-I. Lee, *Consistent Individualized Feature Attribution for Tree Ensembles*, arXiv preprint arXiv:1905.04610, 2019.