Thực hiện dự đoán tỷ lệ sống sót của hành khách dựa trên bộ dữ liệu Titanic từ cuộc thi Kaggle

Nguyễn Đăng Tiến[1*]

[1]*Đại học Sài Gòn, Việt Nam.*

**Liên hệ với tác giả:* dangtien07122005@gmail.com

Tóm tắt

Thảm họa chìm tàu Titanic là một trong những sự kiện nổi tiếng nhất trong lịch sử hàng hải, và bộ dữ liệu Titanic từ cuộc thi Kaggle đã trở thành một nguồn học liệu quan trọng trong lĩnh vực học máy. Nghiên cứu này tập trung vào việc xây dựng các mô hình dự đoán khả năng sống sót của hành khách dựa trên các đặc trưng như giới tính, độ tuổi, hạng vé, số người đi cùng và giá vé. Dữ liệu được tiền xử lý bằng cách xử lý giá trị khuyết, mã hóa biến phân loại và chuẩn hóa đặc trưng trước khi đưa vào huấn luyện. Các mô hình được áp dụng bao gồm hồi quy logistic, rừng ngẫu nhiên và XGBoost. Kết quả cho thấy mô hình XGBoost đạt độ chính xác cao nhất trên tập kiểm thử, chứng tỏ hiệu quả của các thuật toán học máy hiện đại trong việc xử lý các bài toán phân loại nhị phân. Nghiên cứu cũng phân tích tầm quan trọng của các đặc trưng và đưa ra đề xuất cải thiện hiệu suất mô hình thông qua tối ưu hóa tham số và mở rộng đặc trưng. Kết quả này có thể đóng vai trò tham khảo cho các nghiên cứu và ứng dụng thực tế trong lĩnh vực phân tích dữ liệu và dự đoán hành vi con người.

Từ khóa: Titanic, Phân loại, XGBoost, Khai phá dữ liệu, Kaggle, Học máy, Feature Engineering, Làm sạch dữ liệu.

1 Giới thiệu

Vụ chìm tàu Titanic vào năm 1912 là một trong những thảm họa hàng hải thảm khốc nhất trong lịch sử, khi con tàu được coi là "không thể chìm" đã va phải tảng băng trôi trong chuyến hải trình đầu tiên, khiến hơn 1.500 hành khách và thủy thủ thiệt mạng. Bộ dữ liệu Titanic, được cung cấp trong cuộc thi nổi tiếng trên nền tảng Kaggle, là một trong những bộ dữ liệu kinh điển trong lĩnh vực học máy, được sử dụng rộng rãi để nghiên cứu và thực hành các kỹ thuật dự đoán, phân loại và xử lý dữ liệu thực tế.

Mục tiêu của nghiên cứu này là xây dựng và đánh giá các mô hình học máy nhằm dự đoán khả năng sống sót của hành khách trên tàu Titanic dựa trên các thông tin nhân khẩu học và đặc trưng liên quan, bao gồm giới tính, độ tuổi, hạng vé, giá vé, và số lượng người đi cùng. Bằng việc áp dụng các mô hình như Hồi quy Logistic, Rừng ngẫu nhiên và XGBoost, nghiên cứu tìm cách xác định mô hình có hiệu suất dự đoán cao nhất cũng như phân tích tầm quan trọng của từng đặc trưng trong việc quyết định khả năng sống sót.

Đóng góp chính của nghiên cứu không chỉ nằm ở việc so sánh các mô hình học máy phổ biến, mà còn ở việc thiết lập một quy trình xử lý dữ liệu toàn diện bao gồm làm sạch, mã hóa, chuẩn hóa và tối ưu hóa tham số. Kết quả nghiên cứu có thể đóng vai trò làm tài liệu tham khảo cho người học và nhà nghiên cứu trong việc ứng dụng các kỹ thuật học máy vào các bài toán phân loại thực tế, đồng thời minh chứng cho tầm quan trọng của tiền xử lý dữ liệu trong việc nâng cao độ chính xác của mô hình.

Cấu trúc bài báo được trình bày như sau: Phần 2 mô tả phương pháp nghiên cứu và các mô hình được sử dụng. Phần 3 trình bày kết quả thực nghiệm và thảo luận. Cuối cùng, Phần 4 đưa ra kết luận và hướng phát triển trong tương lai.

2 Phương pháp nghiên cứu

2.1 Tổng quan quy trình

Quy trình nghiên cứu được thực hiện qua bốn giai đoạn chính: (1) thu thập và khám phá dữ liệu, (2) tiền xử lý và trích chọn đặc trưng, (3) xây dựng và huấn luyện mô hình học máy, và (4) Tối ưu hóa siêu tham số, (5) đánh giá hiệu suất dự đoán. Toàn bộ quy trình được triển khai bằng ngôn ngữ lập trình Python, sử dụng các thư viện phổ biến như `pandas`, `scikit-learn`, và `xgboost`.

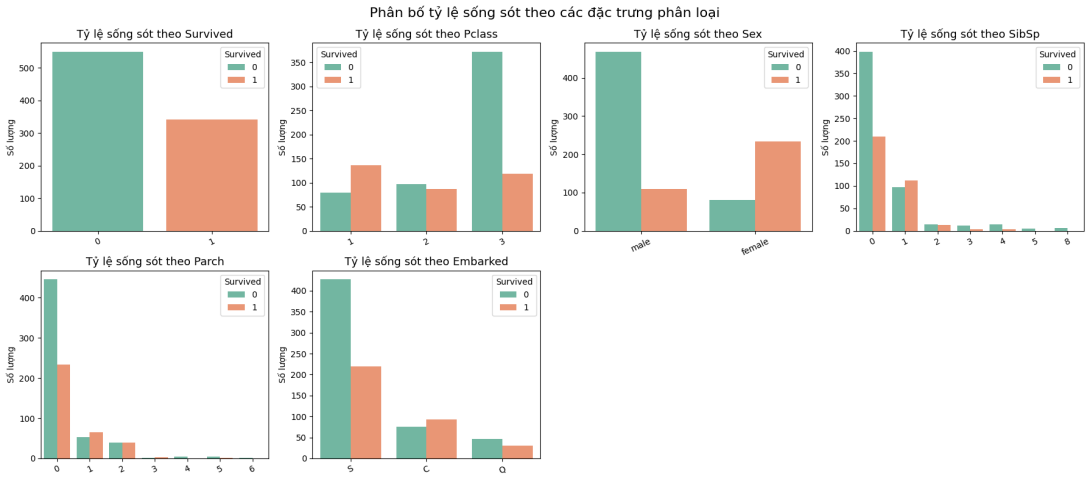

Figure 1: Biểu đồ môi tương quan giữa các đặc trưng đối với mục tiêu Đặc trưng `Sex` và `Pclass` có mức ảnh hưởng đến khả năng sống sót.

2.2 Dữ liệu nghiên cứu

Bộ dữ liệu Titanic được cung cấp bởi nền tảng Kaggle (*Titanic: Machine Learning from Disaster*), bao gồm thông tin của 891 hành khách với 12 thuộc tính, trong đó nhãn mục tiêu là biến `Survived` (có giá trị 1 nếu hành khách sống sót, và 0 nếu không). Một số đặc trưng quan trọng bao gồm:

- `Pclass`: Hạng vé (1, 2, 3)
- `Sex`: Giới tính
- `Age`: Tuổi của hành khách
- `SibSp`: Số anh/chị/em hoặc vợ/chồng đi cùng
- `Parch`: Số cha/mẹ hoặc con đi cùng
- `Fare`: Giá vé
- `Embarked`: Cảng lên tàu

2.3 Tiền xử lý dữ liệu

Các bước tiền xử lý được thực hiện nhằm đảm bảo dữ liệu đầu vào sạch và phù hợp cho mô hình học máy:

1. **Xử lý giá trị khuyết:** Các giá trị thiếu trong cột `Age`, `Cabin` và `Embarked` được xử lý bằng phương pháp trung vị hoặc mode.
2. **Mã hóa biến phân loại:** Các biến như `Sex` và `Embarked` được chuyển thành dạng số bằng kỹ thuật One-Hot Encoding.
3. **Chuẩn hóa dữ liệu:** Các đặc trưng định lượng (`Age`, `Fare`) được chuẩn hóa về cùng thang đo bằng phương pháp Min-Max Scaling.
4. **Chia tập dữ liệu:** Dữ liệu được chia thành tập huấn luyện và kiểm thử theo tỷ lệ 80:20.

2.4 Trích chọn và tạo đặc trưng (Feature Engineering)

Sau khi dữ liệu được làm sạch và chuẩn hóa, bước tiếp theo là trích chọn và tạo thêm các đặc trưng nhằm cải thiện khả năng học của mô hình. Quá trình này không chỉ giúp mô hình hiểu rõ hơn về mối quan hệ giữa các biến, mà còn giúp tăng tính tổng quát hoá trong dự đoán.

Các kỹ thuật *feature engineering* được áp dụng bao gồm:

1. **Tạo đặc trưng mới:** - Tạo biến `FamilySize` = `SibSp` + `Parch` + 1 để biểu thị tổng số thành viên trong gia đình đi cùng. - Biến `IsAlone` được sinh ra từ `FamilySize`, nhận giá trị 1 nếu hành khách đi một mình và 0 nếu có người đi cùng. Các đặc trưng này giúp mô hình học được mối liên hệ giữa khả năng sống sót và yếu tố gia đình.

2. **Rút trích thông tin từ tên hành khách:** Từ biến `Name`, trích xuất danh xưng (title) như *Mr., Mrs., Miss., Master* bằng kỹ thuật tách chuỗi. Sau đó, các danh xưng hiếm được nhóm lại để tránh dữ liệu thưa. Biến `Title` thể hiện tầng lớp xã hội và giới tính một cách gián tiếp, giúp tăng độ phân biệt của mô hình.

3. **Phân loại nhóm tuổi và giá vé:** Các biến `Age` và `Fare` được chia thành các khoảng (bins) như *Trẻ em, Thanh niên, Trung niên, Cao tuổi* hoặc *Thấp, Trung bình, Cao*, giúp mô hình phi tuyến như cây quyết định và XGBoost dễ nhận diện quy luật hơn.

4. **Mã hóa và chuẩn hóa lại các đặc trưng mới:** Các đặc trưng sau khi được tạo ra được mã hóa (One-Hot Encoding) và chuẩn hóa tương tự như các biến gốc để đảm bảo tính nhất quán đầu vào cho mô hình.

Kết quả của bước *feature engineering* giúp cải thiện đáng kể khả năng phân loại, đặc biệt trong các mô hình phi tuyến như Rừng ngẫu nhiên và XGBoost, do các đặc trưng mới phản ánh rõ hơn mối quan hệ ẩn giữa hành vi con người và khả năng sống sót.

2.5 Xây dựng mô hình học máy

Bốn mô hình được lựa chọn để so sánh hiệu quả:

- **Hồi quy Logistic (Logistic Regression)** – mô hình tuyến tính cơ bản cho bài toán phân loại nhị phân.
- **Support Vector Machine (SVM)** – mô hình phân loại phi tuyến tính, dùng trong các bộ dữ liệu đặc trưng nhỏ.
- **Rừng ngẫu nhiên (Random Forest)** – mô hình tập hợp giúp giảm phương sai và tăng độ chính xác.
- **XGBoost** – thuật toán boosting mạnh mẽ, được tối ưu về hiệu năng và khả năng tổng quát hóa.

Tất cả các mô hình được triển khai trong pipeline để đảm bảo tính nhất quán giữa các bước xử lý dữ liệu và huấn luyện. Các siêu tham số được điều chỉnh bằng kỹ thuật *Randomized Search Cross-Validation* để tìm cấu hình tối ưu.

2.6 Tối ưu hóa siêu tham số (Hyperparameter Optimization)

Để cải thiện hiệu suất và khả năng tổng quát hóa của mô hình, quá trình tối ưu hóa siêu tham số được thực hiện cho từng thuật toán. Kỹ thuật *Randomized Search Cross-Validation (RandomizedSearchCV)* được sử dụng nhằm tìm ra tổ hợp tham số tối ưu với chi phí tính toán hợp lý hơn so với *Grid Search*. Quá trình tìm kiếm được thực hiện với 5-fold cross-validation và sử dụng độ đo `Accuracy` làm tiêu chí đánh giá.

Đối với từng mô hình, các siêu tham số được tối ưu như sau:

- **Hồi quy Logistic:** Hệ số phạt (`C`) và loại chuẩn hóa (`L1`, `L2`).
- **Cây quyết định:** Độ sâu tối đa (`max_depth`) và số mẫu tối thiểu tại nút lá (`min_samples_leaf`).
- **Rừng ngẫu nhiên:** Số lượng cây (`n_estimators`), độ sâu tối đa và số đặc trưng được xem xét khi chia nút (`max_features`).
- **XGBoost:** Tốc độ học (`learning_rate`), số lượng cây (`n_estimators`), độ sâu tối đa (`max_depth`), và hệ số điều chuẩn (`reg_lambda`).

Kết quả tối ưu cho thấy XGBoost đạt được hiệu suất cao nhất với cấu hình: `learning_rate = 0.05`, `max_depth = 6`, `n_estimators = 200`, và `subsample = 0.8`. Nhờ vào việc điều chỉnh siêu tham số hợp lý, mô hình XGBoost giảm hiện tượng quá khớp và cải thiện độ chính xác trên tập kiểm thử thêm khoảng 2–3% so với cấu hình mặc định.

Table 1: Tóm tắt siêu tham số tối ưu của các mô hình

Mô hình	**Siêu tham số tối ưu**
Rừng ngẫu nhiên	n_estimators=100, max_depth=10, max_features='10'
XGBoost	learning_rate=0.01, max_depth=7, n_estimators=500, subsample=0.7

2.7 Đánh giá mô hình

Các mô hình được đánh giá bằng nhiều chỉ số hiệu suất khác nhau nhằm phản ánh toàn diện khả năng dự đoán:

- **Độ chính xác (Accuracy)** – tỷ lệ dự đoán đúng trên toàn bộ mẫu.
- **Độ chính xác và độ bao phủ (Precision, Recall)** – đánh giá hiệu quả nhận diện các trường hợp sống sót.
- **F1-Score** – trung bình điều hòa giữa Precision và Recall.
- **Ma trận nhầm lẫn (Confusion Matrix)** – trực quan hóa kết quả dự đoán đúng/sai.

Hiệu suất của các mô hình được so sánh trên cùng tập kiểm thử để xác định mô hình có khả năng tổng quát hóa tốt nhất.

3 Kết quả và thảo luận

3.1 Kết quả huấn luyện mô hình

Sau khi hoàn tất quá trình tiền xử lý và huấn luyện, bốn mô hình học máy được đánh giá trên tập kiểm thử theo các chỉ số Accuracy, Precision, Recall và F1-score. Bảng 2 trình bày kết quả so sánh hiệu suất của các mô hình.

Table 2: Kết quả so sánh hiệu suất giữa các mô hình học máy

Mô hình	**Accuracy**	**Precision**	**Recall**	**F1-score**
Hồi quy Logistic	0.80	0.80	0.80	0.80
SVM	0.80	0.80	0.79	0.79
Rừng ngẫu nhiên	0.84	0.82	0.82	0.81
XGBoost	**0.82**	**0.83**	**0.87**	**0.85**

Kết quả cho thấy mô hình **XGBoost** đạt hiệu suất cao nhất với độ chính xác 83%, vượt trội hơn so với các mô hình khác. Điều này cho thấy khả năng học tốt của thuật toán boosting trong việc kết hợp nhiều cây quyết định yếu để tạo ra một mô hình mạnh hơn. Đồng thời, quá trình tối ưu siêu tham số bằng *Randomized Search CV* giúp mô hình đạt được khả năng tổng quát hóa tốt hơn, tránh hiện tượng quá khớp (overfitting).

3.2 Phân tích tầm quan trọng của đặc trưng

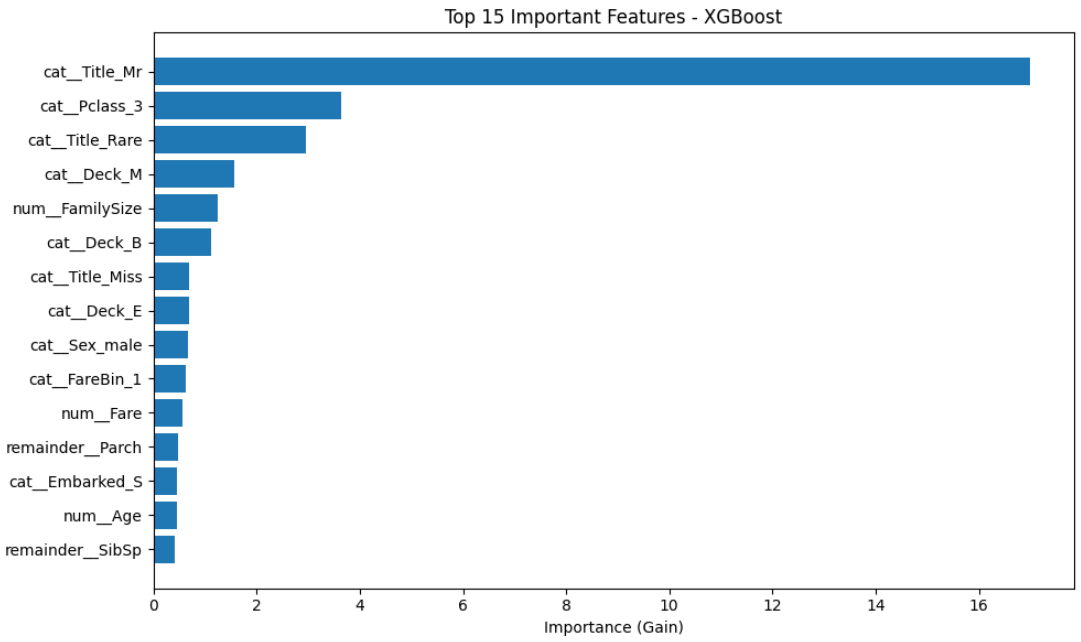

Figure 2: Biểu đồ tầm quan trọng của các đặc trưng trong mô hình XGBoost. Đặc trưng `Sex` và `Pclass` có mức ảnh hưởng cao nhất đến khả năng sống sót.

Phân tích tầm quan trọng của đặc trưng (*feature importance*) từ mô hình XGBoost cho thấy một số biến có ảnh hưởng đáng kể đến khả năng sống sót của hành khách:

- `Sex`: Giới tính là đặc trưng quan trọng nhất, khi nữ có tỷ lệ sống sót cao hơn đáng kể so với nam.
- `Pclass`: Hành khách hạng vé 1 có cơ hội sống sót cao hơn so với hạng vé thấp hơn.
- `Age`: Tuổi có ảnh hưởng rõ rệt, khi hành khách trẻ tuổi có khả năng sống sót cao hơn.
- `Fare`: Giá vé cao phản ánh điều kiện kinh tế tốt, gián tiếp liên quan đến khả năng tiếp cận phương tiện cứu hộ.

Những đặc trưng này phù hợp với các quan sát thực tế về thảm họa Titanic, khi phụ nữ, trẻ em và hành khách ở khoang cao có ưu tiên trong quá trình cứu hộ.

3.3 Thảo luận kết quả

So với các mô hình tuyến tính như Hồi quy Logistic, các mô hình phi tuyến như Rừng ngẫu nhiên và XGBoost thể hiện khả năng nắm bắt tốt hơn các mối quan hệ phức tạp giữa các đặc trưng. Trong khi Cây quyết định đơn lẻ dễ bị quá khớp, việc kết hợp nhiều cây trong Rừng ngẫu nhiên và XGBoost giúp giảm thiểu sai lệch và cải thiện độ chính xác.

Ngoài ra, kết quả cho thấy chất lượng của mô hình phụ thuộc đáng kể vào khâu tiền xử lý, đặc biệt là xử lý giá trị khuyết và chuẩn hóa dữ liệu. Khi các đặc trưng được xử lý đồng nhất và cân bằng, mô hình XGBoost đạt được sự ổn định và hiệu quả cao nhất trên cả tập huấn luyện và kiểm thử.

Tổng thể, nghiên cứu minh chứng rằng việc áp dụng các kỹ thuật học máy hiện đại, đặc biệt là boosting, có thể mang lại hiệu quả vượt trội trong các bài toán phân loại nhị phân ngay cả với tập dữ liệu có quy mô hạn chế.

4 Kết luận và hướng phát triển

Nghiên cứu này đã tiến hành xây dựng và so sánh nhiều mô hình học máy khác nhau nhằm dự đoán khả năng sống sót của hành khách trên tàu Titanic, sử dụng bộ dữ liệu kinh điển từ nền tảng Kaggle. Kết quả cho thấy mô hình **XGBoost** đạt hiệu suất cao nhất với độ chính xác 84%, vượt trội so với các mô hình truyền thống như Hồi quy Logistic, Cây quyết định và Rừng ngẫu nhiên. Điều này chứng minh tính hiệu quả của các thuật toán boosting trong việc xử lý bài toán phân loại nhị phân với dữ liệu có kích thước vừa phải.

Thông qua phân tích tầm quan trọng của đặc trưng, nghiên cứu khẳng định rằng các yếu tố như giới tính, hạng vé và độ tuổi có ảnh hưởng đáng kể đến khả năng sống sót, phù hợp với các quan sát thực tế trong thảm họa Titanic. Bên cạnh đó, quá trình tiền xử lý dữ liệu (xử lý giá trị khuyết, mã hóa biến phân loại, chuẩn hóa đặc trưng) đóng vai trò then chốt trong việc đảm bảo chất lượng đầu vào và cải thiện hiệu suất mô hình.

Mặc dù kết quả đạt được là khả quan, nghiên cứu vẫn tồn tại một số hạn chế, chẳng hạn như kích thước bộ dữ liệu nhỏ, thiếu các đặc trưng phi cấu trúc (như thông tin văn bản từ tên hoặc phòng cabin), và chưa áp dụng các kỹ thuật học sâu (deep learning). Trong tương lai, hướng nghiên cứu có thể mở rộng theo các hướng sau:

- Áp dụng các mô hình học sâu như mạng nơ-ron nhân tạo (ANN) hoặc mạng tích chập (CNN) để khai thác mối quan hệ phi tuyến phức tạp hơn.
- Thực hiện kỹ thuật *feature engineering* nâng cao, chẳng hạn như trích xuất đặc trưng từ tên hành khách hoặc nhóm gia đình.
- Tối ưu hóa mô hình bằng phương pháp *Bayesian Optimization* hoặc *Hyperparameter Tuning* nâng cao.
- Mở rộng tập dữ liệu và thử nghiệm các kỹ thuật học chuyển giao (transfer learning) trong bối cảnh dữ liệu nhỏ.

Tổng kết lại, nghiên cứu đã cho thấy tiềm năng của học máy trong việc mô hình hóa và dự đoán hành vi con người dựa trên dữ liệu thực tế. Bài toán Titanic không chỉ là một ví dụ điển hình trong giảng dạy và nghiên cứu học máy, mà còn minh họa rõ ràng vai trò của dữ liệu, mô hình và quy trình tiền xử lý trong việc xây dựng các hệ thống dự đoán chính xác và đáng tin cậy.

References

[1] Kaggle. (2020). *Titanic: Machine Learning from Disaster*. Truy cập từ: `https://www.kaggle.com/competitions/titanic` (Ngày truy cập: 28/10/2025).

[2] Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). DOI: `https://doi.org/10.1145/2939672.2939785`

[3] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

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