|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM  BÀI TẬP LỚN TRÍ TUỆ NHÂN TẠO  **THUẬT TOÁN RỪNG NGẪU NHIÊN**  **ỨNG DỤNG TRONG BÀI TOÁN PHÂN LỚP**  Ngành: Công nghệ phần mềm  Mã số: 7.48.02.02  *Sinh viên thực hiện*:  **Nguyễn Vũ Khương**  **Nguyễn Quang Trung Nhân**  Lớp: CT6N  *Người hướng dẫn 1*:  **ThS. Nguyễn Minh Đế**  Khoa An toàn thông tin – Học viện Kỹ thuật mật mã  Hồ Chí Minh, 2024 |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM  BÀI TẬP LỚN TRÍ TUỆ NHÂN TẠO  **THUẬT TOÁN RỪNG NGẪU NHIÊN**  **ỨNG DỤNG TRONG BÀI TOÁN PHÂN LỚP**  Ngành: Công nghệ phần mềm  Mã số: 7.48.02.02  *Sinh viên thực hiện*:  **Nguyễn Vũ Khương**  **Nguyễn Quang Trung Nhân**  Lớp: CT6N  *Người hướng dẫn 1*:  **ThS. Nguyễn Minh Đế**  Khoa An toàn thông tin – Học viện Kỹ thuật mật mã  Hồ Chí Minh, 2024 |

Mục lục

[DANH MỤC KÝ HIỆU VÀ VIẾT TẮT ii](#_Toc169039411)

[DANH MỤC HÌNH VẼ iii](#_Toc169039412)

[DANH MỤC BẢNG iv](#_Toc169039413)

[Lời cảm ơn v](#_Toc169039414)

[Lời nói đầu vi](#_Toc169039415)

[Chương 1. GIỚI THIỆU BÀI TOÁN PHÂN LỚP 1](#_Toc169039416)

[1.1. Giới thiệu bài toán 1](#_Toc169039417)

[1.2. Xác định yêu cầu, đầu vào, đầu ra 1](#_Toc169039418)

[1.3. Kết luận 2](#_Toc169039419)

[Chương 2. THUẬT TOÁN RỪNG NGẪU NHIÊN 3](#_Toc169039420)

[2.1. Giới thiệu thuật toán 3](#_Toc169039421)

[2.1.1. Khái niệm cơ bản 3](#_Toc169039422)

[2.1.2. Các bước thực hiện của thuật toán Random Forest 4](#_Toc169039423)

[2.2. Tiền xử lý dữ liệu 6](#_Toc169039424)

[2.2.1. Phân tích thống kê dữ liệu (EDA) 6](#_Toc169039425)

[2.2.2. Feature Engineering 9](#_Toc169039426)

[2.3. Xây dựng mô hình 11](#_Toc169039427)

[2.4. Kết luận 11](#_Toc169039428)

[Chương 3. KẾT QUẢ 12](#_Toc169039429)

[3.1. Đánh giá mô hình 12](#_Toc169039430)

[3.2. Feature Importance 13](#_Toc169039431)

[3.3. Triển khai mô hình dự đoán với Django 14](#_Toc169039432)

[3.3.1. Mô tả dự án 14](#_Toc169039433)

[3.3.2. Triển khai Demo 14](#_Toc169039434)

[3.4. Kết luận 16](#_Toc169039435)

[Chương 4. KẾT LUẬN 17](#_Toc169039436)

[Tài liệu tham khảo 18](#_Toc169039437)

[Phụ lục 19](#_Toc169039438)

DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

DANH MỤC HÌNH VẼ

[Figure 1: Mô tả tập dữ liệu huấn luyện 2](#_Toc169039384)

[Figure 2: Mô tả cách hoạt động của mô hình Random forest 4](#_Toc169039385)

[Figure 3: Thông tin hai hành khách bị thiếu nơi lên tàu 7](#_Toc169039386)

[Figure 4: Xử lý các giá trị Age bị thiếu 7](#_Toc169039387)

[Figure 5: Mô tả vị trí các loại cabin 8](#_Toc169039388)

[Figure 6: Xử lý các biến Cabin bị thiếu 8](#_Toc169039389)

[Figure 7: Gom nhóm biến Fare thành 13 nhóm khác nhau 9](#_Toc169039390)

[Figure 8: Gom nhóm biến Agr thành 10 nhóm khác nhau 9](#_Toc169039391)

[Figure 9: Xử lý Is\_Married 9](#_Toc169039392)

[Figure 10: Xử lý biến Title 10](#_Toc169039393)

[Figure 11: Mô tả mô hình sau khi đã được tiền xử lý 10](#_Toc169039394)

[Figure 12: Mô tả các tham số của mô hình 11](#_Toc169039395)

[Figure 13: Mô tả điểm đánh giá đối với tập dữ liệu train, test fold\_1 12](#_Toc169039396)

[Figure 14: Mô tả các đánh giá độ quan trong của các biến 13](#_Toc169039397)

[Figure 15: Mô tả dữ liệu test 14](#_Toc169039398)

[Figure 16: Kết quả dự đoán đối với hành khách thứ nhất 14](#_Toc169039399)

[Figure 17: Kết quả dự đoán đối với hành khách thứ ba 15](#_Toc169039400)

[Figure 18: Kết quả dự đoán đối với hành khách thứ hai 15](#_Toc169039401)

[Figure 19 Kết quả dự đoán đối với hành khách thứ 4 16](#_Toc169039402)

[Figure 20: Kết quả dự đoán đối với hành khách thứ 5 16](#_Toc169039403)

DANH MỤC BẢNG

[Table 1‑1: Mô tả tập dữ liệu huấn luyện 1](#_Toc169039404)

[Table 2‑1: Mô tả sơ bộ dữ liệu huấn luyện 6](#_Toc169039405)

[Table 2‑2: Mô tả missing value cho từng biến 6](#_Toc169039406)

[Table 2‑3: Chuyển đổi giá trị biến Cabin sang biến Deck 8](#_Toc169039407)

[Table 2‑4: Bảng chuyển đổi biến Family\_Size\_Group 9](#_Toc169039408)

[Table 2‑5: Giải thích các tham số xây dựng mô hình 11](#_Toc169039409)

[Table 3‑1: Điểm đánh giá mô hình cho từng tập data 12](#_Toc169039410)

Lời cảm ơn

Trong quá trình thực hiện bài tập lớn này, em đã nhận được sự giúp đỡ tận tình của cán bộ hướng dẫn là ThS. Nguyễn Minh Đế – Giảng viên Khoa Công nghệ phần mềm Học viện Kỹ thuật Mật mã, sự quan tâm sâu sát của cán bộ Hệ quản lý học viên, sự động viên của người thân và bạn bè.

Xin cảm ơn tất cả mọi người đã tạo những điều kiện tốt nhất để em hoàn thành bài tập lớn này!

|  |  |
| --- | --- |
|  | **SINH VIÊN THỰC HIỆN ĐỒ ÁN**  Nguyễn Vũ Khương  Nguyễn Quang Trung Nhân |

Lời nói đầu

Phân lớp là một trong những vấn đề cơ bản của trí tuệ nhân tạo nói chung và học máy nói riêng, hiện nay có nhiều thuật toán phân lớp phát triển và ứng dụng rất nhiều để giải quyết các bài toán phức tạp khác. Nhóm sử dụng giải thuật Random Forest nhằm mục đích xây dựng một hệ thống dự đoán tỉ lệ sống sót của hành khách đi trên chuyến tàu titanic. Trong quá trình thực hiện đề tài, không thể tránh khỏi các sai sót, em mong xin nhận được góp ý và đánh giá của thầy

|  |  |
| --- | --- |
|  | **SINH VIÊN THỰC HIỆN ĐỒ ÁN**  Nguyễn Vũ Khương  Nguyễn Quang Trung Nhân |

# GIỚI THIỆU BÀI TOÁN PHÂN LỚP

## Giới thiệu bài toán

Vụ đắm tàu Titanic gây ra cái chết cho rất nhiều người, nó gây ra hậu quả rất lớn về người và của, và để lại một bài học quý giá về tầm quan trọng của an toàn hàng hải và sự chuẩn bị đầy đủ trong mọi tình huống khẩn cấp. Tưởng tượng rằng, ta là một hành khách trên một con tàu tương tự về mọi mặt so với Titanic và nó đang sắp chìm vào đại dương, liệu ta có thể dự đoán được rằng mình có thể sống sót được hay không? Và đây cũng là bài toán mà nhóm đặt ra – “Đự đoán khả năng sống sót của hành khách trên tàu Titanic”.

## Xác định yêu cầu, đầu vào, đầu ra

Yêu cầu của bài toán được đặt ra là từ danh sách các thông tin của các hành khách trên tàu Titanic, xây dựng được một mô hình có khả năng dự đoán khả năng sống sót của một hành khách, hay nói cách khác hành khách đó là sống hay sẽ chết nếu lên tàu.

Ta sẽ xây dựng mô hình với hai tập dữ liệu , bao gồm các biến như sau:

Table 1‑1: Mô tả tập dữ liệu huấn luyện

|  |  |
| --- | --- |
| Tên biến | Mô tả |
| Survived | Hành khách đó có sống sót hay không |
| Pclass | Hạng vé |
| Sex | Giới tính |
| SibSp | Tổng số anh chị em và vợ/chồng của hành khách |
| Ticket | Vé |
| Name | Tên hành khách |
| Age | Tuổi |
| Parch | Tổng số cha mẹ hoặc con cái |
| Fare | Giá vé |
| Embarked | Nơi lên tàu |
| Cabin | Mã cabin |

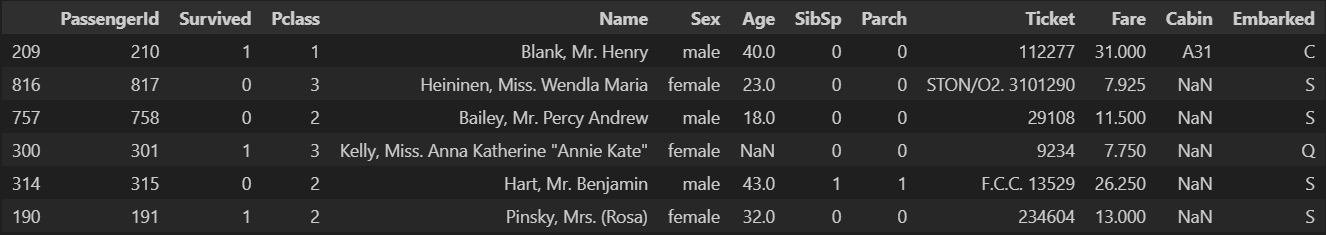


Figure 1: Mô tả tập dữ liệu huấn luyện

Đầu vào của mô hình là một ma trận dữ liệu huấn luyện và một vector nhãn là các biến từ tập sau khi đã được tiền xử lý.

Đầu ra của bài toán là một vector là các dự đoán khi ta truyền vào một tập dữ liệu vào mô hình sau khi đã huấn luyện.

## Kết luận

Như vậy, ở chương 1 nhóm đã xác định yêu cầu bài toán, đầu vào để xây dựng mô hình, đầu ra của thuật toán, từ đây ta sẽ tiếp cận đến với thuật toán Rừng ngẫu nhiên để giải quyết bài toán trên ở Chương 2.

# THUẬT TOÁN RỪNG NGẪU NHIÊN

Như đã trình bày ở Chương 1, đầu ra của bài toán là một vector với các phần tử chỉ chứa các giá trị 0 hoặc 1, rõ ràng đây là một bài toán phân lớp nhị phân điển hình. Ta có thể sử dụng các thuật toán phân lớp như K-Nearest Neighbors (KNN), Decision Tree,… Ở đây để có được hiệu quả cao, giảm thiểu overfitting, mà không quan tâm tới hiệu năng, nhóm sẽ dùng thuật toán Random Forest để giải quyết bài toán này.

## Giới thiệu thuật toán

### Khái niệm cơ bản

Thuật toán Rừng ngẫu nhiên là một kỹ thuật học cây mạnh mẽ trong Machine Learning . Nó hoạt động bằng cách tạo ra một số Cây quyết định trong giai đoạn đào tạo. Mỗi cây được xây dựng bằng cách sử dụng một tập hợp con ngẫu nhiên của tập dữ liệu để đo lường một tập hợp con ngẫu nhiên các tính năng trong mỗi phân vùng. Tính ngẫu nhiên này tạo ra sự biến đổi giữa các cây riêng lẻ, giảm nguy cơ trang bị quá mức và cải thiện hiệu suất dự đoán tổng thể. Trong dự đoán, thuật toán tổng hợp kết quả của tất cả các cây, bằng cách bỏ phiếu (đối với nhiệm vụ phân loại) hoặc bằng cách tính trung bình (đối với nhiệm vụ hồi quy). Quá trình ra quyết định hợp tác này, được hỗ trợ bởi nhiều cây với thông tin chuyên sâu của chúng, cung cấp một ví dụ về kết quả ổn định và chính xác . Rừng ngẫu nhiên được sử dụng rộng rãi cho các chức năng phân loại và hồi quy, được biết đến với khả năng xử lý dữ liệu phức tạp, giảm tình trạng trang bị quá mức và cung cấp dự báo đáng tin cậy trong các môi trường khác nhau.

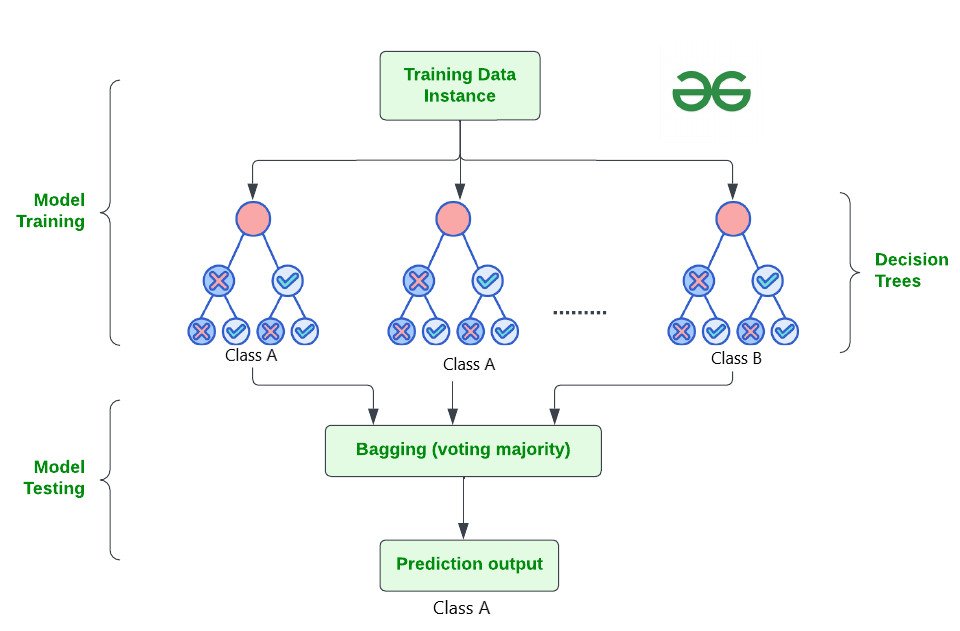


Figure 2: Mô tả cách hoạt động của mô hình Random forest

### Các bước thực hiện của thuật toán Random Forest

#### Bước 1: Tạo tập dữ liệu huấn luyện ngẫu nhiên (Bootstrapping)

Từ tập dữ liệu huấn luyện gốc với nnn mẫu, tạo tập dữ liệu ngẫu nhiên bằng cách chọn ngẫu nhiên có thay thế từ Mỗi tập dữ liệu có cùng kích thước với .

#### Bước 2: Xây dựng các cây quyết định

Đối với mỗi tập dữ liệu ​, xây dựng một cây quyết định. Trong quá trình xây dựng mỗi cây, tại mỗi nút phân chia:

Chọn một tập con ngẫu nhiên đặc trưng từ tất cả đặc trưng.

Chọn đặc trưng tốt nhất từ tập con này để phân chia dữ liệu tại nút.

##### Quá trình xây dựng cây quyết định

Cây quyết định được xây dựng theo phương pháp đệ quy, với các bước chính như sau:

**Chia tách dữ liệu (Splitting)**: Dữ liệu được chia thành các tập con dựa trên một thuộc tính và giá trị ngưỡng (hoặc các giá trị phân loại). Mục tiêu là chọn thuộc tính và giá trị ngưỡng sao cho chất lượng chia tách tốt nhất, tức là làm giảm độ hỗn loạn của dữ liệu sau khi chia tách.

**Chọn thuộc tính và giá trị ngưỡng tốt nhất**: Để đánh giá chất lượng chia tách, chúng ta sử dụng các tiêu chí như:

**Entropy và Thông tin thu được (Information Gain)**:

Trong đó:

* là tập dữ liệu hiện tại.
* ​ là xác suất của lớp thứ trong tập .
* là thuộc tính được xem xét để chia tách.
* ​ là tập con của khi thuộc tính có giá trị .

**Gini impurity**:

Gini impurity đo lường xác suất mà một mẫu được chọn ngẫu nhiên sẽ bị phân loại sai nếu nó được gán nhãn theo phân phối của nhãn trong tập dữ liệu.

##### Đệ quy xây dựng cây

Quá trình chia tách tiếp tục đệ quy cho mỗi tập con cho đến khi đạt điều kiện dừng, chẳng hạn như:

* Tất cả các mẫu trong tập con thuộc cùng một lớp.
* Số lượng mẫu trong tập con nhỏ hơn một ngưỡng nhất định.
* Độ sâu của cây đạt đến một ngưỡng nhất định.

#### Bước 3: Kết hợp các cây quyết định

Sau khi xây dựng xong cây quyết định, sử dụng chúng để dự đoán.

**Phân loại:** Mỗi cây trong rừng ngẫu nhiên đưa ra một dự đoán phân loại và dự đoán cuối cùng là nhãn được đa số các cây bầu chọn.

trong đó là dự đoán của cây thứ và là số cây trong rừng.

**Hồi quy:** Mỗi cây đưa ra một dự đoán giá trị liên tục và dự đoán cuối cùng là trung bình của các dự đoán từ các cây.

## Tiền xử lý dữ liệu

### Phân tích thống kê dữ liệu (EDA)

#### Tổng quan

Đây là bộ dữ liệu có 11 cột và 890 dòng gồm các thông tin được mô tả dưới bảng sau:

Table 2‑1: Mô tả sơ bộ dữ liệu huấn luyện

|  |  |  |
| --- | --- | --- |
| Tên cột | Số lượng giá trị không null | Kiểu dữ liệu |
| PassengerId | 891 non null | int64 |
| Survived | 891 non null | int64 |
| Pclass | 891 non null | int64 |
| Name | 891 non null | object |
| Sex | 891 non null | object |
| Age | 714 non null | float64 |
| SibSp | 891 non null | int64 |
| Parch | 891 non null | int64 |
| Ticket | 891 non null | object |
| Fare | 891 non null | float64 |
| Cabin | 204 non null | object |
| Embarked | 889 non null | object |

#### Dữ liệu thiếu

Table 2‑2: Mô tả missing value cho từng biến

|  |  |
| --- | --- |
| Tên cột | Số lượng giá trị bị thiếu |
| PassengerId | 0 |
| Survived | 0 |
| Pclass | 0 |
| Name | 0 |
| Sex | 0 |
| Age | 177 |
| SibSp | 0 |
| Parch | 0 |
| Ticket column missing values | 0 |
| Fare column missing values | 0 |
| Cabin column missing values | 687 |
| Embarked column missing values | 2 |

##### Embarked

Ta có thể xử lý dữ liệu bị thiếu của Embarked bằng cách xóa hai hàng đó đi hoặc có thể thay thế bằng giá trị mode, ở đây nhóm tìm thông tin của hai người này trên [Martha Evelyn Stone: Titanic Survivor](https://www.encyclopedia-titanica.org/titanic-survivor/martha-evelyn-stone.html) và tìm được hai người này đi cùng nhau và họ lên tàu từ cảng Southampton

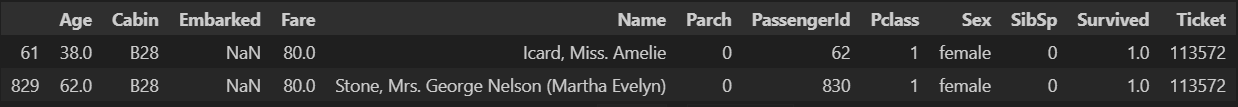


Figure 3: Thông tin hai hành khách bị thiếu nơi lên tàu

##### Age

Ta xử lý các giá trị tuổi bị thiếu bằng cách lấy giá trị trung vị số tuổi của các hành khách theo hạng vé và giới tính như hình dưới:

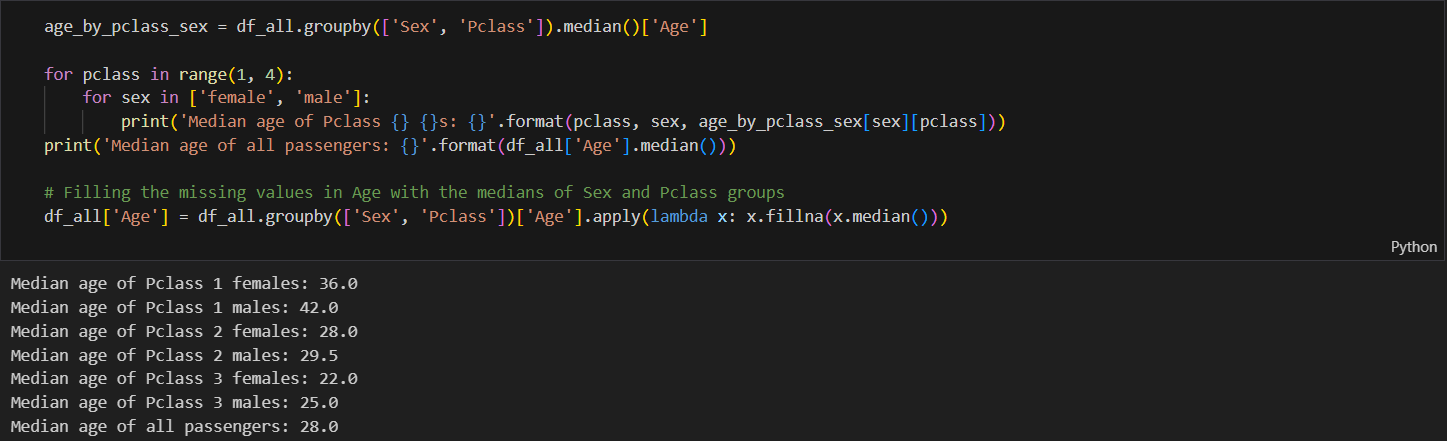


Figure 4: Xử lý các giá trị Age bị thiếu

##### Cabin

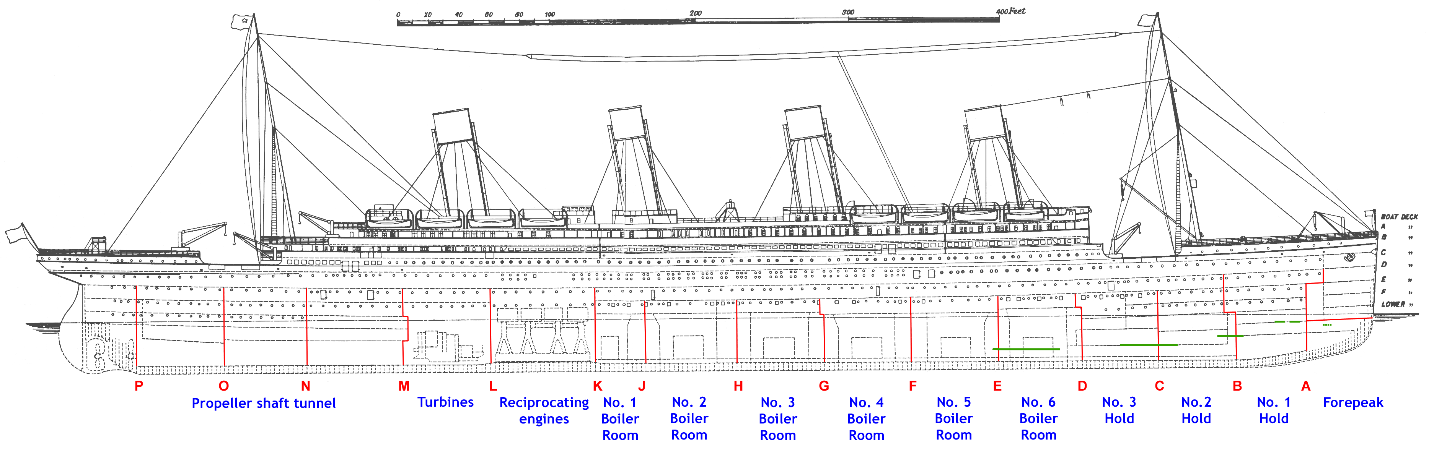


Figure 5: Mô tả vị trí các loại cabin

Các chữ cái đầu tiên trong biến cabin biểu diễn cho các vị trí A,B,C,D,E,F,G,H,J,K,L,M,N,O,P,..

Nhóm xử lý bằng cách tạo thêm một biến Deck và gán giá trị theo như hình dưới đây

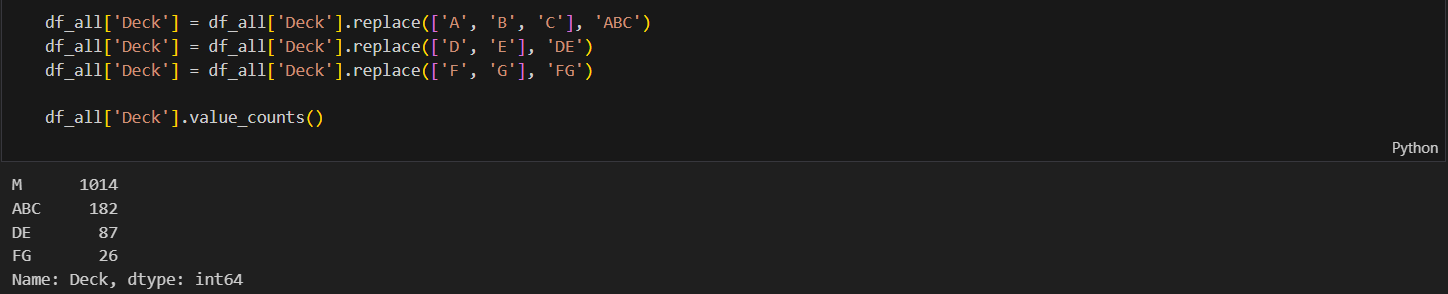


Figure 6: Xử lý các biến Cabin bị thiếu

Trong đó, các hành khách có Cabin là:

Table 2‑3: Chuyển đổi giá trị biến Cabin sang biến Deck

|  |  |  |
| --- | --- | --- |
| Cabin | Deck | Số lượng |
| Mising value | M | 1014 |
| A,B,B | ABC | 182 |
| D,E | DE | 87 |
| F,G | FG | 26 |

### Feature Engineering

#### Góm nhóm biến Fare và Age

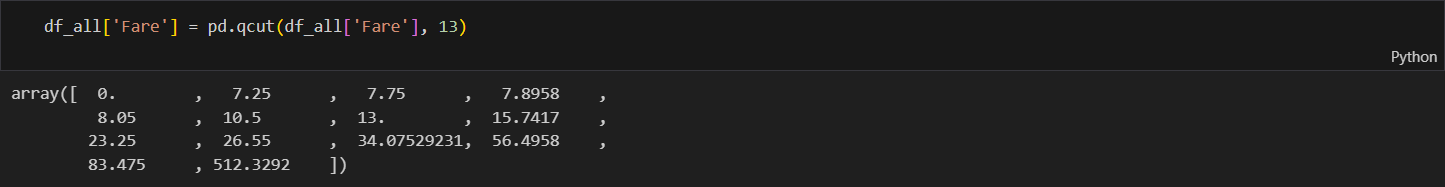


Figure 7: Gom nhóm biến Fare thành 13 nhóm khác nhau

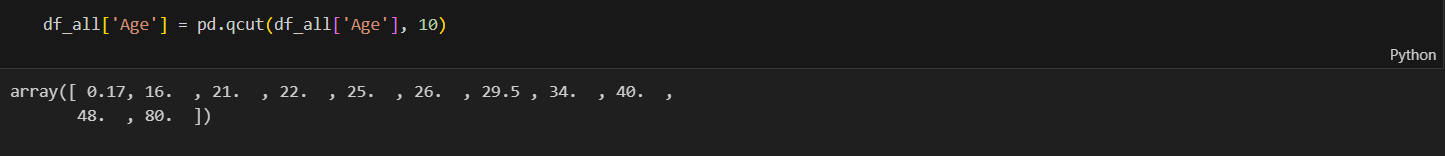


Figure 8: Gom nhóm biến Agr thành 10 nhóm khác nhau

#### Tạo biến Family\_Size\_Group và Ticket\_Frequency

Đầu tiên ta tạo biến , chuyển đổi thành các giá trị theo bảng sau:

Table 2‑4: Bảng chuyển đổi biến Family\_Size\_Group

|  |  |
| --- | --- |
| Số lượng thành viên | Phân loại |
| 1 | Alone |
| 2,3,4 | Small |
| 5,6 | Medium |
| 7,8,11 | Large |

Tiếp theo ta tạo biến Ticket\_Frequency bằng cách tính tổng số người có vé giống nhau bằng câu lệnh:

df\_all['Ticket\_Frequency'] = df\_all.groupby('Ticket')['Ticket'].transform('count')

#### Title và Is\_Married

Tạo cột Is\_Married có giá trị bằng 0, nếu trong tên của hành khách có chữ Mrs thì có giá trị bằng 1:

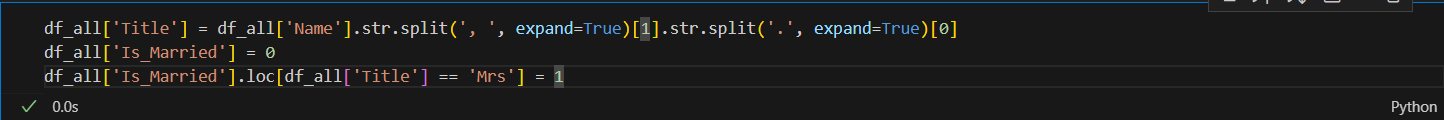


Figure 9: Xử lý Is\_Married

Đối với biến Title ta tạo thành 4 nhóm lớn theo danh xưng như hình dưới đây:

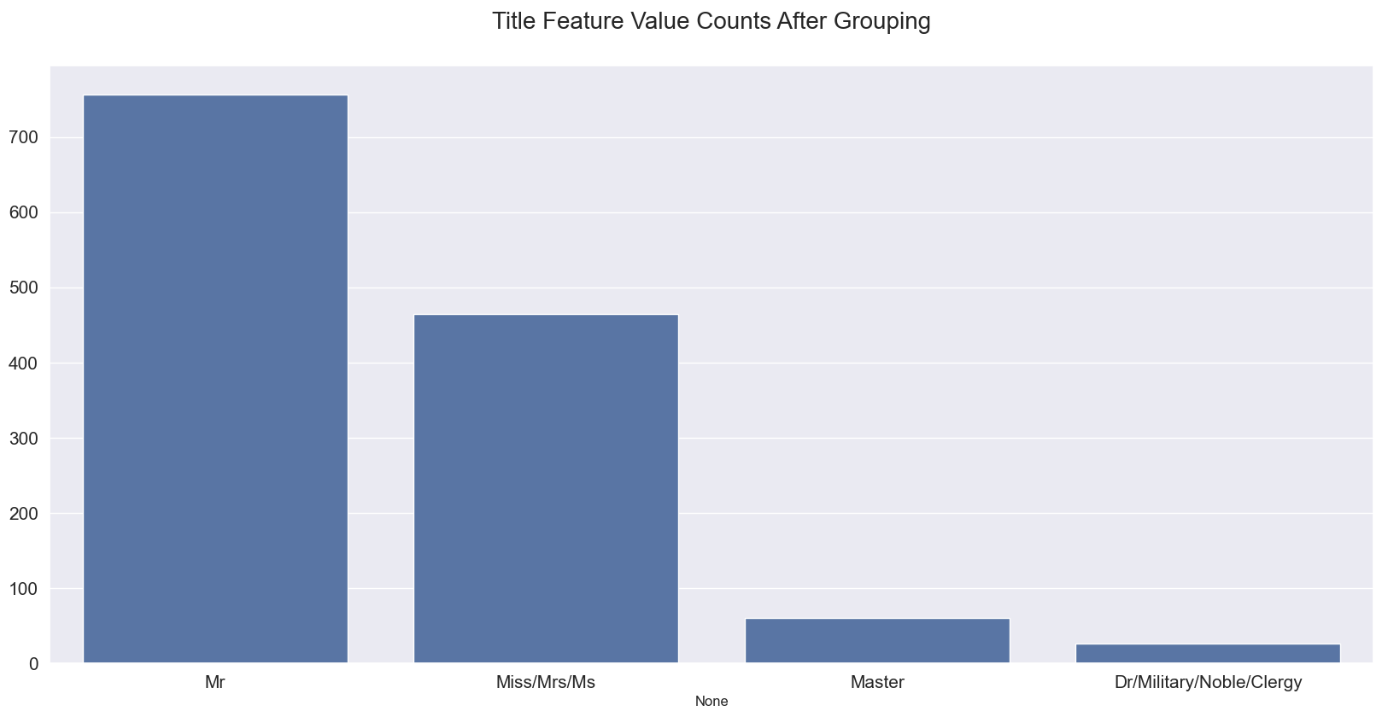


Figure 10: Xử lý biến Title

#### Survived\_Rate

Ta tạo biến Survived bằng cách tính trung bình tỉ lệ sống sót trung bình của từng gia đình và trung bình tỉ lệ số sót của từng nhóm người cùng vé

#### Lable encoding

Ta sử dụng lable encoding có các biến sau:

['Embarked', 'Sex', 'Deck', 'Title', 'Family\_Size\_Grouped', 'Age', 'Fare']

#### One-host encoding

Ta sử dụng One-host encoding cho các biến sau:

['Pclass', 'Sex', 'Deck', 'Embarked', 'Title', 'Family\_Size\_Grouped']

#### Tóm lại

Ta sẽ có được một mô hình gồm 26 cột như sau:

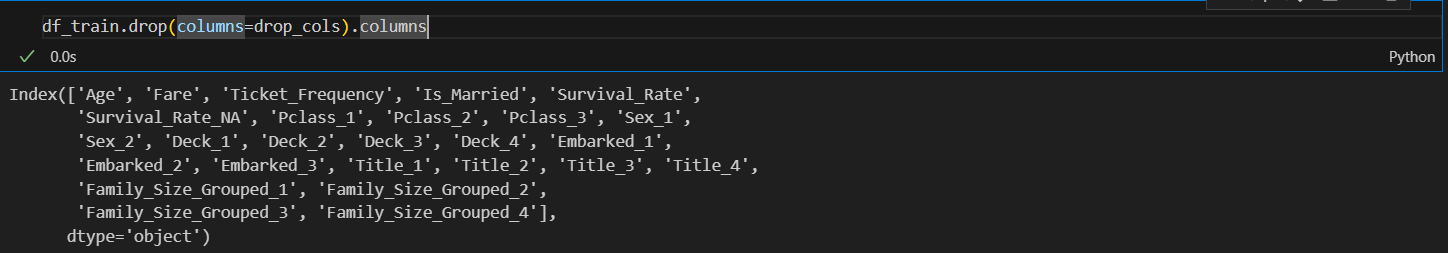


Figure 11: Mô tả mô hình sau khi đã được tiền xử lý

## Xây dựng mô hình

Ta sẽ xây dựng một mô hình với các tham số:

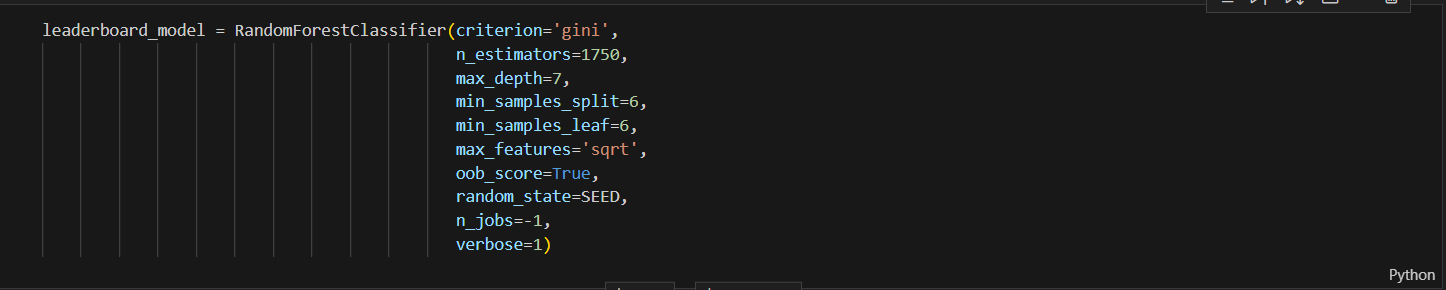


Figure 12: Mô tả các tham số của mô hình

Table 2‑5: Giải thích các tham số xây dựng mô hình

|  |  |  |
| --- | --- | --- |
| Số lượng cây quyết định | 1750 | Số lượng đủ lớn để có thể giảm thiểu overfiting |
| Độ sâu tối đa | 7 | Độ sâu càng lớn, cây quyết định càng chi tiết và có thể mô hình hóa dữ liệu tốt hơn nhưng cũng dễ bị overfitting. Ở đây, mỗi cây được giới hạn ở độ sâu 7 |
| Số lượng mẫu tối thiểu | 6 | Số lượng mẫu tối thiểu cần thiết để chia một nút. |
| Số lượng thuộc tính tối đa được xem xét khi tìm kiếm điểm chia tốt nhất | 'sqrt' | Tại mỗi nút, số lượng thuộc tính được xem xét bằng căn bậc hai của tổng số thuộc tính. Điều này giúp giảm tính toán và làm tăng tính ngẫu nhiên của mô hình, giúp giảm overfitting. |

## Kết luận

Như vậy, ở chương 2 nhóm đã trình bày các bước thực hiện thuật toán Random Forest, đã thực hiện các bước tiền xử lý dữ liệu cần thiết, từ đó xây dựng một mô hình Rừng Ngẫu nhiên. Chương 3 là phần mà nhóm trình bày việc kiểm tra, đánh giá mô hình đã xây dựng.

# KẾT QUẢ

## Đánh giá mô hình

Table 3‑1: Điểm đánh giá mô hình cho từng tập data

|  |  |  |
| --- | --- | --- |
| Fold | Điểm đánh giá cho tập từng tập train | Điểm đánh giá cho từng tập test |
| 1 | 0.9325431591946398 | 0.8658761528326745 |
| 2 | 0.9264253529089005 | 0.9141711229946525 |
| 3 | 0.9308897128510384 | 0.8814171122994654 |
| 4 | 0.9233826047919125 | 0.9140374331550803 |
| 5 | 0.924067599067599 | 0.905464698843239 |

Để đánh giá mô hình có hiệu quả hay không, ta càn kiểm tra điểm đánh giá cho từng tập dữ liệu train và test. Có thể thấy rằng kết quả rất khả quan khi các dự đoán của các tập điều chính xác trên 85% có. Đây là một mức điểm rất tốt và có thể áp dụng để triển khai vào các dự án.

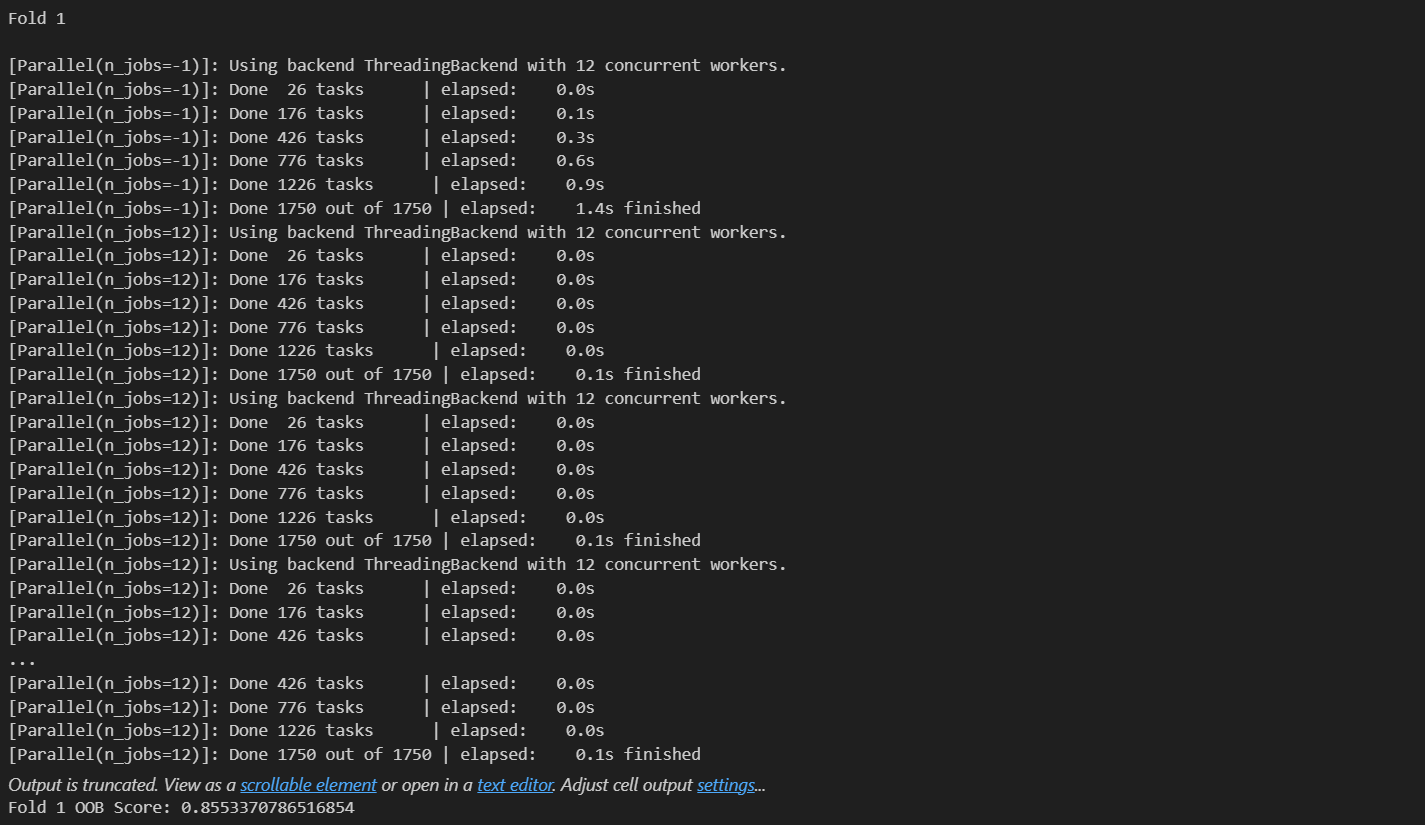


Figure 13: Mô tả điểm đánh giá đối với tập dữ liệu train, test fold\_1

## Feature Importance

Một chức năng rất quan trọng của Random Forest là nó cho phép ta có thể xem đánh giá các giá trị đo độ quan trọng của các biến, từ đó có thể cải thiện độ chính xác của mô hình hơn như hình dưới đây:

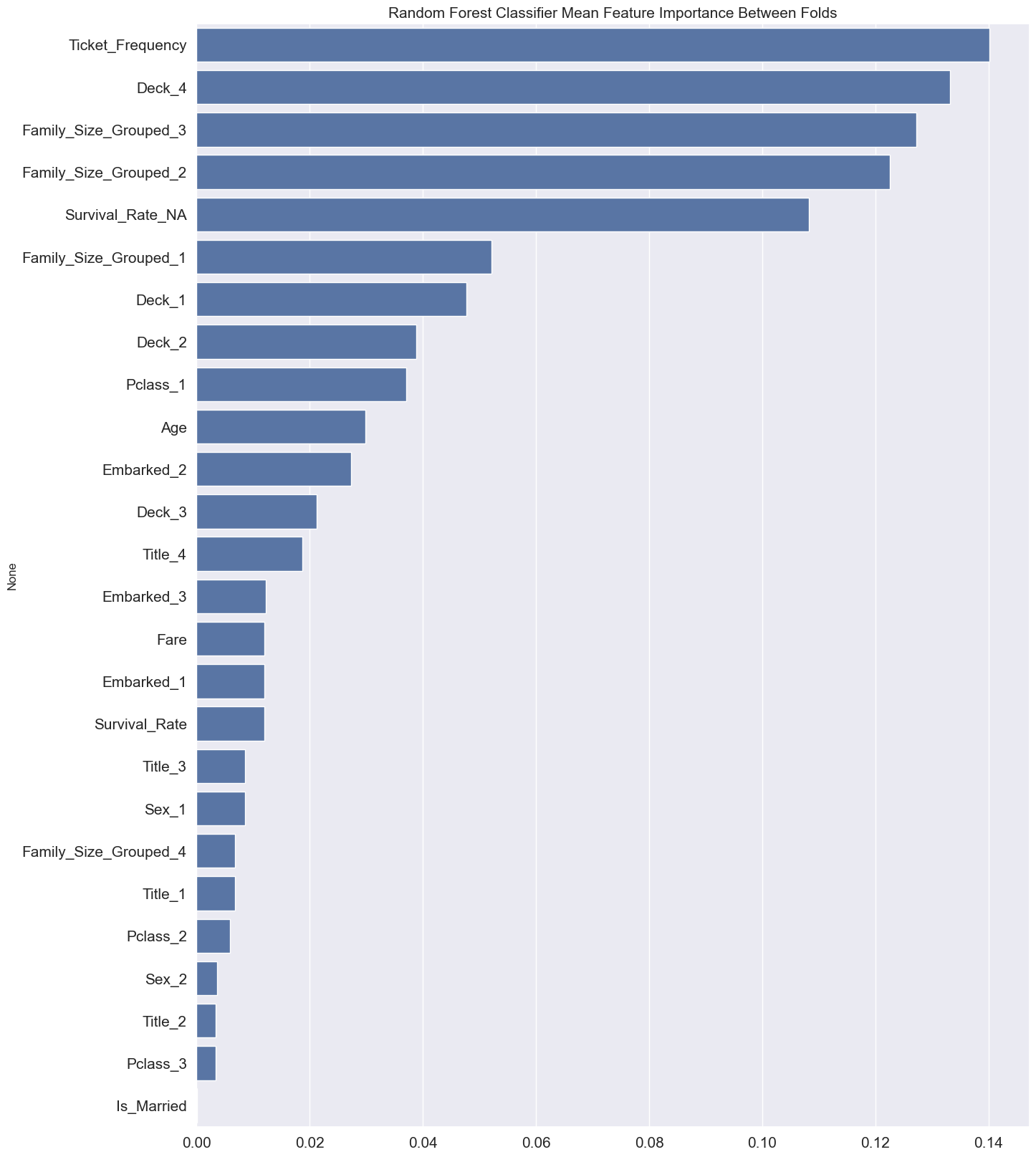


Figure 14: Mô tả các đánh giá độ quan trong của các biến

Ta có thể thấy rằng biến Is\_Married có độ quan trọng không đáng kể, ta hoàn toàn có thể loại bỏ, việc này không làm ảnh hưởng để độ chính xác của mô hình, nhưng giúp tăng tốc độ xử lý.

## Triển khai mô hình dự đoán với Django

### Mô tả dự án

Nhóm xây dựng một trang web ứng dụng mô hình đã xây dựng đơn giản, với chức năng đặt vé tàu. Tại đó, người dùng sẽ nhập các thông tin cần thiết gửi về Server, Server tiếp nhập thông tin, sử dụng mô hình dự đoán người đó sẽ sống hay chết nếu lên tàu và trả về kết quả.

### Triển khai Demo

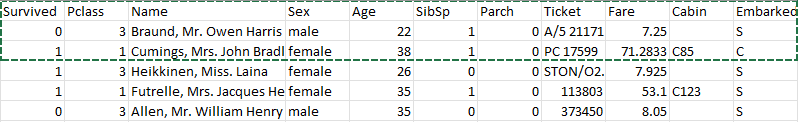


Figure 15: Mô tả dữ liệu test

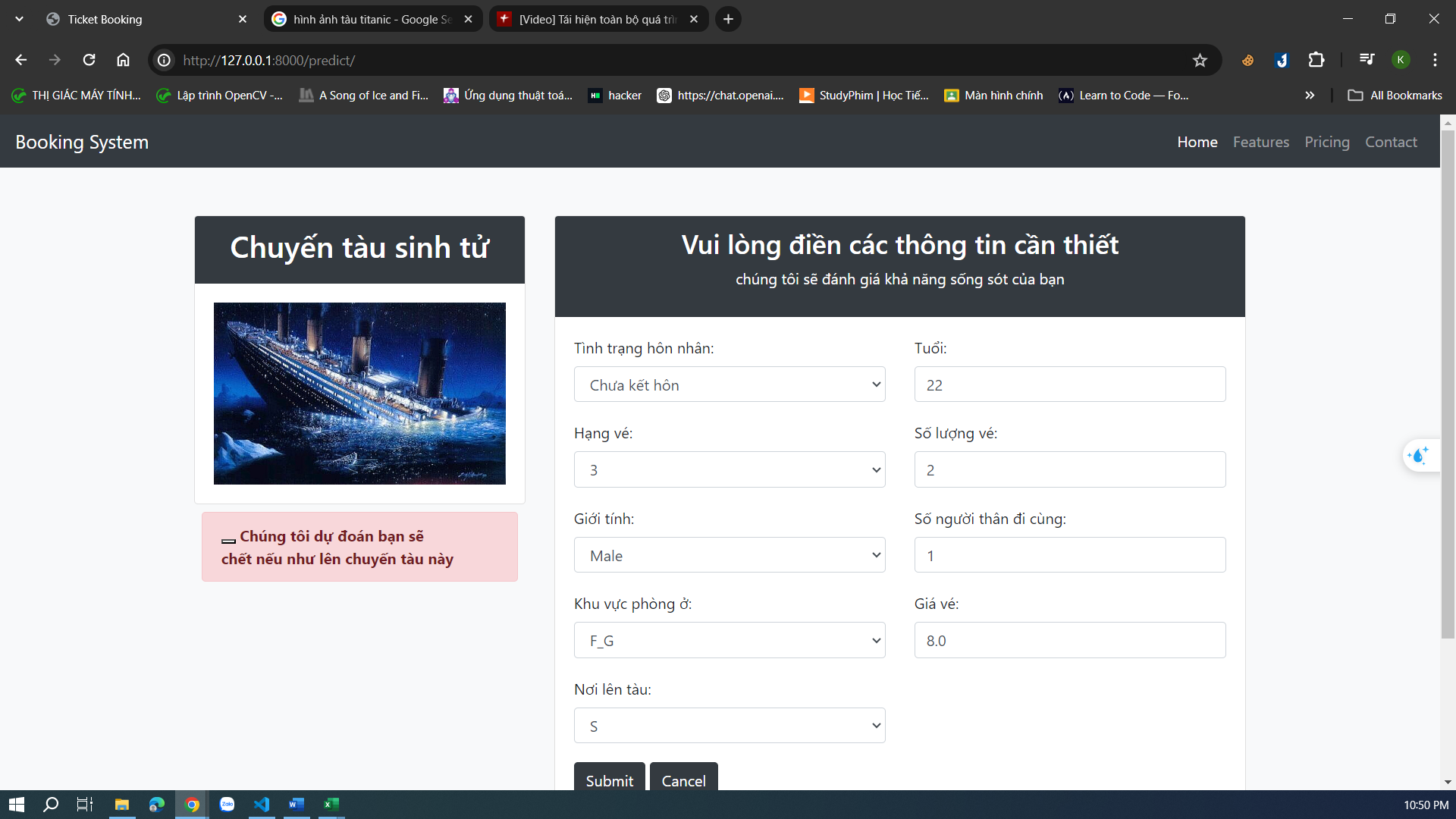


Figure 16: Kết quả dự đoán đối với hành khách thứ nhất

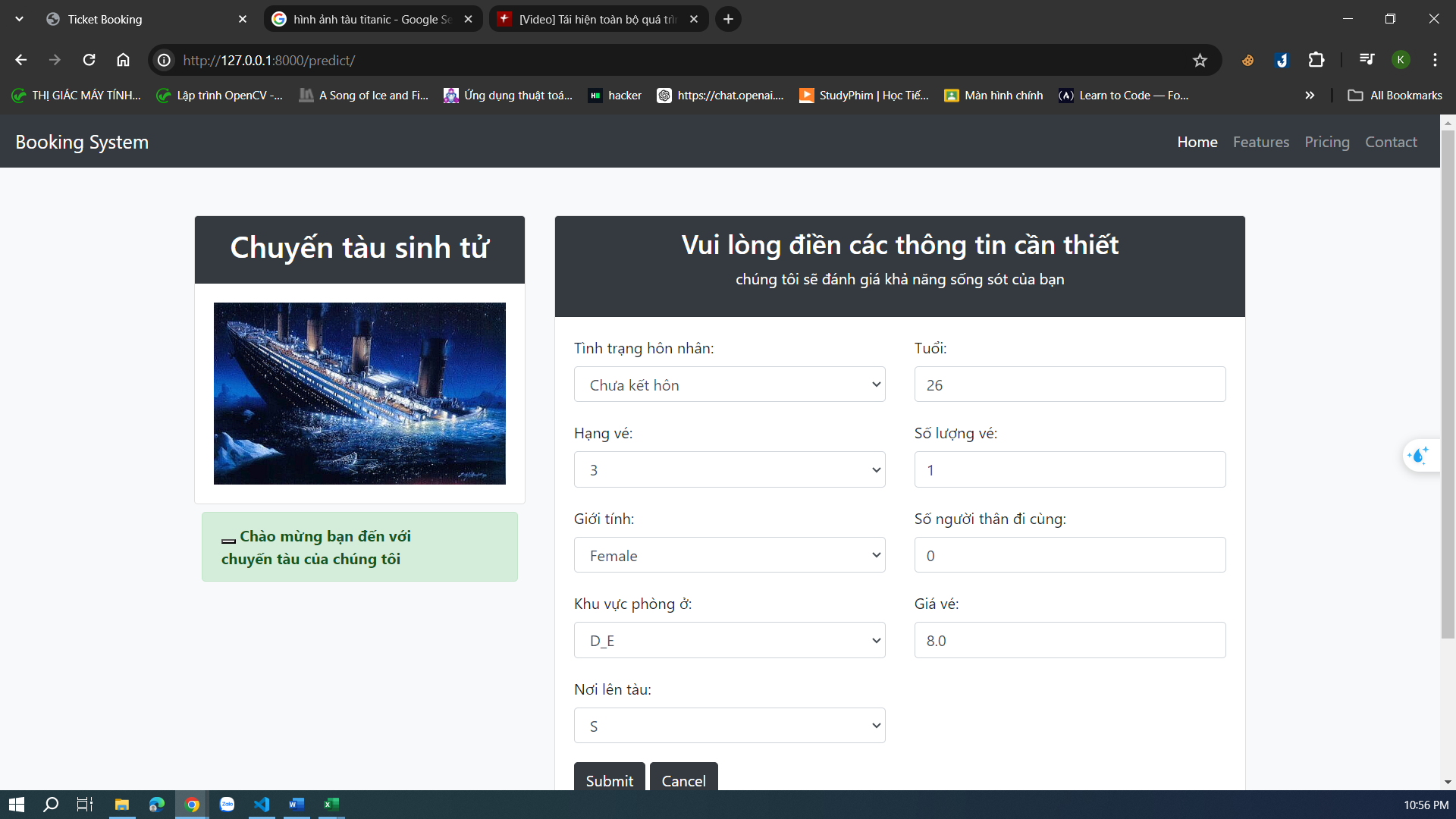


Figure 17: Kết quả dự đoán đối với hành khách thứ ba

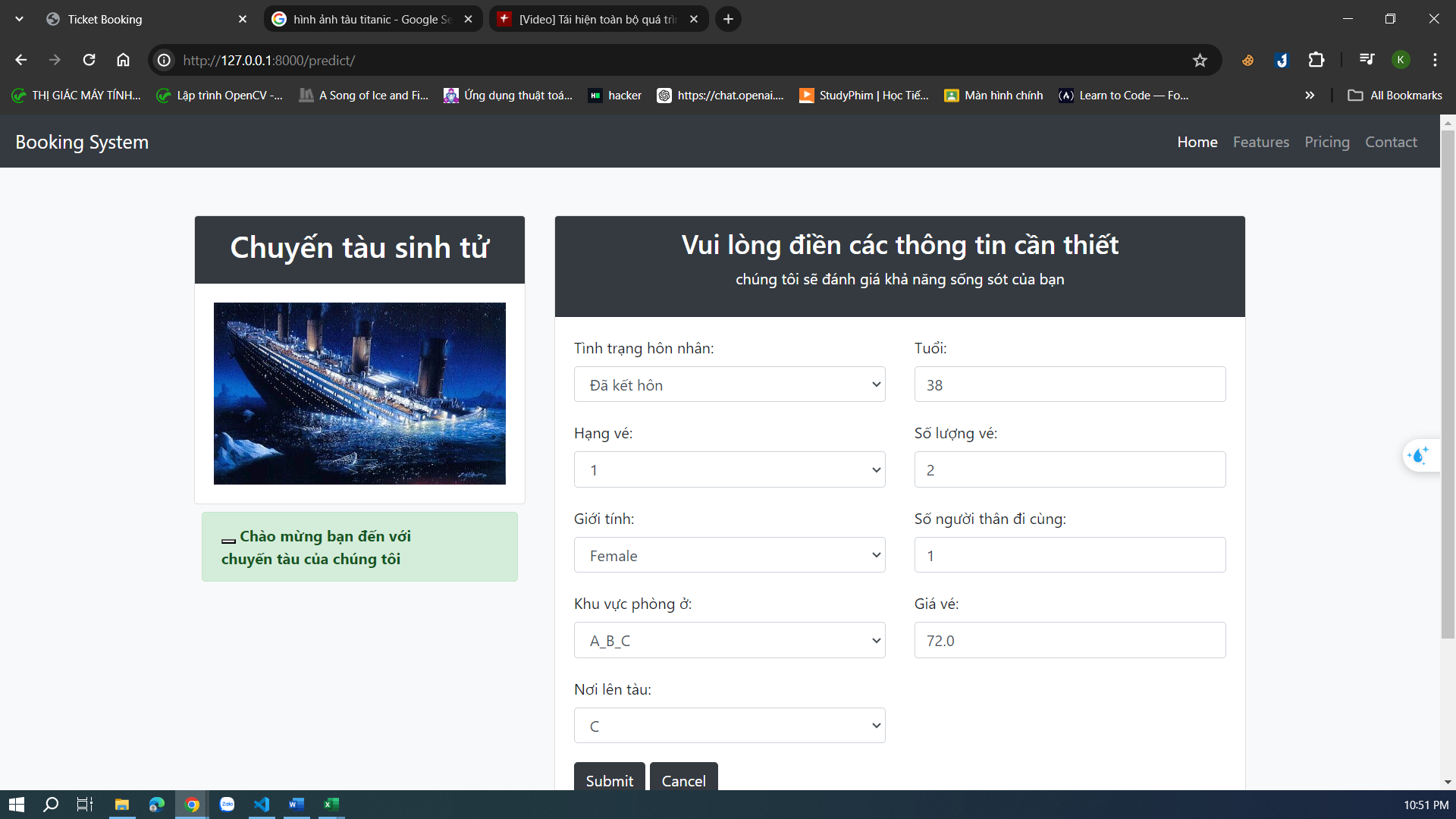


Figure 18: Kết quả dự đoán đối với hành khách thứ hai

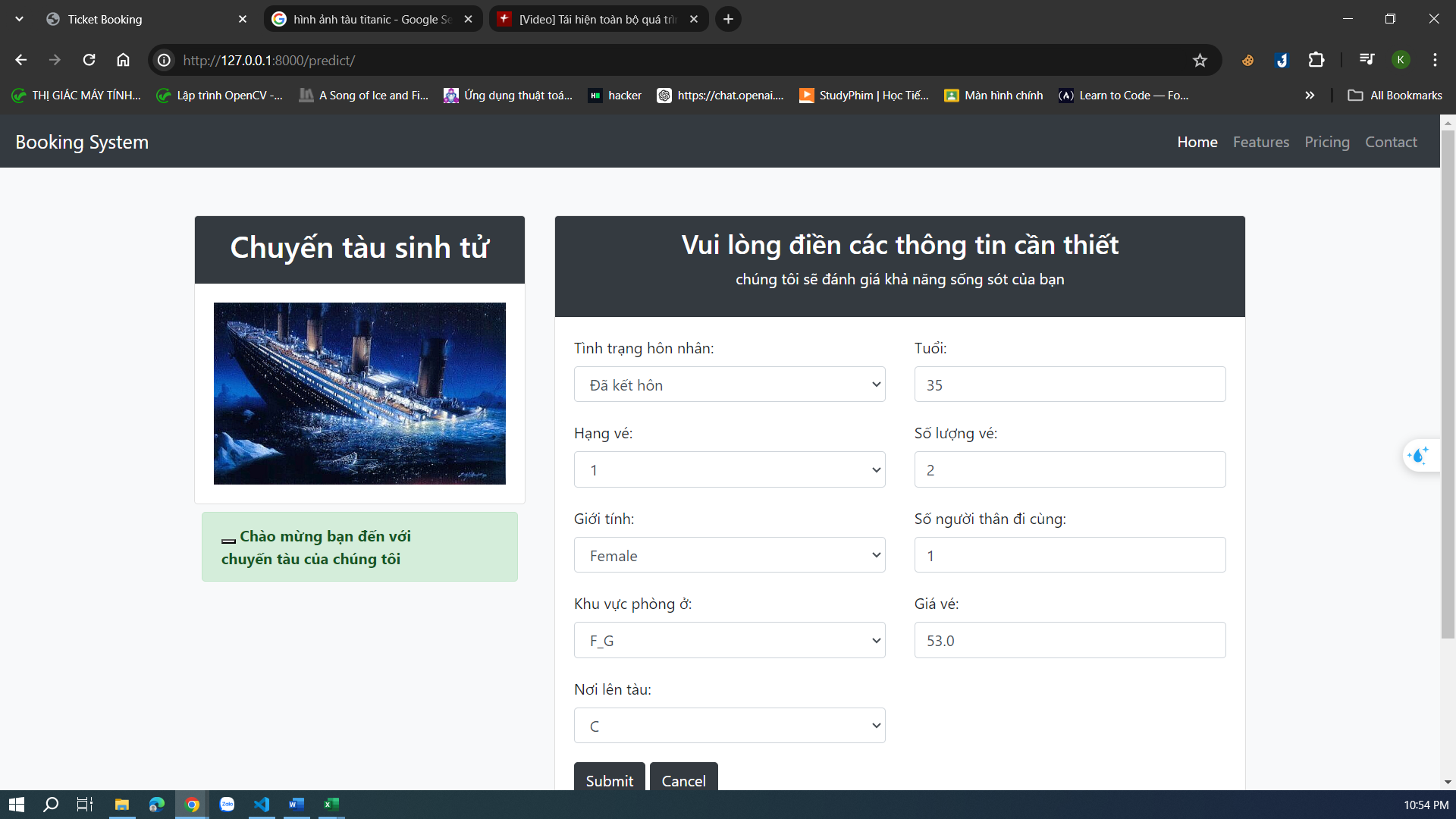


Figure 19 Kết quả dự đoán đối với hành khách thứ 4

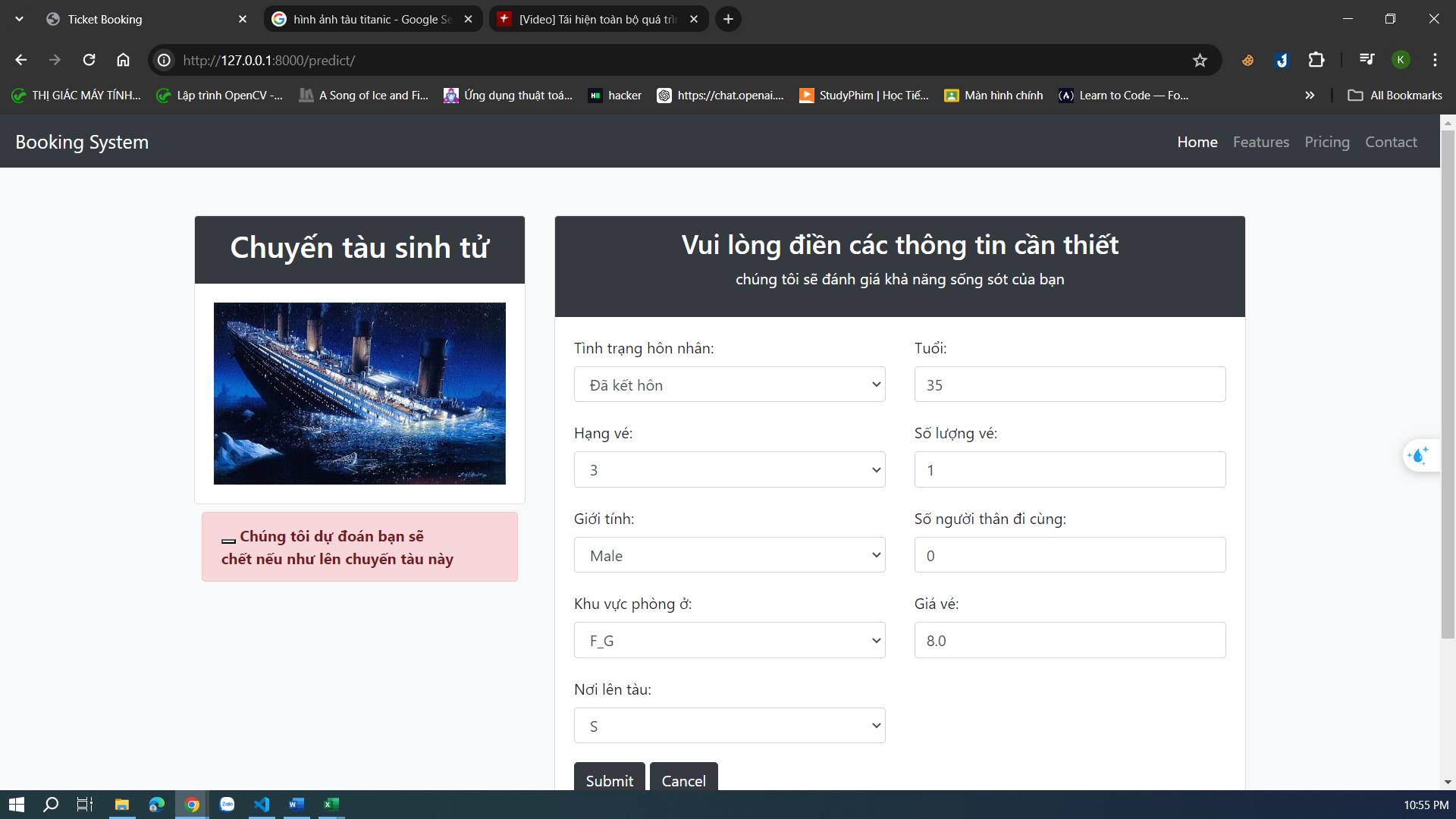


Figure 20: Kết quả dự đoán đối với hành khách thứ 5

## Kết luận

Như vậy với 5 mẫu đầu tiên, ta có thể thấy rằng mô hình đã làm việc khá tốt khi đưa ra dự đoán đúng với tỉ lệ chính xác 100%. Điều này chứng tỏ khả năng dự đoán chính xác của mô hình Random Forest.

# KẾT LUẬN

# Tài liệu tham khảo

Phụ lục