Tối ưu hóa mô hình dự đoán tỉ lệ sống sót Titanic thông qua kĩ thuật Feature engineering lặp lại

Trần Hồ Minh Hải, Trương Văn Thiện, Phan Đức Nhân, Võ Gia Kiệt

1.Giới thiệu

Xác định vấn đề:

- **input:** tập dữ liệu Titanic từ kaggle
- **output :**dự đoán khả năng sống sót của các hành khách trên chuyến tàu Titanic gặp tai nạn

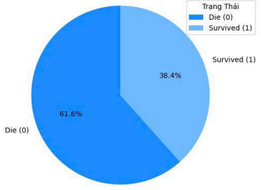
Thách thức:

- Dữ liệu thô chứa nhiều khuyết điểm
- Gồm nhiều cột ít tác động đến taget

Dataset

- 891 dữ liệu train, 418 dữ liệu test
- Nguồn dữ liệu: https://www.kaggle.com/competitions/titanic

Tỷ lệ Sống Sót và Tử Vong trên Tàu Titanic




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
```

2. Phân tích khám phá dữ liệu

Số lượng giá trị bị thiếu

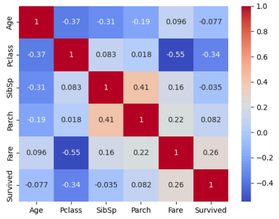
Train

Thuộc Tính	Số lượng Missing	Tỉ lệ
Age	177	19.8%
Cabin	687	77.1%
Embarked	2	nhỏ không đáng kể

Test

Thuộc Tính	Số lượng Missing	Tỉ lệ
Age	86	20.6%
Cabin	327	78.2%
Fare	1	nhỏ không đáng kể

Ma trận tương quan



4 thuộc tính tương quan cao với target

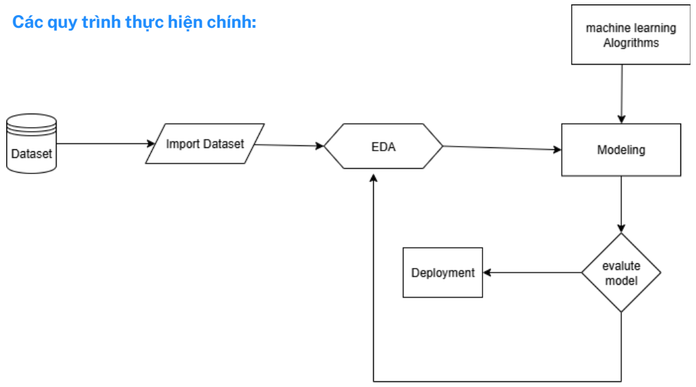
Pclass	-0.34
Fare	0.26
Age	-0.07
Parch	0.08

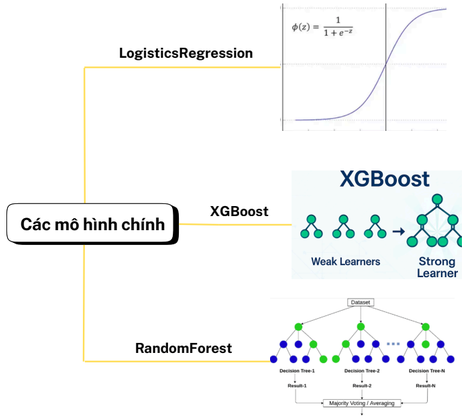
Kết Luận

- Parch tương quan dương yếu không nói lên được gì
- Age tương quan âm yếu không nói lên được gì
- Pclass tương quan âm mạnh những người có Pclass 1 thường có tỉ lệ sống cao hơn
- Fare tương quan dương mạnh: những người có giá vé cao thì thường có khả năng sống sót cao hơn

3.Tổng quan quy trình

Các quy trình thực hiện chính:





4.Phương pháp đề xuất

Đối với các giá trị bị thiếu(missing value):

Age	Fare	Embarked	Cabin
Sử dụng giá trị Median để điền khuyết	Sử dụng giá trị Median để điền khuyết	Sử dụng giá trị Mode (giá trị xuất hiện nhiều nhất)	Drop cột này do quá nhiều dữ liệu bị thiếu

Kỹ thuật Khai thác Đặc trưng (Feature Engineering - FE):

- **Xử lí và mã hóa (Encoding):**

Sex	Pclass và Embarked	Name
male thành 0 và female thành 1	Áp dụng One-Hot Encoding bằng get_dummies	Trích xuất danh xưng (Title) từ cột Name (ví dụ: Mr, Mrs, Master)

- **Rời rạc hóa (Categorical Binning)**

Fare và Age: Chuyển Fare và Age thành dạng Category (Fare_cat và Age_cat) sử dụng các bins

- **Tạo Feature Tổ hợp (Derived Features):**

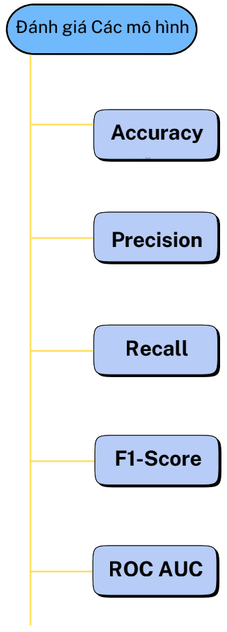
FamilySize = SibSp + Parch + 1

IsAlone: Đánh dấu 1 nếu FamilySize = 1

Family_Survival: Tạo quy tắc dựa trên việc các thành viên trong gia đình (xác định qua Last_Name và/hoặc Ticket) thường có chung số phận sống sót hoặc tử nạn

→ Tổng cộng có 20 features để huấn luyện mô hình

5.Đánh giá mô hình



Accurary biểu thị tỷ lệ phần trăm các dự đoán đúng so với tổng số mẫu dữ liệu được dự đoán. Công thức tính như sau:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision cho biết tỷ lệ các mẫu được dự đoán là dương tính mà thực tế cũng đúng là dương tính, trên tổng số mẫu mà mô hình dự đoán là dương tính. Công thức tính như sau:

$$Precision = \frac{TP}{TP+FP}$$

Recall thể hiện tỷ lệ các mẫu dương tính thực sự được mô hình dự đoán đúng trên tổng số mẫu dương tính trong tập dữ liệu. Công thức tính như sau:

$$Recall = \frac{TP}{TP+FN}$$

F1_Score: mang lại cái nhìn cân bằng hơn về hiệu suất của mô hình khi cân đồng thời tối ưu cả Precision và Recall.

$$F1 = 2 * \frac{recall * precision}{recall + precision}$$

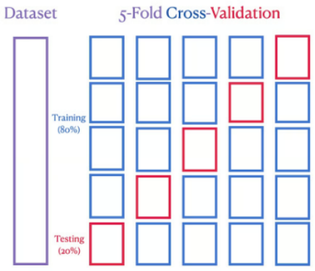
ROC: là công cụ biểu diễn mối quan hệ đánh đổi giữa hai chỉ số: Tỷ lệ Dương tính Đúng (True Positive Rate - TPR) và Tỷ lệ Dương tính Sai (False Positive Rate - FPR) tại tất cả các ngưỡng phân loại khác nhau.

$$TPR = \frac{TP}{TP+FN};$$

$$FPR = \frac{FP}{FP+TN}$$

6.Thử nghiệm mô hình

Chia K-Fold để huấn luyện



Chia tập dữ liệu thành 5 phần bằng nhau mỗi 1 lần huấn luyện sẽ học trên 4 phần 1 phần còn lại đánh giá học cho tới khi đủ 5 lần tiến hành lấy trung bình các độ đo để đánh giá các mô hình

Thí nghiệm 1 - Trên 5 K-Fold

Model	Accurary	F1 Score	ROC_AUC	Public-Score
LogisticRegression	0.8698	0.8226	0.9208	0.79904
Random Forest	0.8687	0.8157	0.9335	0.79425
XGBoost	0.8721	0.8202	0.9351	0.80143
Voting	0.8721	0.8224	0.9329	0.79425
Stacking	0.8732	0.8243	0.9327	0.79425
Blending	0.8743	0.8245	0.9318	0.80143
AvageWeight	0.8721	0.8224	0.9327	0.79425

→ Các tham số tối ưu cho mô hình: 'xgb__colsample_bytree': 0.8, 'xgb__learning_rate': 0.01, 'xgb__max_depth': 5, 'xgb__n_estimators': 200, 'xgb__reg_alpha': 0, 'xgb__reg_lambda': 1, 'xgb__subsample': 1.0

7.Kết luận

Model	Điểm số Tối ưu khi Submit (Accuracy)	Động lực Chính
XGBoost	0.80143	Feature Selection chiến lược
Random Forest	0.79425	Random Forest / Xử lý Missing Value cơ bản
Logistic Regression	0.79904	Feature Family_Survival + Scaling

Mô hình tốt nhất: XGBoost kết quả khi submit lên kaggle