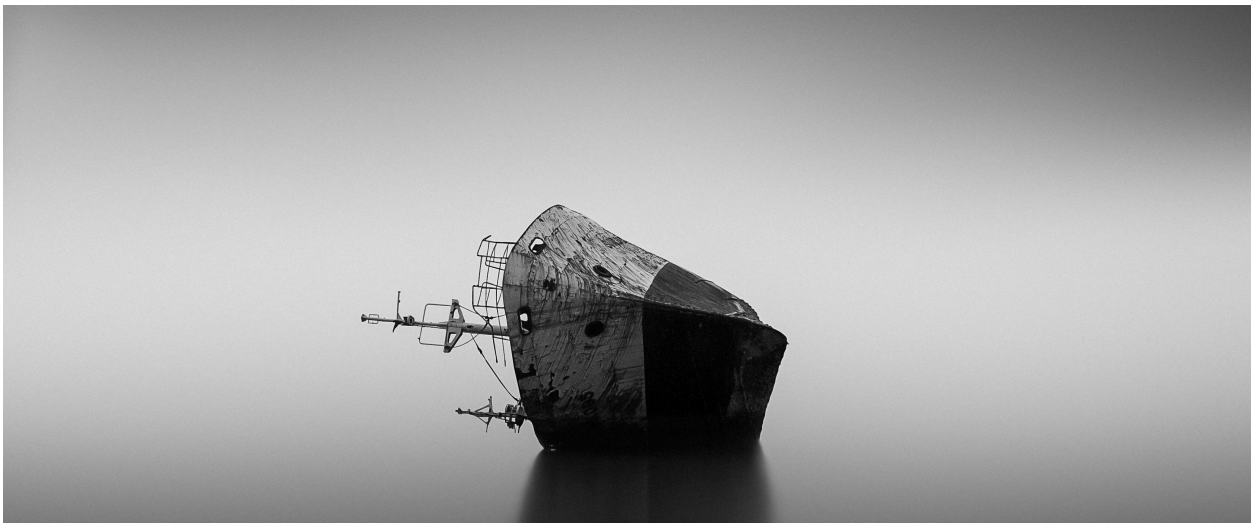


ML Pipeline Project

🚢 Titanic Survival Prediction — Machine Learning Project

1. Project Overview

This project is based on the **Kaggle Titanic Competition: Machine Learning from Disaster**.

The objective is to build machine learning models that predict whether a passenger survived the Titanic shipwreck.

- **Dataset**: Titanic training (`train.csv`) and testing (`test.csv`) sets provided by Kaggle.
- **Goal**: Predict the survival (`0 = did not survive, 1 = survived`).

2. Exploratory Data Analysis

We first explored the dataset to understand variable distributions, missing values, and the relationship between features and survival outcome.

Data Overview

- **Missing values** were identified in `Age`, `Cabin (Deck)`, `Embarked`, and `Fare`.

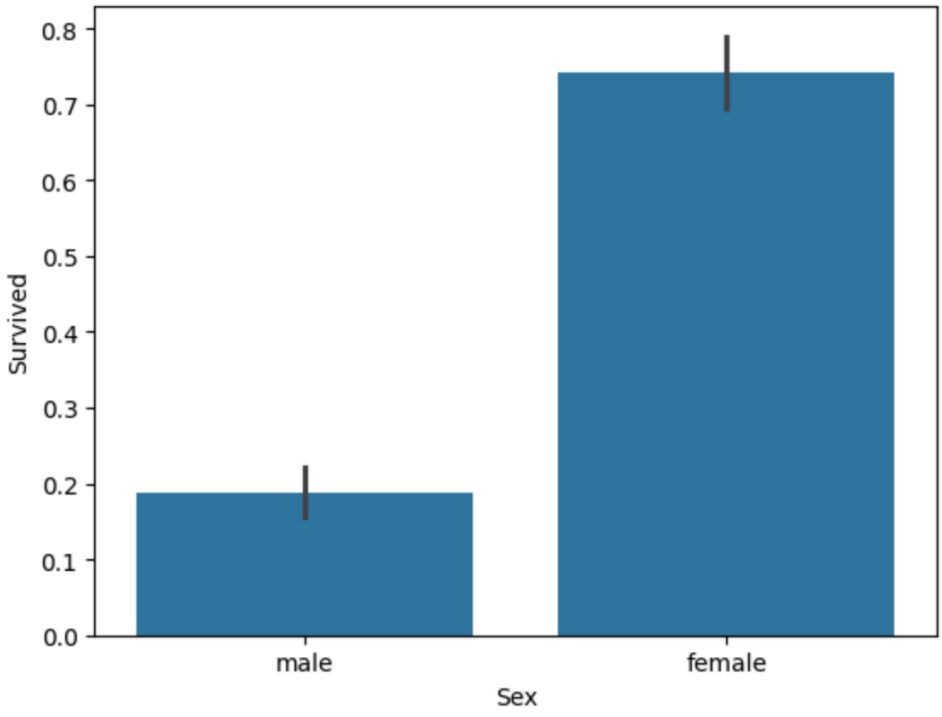
- **Target distribution** (`Survived`):
  - ~38% survived
  - ~62% did not survive

    → indicating an **imbalanced dataset**.

Key Findings with Visualizations

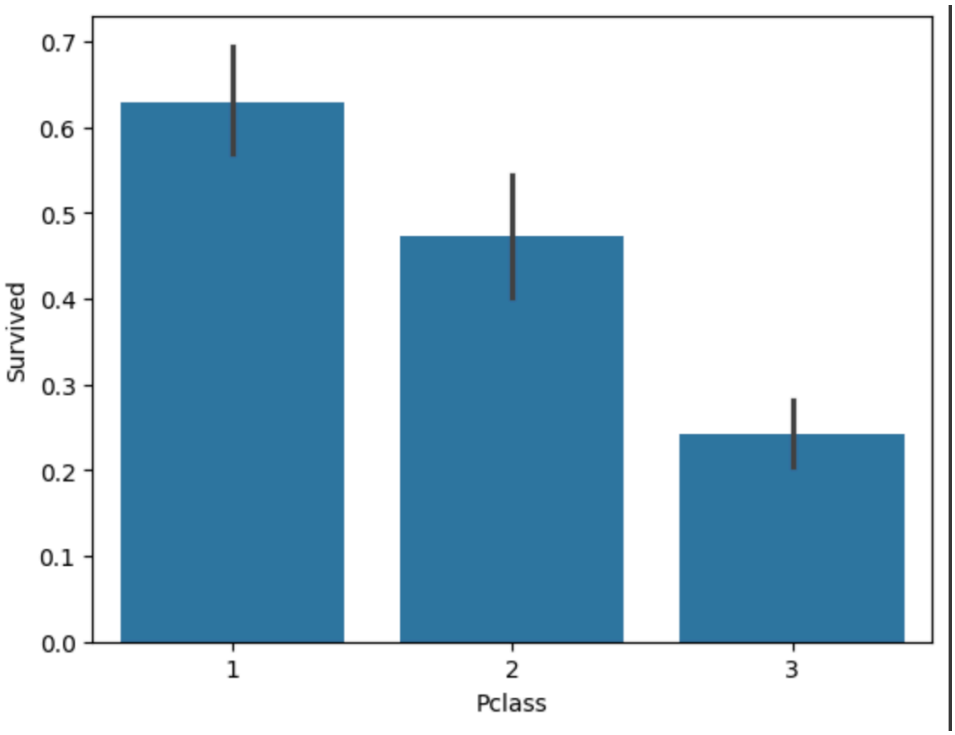
1. Sex vs Survival

- Female passengers had a much higher survival rate (~74%) compared to males (~19%).
- Confirms the strong influence of gender on survival.



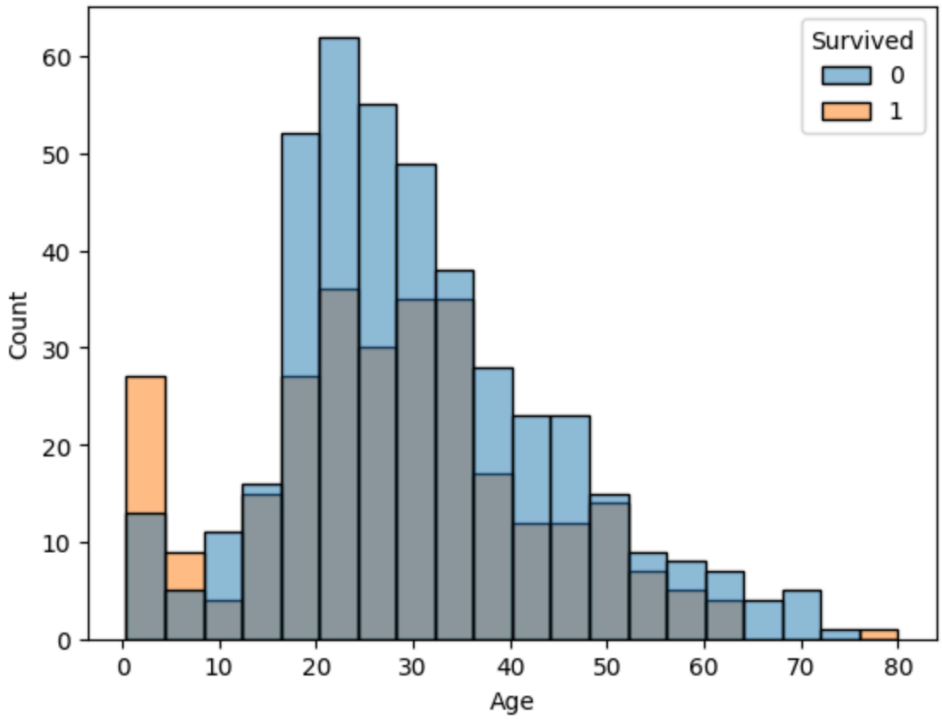
2. Pclass vs Survival

- Survival probability decreased with lower ticket class.
- 1st class passengers had the highest survival rate (~63%), while 3rd class had the lowest (~24%).



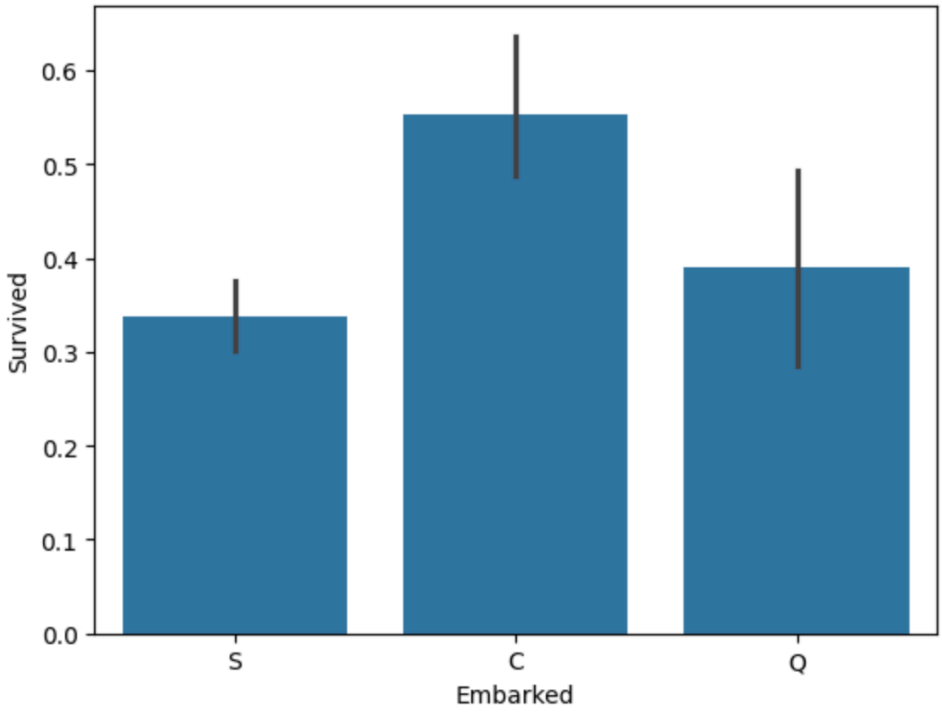
3. Age Distribution

- Survivors were more concentrated among children and young adults.
- Very young passengers (0–10 years) had relatively higher survival rates.



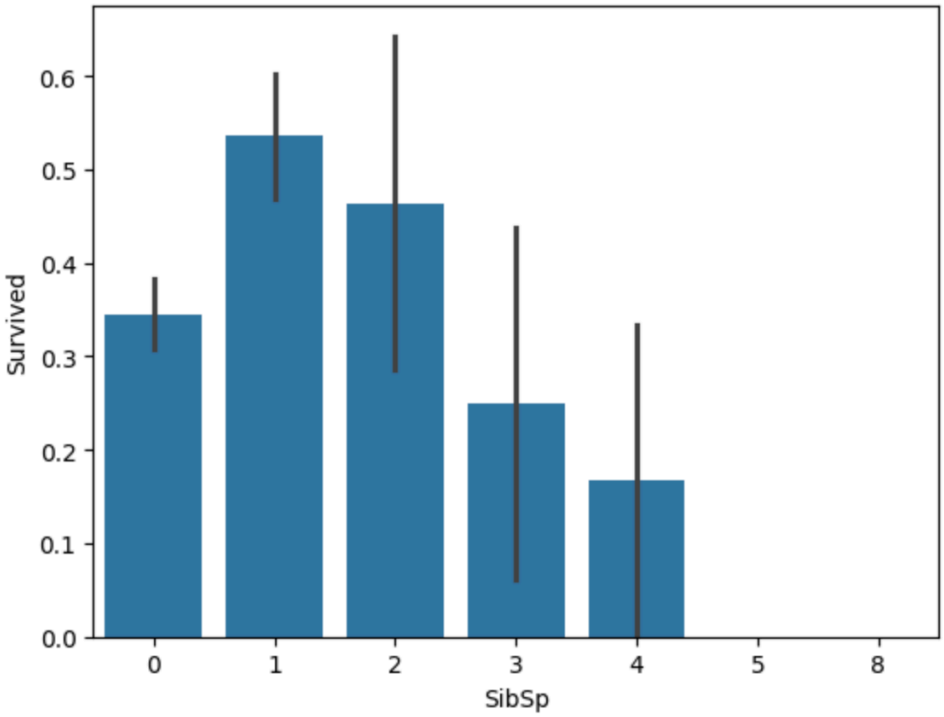
4. Embarked Port vs Survival

- Passengers embarking from **Cherbourg (C)** had the highest survival rate (~55%).
- Southampton (S) had the lowest (~34%).



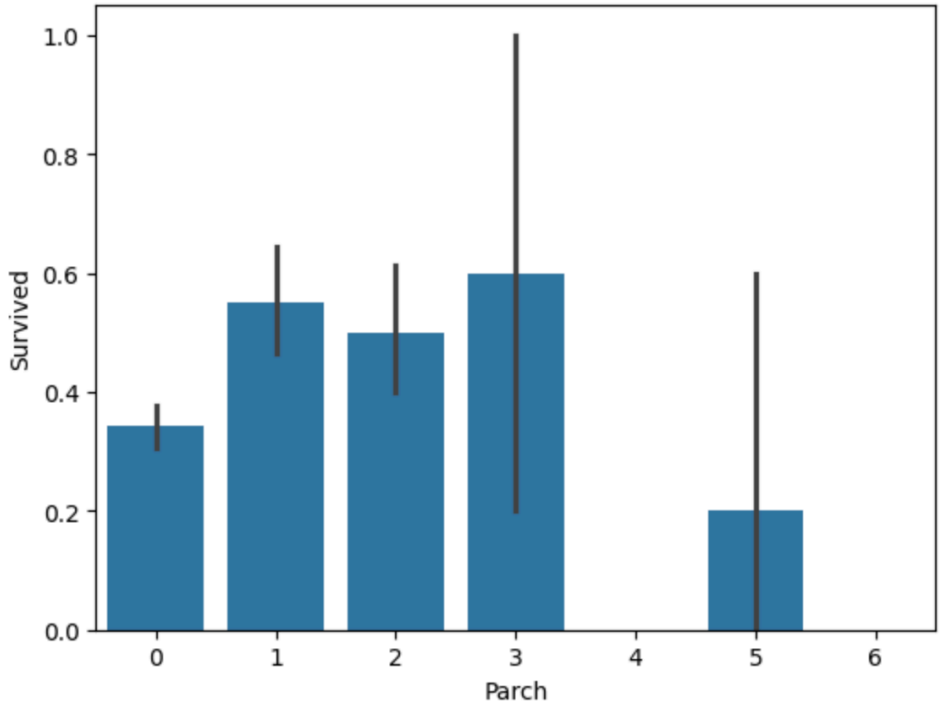
5. SibSp (Siblings/Spouses) vs Survival

- Passengers with **1–2 siblings/spouses** had better survival rates.
- Traveling alone (0) or with large families (≥3) reduced survival probability.



6. Parch (Parents/Children) vs Survival

- Similar to SibSp, survival was higher when traveling with **1–3 parents/children**.
- Huge families had much lower survival rates.



🔑 Insights

- **Sex** and **Pclass** were the most important factors influencing survival.
- **Age** showed that children had a higher chance of survival.
- **Family-related features (SibSp, Parch)** suggest that moderate family size improved survival, while being alone or in large groups reduced it.
- **Embarkation port** also played a role, with Cherbourg passengers surviving at higher rates.

3. Data Preprocessing

📌 Handling Missing Values

- **Embarked**
  - Filled missing values in the training set with the most frequent value (`S` = Southampton).
  - The test set contained almost no missing values for this feature.
- **Fare**
  - Filled missing values with the median fare separately for training and test sets.
  - Using the median reduces the impact of outliers compared to the mean.
- **Age**
  - To provide a more accurate imputation, we used a **grouped median strategy**:
    - Combined training and test sets to calculate median age by `(Pclass, Sex, Title)`.
    - Filled missing ages based on these group medians.
    - Any remaining missing values were filled with the overall median age as a fallback.

📌 Feature Cleanup

- **Cabin → Deck**
  - Extracted the first letter of the `Cabin` string as a new feature `Deck` (values A–G, or NaN).
  - Missing values were filled with `"U"` (Unknown).
  - Dropped the original `Cabin` column due to its high proportion of missing entries (>75%).
- **PassengerId**
  - Preserved from the test set in order to generate the final Kaggle `submission.csv` file.

📌 Encoding Categorical Features

- Converted categorical variables into numeric form using **One-Hot Encoding**:
  - `Sex`, `Embarked`, `Title`, `Deck`, and `Pclass` were expanded into binary indicator columns.
  - This ensured that the model could process categorical information without assuming ordinal relationships.
- To guarantee **column consistency between training and test sets**:
  - Applied one-hot encoding separately to train and test.
  - Re-aligned the resulting dataframes so that they had the same columns, filling any missing categories with `0`.

4. Feature Engineering

We created additional features to capture family structure and grouped continuous variables into meaningful categories.

- **FamilySize** = `SibSp + Parch + 1`
  - Represents the total number of family members aboard (siblings, spouses, parents, children, plus the passenger).
- **IsAlone**
  - Binary variable indicating whether the passenger traveled alone.

- `IsAlone = 1` if `FamilySize == 1`, else `0`.

- **AgeBin**
  - Grouped passenger ages into 5 categories:
    - 0: Child (0–12)
    - 1: Teenager (13–18)
    - 2: Young Adult (19–35)
    - 3: Adult (36–60)
    - 4: Senior (61–80)
- **FareBin**
  - Divided passengers into 4 quartiles based on fare values.
  - Labels assigned from `0` (lowest quartile) to `3` (highest quartile).
- **One-Hot Encoding**
  - Applied to categorical engineered features (e.g., `AgeBin`, `FareBin`, `Title`, `Embarked`, `Deck`) to convert them into numerical format suitable for machine learning models.

5. Modeling & Evaluation

📌 Data Preparation for Modeling

- Separated features (`X`) and target (`y = Survived`).
- Strict validation checks before modeling:
  - No missing values (`NaN`).
  - No object-type columns (all categorical features were encoded).
- Split dataset into **training set (80%)** and **validation set (20%)** with stratification to preserve target distribution.

📌 Baseline Models

Logistic Regression

- Accuracy: **0.8547**
- ROC-AUC: **0.8413**
- Confusion Matrix:

```
[[99 11]
 [15 54]]
```

Random Forest

- Accuracy: **0.8045**
- ROC-AUC: **0.7896**
- Confusion Matrix:

```
[[94 16]
 [19 50]]
```

XGBoost

- Accuracy: **0.9106**
- ROC-AUC: **0.9057**
- Confusion Matrix:

```
[[102   8]
 [  8  61]]
```

📌 Hyperparameter Tuning

- Applied **GridSearchCV** with 5-fold cross-validation, optimizing for **ROC-AUC**.
- Hyperparameter search space included:
  - `n_estimators`: [200, 400, 600]
  - `max_depth`: [3, 4, 5, 6]
  - `learning_rate`: [0.01, 0.05, 0.1]

- `subsample`: [0.8, 0.9, 1.0]
- `colsample_bytree`: [0.8, 0.9, 1.0]

- **Best Parameters Found**:

```
{
  "colsample_bytree": 0.8,
  "learning_rate": 0.05,
  "max_depth": 3,
  "n_estimators": 200,
  "subsample": 0.8
}
```

- **Best Cross-validated ROC-AUC**: **~0.8784**

📌 Final Model & Submission

- Chose **XGBoost with tuned hyperparameters** as the final model.
- Re-trained on the **entire training dataset** (to maximize learning).
- Generated predictions on the Kaggle test set and created the final submission file:

```
submission = pd.DataFrame({
    "PassengerId": test_passengerId,
    "Survived": test_pred
})
submission.to_csv("submission.csv", index=False)
```

📌 Insights from Modeling

- **XGBoost outperformed Logistic Regression and Random Forest** with both higher accuracy and ROC-AUC.
- Logistic Regression still achieved strong results, showing that even simple linear models can capture key survival patterns.

- Random Forest underperformed relative to XGBoost, likely due to less effective handling of categorical interactions and feature importance weighting.
- Hyperparameter tuning further stabilized XGBoost's performance.

6. Feature Importance Analysis

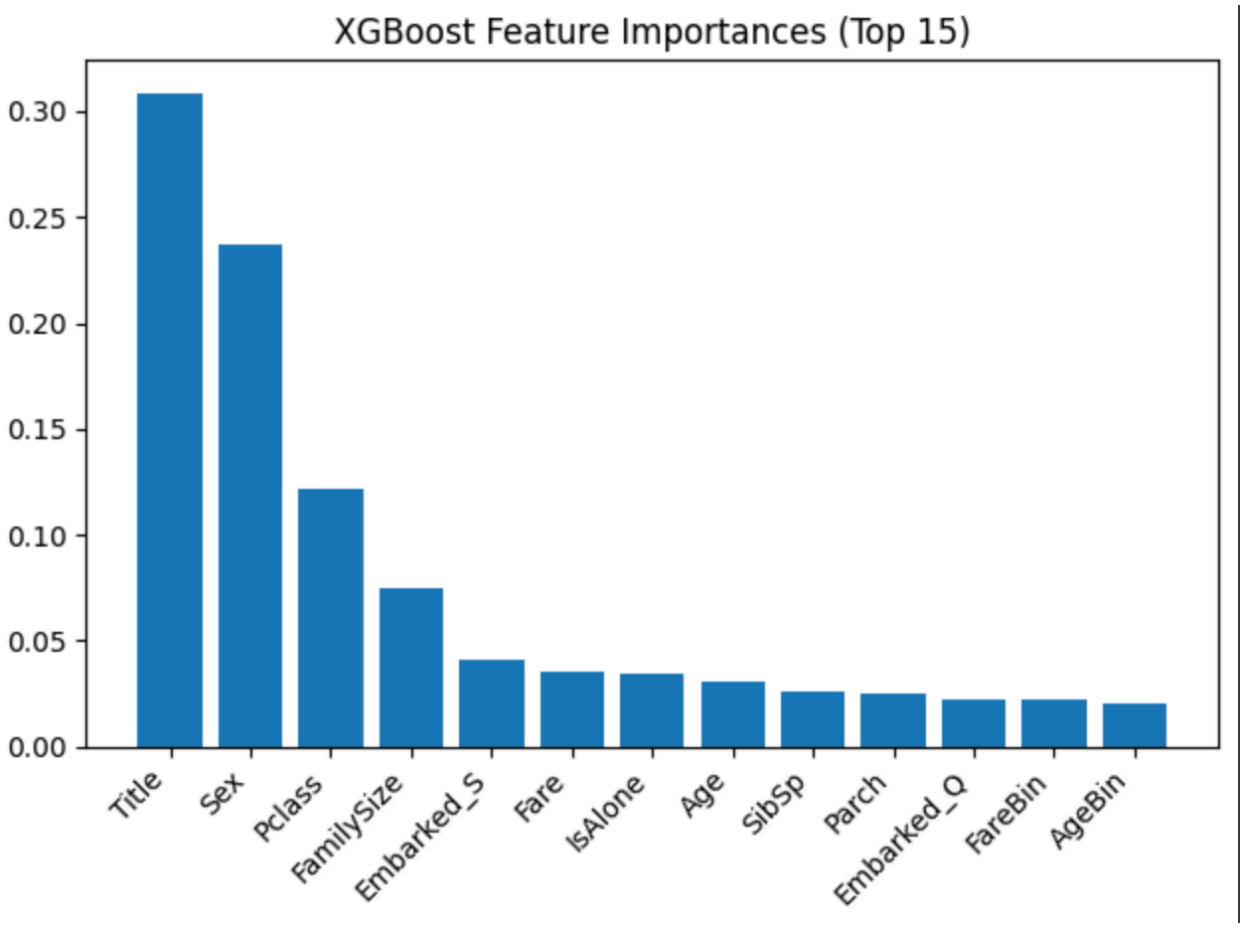
📊 Feature Importance (XGBoost)

To interpret the decision-making process of the final XGBoost model, we extracted the **feature importances** and plotted the top 15 contributors.

Key Findings:

- **Sex** and **Pclass** were the most critical features, confirming earlier EDA results that gender and passenger class strongly influenced survival.
- **Engineered features** such as `IsAlone`, `FamilySize`, `FareBin`, and `AgeBin` ranked among the top contributors, validating the impact of our feature engineering process.
- **Title** (extracted from passenger names) also showed predictive strength, adding social status and cultural context as useful signals.

The feature importance analysis demonstrates that **data preprocessing + feature engineering** significantly improved model performance and interpretability.



7. Overview

1. **Sex** and **Pclass** were the strongest predictors of survival, aligning with historical accounts of the Titanic disaster.
2. **Feature engineering added substantial value**: engineered variables such as `IsAlone`, `FamilySize`, `FareBin`, and `AgeBin` significantly improved model performance.
3. **XGBoost outperformed other models**, achieving the highest accuracy and ROC-AUC compared to Logistic Regression and Random Forest.
4. **Interpretability matters**: feature importance analysis confirmed that both raw and engineered features contributed meaningfully to predictions.