Machine Learning Predictive Models: Titanic Survival Rates

Author: Alysen Casaccio MDS, BHA, RN-BC, CSHIMS

Data Science Portfolio

Problem Statement

The Titanic Survival Predictor is a machine-learning competition hosted by Kaggle.com. The objective is to develop a machine-learning model to predict survival outcomes for passengers aboard the Titanic based on a dataset that includes details such as passenger names, ages, genders, socio-economic class, and other factors. This project explores machine-learning approaches, feature engineering, and evaluation metrics to optimize prediction accuracy while maintaining model interpretability and balance.

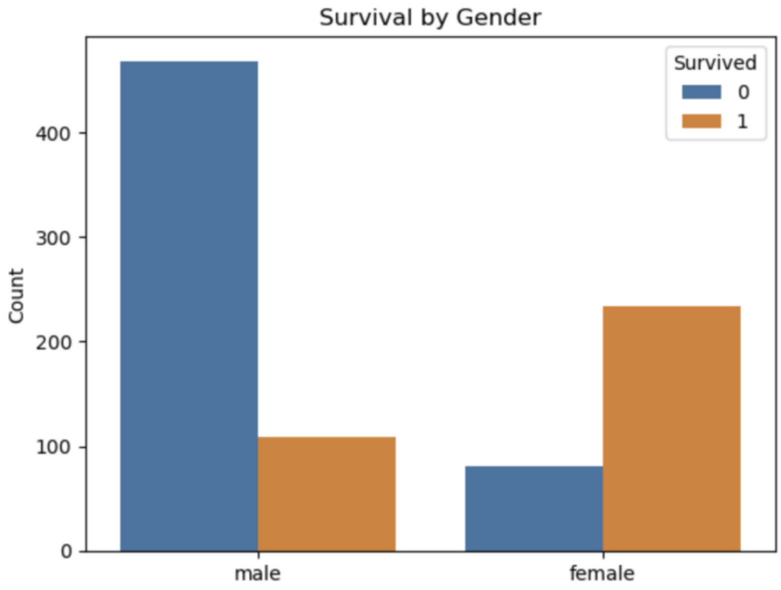
Data Selection

The Titanic dataset consists of two primary files: training and test data. The training data includes 891 passenger records, each labeled with the survival outcome (Survived: 1 = survived, 0 = did not survive). These records provide information about various passenger characteristics, including demographic details such as Age and Sex, socio-economic indicators like Pclass (ticket class) and Fare, and travel-related information, including SibSp (number of siblings or spouses aboard), Parch (number of parents or children aboard), and Embarked (port of embarkation). The target variable, Survived, is the model building and evaluation classification label.

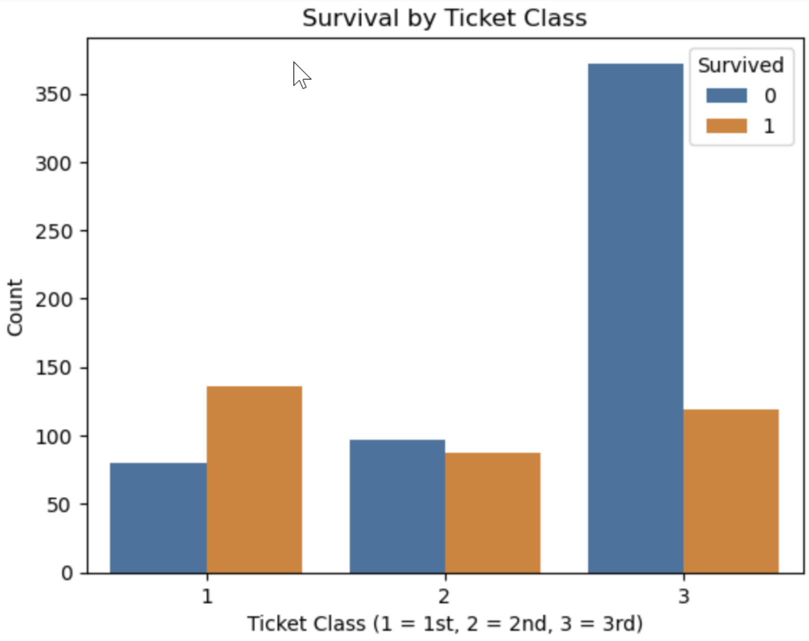
The test data contains 418 passenger records, which lack survival labels and are reserved for final predictions submitted to Kaggle. The dataset's features collectively offer insights into the factors influencing survival, capturing demographic, economic, and travel characteristics. These features were carefully explored and preprocessed to ensure they contributed effectively to the machine-learning models.

Exploratory Data Analysis

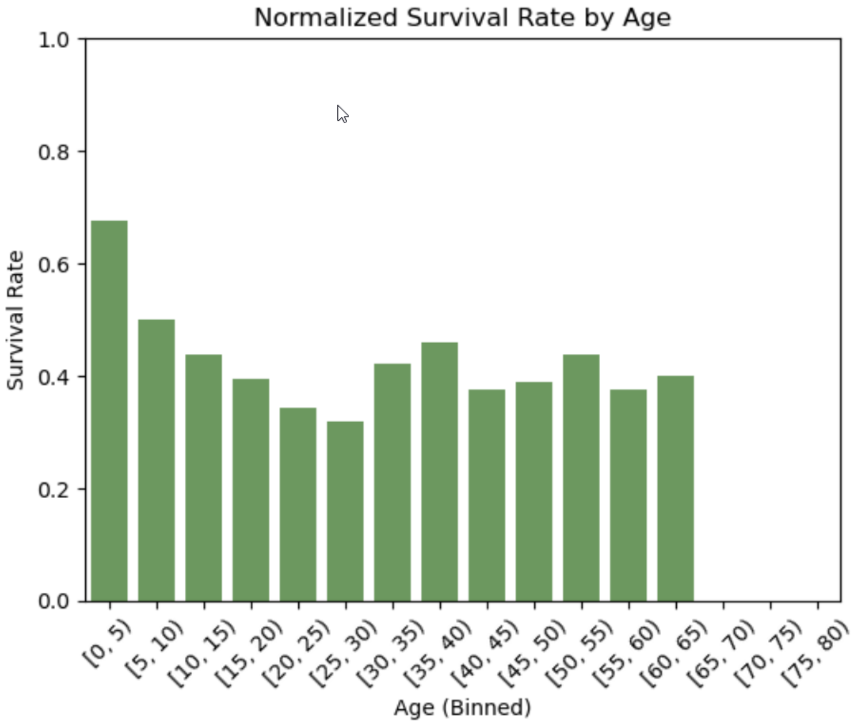
Exploration of the Titanic dataset revealed several critical insights into survival patterns. Gender emerged as a significant factor, with female passengers showing substantially higher survival rates than males, underscoring the importance of the Sex feature.



The socioeconomic class also played a pivotal role, as first-class passengers exhibited the highest survival rates, while third-class passengers faced the most significant mortality risk.



Age was another influential factor, with children and younger passengers demonstrating higher survival probabilities. This likely reflects evacuation prioritization for families and young passengers.



Analysis of cabin assignments highlighted socio-economic disparities. Missing cabin data was predominantly associated with third-class passengers, consistent with the lack of private accommodation in steerage.

Finally, embarkation points provided additional insight, with passengers boarding at Cherbourg having better survival rates than those embarking at Southampton and Queenstown. These findings guided feature selection and engineering efforts to enhance predictive performance.

Data Preparation

Feature engineering was a crucial step in enhancing the predictive power of the Titanic survival models while effectively handling missing data. Several new features were created to capture meaningful patterns in the dataset. For instance, a binary feature called Has_Cabin was introduced to indicate whether a passenger had an assigned cabin. Additionally, the Deck_Label feature was derived by extracting the first letter of cabin values, with missing values filled as "U" to signify "Unassigned."

To ensure numerical features were on comparable scales, Age and Fare were normalized using StandardScaler, which improved the models’ ability to converge and make accurate predictions. Categorical variables, such as Sex, Embarked, and Deck, were encoded using label encoding, converting them into numeric formats while preserving their inherent order or distinctions. This systematic preprocessing provided a strong foundation for building effective machine-learning models.

Multicollinearity Investigation

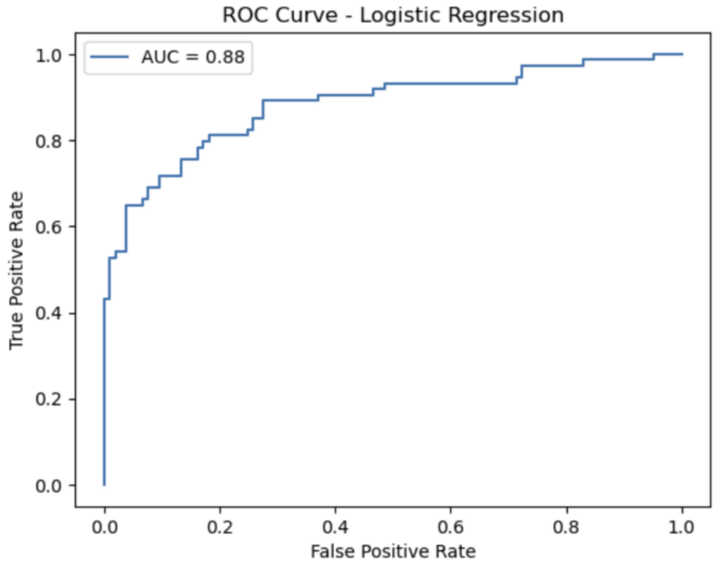
A Variance Inflation Factor (VIF) analysis was conducted on the numeric features to address redundancy in the dataset. This process helped identify multicollinearity, a condition where features are highly correlated, potentially reducing the model's effectiveness. The analysis revealed that the Deck_Label feature exhibited the highest VIF value, indicating strong multicollinearity with Pclass.

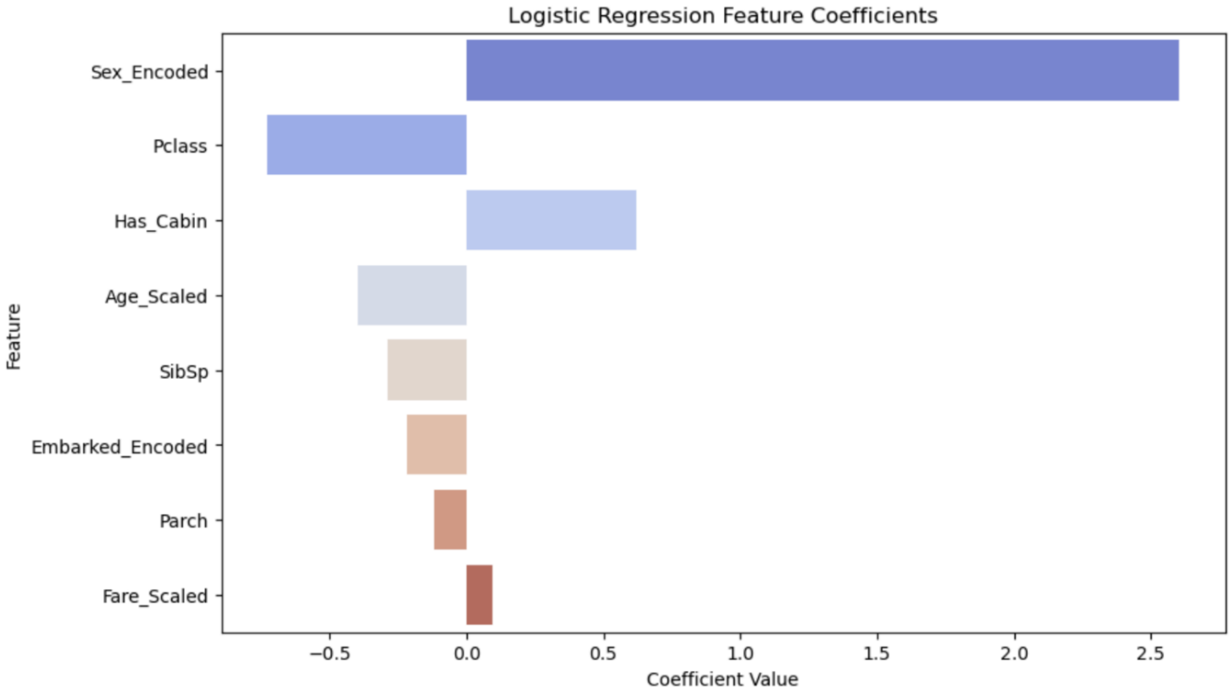
To mitigate this issue, Deck_Label was removed from the dataset. This decision reduced redundancy while preserving the integrity of the remaining features. By removing Deck_Label, the dataset became cleaner and more interpretable, with minimal loss of predictive information. This step ensured that the models could focus on genuinely independent and informative features.

Model Building and Interpretation

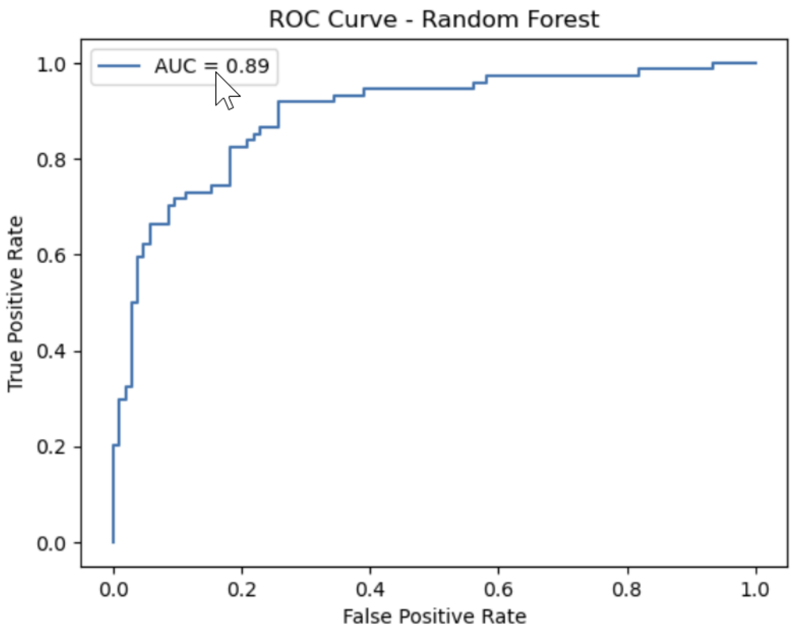
Three distinct models—Logistic Regression, Random Forest, and XGBoost—were selected and evaluated to predict survival outcomes. Each model offered unique strengths, making it well-suited to the Titanic dataset's characteristics. The selection rationale balanced simplicity, interpretability, and the ability to capture complex patterns within the data.

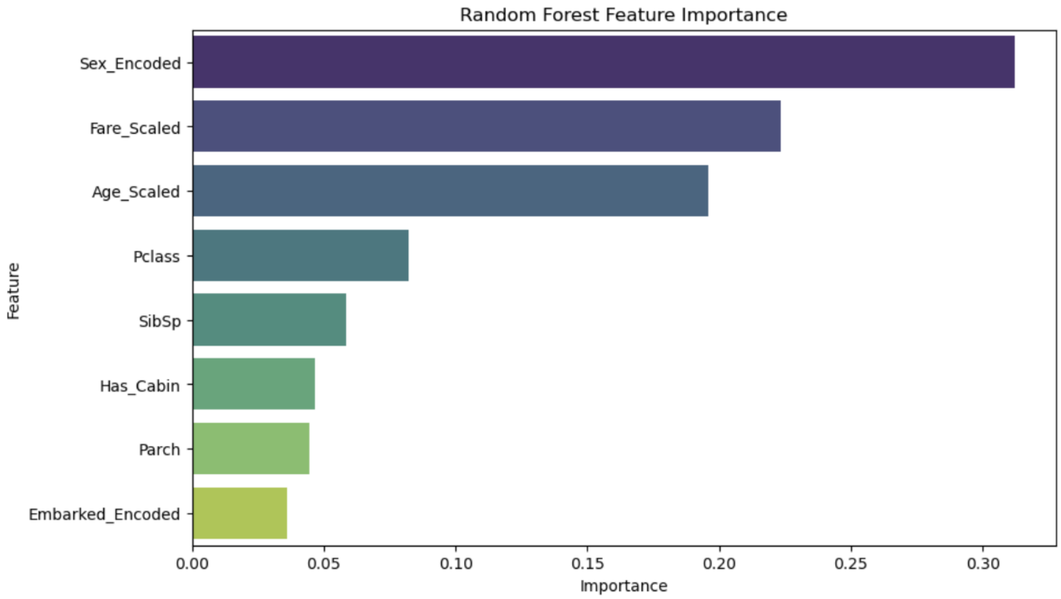
Logistic Regression was chosen as a baseline model because of its simplicity and high interpretability. Logistic regression assumes a linear relationship between features and the target variable, which makes it easy to understand and explain the impact of individual features. The calculated feature coefficients provided direct insights into factors most strongly influenced survival, such as gender (Sex_Encoded) and ticket class (Pclass). This model's performance was a benchmark for evaluating the more advanced algorithms.



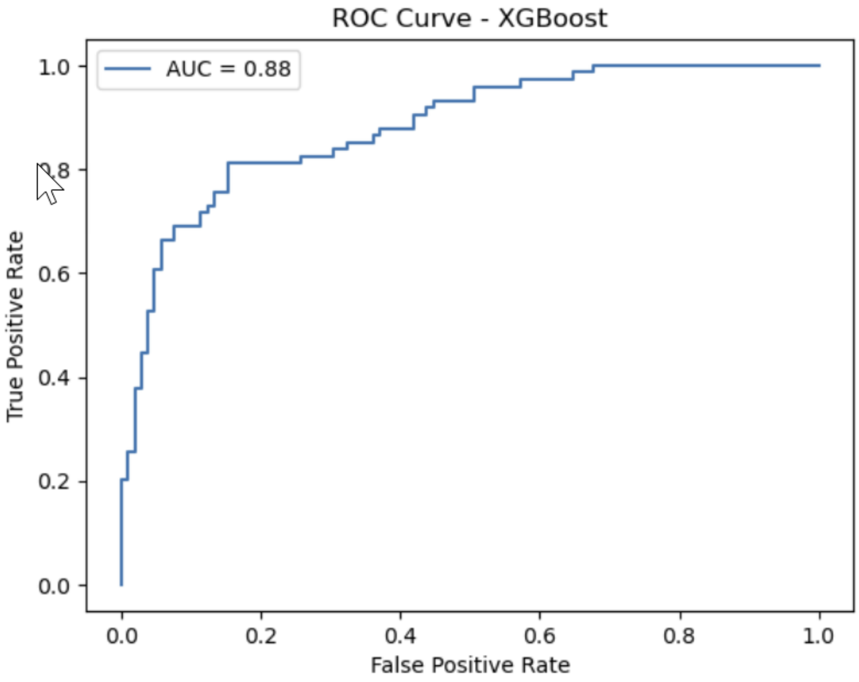


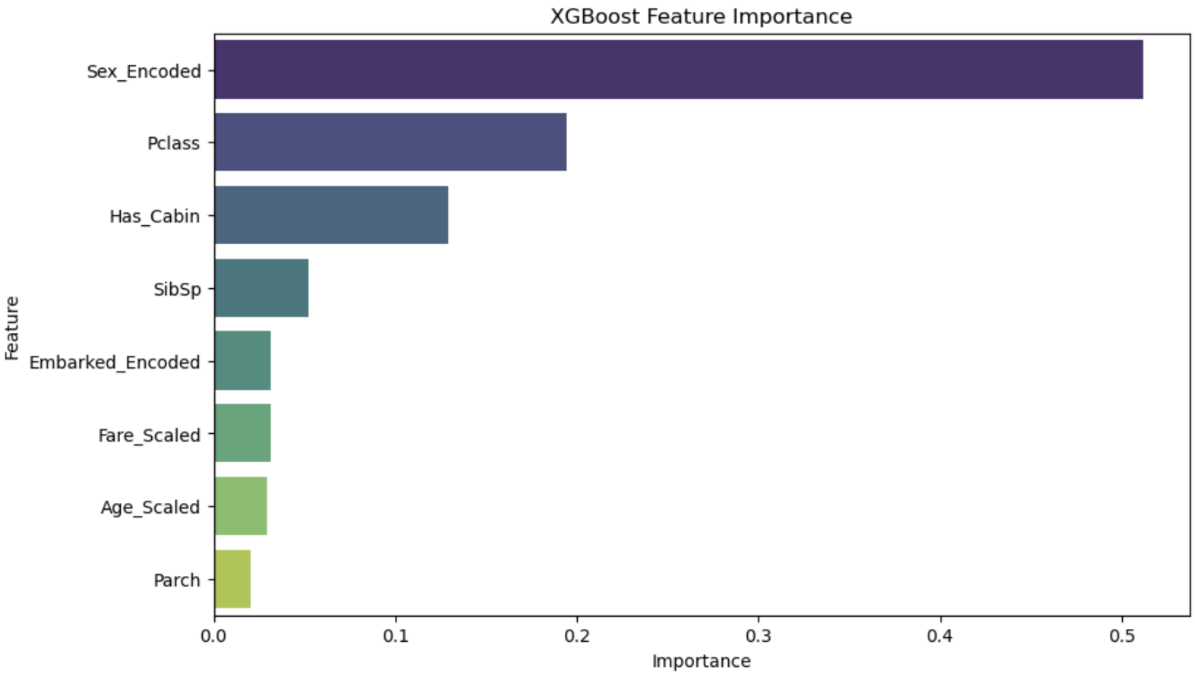
Random Forest was selected for its ability to capture non-linear relationships and its resilience to multicollinearity, a potential issue given the interplay between socio-economic and demographic features in the dataset. This ensemble-based model aggregated predictions from multiple decision trees, reducing overfitting and improving generalization. The Random Forest model also offered a strong mechanism for ranking feature importance, consistently highlighting Sex_Encoded, Fare_Scaled, and Age_Scaled as the most influential predictors of survival.





XGBoost was included as a gradient-boosting algorithm, capable of refining predictions iteratively by focusing on errors made by prior models. This model's strength lies in optimizing performance while managing complex feature interactions and noisy data. Among the three models, XGBoost demonstrated the highest AUC, reflecting its superior discriminatory power in distinguishing survivors from non-survivors.





By combining these three approaches, the project leveraged both interpretable baselines and advanced methods capable of capturing nuanced patterns. This strategic selection ensured a comprehensive dataset evaluation while balancing complexity and predictive power.

Evaluation

The models were evaluated using accuracy, precision, recall, F1-score, and AUC. The results are summarized below:

Metric	Logistic Regression	Random Forest	XGBoost (Default)
Validation Accuracy	82%	82%	82%
AUC	0.88	0.88	0.89
Precision (Survived)	80%	83%	82%
Recall (Survived)	76%	74%	74%
F1-Score (Survived)	78%	78%	77%

Conclusion

The project successfully applied machine-learning techniques to predict Titanic's survival with accuracy and balance. The consistent performance across models (82% validation accuracy) suggests that the dataset's predictive limit has been reached. While XGBoost demonstrated slight improvements in AUC, logistic regression remains a compelling choice for simplicity and interpretability. Future enhancements could explore ensemble methods or advanced feature engineering, though gains may be marginal without additional data.

References

- Kaggle Titanic Competition: *Kaggle.com*. Retrieved from: https://www.kaggle.com/competitions/titanic

- Titanic Historical Details: *Encyclopedia-Titanica.com*. Retrieved from: https://www.encyclopedia-titanica.org
- Python Documentation: *Python.org*. Retrieved from https://www.python.org/doc/
- Scikit-learn Documentation: *Scikit-learn.org*. Retrieved from: https://scikit-learn.org/stable/index.html