Machine Learning Model Comparison Report: Titanic Survival Prediction

By Arya Kumari
Date- 25/01/26
AI Intern at OWL-Ai

1. Executive Summary

The objective of this project was to perform an exploratory data analysis and build a predictive system to determine passenger survival on the Titanic. We implemented two distinct machine learning algorithms: **Logistic Regression** (a linear statistical model) and **Random Forest** (an ensemble of decision trees). While both models yielded an identical initial accuracy of **80.45%**, deeper evaluation through cross-validation and error analysis reveals that the **Random Forest** model is the more robust solution for this specific classification problem.

2. Data Preprocessing & Methodology

High-quality model performance relies on structured data. The following steps were executed within the Google Colab environment:

- **Feature Selection:** We focused on four key variables: `pclass` (socio-economic status), `sex` (gender), `age`, and `fare` (ticket cost).

- **Imputation:** Missing data in the `age` column was addressed by applying the **median** value to prevent bias from outliers.
- **Encoding:** Categorical text data (`sex`) was transformed into numerical format (0 and 1) to allow for mathematical processing by the algorithms.
- **Validation Strategy:** The data was partitioned using an 80/20 split. To ensure results weren't a result of "luck" in the split, a **5-fold Cross-Validation** was performed, testing each model on five different subsets of the data.

3. Comparative Analysis of Results

A. Global Accuracy and F1-Score

Both models reached a baseline accuracy of **80.45%**. However, accuracy can be misleading in datasets where classes are slightly imbalanced. We utilized the **F1-Score** to measure the balance between Precision and Recall.

- **Logistic Regression F1-Score:** 0.7632
- **Random Forest F1-Score:** 0.7534

B. Confusion Matrix Breakdown

Analyzing the "Errors" (False Positives vs. False Negatives):

- **Logistic Regression:** Demonstrated a slightly higher **Recall**. It was more "sensitive" to finding survivors, meaning it had fewer False Negatives (actual survivors who were predicted to die).
- **Random Forest:** Demonstrated higher **Precision**. When this model predicted someone survived, it was correct more often, though it was slightly more "conservative" in its predictions.

C. Model Stability (Cross-Validation)

The ultimate tie-breaker was the Cross-Validation (CV) score. This metric averages performance over multiple trials:

- **Logistic Regression CV:** 78.57%
- **Random Forest CV:** 79.69%

The higher CV score for Random Forest proves that it generalizes better to new data and is less likely to "overfit" to the specific training set.

4. Feature Importance

Using the Random Forest's internal ranking system, we identified the drivers of survival. The model determined that **Sex** and **Fare** were the most influential predictors, followed by **Age** and **Class**. This aligns with historical accounts of the "women and children first" protocol and the prioritization of higher-class cabins.

5. Final Verdict and Recommendations

Winning Model: Random Forest Classifier

Justification:

1. **Superior Generalization:** The higher Cross-Validation score confirms that the Random Forest is more dependable for real-world application.
2. **Non-Linearity:** The Titanic dataset contains complex interactions (e.g., young children in 3rd class having different survival rates than adults in 3rd class). Random Forest captures

these "if-then" logic branches better than the linear nature of Logistic Regression.

3. **Scalability:** As more features are added to the dataset (such as 'Cabin' or 'Family Size'), the Random Forest will likely widen its lead over Logistic Regression due to its ability to handle high-dimensional data without requiring strict statistical assumptions.

Conclusion:

For a production environment or a research paper, the **Random Forest** model should be the selected deployment model. Future work should involve **Hyperparameter Tuning** to optimize the `max_depth` and `n_estimators` to potentially push accuracy beyond the current 80.45% threshold.