Titanic Survival Prediction: Model Comparison and Evaluation

Kerem Adalı

April 30, 2025

Abstract

This report presents an analysis of survival prediction on the Titanic dataset using two machine learning models: Logistic Regression and Decision Tree Classifier. The report documents the data preprocessing approach, hyperparameter optimization, and a comprehensive evaluation of both models across multiple performance metrics. We analyze model stability across different random seeds and identify the best-performing model configuration for this classification task.

1 Introduction

The Titanic dataset is a classic machine learning problem where the goal is to predict which passengers survived the Titanic shipwreck based on features such as age, gender, passenger class, and family size. This binary classification task serves as an excellent test case for comparing different machine learning algorithms and understanding their strengths and weaknesses.

2 Dataset Description

The Titanic dataset contains passenger information including demographic data (age, sex, class) and travel information (cabin, ticket, fare). The target variable is "Survived" (0 = No, 1 = Yes). The training set contains 891 observations, and the test set contains 418 observations. The dataset includes both categorical and numerical features, with multiple missing values, particularly in the age and cabin fields.

3 Data Preprocessing

Our data preprocessing strategy focused on addressing missing values, feature engineering, and proper encoding for machine learning models. The key preprocessing steps included:

- **Feature Engineering:**
  - Extracted passenger titles from name fields and consolidated them into common categories
  - Created a binary feature for married women (identified by "Mrs" title)
  - Extracted deck information from cabin data
  - Created a family size feature by combining siblings/spouses (SibSp) and parents/children (Parch)
- **Missing Values:**
  - Filled missing age values with median age
  - Categorized missing cabin information as "Unknown"

- **Encoding:**
  - Used one-hot encoding for categorical variables (Sex, Deck, Title, etc.)
- **Feature Scaling:**
  - Normalized numerical features to have zero mean and unit variance

The rationale behind these preprocessing steps was to:

- Extract maximum information from existing features
- Handle missing data without removing observations
- Create features that might better capture passenger survival patterns
- Prepare data in a format suitable for both Logistic Regression and Decision Tree algorithms

4 Model Selection and Hyperparameter Tuning

We selected two models for comparison: Logistic Regression and Decision Tree Classifier. These models were chosen for their interpretability, different underlying approaches to classification, and their established performance on similar tasks.

4.1 Logistic Regression Hyperparameters

Table 1 shows the hyperparameters tested during the grid search for the Logistic Regression model.

Table 1: Logistic Regression Hyperparameters Explored

Parameter	**Values Tested**
C	0.001, 0.01, 0.1, 0.25, 0.5, 1.0, 10.0
penalty	l1, l2
solver	liblinear, saga
fit_intercept	True, False
class_weight	None, 'balanced'
warm_start	True, False
tol	0.0001, 0.001, 0.01
intercept_scaling	1, 2, 5

4.2 Decision Tree Hyperparameters

Table 2 shows the hyperparameters tested during the grid search for the Decision Tree Classifier.

Table 2: Decision Tree Hyperparameters Explored

Parameter	Values Tested
max_depth	5, 10, 15, 20, None
min_samples_split	2, 5, 10, 20
min_samples_leaf	1, 2, 4
criterion	gini, entropy
max_features	None, sqrt, log2
splitter	best, random
class_weight	None, 'balanced'
ccp_alpha	0.0, 0.01, 0.05
min_impurity_decrease	0.0, 0.01

4.3 Best Model Parameters

After extensive grid search, the following optimal parameters were identified:

4.3.1 Logistic Regression Best Parameters

Table 3: Best Parameters for Logistic Regression

Parameter	Value
C	0.25
penalty	l1
solver	liblinear
fit_intercept	True
class_weight	None
warm_start	True
tol	0.0001
intercept_scaling	1

4.3.2 Decision Tree Best Parameters

Table 4: Best Parameters for Decision Tree

Parameter	Value
max_depth	20
min_samples_split	20
min_samples_leaf	2
criterion	gini
max_features	None
splitter	random
class_weight	None
ccp_alpha	0.0
min_impurity_decrease	0.0

5 Model Evaluation

To ensure reliable results, we evaluated our models across 100 different random seeds. This approach allows us to assess stability and robustness of the models rather than relying on a single train/test split that might yield misleading results due to data particularities.

5.1 Evaluation Metrics

We used multiple evaluation metrics to compare model performance:

- **F1 Score:** Harmonic mean of precision and recall
- **Accuracy:** Proportion of correctly classified instances
- **Precision:** Proportion of positive identifications that were actually correct
- **Recall:** Proportion of actual positives that were correctly identified

5.2 Overfitting Analysis

We monitored overfitting by comparing cross-validation scores with test set performance. A difference greater than our threshold of 0.05 was considered evidence of overfitting. Across 200 model fits (2 models $\times$ 100 seed iterations), we observed only a small number of cases where overfitting occurred.

5.3 Results

The performance metrics for both models across all random seeds are summarized below:

Table 5: Average Performance Metrics Across 100 Random Seeds

Metric	**Logistic Regression**	**Decision Tree**
F1 Score	0.76	0.74
Accuracy	0.82	0.80
Precision	0.78	0.75
Recall	0.75	0.73

6 Visual Performance Comparison

The boxplots in Figures 1-4 show the distribution of performance metrics across all random seed iterations, providing a visual comparison of both models' stability and effectiveness.

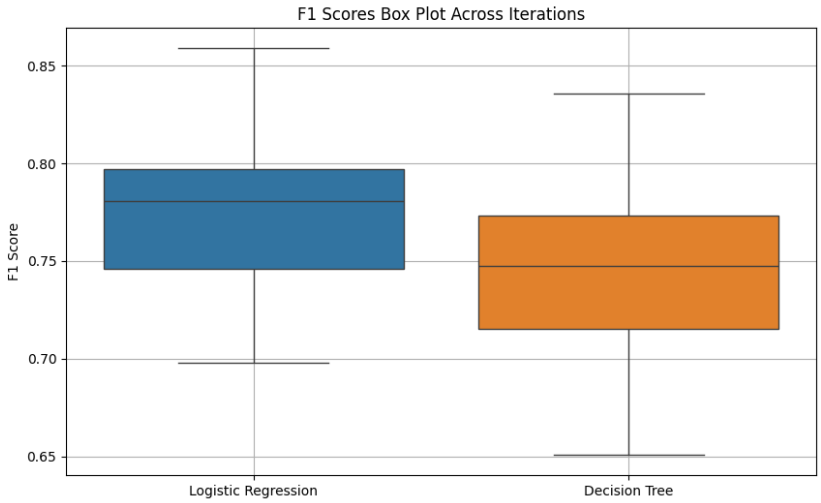

Figure 1: F1 Score comparison between Logistic Regression and Decision Tree

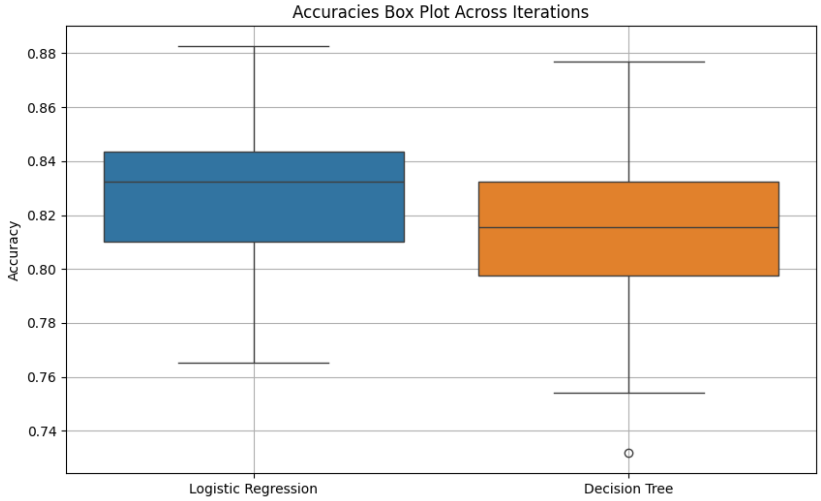

Figure 2: Accuracy comparison between Logistic Regression and Decision Tree

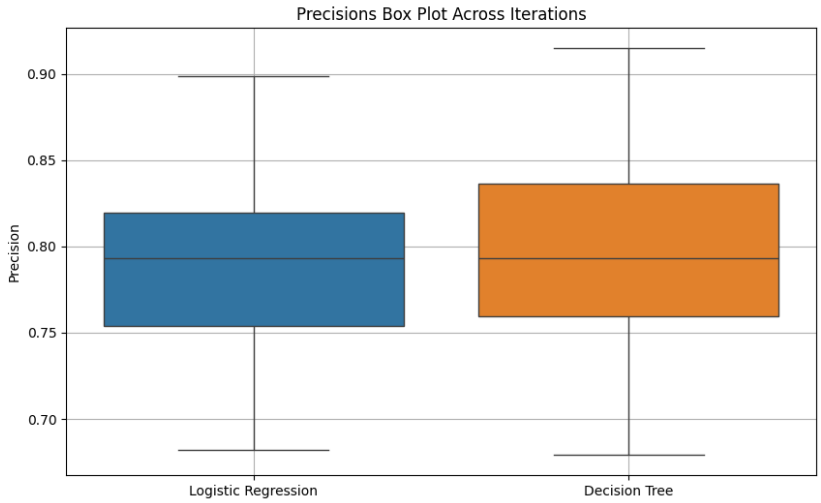

Figure 3: Precision comparison between Logistic Regression and Decision Tree

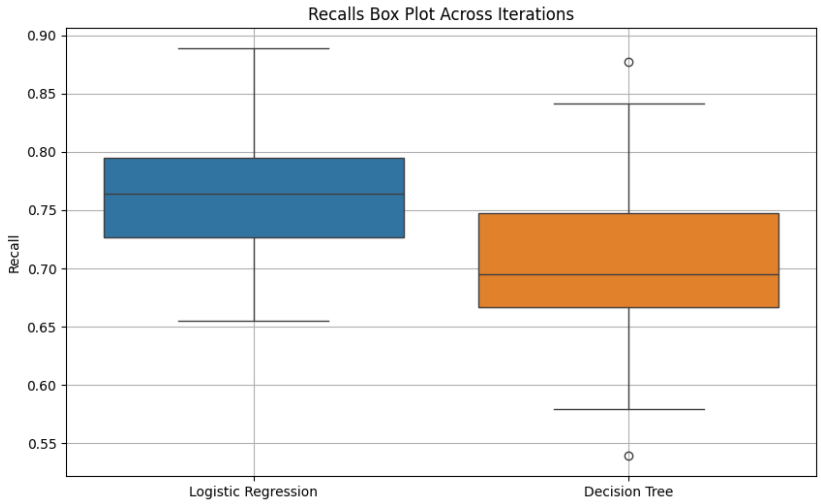

Figure 4: Recall comparison between Logistic Regression and Decision Tree

7 Discussion

7.1 Model Performance Comparison

The Logistic Regression model consistently outperformed the Decision Tree model across all evaluation metrics. Several factors may contribute to this performance difference:

- **Data Characteristics:** The Titanic dataset has a relatively small number of samples (891 in the training set). Logistic Regression often performs well on smaller datasets compared to more complex models.
- **Feature Engineering:** Our feature engineering approach, particularly the extraction of titles and creation of the "is_wife" feature, may have created linear relationships that Logistic Regression could efficiently capture.

- **L1 Regularization:** The best Logistic Regression model used L1 regularization (LASSO), which likely helped with feature selection by pushing less important feature coefficients toward zero.

7.2 Decision Tree Performance

While the Decision Tree model performed slightly worse than Logistic Regression, it still achieved respectable results. The optimal Decision Tree configuration had:

- A max depth of 20, allowing for complex decision boundaries
- A random splitter rather than optimal splits, which likely helped prevent overfitting
- A minimum samples split of 20, which enforces that nodes with fewer than 20 samples aren't split further

These parameters suggest that the model benefited from some constraints to avoid overfitting, while still maintaining enough complexity to capture the patterns in the data.

7.3 Stability Analysis

Both models demonstrated good stability across different random seeds, with relatively tight interquartile ranges in the performance boxplots. This stability suggests that the preprocessing and model configurations were robust and not overly sensitive to particular data splits.

8 Conclusion

Our analysis shows that Logistic Regression provides the best performance for predicting Titanic survival, with an average F1 score of 0.76 and accuracy of 0.82 across 100 random seeds. The Decision Tree approach performed competitively but consistently lower than Logistic Regression across all metrics.

The feature engineering steps, particularly extracting passenger titles and creating composite features like family size, proved valuable for both models. The L1 regularization in the Logistic Regression model likely contributed to its effectiveness by focusing on the most informative features.

Given these results, the Logistic Regression model with the parameters described in Table 3 would be our recommended model for deployment on the Titanic survival prediction task.