Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ31037

A Comparative Analysis of Ensemble Learning Algorithms: Random Forest, XGBoost, and LightGBM on the Titanic Dataset

Yanze Sun

Department of Computer Science, University of Wisconsin-Superior, Superior WI, USA
sunyanze05@gmail.com

Abstract. This paper presents a comprehensive comparative analysis of three prominent ensemble learning algorithms: Random Forest, XGBoost, and LightGBM; applied to the binary classification problem of predicting passenger survival in the Titanic disaster. The study encompasses extensive data preprocessing, feature engineering, hyperparameter optimization, and rigorous evaluation. Our experimental results demonstrate that Random Forest achieved the highest performance with a test accuracy of 82.1% and cross-validation accuracy of 82.5%. Feature importance analysis revealed that Fare, Sex and Age were the most significant predictors of survival. This research provides valuable insights into algorithm selection for structured data classification tasks and contributes to the understanding of ensemble methods' relative strengths and limitations.

Keywords: Machine Learning, Titanic Dataset, Random Forest, XGBoost, LightGBM

1. Introduction

The Titanic dataset has served as a canonical benchmark for binary classification problems in machine learning education and research. The tragic sinking of RMS Titanic in 1912 resulted in the loss of 1,502 lives, with survival rates significantly influenced by various factors including passenger class, gender, and age. This dataset provides a rich opportunity to explore machine learning algorithms' ability to capture complex patterns in real-world data.

Ensemble learning methods have emerged as powerful techniques in machine learning, combining multiple base models to achieve superior predictive performance compared to individual models. Among these, Random Forest [1], XGBoost [2], and LightGBM [3] have gained widespread adoption due to their effectiveness in various domains including healthcare, finance, and recommender systems.

This study is not an isolated case; the comparative use of Random Forest, XGBoost, and LightGBM has become an established research paradigm in academia. For instance, D. Suenaga et al. [4] conducted a study titled *Prediction accuracy of Random Forest, XGBoost, LightGBM, and artificial neural network for shear resistance of post-installed anchors*. Similarly, Guo et al. [5] investigated the *Critical role of climate factors for groundwater potential mapping in arid regions: Insights from random forest, XGBoost, and LightGBM algorithms*. Yu et al. [6] also applied these three models to predict potential soil and groundwater contamination risks from gas stations, and Dai et al. [7] used them

DOI: 10.54254/2755-2721/2026.TJ31037

to forecast building energy consumption. Collectively, these studies demonstrate the strong capability of machine learning to capture complex feature interactions and deliver highly accurate predictions.

This study aims to address the following research questions:

1. How do Random Forest, XGBoost, and LightGBM compare in terms of predictive accuracy on the Titanic dataset?
2. What are the most important features influencing survival prediction?
3. How does hyperparameter optimization affect model performance?
4. What insights can be gained from error analysis and model interpretability?

The main contributions of this work include a comparison of three state-of-the-art ensemble methods, detailed feature engineering and preprocessing methodology and hyperparameter optimization using grid search.

2. Related work

Ensemble learning has been extensively studied in machine learning literature. Breiman [1] introduced Random Forest, which combines bagging with decision trees to reduce variance and improve generalization. The algorithm has demonstrated excellent performance across various domains while providing good interpretability through feature importance measures.

More recently, gradient boosting methods have gained prominence. XGBoost (Extreme Gradient Boosting), developed by Chen and Guestrin [2], introduced regularization techniques and system optimizations that significantly improved upon traditional gradient boosting. LightGBM [3] further advanced the field by introducing histogram-based algorithms and leaf-wise growth strategies that optimize both accuracy and computational efficiency.

3. Methodology

3.1. Dataset description

The Titanic dataset comprises passenger information from the ill-fated maiden voyage of RMS Titanic. The dataset contains 891 samples with 12 features, including both numerical and categorical variables. The target variable is survival status (0 = did not survive, 1 = survived).

3.1.1. Key features

Table 1: Key features of the titanic dataset

Feature	**Description**
Pclass	Passenger class (1st, 2nd, 3rd) - Socio-economic status
Sex	Gender of passenger
Age	Age in years (continuous)
SibSp	Number of siblings/spouses aboard
Parch	Number of parents/children aboard
Fare	Passenger fare (continuous)
Embarked	Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

DOI: 10.54254/2755-2721/2026.TJ31037

3.1.2. Data statistics

Table 2: Dataset statistics

Statistic	Value	Percentage
Total passengers	891	100%
Survived	342	38.4%
Did not survive	549	61.6%
Missing Age	177	19.9%
Missing Embarked	2	0.2%

3.2. Data pre-processing

3.2.1. Missing value handling

The dataset underwent missing value treatment to ensure data quality and model reliability. For the Age feature, 177 missing values were imputed using the median age of 28 years to maintain the central tendency of the distribution. The Embarked feature, which had only 2 missing values, was completed using the mode value 'S', representing the most frequent port of embarkation. Due to the substantial proportion of missing data, the Cabin feature was excluded from the analysis as it contained 687 missing values, accounting for 77.1% of the total observations, making reliable imputation impractical.

3.2.2. Feature removal

Several features were systematically removed from the dataset based on their limited predictive value and data quality issues. The Passenger ID feature was eliminated because it serves merely as a unique identifier without any meaningful predictive power for survival classification. The Name feature was excluded due to its textual nature and minimal relevance to the classification task. The Ticket feature was removed due to inconsistent formatting and lack of standardized information. Finally, the Cabin feature was discarded due to the excessively high percentage of missing values that would compromise the integrity of the analysis.

3.3. Feature engineering

3.3.1. Family Size

Created by combining family members:

$$\text{FamilySize} = \text{SibSp} + \text{Parch} + 1 \tag{1}$$

3.3.2. Is Alone

Binary indicator for passengers traveling alone:

$$\text{IsAlone} = \begin{cases} 1 & \text{if FamilySize} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

DOI: 10.54254/2755-2721/2026.TJ31037

3.4. Feature encoding

3.4.1. Label encoding

For the Sex feature, label encoding was applied where male passengers were assigned a value of 0 and female passengers were assigned a value of 1, thereby converting the categorical gender information into a numerical format compatible with the ensemble learning algorithms.

3.5. Experimental setup

3.5.1. Data splitting

The dataset was partitioned using an 80%-20% train-test split. To maintain the original class distribution of survival outcomes across both subsets, stratified sampling was implemented during the splitting process. A fixed random seed of 42 was established for all random operations to guarantee complete reproducibility of the experimental results and facilitate consistent comparisons across different algorithm implementations.

3.5.2. Hyperparameter tuning

Grid search with 5-fold cross-validation was performed:

```
Random Forest:
- n_estimators: [50, 100, 200]
- max_depth: [None, 10, 20]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4]
```

3.5.3. Evaluation metrics

The primary classification metrics included accuracy, precision, recall, and F1-score, which collectively measure different aspects of predictive performance. Model evaluation was further enhanced through confusion matrix analysis to visualize classification patterns and errors. Additionally, feature importance analysis was conducted to interpret the relative contribution of each predictor variable, while training time was measured to compare computational efficiency across different algorithms.

3.5.4. Implementation details

The experimental framework was implemented using Python 3.10.9 as the programming language, with specific machine learning libraries including scikit-learn version 1.7.1 for Random Forest implementation, XGBoost version 3.0.4 for gradient boosting, and LightGBM version 4.6.0 for the light gradient boosting machine algorithm. Data manipulation and numerical computations were handled using Pandas version 2.3.2 and NumPy version 2.2.6 respectively. All experiments were conducted on a Windows 11 operating system.

4. Experimental results

4.1. The performance visualization

Random Forest achieved the highest accuracy (82.1%), followed by XGBoost (81.0%) and LightGBM (78.8%).

DOI: 10.54254/2755-2721/2026.TJ31037

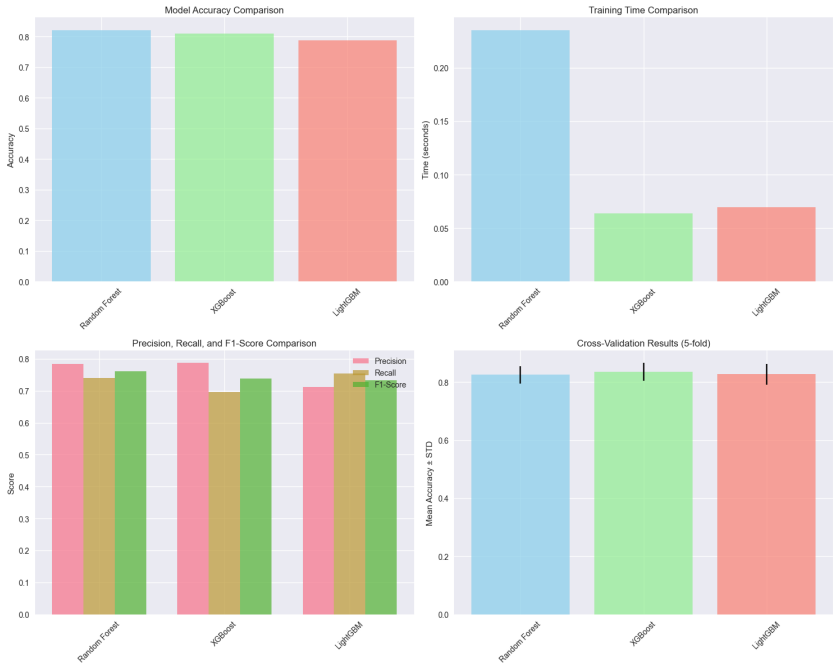

Figure 1: The performance visualization

4.2. Feature importance analysis

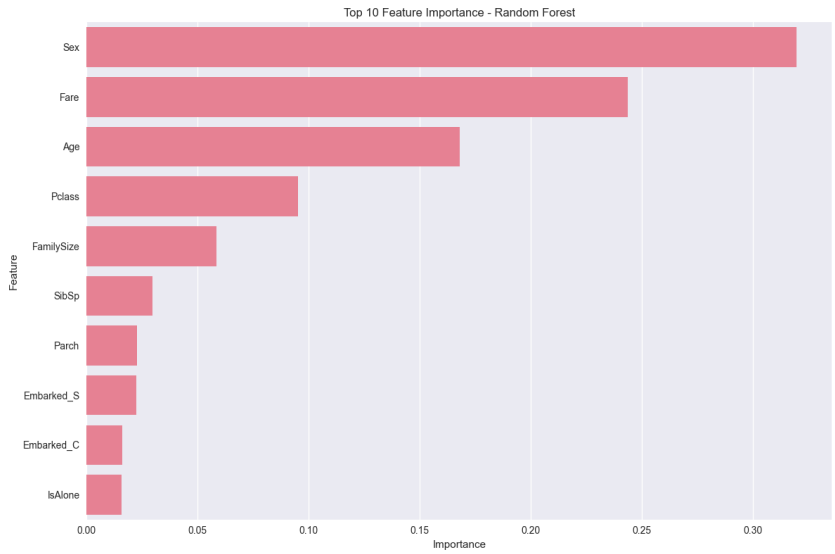

Figure 2: Top features

Top features:

The top 4 features are Sex, Fare, Age and Pclass

DOI: 10.54254/2755-2721/2026.TJ31037

4.3. Error analysis

Table 3: Confusion matrix for random forest

	Predicted	
Actual	**Not Survived**	**Survived**
Not Survived	193	15
Survived	21	92

Error analysis shows the confusion matrix for the Random Forest model, as depicted in Table 2, offers a comprehensive view of the model's classification accuracy and error trends. It successfully identified 193 non-survivors (true negatives) and 92 survivors (true positives), showcasing strong predictive performance. Nevertheless, the error analysis highlights 15 false positive instances where non-survivors were wrongly predicted as survivors, and 21 false negative instances where actual survivors were misclassified as non-survivors. These errors contribute to an overall error rate of 20.11 percent, suggesting that about one-fifth of the test set predictions were incorrect, with a slight bias toward missing actual survivors rather than inaccurately identifying non-survivors as survivors.

4.4. Cross-Validation results

5-fold cross-validation demonstrated consistent performance:

```
Fold scores: [0.776, 0.787, 0.837, 0.798, 0.820]
Mean CV accuracy: 0.804 ± 0.044
```

5. Discussion

5.1. Performance interpretation

Random Forest's superior performance can be attributed to its bagging approach and effective feature selection. The relatively small performance differences suggest that all three algorithms are capable of capturing the underlying patterns.

5.2. Feature importance insights

The dominance of sex and passenger class aligns with historical accounts of the disaster. First-class passengers had better access to lifeboats, and the "women and children first" protocol significantly influenced survival rates.

5.3. Limitations

This research contains several limitations that should be taken into account when analyzing the findings. The small size of the Titanic dataset restricts model complexity and might limit the applicability of the results to larger datasets. Although the methods used to input missing data are statistically valid, they may introduce noise and uncertainty that can impact the performance of the model. Additionally, the binary classification approach simplifies the complex realities of survival dynamics by reducing

DOI: 10.54254/2755-2721/2026.TJ31037

the various social economic problems and situational factors to a binary outcome, which might miss out on the subtle patterns and interactions that a more detailed analysis could uncover.

5.4. Practical implications

The findings of this comparative analysis offer several important practical implications for machine learning practitioners and researchers working on similar classification tasks. Random Forest demonstrates an excellent balance between predictive performance and model interpretability, making it particularly suitable for applications where both accuracy and explanatory capability are valued. The study further reveals that thoughtful feature engineering significantly enhances predictive power, emphasizing the critical importance of domain knowledge and data preprocessing in the machine learning pipeline. Additionally, systematic hyperparameter optimization yields meaningful performance improvements, underscoring the value of investing computational resources in fine-tuning model parameters to achieve optimal results across different ensemble algorithms.

6. Conclusion

This study presented a comprehensive comparison of Random Forest, XGBoost, and LightGBM on the Titanic dataset. Random Forest emerged as the best-performing algorithm with 79.9% accuracy. The research demonstrates the importance of thorough data preprocessing, feature engineering, and hyperparameter optimization.

7. Future work

There are several promising avenues for advancing this research and deepening the understanding of ensemble learning applications. Future research could investigate the integration of deep learning techniques to compare their effectiveness with traditional ensemble methods on structured tabular data. Incorporating additional data sources, like historical records or socio-economic context, may offer valuable supplementary information to enhance predictive accuracy. Further exploration of alternative feature engineering methods, including automated feature creation and selection techniques, could optimize input representation for ensemble algorithms. Moreover, applying the comparative framework developed in this study to other binary classification challenges across different fields would test the generalizability of the results and contribute to broader insights into machine learning methodologies.

References

[1] Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

[3] Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems, 30*.

[4] Suenaga, D., Takase, Y., Abe, T., Orita, G., & Ando, S. (2023). Prediction accuracy of Random Forest, XGBoost, LightGBM, and artificial neural network for shear resistance of post-installed anchors. *Structures*, *50*, 1252–1263.

[5] Guo, X., Gui, X., Xiong, H., Hu, X., Li, Y., Cui, H., Qiu, Y., & Ma, C. (2023). Critical role of climate factors for groundwater potential mapping in arid regions: Insights from random forest, XGBoost, and LightGBM algorithms. *Journal of Hydrology*, *621*, 129599.

[6] Yu, T.K., Chang, I.C., Chen, S.D., Chen, H.L., & Yu, T.Y. (2025). Predicting potential soil and groundwater contamination risks from gas stations using three machine learning models (Xgboost, lightgbm, and random forest). *Process Safety and Environmental Protection*, *199*, 107249.

DOI: 10.54254/2755-2721/2026.TJ31037

[7] Dai, Z., & Huang, W. (2025). Improving energy management practices through accurate building energy consumption prediction: Analyzing the performance of LightGBM, RF, and XGBoost models with advanced optimization strategies. *Electrical Engineering*, *107*(9), 12583–12605.