Who's Going To Swim With The Fish?

A Titanic Analysis

By Austin Coulter, Peter Nguyen, Preethi Vezhavendan

Summary: This report presents a comprehensive comparative analysis of machine learning models applied to predicting Titanic passenger survival. The analysis evaluated nine distinct models across four different methodological approaches: logistic regression, Support Vector Machine(SVM) kernel methods, tree-based ensembles, and discriminant analysis. Using standardized preprocessing and consistent evaluation metrics, we identified an optimized Random Forest and SVM with Radial Basis Function Kernal as the optimal models, both achieving 83.413% test accuracy. The findings demonstrate that tree-based ensemble methods substantially outperform linear approaches for this classification task, suggesting that survival patterns in the Titanic dataset are inherently non-linear and complex.

1 Introduction

Classification problems represent one of the most fundamental and pervasive challenges in machine learning, with applications spanning medical diagnosis, financial fraud detection, image recognition, natural language processing, and risk assessment across virtually every industry. At its core, classification seeks to assign observations to discrete categories based on their characteristics—a task that humans perform intuitively but that requires sophisticated mathematical frameworks to automate at scale. Supervised machine learning methods have emerged as the dominant application for addressing these challenges, leveraging labeled historical data to learn patterns that generalize to unseen cases.

This analysis undertakes a comprehensive evaluation of supervised machine learning methods applied to the historical Titanic disaster. On April 15, 1912, the RMS Titanic struck an iceberg in the North Atlantic, leading to one of history's most infamous maritime disasters. Of the approximately 2,224 passengers and crew aboard, only about 710 survived, representing a survival rate of roughly 32%. Critically, survival was not random. Historical accounts document systematic patterns. The "women and children first" evacuation protocol, stark class-based disparities in lifeboat access, and varying survival rates by age, family composition, and socioeconomic status. These patterns suggest that survival was predictable based on measurable passenger characteristics. This is precisely the type of problem where supervised classification excels.

Our analysis evaluates nine distinct supervised learning models. Logistic Regression, Support Vector Machines with linear, RBF, and sigmoid kernels, Random Forests, Gradient Boosting, Bagging, and Linear and Quadratic Discriminant Analysis. Each model offers different approaches to the classification problem. Linear methods assume that decision boundaries can be represented as hyperplanes, offering interpretability but potentially insufficient flexibility. Kernel methods implicitly map data to higher-dimensional spaces where complex patterns become linearly separable, providing powerful non-linear capabilities. Tree-based ensembles partition feature space through recursive splits and aggregate multiple models to reduce variance and improve generalization. Discriminant analysis takes a probabilistic approach, modeling class-conditional distributions and applying Bayes' theorem for optimal classification under normality assumptions.

We investigate which model achieves the highest accuracy, why certain approaches outperform others, which features drive predictions, and what trade-offs exist between predictive performance, and interpretability. The analysis reveals that tree-based ensemble methods, particularly Random Forests, achieve superior performance compared to linear approaches, suggesting that Titanic survival patterns exhibit non-linear interactions that simpler models cannot capture. Vector Machines with RBF kernels perform comparably to ensembles, validating the hypothesis that non-linear decision boundaries are essential for this task.

2 Data

The Titanic dataset, contains features capturing demographics, family relationships, ticket information, and spatial cabin assignments, and provides an ideal testbed for comparing algorithmic approaches. In total, there were 891 observations, and 26 variables in the titanic dataset. Of the observations, 342 were survivors, and 549 were deceased. From our dataset, we have a survival rate of about 38.4%, which is slightly higher than the true survival rate. This might lead to some bias, but for the sake of this project and keeping our sample size high, we will ignore it. To assess whether someone survived or not, we used 20 different independent variables in our analysis to model survival. Although we omit details of these 20 independent variables for the sake of brevity, they represent different demographic and socioeconomic information on the passengers aboard that are useful for classification purposes.

We implemented standardized preprocessing pipelines, maintained identical train-test splits, and evaluated all models using consistent metrics of accuracy, precision, recall, F1-score, and ROC-AUC. This study provides an apples-to-apples comparison of all models we created. We started out by removing the Cabin, cabin room number, title, PassengerId, Name, and Ticket columns from the dataset because they either had too many null values or were not useful in prediction. Then we filled in the missing age and Embarked rows with the median age and the mode of Embarked. There wasn't too many missing rows, so we wont worry about it affecting the data. Once there were no more missing data, we binary coded Sex to 1 and 0. For the tree based methods, we used label encoding to convert the Embarked, title group, and cabin deck columns to numerical values. For the other methods, we used one hot encoding to turn each column into p-1 boolean columns, where p is the number of distinct values in each column. Once the data was ready, we started fitting the models. For these non tree based methods, we also use standard scaling before fitting the models.

3 Analysis

To carry out our analysis, we begin by carrying out hyperparameter Tuning Methodology for all individual analyses by using 5-fold cross-validation within GridSearchCV, systematically exploring the following parameter combinations.

Model Hyperparameter Search

Discriminant Analysis

LDA solver: `svd`, `lsqr`, `eigen`

QDA regularization: $\{0.0, 0.1, 0.5\}$

Grid search: Not applied

Logistic Regression

C**:** $\{0.001, 0.01, 0.1, 1, 10, 100\}$

Penalty: $\{\ell_1, \ell_2\}$

Solver: `liblinear`, `saga`

Class weight: `None`, `balanced`

Metric: ROC–AUC

Support Vector Machines (SVM)

Linear kernel C: $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$
RBF kernel: $C \in \{0.1, 1, 10, 100, 1000\}$, $\gamma \in \{\texttt{scale}, \texttt{auto}, 0.001, 0.01, 0.1, 1\}$
Sigmoid kernel: $C \in \{0.1, 1, 10, 100\}$, $\gamma \in \{\texttt{scale}, \texttt{auto}, 0.01, 0.1, 1\}$
coef$_0$: $\{0, 0.5, 1\}$
Metric: Accuracy

Tree-Based Ensembles

Max depth: $\{5, 10, 15\}$
Min samples split: $\{10, 20, 30\}$
Min samples leaf: $\{5, 10, 20\}$
 # estimators: $\{50, 100, 200\}$
Metric: Accuracy with F1-score consideration

After running the cross validation and grid search on the parameters in question, the optimal model parameters for the different models are as such.

Optimal Model Parameters

Logistic Regression: {C: 0.1, class weight: None, penalty: l2, solver: saga}

Linear Discriminant Analysis: { Solver: svd }

Quadratic Discriminant Analysis: {regularization: 0.0 }

SVM Linear Kernel: {C: 1 }

SVM RBF Kernel: {C: 10, γ : 0.01 }

SVM Sigmoid Kernel: {C: 0.1, coef0: 1, γ : scale }

Random Forest: {max depth: None, max features: sqrt, min samples leaf: 1, min samples split: 10, n estimators: 100}

Gradient Boosting: {learning rate: 0.05, max depth: 3, min samples leaf: 5, min samples split: 20, n estimators: 300, subsample: 0.9}

This table summarizes the performance of several classification models evaluated using Accuracy, Precision, Recall, F1 Score, and ROC AUC. In general, ensemble and nonlinear models tend to achieve the strongest predictive performance, while simpler linear models provide competitive results with greater interpretability.

Model	Test Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic	0.7892	0.73	0.77	0.7458	0.8476
LDA	0.8027	0.7528	0.7528	0.7528	0.8602
QDA	0.7892	0.6981	0.8315	0.76	0.8304
SVM Linear	0.8296	0.8000	0.7640	0.7816	0.838756
SVM RBF	0.8341	0.8171	0.7528	0.7836	0.844877
SVM Sigmoid	0.7758	0.7191	0.7191	0.7191	0.847895
Random Forest	0.8341	0.7889	0.7978	0.7933	0.8814
Bagging	0.8161	0.8158	0.6966	0.7515	0.8647
Gradient Boosting	0.8117	0.7901	0.7191	0.7529	0.8757

4 Discussion

Among all models, Random Forest and SVM with an RBF kernel achieve the highest test accuracy (0.8341). Random Forest also obtains the best ROC-AUC (0.8814) and the highest F1-Score (0.7933), indicating a strong balance between precision and recall and excellent discrimination capability across classification thresholds. This suggests that Random Forest is particularly effective at capturing nonlinear interactions and complex feature relationships in the data. One reason why these models are more effective than QDA, is because QDA makes strong assumptions, limiting its ability to capture the complex relationships even though it fits a nonlinear model.

The SVM RBF model performs comparably, with high precision (0.8171) and strong F1-Score (0.7836). This reflects the power of kernelized SVMs to learn nonlinear decision boundaries. However, its recall is slightly lower than that of Random Forest, implying a more conservative classification strategy. Tree-based ensemble methods such as Gradient Boosting and Bagging also perform well, with ROC-AUC values above 0.86. Gradient Boosting shows a good tradeoff between bias reduction and variance control, while Bagging improves stability over single decision trees but achieves lower recall compared to other ensemble methods. Another reason why ensemble methods generally work better, is because they can support non-continuous decision boundaries compared to the other models.

Among simpler models, Linear SVM stands out, achieving high precision (0.8296), strong precision (0.8000), and a F1-Score (0.7816) close to that of more complex models. This suggests that much of the class separation can be captured by a linear boundary. Logistic Regression and LDA perform similarly, with LDA slightly outperforming Logistic Regression in accuracy and ROC-AUC. The stronger performance of LDA reflects its use of class-conditional distributions under Gaussian assumptions. QDA achieves notably high recall (0.8315), indicating sensitivity to the positive class, but at the cost of reduced precision. The SVM with a sigmoid kernel performs poorly relative to other SVM variants, indicating that this kernel may not be well matched to the data structure.

There is a clear tradeoff between interpretability and flexibility across models. Logistic Regression, LDA, and Linear SVM are highly interpretable. Coefficients and decision boundaries can be directly examined, making them suitable for settings where transparency is critical. However, their limited flexibility may prevent them from capturing complex nonlinear patterns. In contrast, Random Forests, Gradient Boosting, and RBF-kernel SVMs offer greater flexibility at the expense of interpretability. Their superior predictive performance suggests that the underlying data relationships are at least partially nonlinear. However, understanding individual predictions becomes more difficult.

The results also illustrate the classic bias–variance tradeoff. Simpler models exhibit higher bias but lower variance, leading to stable but less flexible decision boundaries. More complex models (QDA, RBF SVM) reduce bias but increase variance, which can improve recall or accuracy but risks overfitting. Ensemble methods strike a balance. Random Forest reduces variance through averaging, while Gradient Boosting incrementally reduces bias, leading to strong generalization performance. The superior results of Random Forest suggest that variance reduction was particularly beneficial for this dataset.

5 Conclusion

Overall, the results demonstrate that no single model is universally optimal; rather, model performance depends on the balance between predictive accuracy, robustness, interpretability, and tolerance for variance. Among the evaluated approaches, Random Forest emerges as the strongest overall performer, achieving the highest ROC-AUC and F1-Score while maintaining competitive accuracy. The strong recall and balanced precision further suggest that Random Forest provides reliable classification across both positive and negative classes.

While Random Forests and kernelized SVMs offer superior predictive power, their decision processes are less transparent, making them less ideal in contexts where model explainability, accountability, or regulatory compliance is required. In contrast, Logistic Regression, LDA, and Linear SVM provide highly interpretable decision boundaries and parameter estimates. Although their performance is slightly inferior, particularly in terms of ROC-AUC and F1-Score, the gap is relatively modest. This suggests that for applications prioritizing transparency and ease of explanation, simpler linear models may represent a reasonable and defensible compromise.

In conclusion, while Random Forest is the preferred model when predictive performance is the primary objective, linear and discriminant models remain attractive alternatives in interpretability-driven or resource-constrained settings. The results underscore the importance of evaluating models not only on raw performance metrics but also on their practical implications, reinforcing that model selection should be guided by both statistical performance and application-specific priorities.

References

1. James, Gareth, Witten, Daniela, Hastie, Taylor, and Tibshirani, Robert (2023). *An Introduction to Statistical Learning: With Applications in Python*. Springer, New York.

2. Goodfellow, Bengio, Courville (2016). *Deep Learning*, MIT Press.

3. Wikimedia Foundation. (2026). Titanic. Wikipedia. https://en.wikipedia.org/wiki/Titanic

Appendix

A1. Further explaination on Discriminant analysis

In figure 1 below, we see the how LDA and QDA are used to fit the data. The LDA and QDA predicted probability distribution plots show histograms of the models' confidence in their survival predictions, separated by actual outcome. The vertical dashed line at 0.5 represents the decision boundary. Predictions above 0.5 are classified as "survived" and below 0.5 as "did not survive." Ideally, you want to see good separation between the two distributions, with the red bars clustered near 0 and green bars clustered near 1. Overlap between the distributions indicates uncertainty or cases where the model struggles to confidently classify passengers. The LDA plot generally shows cleaner separation with more extreme probabilities, suggesting the model makes more confident predictions, while QDA might show slightly more overlap and intermediate probabilities, indicating areas of uncertainty where the quadratic boundaries don't separate classes as cleanly. This visualization helps assess not just accuracy but also model calibration and confidence, which is crucial for understanding when the model is certain versus when it's making borderline decisions.

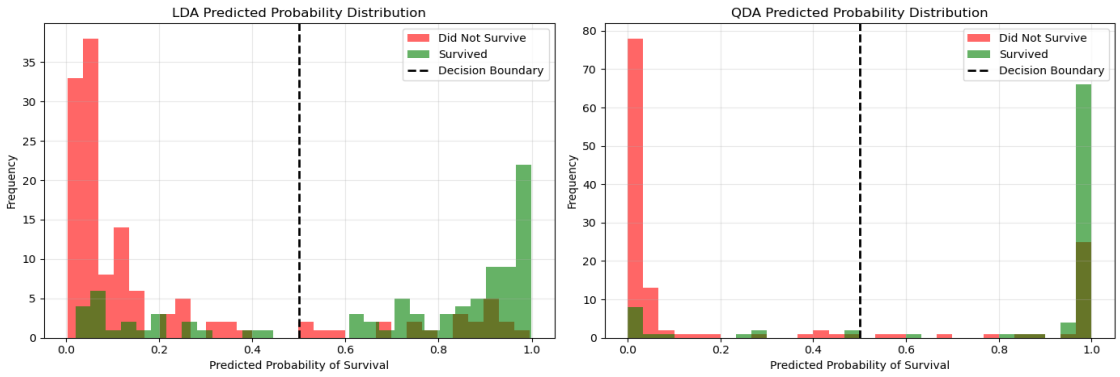

Figure 1: Decision Boundary for Discriminant Analysis

A2. Further explanation on ensemble methods

In figure 2 below, we see the model comparison of all the ensemble methods. The Model Performance Comparison and Overfitting Analysis plots provide a comprehensive visual assessment of how the four tree-based models perform across multiple evaluation metrics. The left bar plot displays five key metrics, training accuracy, test accuracy, precision, recall, and F1-score, allowing you to quickly compare how each model balances overall correctness, false positive avoidance, true positive detection, and their harmonic mean. Models with consistently high bars across all metrics are generally superior performers. The right scatter plot specifically diagnoses overfitting by plotting each model's training accuracy (x-axis) against its test accuracy (y-axis), with a red dashed diagonal line representing perfect generalization where training and test performance are identical. Models plotting close to this line demonstrate good generalization, while those significantly above the line indicate overfitting. Conversely, models below the line would suggest

underfitting, and models in the lower-left region indicate overall poor performance. This dual visualization helps identify not just which model performs best, but whether that performance is robust and trustworthy for real-world predictions or simply an artifact of memorizing training examples.

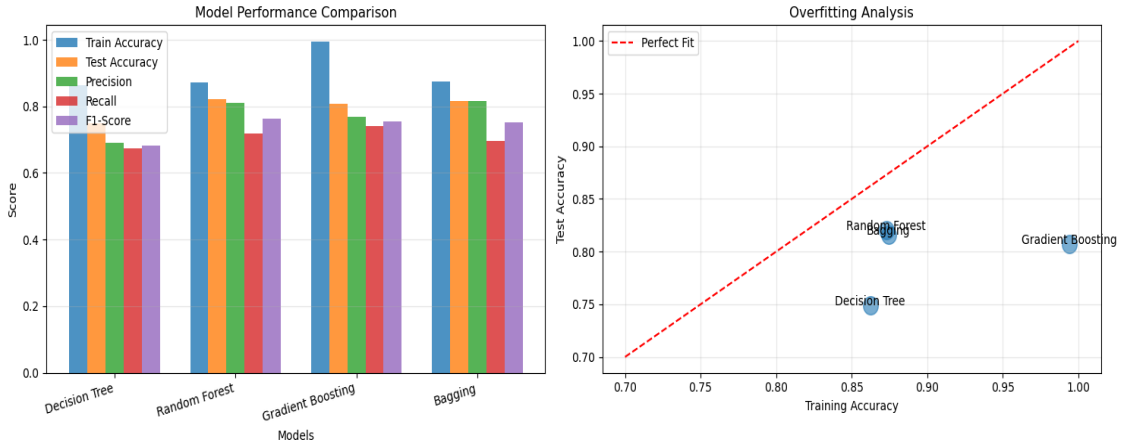

Figure 2: Model Comparison Between Ensemble Methods

A3. Further analysis on Feature Importance

In figure 3 and 4 below, we see the feature importance for the Random Forest model and Logistic Regression models. The Feature Importance plots provide insights into which variables the models rely on most heavily when making survival predictions. The left bar plot displays each feature's relative importance score calculated from how much each feature reduces impurity across all trees in the forest, sorted from most to least influential. The right cumulative importance plot shows the running sum of feature importance as you add features from most to least important, with a red dashed line at 95% indicating the threshold where most of the model's predictive power is captured. The point where the curve crosses this line tells you the minimum number of features needed to explain 95% of the model's decision-making, which is useful for feature selection and understanding whether the model relies on a few dominant features or distributes importance across many variables. Together, these plots help identify which passenger characteristics are most critical for predicting Titanic survival and reveal whether simpler models with fewer features might perform nearly as well.

The difference in important features between Random Forest and Logistic Regression stem from fundamental algorithmic differences. Logistic Regression assumes linear, additive relationships and calculates a weighted sum where each coefficient represents a feature's independent effect on survival, and it's sensitive to multicollinearity. Random Forest measures importance by how much each feature reduces impurity across tree splits, naturally capturing non-linear relationships, feature interactions, and threshold effects without any linearity assumption. Because Random Forest uses bootstrap sampling and random feature subsets, it handles correlated features better.

Therefore, Logistic Regression favors features with strong linear discriminative power, while Random Forest prioritizes features that create optimal splits or interact with others, even when those relationships aren't linear.

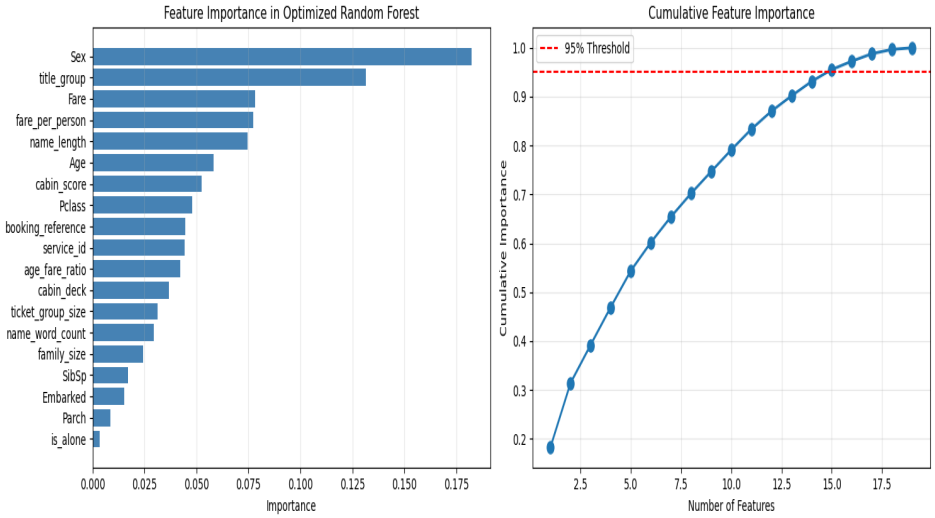

Figure 3: Random Forest Feature Importance

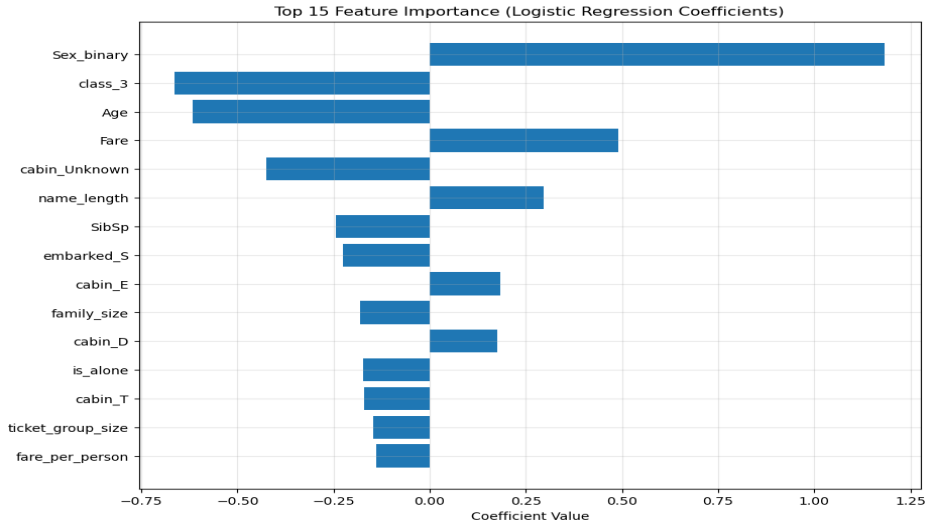

Figure 4: Logistic Regression Feature Importance