MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبد الرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

Final Report Machine Learning

Predicting Survival on the Titanic Using Machine Learning Techniques

ARTI 406 – Machine learning
Academic Year (2024 - 2025)
Second Semester

Group Members

Fajer Mohammed Alzamanan (Leader)
Fellwa Khalid Alhudaithi
Alhanouf Abdullah Alqahtani
Sara Sultan Alzahrani
Maram Mohammed Alnabrees
Sajedah Abdulelah Alqudaihi
Hanan Alshumrani

Supervised by:

Dr. Rabab Alkhalifa

ARTI406 - FA01

Table of content

Table of table

Table of figure

Table 1 Table of Acronyms

Acronym	Meaning
AUC	Area Under Curve
ROC	Receiver Operating Characteristic
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
PCA	Principal Component Analysis
RF	Random Forest
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations (for model interpretability)
LIME	Local Interpretable Model-Agnostic Explanations
RF	Random Forest
SVM	Support Vector Machine
XGBoost	Extreme Gradient Boosting
RBF	Radial Basis Function
KNN	K- Neural Network
ANN	Artifical Neural Network
ADA	ADABoost

Abstract

This study offers an effective machine learning approach to predicting passenger survival in the Titanic disaster and addressing gaps observed in the earlier studies. Our goal in this project is to identify how predictive models can discover patterns and relations in historical data, improve the prediction accuracy, and discover the most important feature in survival. While previous studies have attempted similar predictions, they rely on traditional machine learning models without using advanced ensemble techniques, they also applied minimal feature engineering steps, and poorly handled the missing data. Our study responds to these gaps by exploring broader models, including ensemble classifiers, conducting rich feature engineering and selection processes, handling the dataset effectively, and extensive hyperparameter tuning to enhance performance. The Support Vector Machine algorithm was selected due to the model's ability to find the best decision boundaries in a complex dataset. Random Forest was chosen for its strong ability to minimize overfitting by utilizing ensemble learning through several decision trees. Our results show that the Support Vector Machine performed well; however, among the other individual models tested, the Random Forest achieved the best accuracy value. The results also show the effectiveness of using advanced machine learning techniques, such as the Stacking classifier, which achieved the highest accuracy among all other models tested when tested on the full set of features with strong data preprocessing and hyperparameter tuning steps. Additionally, our study also shows that Sex,Fare and Pclass are the most critical features in survival. The insight from this project can contribute to the wider field of machine learning in historical data analysis and may have implications for improving risk assessment and decision-making processes in modern disaster management.

1. Introduction

Machine learning is now a critical predictive analytics technique enabling data-driven decisions in a wide range of domains. Perhaps the most widely used dataset to practice classification problems is the Titanic dataset, which contains granular passenger data such as demographics, ticket class, fare, and survival. The prediction of passenger survival is a fundamental machine learning problem illustrating essential concepts such as classification, feature selection, and model evaluation. This study examines the performance of two widely used classification algorithms Random Forest (RF) and Support Vector Machine (SVM) in predicting survival outcome based on the Titanic dataset. In addition to measuring prediction accuracy, the aim is also to identify the most significant features in survival. RF, which is an ensemble method, builds an ensemble of decision trees to mitigate overfitting and enhance generalization. SVM is, however, a robust supervised machine learning algorithm that constructs the best hyperplane for tagging data, especially optimal for binary classifications such as survival prediction.

Previous research has shown the ability of RF and SVM in survival prediction problems. SVM attained a highest accuracy of 83.9% as per Rajesh M. [6], outperforming RF (82.5%) and GBM (77.6%), exhibiting the ability of SVM in binary classification problems when supported with rigorous preprocessing. Likewise, Kakde et al. [4] established that even though Logistic Regression performed with the highest accuracy rate (83.7%), SVM and RF followed closely behind, affirming their superior performance in classifying tasks With well-structured historical data. A systematic review by Ekinci et al. [5] compared 14 ML techniques and concluded that ensemble models, namely Gradient Boosting and RF, had superior prediction performance (up to 0.82 F-measure). The study highlighted the importance of hyperparameter tuning and feature engineering. Likewise, Adinoyi et al. [7] indicated the efficacy of RF and XGBoost, while RF achieved an 88.12% accuracy, and SVM was found to be closely comparable in other research referred to. Nair [1] also demonstrated RF's utility in enhancing decision stability and classification accuracy, although Logistic Regression was most accurate in that particular study (93.54%). Though NB was also outstanding at 91.3%, its assumption of feature independence limited its application. Further, Dasgupta et al. [8] reported RF as the best performer (80.41%) among the models tested, which further endorsed the application of the ensemble method in intricate survival prediction tasks. Aakriti Singh et al. [2] also compared other models such as RF and SVM and concluded that Logistic Regression provided the best results, although RF provided consistent accuracy, proving the larger conclusion that ensemble methods are good classifiers for this data. Comparison of models in this study focuses on precision, recall, and accuracy in contrast of RF and SVM. The findings in the initial analyses reveal that RF and SVM perform equally well in identifying key survival predictors particularly gender, class, and age—but are relatively different in terms of

prediction accuracy and recall, as reported by Bisht et al. [14] and Liang [11], who also reported minimal variations in model performance between RF and Decision Trees.

In general, literature on the support of RF and SVM on survival classification tasks has been established, with past research confirming their effectiveness and robustness. In this research, the current study contributes to such existing studies by means of a brief evaluation of RF and SVM based on the Titanic dataset by means of lessons learned through past research, as well as careful preprocessing and assessment methods. These findings can further inform current safety measures and improved risk assessment strategies in transport and disaster management.

The remaining part of this work is organized as follows . At the beginning abstract . Section 1 contains introduction . Section 2 contains Review of related literature include gap .Section 3 contains project deliverables of the team. Section 4 contains a description of the proposed techniques include RF and SVM . Section 5 contains empirical studies include description of dataset by statistical analysis and experimental setup , performance measures include classification performance metrics , feature importance and interpretability include feature for RF and SVM and optimization strategy. Section 6 contains the result and discussion include discussion of final results, further discussions. Section 7 contains the conclusion and recommendations . Finally acknowledgment .

Key word : Predicting survival of passengers using various ML models, analysis and preprocessing of the Titanic dataset, model evaluation and comparison.

2.0 Review Of Related Literature

A research by Dr.Prabha Shreeraj Nair [1]. The researches are used ML algorithms to predict Titanic passenger survival based on various factors. Data are from Kaggle, 891 training samples, and 418 test samples with attributes in each sample consisting of age, sex, passenger class, fare paid, and family relations. The experiment compares the performance of four models: NB, LR, Decision Tree, and RF. Feature preprocessing involved missing data handling, one-hot encoding for categorical features, and feature creation, which facilitated improved predictive ability of the features, e.g., "Mother" and "Children" .Feature selection analysis was performed to reach a conclusion that sex, passenger class, and age are significant factors in influencing survival. The models were then evaluated using accuracy and false discovery rate, where LR attained the highest accuracy of 93.54%and the lowest false discovery rate of 8.60%. The approach included cleaning the data, engineering features, and training different classification models to determine the best way of predicting survival. Decision trees gave hierarchical decision rules about survival probability, whereasRF improved the stability of such predictions by averaging multiple decision trees. Although NB provided the best performance at 91.3%, this method assumes that the features are independent of one another, which is not necessarily so. Some of the limitations include: cabin data was partial, feature selection was biased, and constraints over dataset size. This can involve future projects based on other attributes, cross-validation, and experimenting with more sophisticated models such as SVM and KNN. These results again bring out the importance of feature selection and model comparison when performing predictive analytics and how machine

learning insights can enable us to learn some things about certain events in the past, such as the Titanic tragedy.

A paper by Aakriti Singh et al. [2] provides an overview of the Kaggle dataset on the Titanic tragedy of 1912 which consists of a training set with 891 rows and a test set with 418 rows. Each row outlines certain features about a single passenger, such as PassengerID, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Cabin, and Embarked, as well as Fare. The data also analyzes other variables like age, sex, passenger class, and fare to see how they impacted the chances of surviving, using naive bayes, linear SVM, logistic regression, random forest, and CatBoost. Out of these techniques, logistic regression was found to provide the best results along with the lowest false discovery rate. Additionally, the study incorporates works that point out sex as a major determiner of survival and consider social norms, finances, and class as contributing factors. The methodology primarily consisted of extensive data cleaning to deal with missing critical age and cabin columns, followed by feature selection and engineering such as recording gender from a categorical to a numerical variable to improve the models' predictive strength. Perceived model performance was evaluated using confusion matrices focusing on accuracy and false discovery rate. Still, some limitations comprise the dataset's incompleteness, overfitting that may arise due to small sample size, narrow feature set which may ignore other pertinent factors, limited range of metrics evaluation which do not include precision, recall, F1-score, and the likelihood of bias through data transformation processes. The study also points out gaps in the literature on the need for more systematic attempts regarding the comparison of different machine learning algorithms with varying advanced feature engineering techniques, more robust attempts to mitigate bias, and, most importantly, attempt to validate these claims with broader datasets and real-life situations in order to further these findings.

In the paper of Rajib Hossain Khan et al. [3], the authors applied Weka data mining techniques for predicting Titanic passenger survival patterns. The data had class, age, and gender attributes, and the study applied classification and clustering techniques, i.e., the J48 decision tree and SimpleKMeans clustering algorithm. J48 classifier indicated that women and children in first and second class survived the most and men, particularly in the third class, survived the least. The model accuracy was 79.78%, confirming the findings from clustering which indicated that the cluster in which women were predominant had higher chances of survival. However, there were numerous limitations to the research: there was no feature engineering, no appropriate handling of missing values, and only J48 and KMeans were tried without attempting more sophisticated models like XGBoost or Deep Learning. The dataset was also small, and cross-validation was not attempted, which would have provided more robust results. Additional features must be added in future research, ensemble methods experimented with, and cross-validation implemented to make the models more robust. Despite these limitations, the study determines how data mining can be applied to historical data sets for predictive modeling.

A study was conducted by Kakde et al. [4] explored the use of Exploratory Data Analysis (EDA) and machine learning techniques to predict survival rates among Titanic passengers. The significance of feature engineering to identify the key factors like age, gender, ticket class, and fare was emphasized in this study as essential survival factors. The study applied machine learning techniques, including Logistic Regression, Support Vector Machine (SVM), Random Forest, and Decision Tree to evaluate the chances whether passengers will survive. The results showed that Logistic Regression was the best performing model with an accuracy of approximately 83.7%, then Support Vector Machine (SVM), Random Forest, and Decision Tree. The findings indicate that females, children, family of 3 or less, and those who have higher fare and more cabins are more likely to survive. More studies need to be conducted using deep learning algorithms to improve the prediction accuracy.

The study by Ekinci et al. [5] presents a comparative analysis of machine learning techniques using the Titanic dataset with the goal of understanding factors affecting passenger survival. This paper puts to work fourteen varied machine learning methods, namely: Logistic Regression (LR), k-Nearest Neighbors (kNN), Naïve Bayes (NB), Support Vector Machines (SVM), Decision Tree (DT), Bagging, AdaBoost, Extra Trees, Random Forest (RF), Gradient Boosting (GB), Calibrated GB, Artificial Neural Networks (ANN), and two voting approaches that combine multiple classifiers. Above methodology has been followed to select most influencing features that analyze the survival possibility in the Titanic dataset provided on Kaggle. Feature engineering has been performed by adding new variables and treating missing data so that the model's performance may improve. The experimental results indicated that the most accurate model is the voting ensemble of GB, ANN, and kNN, with an F-measure of 0.82, beating the benchmark in Kaggle. The best individual model is GB, with an F-measure of 0.815. Dominant factors affecting the survival predictions include gender, passenger class, age, and fare. It also presents the accuracy of various models with their F-measure values, and among them, it is observable that generally, the ensemble methods tend to outperform other classifiers. It also infers that on historical datasets, machine learning techniques can indeed make quite accurate predictions of survival probabilities, especially for the ensemble methods. The insights brought out are feature engineering, hyperparameter tuning, and model selection as very important to realize high prediction accuracy.

In Rajesh M.'s [6] paper, the study applied machine learning algorithms in the prediction of the survival of Titanic passengers using Random Forest, Support Vector Machine (SVM), and Gradient Boosting Machine (GBM). The data, which was downloaded from Kaggle, consisted of 891 training data and 418 test data with the most significant features including passenger class, age, gender, fare, and family relationships. Data preprocessing included removal of irrelevant features (PassengerID, Name, Ticket, and Cabin) and replacement of missing values in Age and Fare with the median. EDA revealed that females and children survived more, and lower-class male passengers were least likely to survive. Model performance was evaluated using accuracy and Kappa scores, with SVM being the best performer at 83.9% accuracy, followed by Random Forest (82.5%) and GBM (77.6%). Feature selection and preprocessing were highlighted as being

instrumental to the improvement of model performance. Limitations stated were the lack of deep learning models, lack of cross-validation, and small size of the dataset, which could affect generalizability. Future research must explore ensemble methods, additional survival-related features, and deep learning models to boost predictive performance. Despite the limitations, the study demonstrates the feasibility of machine learning for survival prediction and provides a comparison of classification models.

In paper by A. Adinoyi et al. [7]. The researchs are concerned with Titanic passenger survival prediction based on ML models such as Decision Trees, Random Forest, XGBoost, and CatBoost. The dataset, which was obtained from Kaggle, has 891 training instances and 471 test instances with 12 features including age, gender, ticket class, and fare. EDA identified the most important survival factors, and it was noted that women, children, and first-class passengers survived at a higher rate. Imputation techniques were applied to handle missing values in attributes like Age, Cabin, and Embarked. Data preprocessing included one-hot encoding for categorical features and numeric value normalization. Training data and test data were divided, and grid search was employed to optimize hyperparameters. The performance of the models was compared in terms of accuracy, AUC, precision, recall, and confusion matrices. The best performance has come with XGBoost: 91% accuracy, 0.89 AUC-it has emerged as superior in predictive capability. Random Forest and Decision Trees show a performance of 88.12%, while CatBoost shows a performance of 87.21% with increased computation. ROC curves and boxplots were drawn for performance comparison to establish the models' performance. While XGBoost had a very high performance, the study is constrained by the limited sample size, which may affect generalizability. The complexity of the boosting algorithms, i.e., XGBoost and CatBoost, makes it more prone to overfitting, which must be mitigated through proper tuning of hyperparameters. Also, the information does not provide the key survival attributes, i.e., health and survival details of the passengers, due to which prediction is not so accurate. Even though accuracy and AUC measures ensure the reliability of models, crossvalidation techniques could further support it. According to the research study, one can conclude that this dataset should be extended, and other classifiers, i.e., DL models, may provide better results. Ensemble approaches and the inclusion of more passenger attributes will, in future studies, further improve the predictions. While XGBoost turned out to be the best, accuracy versus interpretability remains a trade-off. It has been made evident through results how feature selection and preprocessing of data are really important in the case of classification problems. In general, this research shows how ML can be applied for the analysis of historic events and predictive modeling for other datasets different from those on the Titanic.

In paper by Anasuya Dasgupta et al. [8] investigated the application of machine learning to predict survival outcomes following the Titanic accident. The study focuses on the historical background of social stratification throughout this time period, emphasizing how gender, age, and socioeconomic status all had a substantial impact on survival. The authors use exploratory data analytics (EDA) and rigorous feature engineering to improve forecast accuracy by examining critical characteristics such as age, gender, ticket class, and family size. By comparing multiple

machine learning models, including Decision Trees and Random Forest, they show that the Random Forest model outperformed the other models with 80.41% accuracy, demonstrating the usefulness of ensemble approaches in complicated datasets. The study highlights the significance of data preprocessing and feature selection and recommends further investigation into similar historical events to enhance the findings' applicability. Overall, it effectively combines historical context with modern data science techniques to provide insights into survival dynamics during disasters.

In paper by Haque et al. [9] present the analysis of data related to passengers who travelled on the ship Titanic, identifying the features through machine learning that decided the survival rate of the passengers. Feature engineering is accomplished by converting the categorical values into numerical, calculating family size, and extracting the title from name and deck from ticket number. Classification by the Decision Tree algorithms classifies passengers into groups either survived or not survived. Data clustering is done in the KMeans algorithm implementation, too. To implement the Decision Tree used within the study at hand, for clustering, programming in R programming and Python had been used. First, the data is downloaded from Kaggle. Then, this dataset is preprocessed and divided into training and test sets. Feature engineering is done to enrich the data by filling in the missing values and categorizing the titles and surnames for studying the family structures. Decision tree classification shows the various levels of survival probability with respect to parameters like age, passenger class, and size of family. It seems that passengers who traveled in small families of 2-4 members had a better survival rate than single passengers or those who traveled in large families. The accuracy of the Decision Tree model is 85.52%. This paper concludes that machine learning models efficiently predict the survival rate of the people in the Titanic according to their feature characteristics. Improvements include the refinement of feature engineering techniques and applications of other algorithms for classification and clustering. The paper lastly suggests an extension of such methodologies to any other real-time data, say, natural calamity or pandemic data, to conduct further analyses and inferences.

In the research by Anshika Gupta et al. [10] compared several machine learning algorithms and found that Random Forest achieved an accuracy of 82%, outstanding Logistic Regression and Decision Tree models, which had accuracies of 78% and 79%, respectively. Their findings highlighted the importance of data cleaning and feature engineering in improving forecast accuracy. In contrast, Nair's research concentrated on four models—Naive Bayes, Logistic Regression, Decision Tree, and Random Forest—and demonstrated LR's highest accuracy of 93.54%. Both studies noted that feature selection and preprocessing had a huge impact on survival estimates. Despite their insights, major constraints include missing data and inherent biases in feature selection. The studies underscore the need for future research to explore additional attributes and advanced models, reminding us of the importance of predictive analytics in understanding historical events like the Titanic disaster. This body of work highlights not only the capabilities of ML techniques but also the critical insights they can provide for contemporary safety protocols.

The paper by Wenqing Liang [11] studies the most important features of passengers on the steamship Titanic. The study states that characteristics such as the passengers' age, sex, and ticket class were the most significant determinants of their survival. The study aims to accurately predict the survival rate of travelers on the Titanic using a dataset from the Kaggle website comprising 891 rows and 12 columns, each containing various features of the passengers like the name, age, gender, ticket class, and other pieces of information. The paper used the Random Forest algorithm and the Decision Tree algorithm to predict the survival rate of passengers due to the ability of these machine learning algorithms to predict with high accuracy and compatibility with other decision-making techniques. The study detected that the decision tree algorithm performed better compared with the random forest algorithm in predicting the survival rate of titanic passengers, with accuracy scores of 0.761 and 0.759 respectively.

A study by Yufan Ai [12] examines key features and characteristics that influence survival rates of passengers on the steamship Titanic, features as gender, age, and social class, through exploratory data analysis and performance evaluation of the models. The study used two datasets from the Kaggle website to train and test the model. It utilized a variety of machine-learning Methods for predicting the survival rates of passengers. The models include the k-nearest neighbor algorithm, support vector machine (SVM), binary logistic regression, and artificial neural network (ANN). The study achieved a prediction accuracy of 82.82% using the (SVM) model. The (ANN) also obtained a prediction accuracy of 82.16%. The research emphasized the importance of demographic data, such as age and social class, in predicting survival outcomes.

A study by Wang [13] was constructed on a Kaggle dataset to evaluate Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) for predicting Titanic survival. The results showed that RF with (depth 45 & 75) has reached the highest precision (0.919) and recall (0.872) which makes it the best performing model among others. On the other hand, DT (depth 5) achieved precision (0.905) and recall (0.812). Lastly, LR had the lowest precision (0.882) and recall (0.829). In addition, Old researches align with these results. Lam and Tang (2012) confirmed that class and gender were the strongest predictors with DT, Support Vector Machine (SVM), and Naïve Bayes. Additionally, Shawn et al. (2014) has found that the most significant factor was the gender, and it was proven using DT and clustering. Also, on (2017) Chatterjee compared DT, RF, and LR and each got accuracy of 84%, 81%, and 80.76% respectively, on the same year Singh et al. (2017) mentioned the significant impact of age, gender, and class on the accuracy of the model.

Lastly, Wang’s research reinforces that RF with (depth 45 & 75) has outperformed the other models because of its ensemble learning capabilities. Hyperparameter tuning and feature engineering had a huge role in improving the classification performance which was highlighted by

Wang's research. Hybrid models and deep learning techniques could be explored by future research to enhance survival prediction.

A research by Bisht, M. et al. [14] tackles the Titanic dataset, which contains information about 891 people, with the aim of predicting which passengers survived based on features such as passenger class (Pclass), gender, age, and family structure (e.g., if they had dependents). Study uses multiple machine learning approaches, Like Naive Bayes, Linear Support Vector Machines (SVM), Logistic Regression, Random Forest, and CatBoost. Logistic regression is highlighted for being the most accurate in minimizing false predictions and CatBoost is noted to have the best accuracy overall, achieving the highest score. The approach to this study includes extensive data cleansing and processing, such as Age imputation and Cabin column removal, followed by Feature selection where Age, Passenger class, Sex, Name, Embarked and Fare are used, and evaluates model performance through confusion matrices that track accuracy, true positives, and false negatives. However, the study has its flaws, such as the risks of biases due to the management of absent information, limitations imposed by a restricted feature set which are likely to miss other important predictors, and a higher risk of overfitting due to the limited dataset which hurts the models' generalizability. Additionally, the study reveals gaps within the literature which reveal the need for a more systematic study of different machine learning algorithms on the Titanic survival prediction dataset, a more detailed study on feature engineering, an assessment of the biases that occur during data transformation, as well as supplementary studies to check the results on other datasets and actual situations.

2.1 Gap Identification

We noted that numerous studies predicting Titanic survival using machine learning encounter specific limitations, which will be touched upon but not explored in depth here.

Papers [5], [9] share several limitations: applying feature engineering without analyzing its impact, neglecting deep learning (CNNs, RNNs), dataset bias handling, model explainability (SHAP, LIME), hyperparameter optimization, and hybrid models, and lacking generalization testing. Papers [4], [6], [11], [12], [14] and [2] share a reliance on basic machine learning algorithms (Logistic Regression, SVM, Random Forest, Decision Tree) instead of advanced techniques (neural networks, deep learning).

Paper [6] and [7] suffers from minimal feature engineering, insufficient crossvalidation, basic missing value imputation, and a lack of exploration of deep learning or ensemble methods, also overlooking data distribution bias and model interpretability. Paper [1] omits potentially valuable features (health, social status), excludes the "Cabin" feature instead of addressing missing values, primarily uses simple algorithms (Logistic Regression, Decision Trees) instead of advanced techniques (DL, SVM), doesn't investigate interaction effects between variables, and uses a small dataset without cross-validation, limiting robustness. Paper [10] lacks detail on data preprocessing (missing value handling), hyperparameter tuning, feature selection/engineering impact, and

advanced ensemble methods beyond Random Forest, also omitting in-depth result analysis (limitations, overfitting) and model interpretability.

Paper [8] lacks a detailed explanation of hyperparameter tuning, comparison of model efficiency/scalability, discussion of dataset size limitations, in-depth feature importance analysis, exploration of dataset biases, and consideration of ethical implications/modern applications. Specifically, paper [13] also didn't analyze the impact of hyperparameter tuning beyond tree depth. Papers [4], [14] and [2] could use principal component analysis (PCA) to enhance feature engineering.

Papers [11] and [12] could benefit from a deeper analysis of other passenger features (siblings/parents, socioeconomic status, behavior), and paper [12] didn't discuss the implications of missing data handling on predictive accuracy. Paper [3] relied primarily on basic feature engineering, using only class, age, and gender, without even considering such features as health condition, social status, or experience in survival, which could have predicted the outcome better.

Finally, papers [14] and [2] suggest assessing data transformation biases and conducting supplementary studies on other datasets/real-world situations.

Closing these gaps by supplementing feature engineering, introducing deep learning, improving missing data handling, employing cross-validation, and mitigating dataset bias would significantly boost the fairness, accuracy, cross-validation, and access to a larger set of data and generalizability of survival prediction models so that the results would be more usable beyond the Titanic dataset.

Table 2 Data Extraction for Gap Analysis

Ref.	Title	Author/s	Year	Dataset	ML	Result	Notes
[1]	Machine Learning-Based Analysis of Titanic Passenger Survival	Dr.Prabha Shreeraj Nair	2017	Data are from Kaggle, 891 training samples, and 418 test samples with attributes in each sample consisting of age, sex, passenger class, fare paid, and family relations.	XGBoost, CatBoost, Random Forest, and Decision Trees	**XGBoost** achieved the highest accuracy (91%) and **AUC** (0.89), with passenger class, sex, and age as key survival factors.	-
[2]	Analyzing Titanic Disaster using Machine Learning Algorithms	Aakriti Singh; Shipra Saraswat; Neetu Faujdar	2017	Dataset used Kaggle website. The data consists of 891 rows	Logistic Regression, Naive Bayes, Decision Tree, Random Forest	**(Algorithms - Accuracy - False discovery Rate)** Nave Bayes - 91.3% - 15.47% Logistic Regression - 94.26% - 7.90% Decision Tree - 93.06% - 9.26% Random Forest - 91.86% - 10.66%	• Comparing the four techniques used in this research work two metrics are used. First metric is accuracy, and the second metric is false discovery. • Future work might include potentially validating more using pruning techniques that is to see if a

							shallower tree with same or improved accuracy can be achieved.
[3]	Mining bookstore and Titanic data by Weka for understanding promotional strategy and predicting survival pattern	HOSSAIN Khan Rajib, Abedi Sohrforouzani Mana, Darvishi, Shahrzad, Claire Ukwishaka Marie	2018	Titanic Kaggle Dataset	J48 Decision Tree, Simple K-Means Clustering	**Titanic Dataset (J48 Classifier):** ▪ J48 classifier had moderate accuracy (almost 80%), with a higher misclassification rate for survivors. **Titanic Dataset (Simple K-Means Clustering):** ▪ Cluster 1 (mostly women) had a higher survival rate. Cluster 0 (mostly men) had a lower survival rate.	Enhance feature engineering (include health, family relations, socio-economic indicators).
[4]	Predicting Survival on Titanic by Applying Exploratory Data Analytics and Machine Learning Techniques	Yogesh Kakde, Shefali Agrawal	2018	The dataset wasn't mentioned explicitly	Logistic Regression, Decision Tree (DT), Random Forest (RF) , Support Vector Machine (SVM)	**(Algorithm - Accuracy)** Logistic Regression (Best performing model) - 83.7% Support Vector Machine (SVM) - 83.1% Random Forest - 82.6% Decision Tree (Worst performing model) - 81.7%	• Feature Engineering played an important role in improving model performance. • Future research is needed to improve the models.

[5]	A Comparative Study on Machine Learning Techniques using Titanic Dataset	Ekin Ekinci, Sevinç İlhan Omurca, Neytullah Acun	2018	Titanic Kaggle Dataset	Logistic Regression, kNN, Naïve Bayes, SVM, Decision Tree, Bagging, AdaBoost, Extra Trees, Random Forest, Gradient Boosting, ANN, Voting	**Voting (GB, ANN, kNN)** achieved the highest F-measure score of 0.82	Feature engineering techniques applied to improve classification performance
[6]	Prediction of Survivors in the Titanic Cruise	Rajesh M.	2019	Titanic Kaggle Dataset	Random Forest, SVM, Gradient Boosting (GBM)	**Support Vector Machine (SVM)**: Accuracy **83.9%,** Error **16.1%** **Random Forest (RF)**: Accuracy **82.5%,** Error **17.5%** **Gradient Boosting Machine (GBM)**: Accuracy **77.6%,** Error **22.4%** SVM achieved the highest accuracy and the lowest error rate.	-
[7]	Analyzing of Titanic Disaster using Machine Learning Algorithms	Abdullahi Adinoyi Ibrahim,IAENG and Rabiat Ohunene Abdulaziz	2020	from Kaggle, has 891 training instances and 471 test instances with 12 features including age, gender, ticket class, and fare.	Decision Trees, Random Forest, XGBoost (Extreme Gradient Boosting), CatBoost (Categorical Boosting)	The highest-performing algorithm in Titanic survivor prediction in this research was **XGBoost**, which performed better than other models and achieved the highest accuracy of 91%	-
[8]	Predicting the Likelihood of Survival of	Anasuya Dasgupta, Ved Prakash Mishra,	2021	Titanic passenger data set of	Logistic Regression, Decision	**The Decision Tree Classifier** had the	• The study explored

	Titanic's Passengers by Machine Learning	Sanjiv Jha, Bhopendra Singh, and Vinod Kumar Shukla		891 rows and 15 columns.	Tree Classifier, Random Forest Classifier.	most accurate result of 99.29%. **Logistic Regression Model** with an accuracy of only 81.11% on the testing data. **Random Forest Classifier** precision of a whopping 97.53% and 80.41% on the training data and testing data respectively.	how factors like age, sex, class, and family size influenced survival. • Feature engineering was considered crucial for model performance. • Cross-validation was likely used during the Random Forest model evaluation. • The authors tested their own survival chances using the model.
[9]	Passenger data analysis of Titanic using machine learning approach in the context of chances of surviving the disaster	Md Arfinul Haque	2021	Kaggle Titanic Dataset	Decision Tree, K-Means Clustering	**Decision Tree** achieved an accuracy of 85.52%	Feature engineering includes extracting title, family size, and deck label
[10]	Exploratory Data Analysis of Titanic Survival Prediction using Machine Learning Techniques	Anshika Gupta, Deepak Arora, Shivam Tiwari	2023	Titanic dataset obtained from Kaggle	Logistic Regression, Random Forest, Stochastic Gradient Descent, Decision Tree, and	**The Random Forest** algorithm performed the best with an accuracy of 82%, an F1-score of 0.82, recall of 0.81, and precision of 0.82. **Logistic Regression and**	The study emphasizes the importance of data preprocessing, feature engineering, and the use of various

					K-nearest neighbor.	**Decision Tree** algorithms also performed well with accuracies of 78% and 79%, respectively. **Stochastic Gradient Descent** showed poor performance with an accuracy of 58%, while **K-nearest neighbor** performed moderately with an accuracy of 66%.	evaluation metrics such as accuracy, F1-score, recall, and precision to compare the performance of different machine learning models.
[11]	Titanic Disaster Prediction Based on Machine Learning Algorithms	Wenqing Liang	2023	The dataset from Kaggle includes various features such as passenger survival status, class (Pclass), name, sex, age and many other information.	The Random Forest and Decision Tree algorithms.	**Decision Tree** algorithm outperformed the **Random Forest** algorithm in predicting the survival rate of Titanic passengers, with accuracy scores of 0.761 and 0.759 respectively	-
[12]	Predicting Titanic Survivors by Using Machine Learning	Yufan Ai	2023	The dataset from the Kaggle website and consists of 891 rows and 12 columns.	Random Forest, K-nearest neighbor algorithm, Vector Machine (SVM)	The study achieved a prediction accuracy of 82.82% on Kaggle using the **SVM model**	-
[13]	Survival Prediction and Comparison of the Titanic based on Machine Learning Classifiers	Tony Wayne Wang	2024	Kaggle Titanic Dataset	Logistic Regression (LR), Decision Tree (DT), Random Forest (RF)	**Logistic Regression (LR):** Achieved cross-validation score: 0.789 **Decision Tree (DT) with the Best performance (depth 5):** Cross-validation score: 0.7924	-

						Decision Tree (DT) with (depth 45 & 75): Cross-validation score: 0.7724 **Random Forest (RF) with the Best performance (depth 45 & 75):** Cross-validation score: 0.798 **Random Forest (RF) with (depth 5):** Cross-validation score: 0.8134	
[14]	Analysis of Machine Learning Algorithms for Predicting Titanic Disaster Survival Rate	Manas Bisht, Akash Singh, Gautam Tripathi, Kumar Shantanu, Amit Gupta, Richa Gupta	2024	The dataset from the Kaggle website and consists of 891 rows and 12 columns, containing various features.	Naive Bayes, Linear Support Vector Machines (SVM), Logistic Regression, Random Forest, CatBoost	**(Algorithm - Accuracy)** CatBoost - 78.42% Logistic Regression - 77.45 % Linear SVM - 76.74% Random Forest - 75.54% Naïve Bayes - 74.10%	CatBoost has a somewhat greater prediction capability than other algorithms to predict survival rate accuracy. Future work might consider using more tree-based ensemble algorithms such as XGBoost and Gradient Boost algorithm.

3.0 Project Deliverables of the team

Table 3 Project Deliverables of the team

Deliverable	To whom	Delivery Media	Duration	Date
Literature Review (Homework-1)	Dr. Rabab Alkhalifah	Softcopy	2 weeks	23/2/2025
Project Proposal	Dr.Rabab Alkhalifah	Softcopy	6 days	2/3/2025
Project Proposal Presentation	Dr. Rabab Alkhalifah	Softcopy	6 days	2/3/2025
Description of Selected ML Algorithms	Dr. Rabab Alkhalifah	Softcopy	3 days	25/3/2025

Final Project Report	Dr. Rabab Alkhalifah	Softcopy	12 days	20/4/2025
Final Project Presentation	Dr. Rabab Alkhalifah	Softcopy		

4.0 Description of the Proposed Techniques

4.1 Random Forest

Random forest is a supervised classifier algorithm in machine learning that generates several decision trees to reach a prediction. The algorithm is flexible as it can be used for both predictive modeling in regression and categorization in classification tasks. The decision trees generated within the algorithm are used to train separate parts of the dataset, with different features in each tree through a process known as feature bagging, which increases model diversity. The final prediction is determined by integrating the results from decision trees, as shown in Figure 1. For the classification tasks, the final prediction is determined by taking most of the predictions made by the decision trees (Majority voting). In contrast, in regression tasks, we reach the final prediction by calculating the average of the results predicted by the trees [15]. The random forest method has three important hyperparameters to be determined before the training step. The number of trees to be generated, the size of the node, and the number of features. Then the algorithm can be applied to solve the task [16].The method also can determine feature importance by using a few ways including the Gini importance, and mean decrease in impurity (MDI) shown in Figure 2. Random forest algorithm is a highly effective and accurate algorithm, as it can deal with high-dimensional datasets easily. Also, the randomness in each decision tree helps to prevent overfitting, it also increases the training process speed because

eachdecision tree can be trained independently and simultaneously [17].

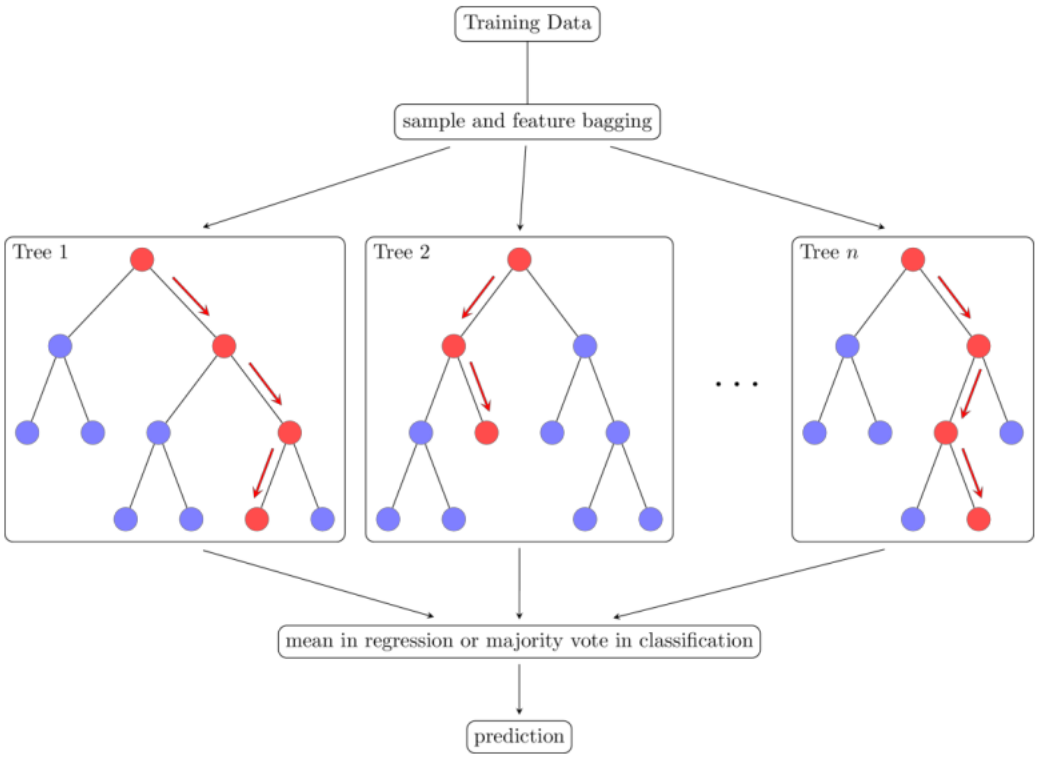

Figure 1 Random Forest Visualization

```
from sklearn.ensemble import RandomForestClassifier

# Assume X_train and y_train are the training features and labels respectively
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Calculate feature importances using MDI
importances = model.feature_importances_
```

Figure 2 Implementing MDI in Python

Random Forest is a great choice for solving the Titanic classification problem as it can handle missing data, particularly in the age column, determine nonlinear relationships between features like Age, Sex which will impact the survival probability, and it can handle both numerical and categorical data. Finally, the Random Forest showed high accuracy, indicating its potential to enhance survival predictions in disaster rescue operations. Figure 3 shows the implementation of Random Forest in Python.

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

Figure 3Implementing Random Forest with RandomForestClassifier in Python

4.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised learning technique that is extensively used in classification issues, even more in binary classification. SVM operates by identifying an optimal hyperplane that maximally separates data points into various classes with the maximum margin between the nearest data points of each class as it shown in Figure 4. Thus, it is highly pertinent from this point onwards to classify the survival rates of Titanic passengers based on different attributes, such as age, sex, ticket class, and fare.

- SVM can use a range of kernel functions to enhance its capability to classify, particularly for data that is not linearly separable:

  a) Linear Kernel: Applied in cases where the data can be separated linearly

  b) Polynomial Kernel: Transforms data into a higher-dimensional polynomial space to increase separability.

  c) Radial Basis Function (RBF) Kernel :allows the mapping of data to a higher space using a Gaussian function and is thus best for complex and nonlinear data relationships.

- The performance of SVM is greatly dependent on hyperparameter optimization as it shown in Figure 5. The two most critical hyperparameters are[18]:

  1) Regularization Parameter (C): Controls the trade-off between achieving a low error rate and allowing misclassification. A higher value for C ensures lower bias but may result in overfitting.

  2) Kernel Coefficient (Gamma γ): Defines the influence of a single training example, on the smoothness of the decision boundary.

```
from sklearn.svm import SVC
svm = SVC(kernel='rbf', C=1, gamma='scale')
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
```

Figure 4 Implementing SVM with SVC in Python

4.2.1 Feature Importance in SVM

As compared to decision-tree-based models like Random Forest, SVM does not natively provide feature importance rankings. Feature importance can be inferred from the weight vectors of the model, which reflect the extent to which each feature is used for classification. Furthermore, model-agnostic explanation methods such as SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model-agnostic Explanations) can be used in SVM decision explanation, which can make the model more interpretable in Titanic survival prediction Recent studies have confirmed that a fusion of Support Vector Machines (SVM) and feature selection techniques, i.e., Principal Component Analysis (PCA), can play a significant role in improving model accuracy by reducing noise and dimensionality as it shown in Figure 5. Furthermore, ensembleb techniques that combine SVM with other classification models, i.e., Random Forest and XGBoost, have also been found to have high potential in improving classification performancein survival prediction problems[19].

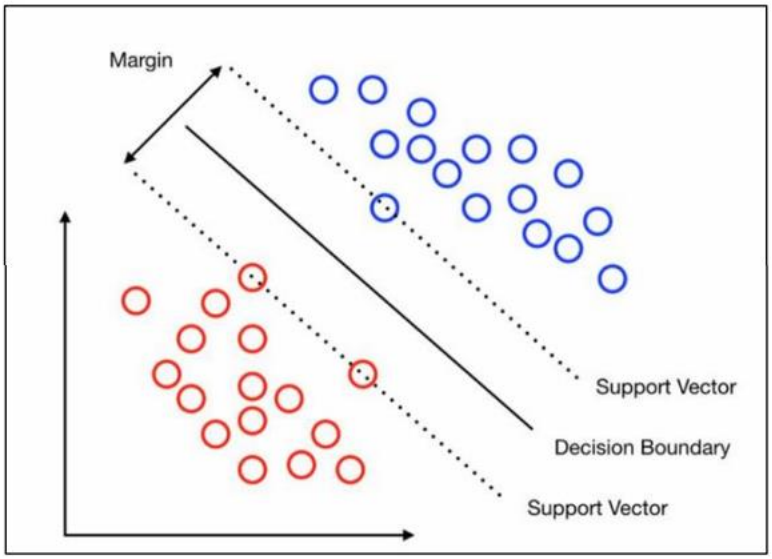

Figure 5 Conceptual SVM Model Visualization

5.0 Empirical Studies

5.1 Description of dataset

The Titanic dataset from Kaggle's "Titanic: Machine Learning from Disaster" . The dataset is split for the purpose of model training and testing, where the goal is to build a predictive model using the training data and evaluate its performance on the test set.The features in the dataset provide a mix of numerical and categorical data, such as passenger class (Pclass), gender (Sex), age (Age), number of siblings/spouses aboard (SibSp), and fare paid (Fare) , number of parents / children aboard the Titanic (parch). Other features include the passenger's name, ticket number, cabin, and embarkation port (Embarked). Some features have missing values, especially Age, Cabin, and Embarked, we will handling during preprocessing.

The training set contains 891 instances (passengers) with 12 columns, including the target variable Survived, which indicates whether a passenger survived (1) or not (0). The test set includes 418 instances and 11 features, excluding the target variable. The dataset also provides a sample submission file (gender_submission.csv a set of predictions that assume all and only female passengers survive.

And can download from hare [1].

5.1.1 Statistical Analysis of the Dataset

The following table (4) summarizes the basic statistics of the dataset after preprocessing, and table (5) show Correlation between each Attribute and the Target attribute .

Table 4 Statistical Analysis of the dataset

Feature	Mean	Median	Std. Dev	Max	Min
Pclass	2.308	3	0.836	3	1
Age	29.70	28	14.53	80	0.42
SibSp	0.523	0	1.10	8	0
Parch	0.382	0	0.806	6	0
Fare	32.20	14.45	49.69	512.33	0.00

Table 5 Correlation between each Attribute and the Target attribute

Feature	Correlation with Survived
Fare	**+0.257**
Parch	+0.082
Age	−0.077
SibSp	−0.035
Pclass	**−0.338**

5.2 Experimental setup

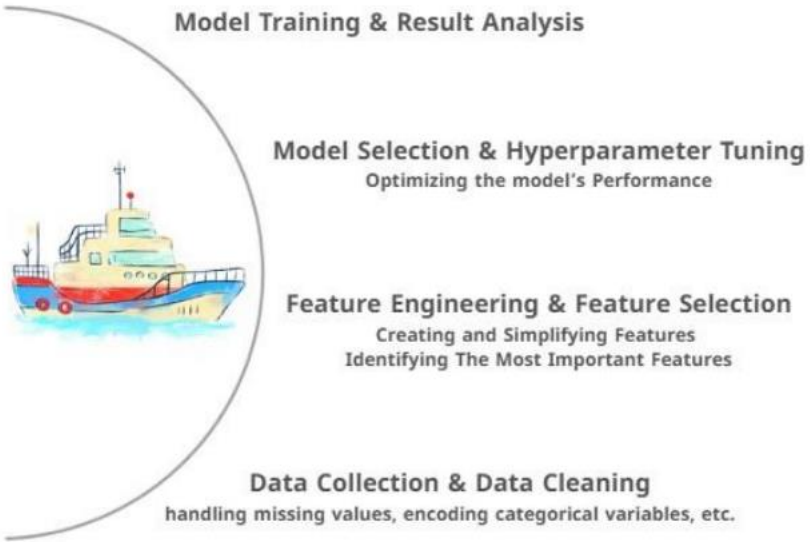

Figure 6 Machine Learning Model Development Steps

This section outlines our model development steps as shown in Figure 6. First, the dataset obtained from the Kaggle website was cleaned by handling the missing values, encoding the categorical features, and dropping irrelevant ones. Both feature engineering and feature selection were performed to enhance the quality of the features and to select and keep only relevant features in our dataset. Also, for our prediction project, we used many machine learning algorithms, including Support Vector Machine, Random Forest, Gradient Boosting, XGBoost, Logistic Regression, and Neural Network. Additionally, we used ensemble learning methods, which add multiple models, like the Stacking and Voting classifiers. Moreover, our models were tuned to improve their performance, then trained on the training set and tested on the testing set to determine their accuracy. Our dataset was split into a training and a testing set. 80% of the dataset represents the training set, while the rest 20% of the data represents the testing set. Also, the stratified sampling was used to ensure that the balance between both of the classes is achieved. During the tuning stage, the training set is further divided into 5 folds to evaluate model performance more reliably.

```
# Family size & Alone
df_processed['FamilySize'] = df_processed['SibSp'] + df_processed['Parch'] + 1
df_processed['IsAlone'] = (df_processed['FamilySize'] == 1).astype(int)

# Extract and simplify titles
df_processed['Title'] = df_processed['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
title_map = {
    "Mr": "Mr", "Miss": "Miss", "Mrs": "Mrs", "Master": "Master",
    "Mlle": "Miss", "Ms": "Miss", "Mme": "Mrs",
    "Dr": "Rare", "Rev": "Rare", "Col": "Rare", "Major": "Rare",
    "Countess": "Rare", "Lady": "Rare", "Jonkheer": "Rare",
    "Don": "Rare", "Dona": "Rare", "Capt": "Rare", "Sir": "Rare"
}
df_processed['Title'] = df_processed['Title'].map(lambda x: title_map.get(x, "Rare"))

# Fill missing values
df_processed['Age'] = df_processed.groupby('Title')['Age'].transform(lambda x: x.fillna(x.median()))
df_processed['Fare'] = df_processed.groupby('Pclass')['Fare'].transform(lambda x: x.fillna(x.median()))
df_processed['Embarked'] = df_processed['Embarked'].fillna(df_processed['Embarked'].mode()[0])
```

Figure 7 Handling the Dataset

```
# Encode categorical features
le = LabelEncoder()
df_processed['Sex'] = le.fit_transform(df_processed['Sex'])
df_processed['Embarked'] = le.fit_transform(df_processed['Embarked'])

# Interaction & composite features
df_processed['Sex_Pclass'] = df_processed['Sex'] * df_processed['Pclass']
df_processed['Age_Pclass'] = df_processed['Age'] * df_processed['Pclass']
df_processed['Fare_Per_Person'] = df_processed['Fare'] / df_processed['FamilySize']
df_processed['Age_Class_Fare'] = df_processed['Age'] * df_processed['Pclass'] * df_processed['Fare']
df_processed['Fare_log'] = np.log1p(df_processed['Fare'])
df_processed['Age_log'] = np.log1p(df_processed['Age'])
df_processed['Family_Survival'] = df_processed['FamilySize'] * df_processed['Survived'].mean()

# Cabin features
df_processed['Has_Cabin'] = df_processed['Cabin'].notnull().astype(int)
df_processed['Cabin_Letter'] = df_processed['Cabin'].str[0].fillna('U')

# One-hot encode titles, pclass, cabin letters
df_processed = pd.get_dummies(df_processed, columns=['Title', 'Pclass', 'Cabin_Letter'], drop_first=True)

# Drop unused columns
df_processed.drop(columns=['Name', 'Ticket', 'Cabin', 'PassengerId'], inplace=True)
```

Figure 8 Some of the Feature Engineering Steps

Both Figure 7 and Figure 8 show the steps we did to handle the missing values, encode the categorical features, and perform feature engineering steps. These feature engineering processes consist of creating and transforming features, creating interaction features, generating family size columns, and extracting and mapping titles. All these steps were performed to improve our model's predictive ability.

```python
# Hyperparameter tuning for Gradient Boosting
gb_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.7, 0.8, 1.0]
}
gb = GradientBoostingClassifier(random_state=42)
gb_search = RandomizedSearchCV(gb, gb_params, n_iter=15, cv=5, scoring='accuracy', random_state=42, n_jobs=-1)
gb_search.fit(X_train, y_train)
best_gb = gb_search.best_estimator_
print("Best Gradient Boosting params:", gb_search.best_params_)

# Hyperparameter tuning for SVM
svm_params = {
    'C': [0.1, 1, 10],
    'kernel': ['rbf', 'linear'],
    'gamma': ['scale', 'auto']
}
svm = SVC(probability=True, random_state=42)
svm_search = RandomizedSearchCV(svm, svm_params, n_iter=10, cv=5, scoring='accuracy', random_state=42, n_jobs=-1)
svm_search.fit(X_train, y_train)
best_svm = svm_search.best_estimator_
print("Best SVM params:", svm_search.best_params_)

# Hyperparameter tuning for XGBoost
xgb_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.7, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0]
}
xgb = XGBClassifier(random_state=42)
xgb_search = RandomizedSearchCV(xgb, xgb_params, n_iter=15, cv=5, scoring='accuracy', random_state=42, n_jobs=-1)
xgb_search.fit(X_train, y_train)
best_xgb = xgb_search.best_estimator_
print("Best XGBoost params:", xgb_search.best_params_)

# Hyperparameter tuning for Logistic Regression
lr_params = {
    'C': [0.1, 1, 10],
    'penalty': ['l2'],
    'solver': ['liblinear', 'saga']
}
lr = LogisticRegression(random_state=42)
lr_search = RandomizedSearchCV(lr, lr_params, n_iter=10, cv=5, scoring='accuracy', random_state=42, n_jobs=-1)
lr_search.fit(X_train, y_train)
best_lr = lr_search.best_estimator_
print("Best Logistic Regression params:", lr_search.best_params_)

# Hyperparameter tuning for MLP
mlp_params = {
    'hidden_layer_sizes': [(100,), (100, 50), (150,)],
    'activation': ['relu', 'tanh'],
    'solver': ['adam', 'sgd'],
    'alpha': [0.0001, 0.001],
    'learning_rate': ['constant', 'adaptive']
}
mlp = MLPClassifier(random_state=42)
mlp_search = RandomizedSearchCV(mlp, mlp_params, n_iter=10, cv=5, scoring='accuracy', random_state=42, n_jobs=-1)
mlp_search.fit(X_train, y_train)
best_mlp = mlp_search.best_estimator_
print("Best MLP params:", mlp_search.best_params_)
```

Figure 9 Hyperparameter Tuning for All Models

```python
# Voting Classifier
voting_clf = VotingClassifier(
    estimators=[('rf', rf), ('gb', gb), ('svm', svm), ('xgb', xgb), ('lr', lr)],
    voting='soft'
)

# Stacking Classifier
stacking_clf = StackingClassifier(
    estimators=[('rf', rf), ('gb', gb), ('svm', svm), ('xgb', xgb), ('mlp', mlp)],
    final_estimator=LogisticRegression(random_state=42)
)
```

Figure 10 Building the Ensemble Classifiers

The steps shown in Figure 9 represent the hyperparameter tuning process to improve the quality and performance of our models. Moreover, Figure 10 shows our Ensemble Classifiers that combine individual models to multiply their strengths.

The correlation heatmap shown in Figure 11 represents the relationships between the numerical features in our Titanic dataset. This graph provides a clear view of which features are highly or minimally correlated with one another.

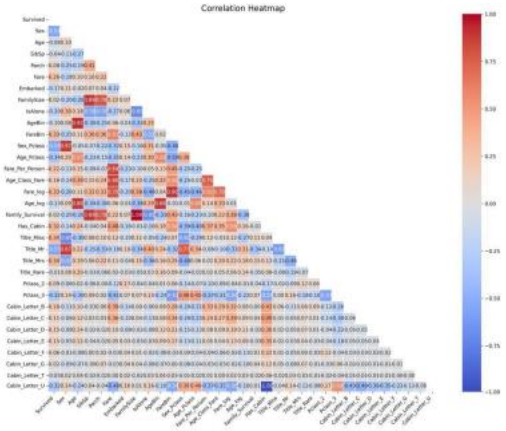

Figure 11 Correlation Heatmap Showing Relationships Between Features

The graph in Figure 12 below represents a correlation bar plot showing how strongly each feature in the dataset is linearly related to the target variable 'Survived'. It instantly shows which features are important predictors of survival.

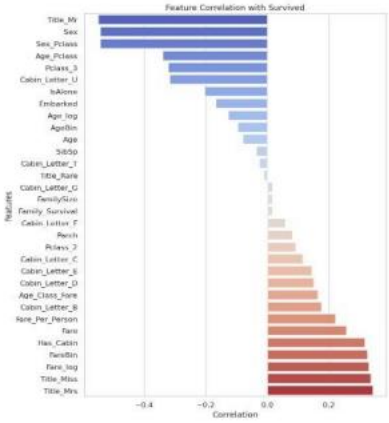

Figure 12 Correlation Bar Plot to Show the Relation Between Features and Target Variable

5.3 Performance Measures

To check the effectiveness of Random Forest and Support Vector Machine (SVM) in classification of survival patterns for passengers, certain crucial measures of performance and variables will be checked. The tests will ensure accuracy, interpretability, computational speed, and real-world effectiveness of the models for disaster readiness and emergency preparation.

5.3.1 Classification Performance Metrics

The accuracy of the classification models will be measured using the following key performance indicators:

a. Accuracy (Overall Model Performance)

- Accuracy is the ratio of correctly classified instances to the total number of instances.
- It provides a general overview of how well the model is performing but be sufficient if the dataset is imbalanced.

b. Precision, Recall, and F1-Score (Class-Specific Performance)

- Precision (Positive Predictive Value): Measures how many of the instances classified as "Survived" are actually correct.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

c. Recall (Sensitivity or True Positive Rate): Measures how well the model identifies all actual survivors.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

d. Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC)

- **ROC Curve:** Plots the true positive rate against the false positive rate at various classification thresholds.
- **AUC Score:** Measures the overall ability of the model to distinguish between survived and non-survived passengers, with a score close to 1 indicating a highly effective model.

5.3.2 Feature Importance and Interpretability

Both Random Forest and SVM allow for an analysis of feature importance to understand which factors play the most significant role in predicting survival.

5.3.2.1 Feature Importance in Random Forest

- Random Forest provides a built-in feature importance measure based onThe number of times a feature is used in decision splits across multiple trees.
- Important features may include:
  - Passenger class (1st, 2nd, or 3rd class).
  - Age (older passengers may have lower survival rates).
  - Gender (historically, women had higher survival rates).
  - Fare (higher fares might indicate privileged access to lifeboats).

5.3.2.2 Feature Influence in SVM

- In SVM, feature importance is derived from the weight coefficients of the support vectors. The kernel function (linear, polynomial, radial basis function - RBF) plays a crucial role in.
- identifying non-linear relationships in survival determinants.

5.4 Optimization strategy

We wish to determine the best and most precise model to predict survival. To that end, we used some Optimization strategys :

1- Advanced Feature Engineering :Creating new, pertinent features from the data given to allow the model to learn more effective patterns.

Examples:

- Names from names extraction (Mr., Mrs., Miss, etc.) can be correlated with gender, age, or social class
- FamilySize: SibSp + Parch + 1 enables the model to identify if a person is traveling alone or with family.
- Cabin Section: Utilize the first letter of the cabin to find on the ship.
- Fare Binning: Binning fare into bins reduces skew.
- Age Grouping: Convert continuous age to categories (child, adult, senior).

Why it improves accuracy:

Raw data hides patterns. New features reveal concealed information not exposed to models. Enables tree-based model relationships, introduces awareness in SVMs and MLPs of non-linear boundaries.

2- Feature Selection: choosing only the most helpful features to train the model.

Common techniques use it :

- **Correlation heatmaps**: Remove highly correlated features (multicollinearity).
- **Feature importance from RandomForest or XGBoost**.
- **Recursive Feature Elimination (RFE)**: Eliminate least important features recursively.
- **Variance Threshold**: Remove features with low or no variability.

Why it improves accuracy:

- Reduces noise and overfitting.
- Makes models easier to interpret, faster, and simpler.
- Allows models to focus on strong predictors better generalization

3- Hyperparameter Tuning : Optimizing the configuration settings for each model.

- Random Forest: n_estimators, max depth, min_samples_split, min_samples_leaf, max_features.
- SVM: Regularization (C), kernel type, gamma.
- Gradient Boosting/XGBoost: subsample, n_estimators, max_depth, learning_rate, colsample_bytree.
- Neural Network (MLP): solver, learning_rate, hidden_layer_sizes, alpha, activation.
- Logistic Regression: solver,penalty, C.
- Gradient Boosting: subsample, n_estimators, max_depth, learning_rate.

Why it improves accuracy:

- Default parameters are generic.
- Properly tuned models are less likely to underfit or overfit.
- You fine-tune them for your specific dataset, giving maximum performance

we tuned the hyperparameters for each model to improve their performance. Table 6 shows the Optimal parameters for the proposed Gradient Boosting model .Table (7) shows the best settings for Random Forest, like setting the maximum depth to 5 and using 100 trees. Table (8) shows the best values for the SVM model, like using a linear kernel and setting C to 10. Table (9) shows the best settings for XGBoost, such as a subsample of 0.8 and a learning rate of 0.01. Table (10) shows the optimized settings for Logistic Regression, like using the saga solver and penalty of 12. Finally, Table (11) shows the best parameters for the MLP model, like using the adam solver and relu activation. Finding the right values helps the models make better predictions and reduces mistakes.

Table 6 Optimal parameters for the proposed Gradient Boosting model

Parameters	Optimal value chosen
subsample	0.7
n_estimators	100
max_depth	5
learning_rate	0.01

Table 7 Optimal parameters for the proposed RF model

Parameters	Optimal value chosen
max_depth	5
n_estimators	100
min_samples_split	5
min_samples_leaf	1
max_features	log2

Table 8 Optimal parameters for the proposed SVM model

Parameters	Optimal alue chosen
C	10
kernel	linear
gamma	scale

Table 9 Optimal parameters for the proposed XGBoost model

Parameters	Optimal Value chosen
subsample	0.8
n_estimators	100
max_depth	3
learning_rate	0.01
colsample_bytree	1

Table 10 Optimal parameters for the proposed Logistic Regression model

Parameters	Optimal Value chosen
solver	saga
penalty	12
C	1

Table 11 Optimal parameters for the proposed NN model

Parameters	Optimal Value chosen
solver	adam
learning_rate	constant
hidden_layer_sizes	(100, 50)
alpha	0.0001
activation	relu

Graph 13 displays accuracy vs hyperparameter in Random Forest model. Accuracy starts around 0.883 and rises to about 0.932 .

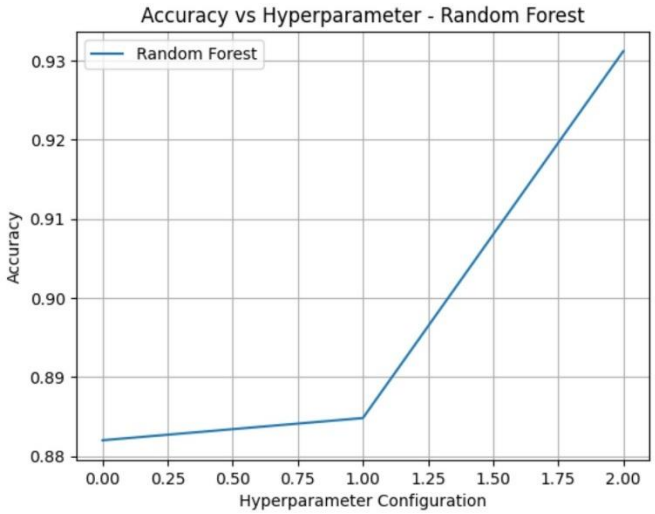

Figure 13 Accuracy vs hyperparameter – Random Forest

Graph 14 displays accuracy vs hyperparameter in SVM model. Accuracy among all hyperparameter is 0.82 .

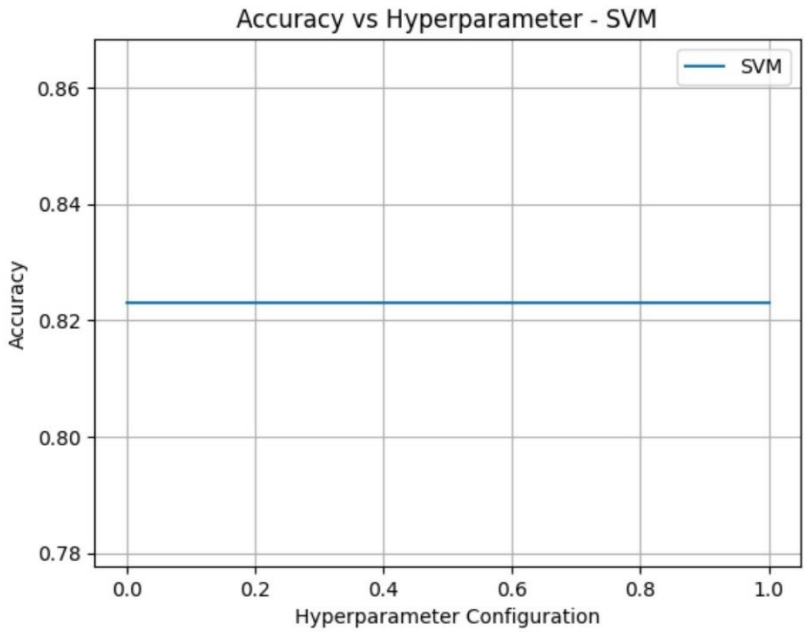

Figure 14 Accuracy vs hyperparameter – SVM

Graph 15 displays accuracy vs hyperparameter in XGBoost model. Accuracy starts around 0.883 and rises to about 0.90 .

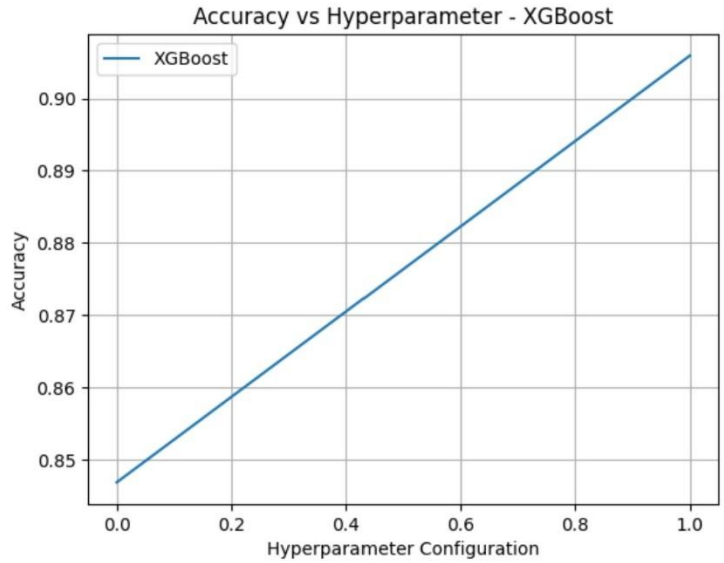

Figure 15 Accuracy vs hyperparameter –XGBoost

Graph 16 displays accuracy vs hyperparameter in Neural Network model. Accuracy among all hyperparameter is 0.86 .

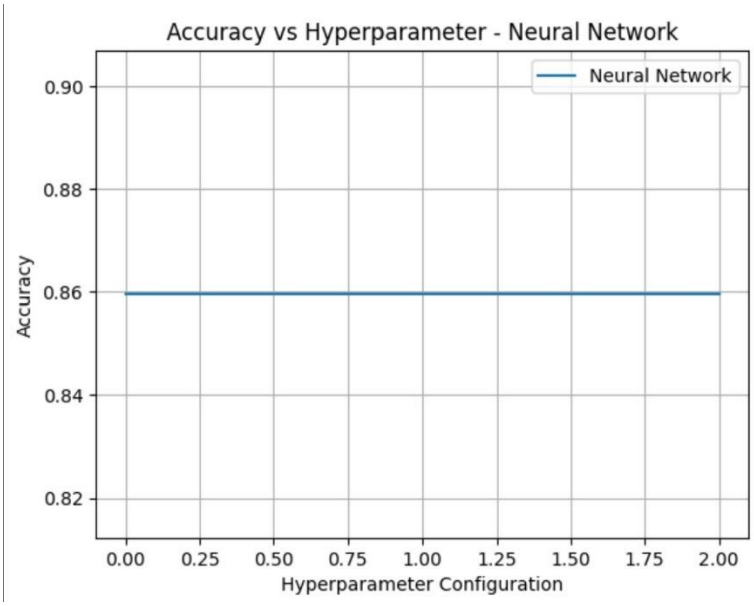

Figure 16 Accuracy vs hyperparameter – Neural Network

Graph 17 displays accuracy vs hyperparameter in Gradient Boosting model. achieves the highest accuracy among the three models, reaching up to 0.98.

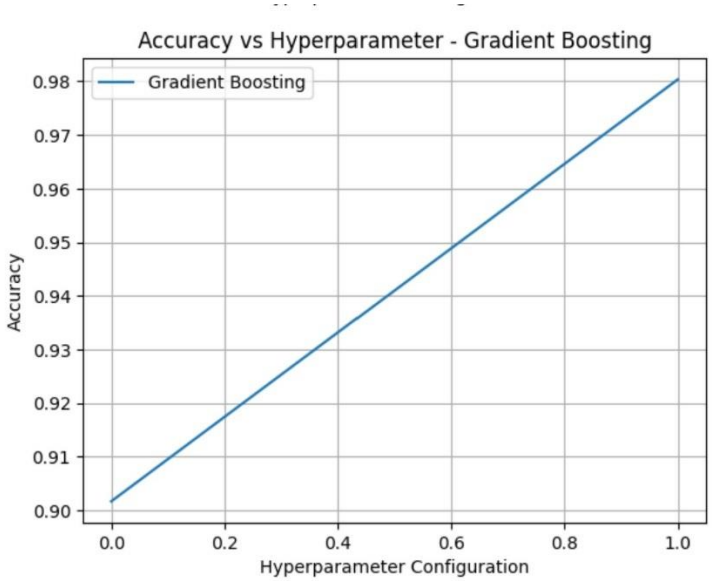

Figure 17 Accuracy vs hyperparameter – Gradient Boosting

Graph 18 displays accuracy vs hyperparameter in Logistic Regression model. Accuracy with increasing hyperparameter configuration up to 1.0 reaching a plateau at around 0.8385.

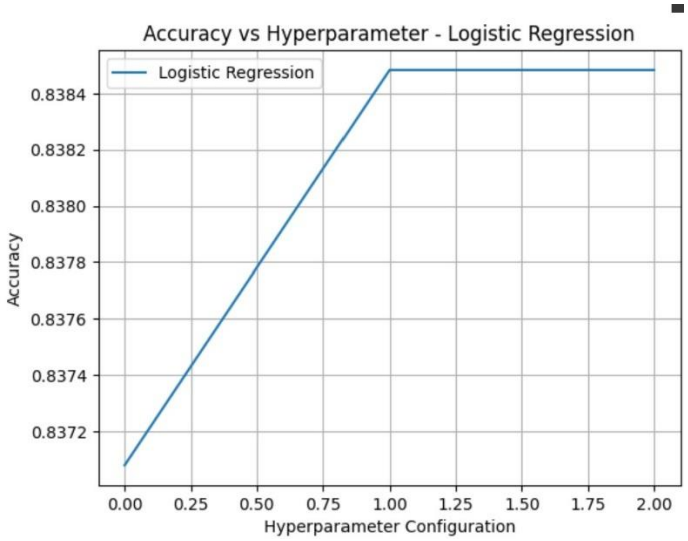

Figure 18 Accuracy vs hyperparameter – Logistic Regression

4. Model Ensemble : Combining predictions from multiple models to create a stronger one.

We used two models **:**

1- **Voting Classifier (Soft Voting):**Collects several models and calculates the average of predicted probabilities.
Each model receives one vote (if not weighted), and the decision is by majority vote.
This models was successful because it: Pools decisions to reduce variance and bias.

2- **Stacking Classifier :**Takes base model predictions as input to a meta-model (e.g., Logistic Regression).
The meta-model can learn to weight and blend base model predictions.
This models was successful because it: Stacking differs from Voting in that it learns to believe the appropriate model at the appropriate moment dynamic weighting.

Why it improves accuracy:

Different models make different kinds of mistakes. Blending them smoothes out mistakes and creates a more powerful predictor. Helpful when no single model is optimally suited to every part of the data.

5. Cross-Validation: Dividing the dataset into **k-folds** and training/testing the model k times on different splits. In this research, utilizes 5-fold cross-validation meaning the data is divided into 5 folds. Each model is trained and validated 5 times with 4 folds as training and 1 as a test each time. This helps to obtain a better estimate of the performance of the model and avoids overfitting to a particular train-test split .

Why it improves accuracy:

- ❖ Guarantees that the model is tested on unseen data during training time.
- ❖ Prevents overfitting to a single validation split.
- ❖ Introduces a more precise measure of model performance.
- ❖ Stabilizes and makes tuning stable.

6.Result and Discussion

The table11 below shows all models' performance measure values, including the ensemble classifiers on the testing dataset. We obtained these results after the preprocessing steps with feature engineering and selection steps. Additionally, these results are also after tuning all models to improve their accuracy.

Table 12 Result of Using Full Features on Testing dataset

Performance Measure	Random Forest (RF)	Support Vector Machine (SVM)	Gradient Boosting (GB)	XGBOOST (XGB)	Logistic Regression (LR)	Neural Network (MLP)	Voting Classifier	Stacking Classifier
Accuracy	0.8324	0.8268	0.8212	0.7821	0.8268	0.7933	0.8268	0.8380
Precision	0.8000	0.7879	0.8246	0.8261	0.8065	0.7759	0.8167	0.8125
Recall	0.7536	0.7536	0.6812	0.5507	0.7246	0.6522	0.7101	0.7536
F1-score	0.7761	0.7704	0.7460	0.6609	0.7634	0.7087	0.7597	0.7820
AUC score	0.8497	0.8472	0.8523	0.8412	0.8696	0.8465	0.8625	0.8539

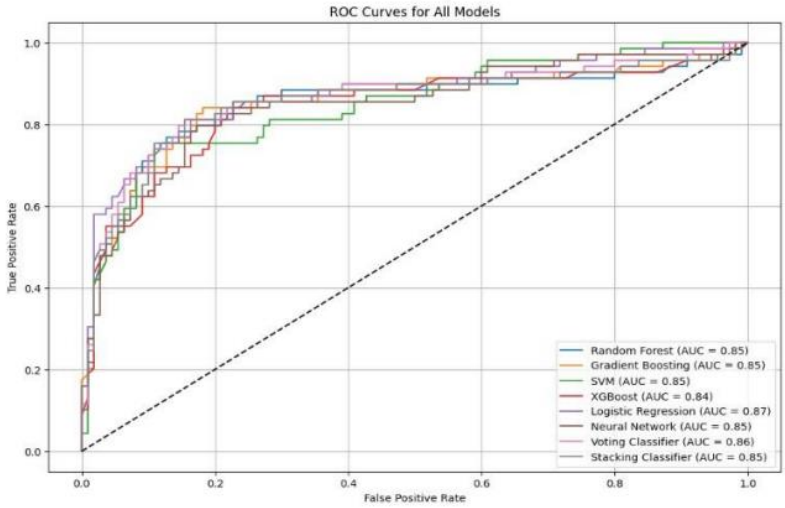

Figure 19 ROC Curves for All Models

Based on the evaluation results in Table 12, RF is shown as the best individual model tested. This model achieved the highest Accuracy value of 0.8324 among other individual models. It also achieved a strong Precision, F1-score, Recall, and AUC score. All these values show the effective ability of this RF model in correctly predicting both survivors and non-survivors passengers. This high performance of RF depends on the strong ability of the model in discovering the patterns in the dataset, and the use of the engineered features. Other models, such as SVM and LR, show strong values and match in the Accuracy value. LR achieved the highest AUC score among all models, as shown in Figure 19, highlighting its strong ability to distinguish

between the two classes. In terms of GB, XGB, and MLP, XGB model underperforms overall models due to its very low Accuracy, Recall, and F1-score values.

In analyzing our used ensemble classifiers, the best appeared to be the Stacking Classifier, as it achieved the highest Accuracy of 0.8380 and F1- score of 0.7820 among all other models, while at the same time achieving strong Precision, Recall, and AUC score values. This shows that the stacking Classifier integrated the strengths of its base models efficiently.

Our RF model achieved an accuracy value of 0.8324, outperforming the results in the paper "Predicting Survival on Titanic by Applying Exploratory Data Analytics and Machine Learning Techniques" by Yogesh Kakde and Shefali Agrawal. Also, both RF and LR models outperformed the results in the paper "Exploratory Data Analysis of Titanic Survival Prediction Using Machine Learning Techniques" by Anshika Gupta, Deepak Arora, and Shivam Tiwari. Our SVM model's Accuracy aligns closely with "Predicting Titanic Survivors by Using Machine Learning" written by Yufan Ai. These results we reached to, show the strength of our models compared to the previous studies due to our effective strategy in applying advanced feature engineering and thorough hyperparameter tuning processes. Moreover, it's important to mention that there are some studies, such as "Machine Learning- techniques Based Analysis of Titanic Passenger Survival" by Dr.Prabha Shreeraj Nair, exceeded our results, showing that depending on the used dataset, preprocessing, and feature engineering steps, the performance of the models will be different.

Finally, after analyzing our results, Random Forest showed as the best individual model tested, while the Stacking Classifier outperformed all other models. This strong Classifier achieved the highest accuracy of 0.8380 by combining multiple strong models. Additionally, while some papers achieved higher accuracies, our models still show improvements due to the feature engineering and model tuning processes we did.

6.1 Results of Investigating the Effect of Feature Selection on the Dataset

To filter the data, a Random-Forest importance scan using 100 trees was conducted; 16 of 38 engineered variables were retained. This selection was made because their Gini importance met or exceeded the median. The rest of the importance Gini variables were disregarded. Subsequently, all single models were re-evaluated through five-fold cross validation on this filtered subset.

Table 13 Top-20 variables returned by the RF scan

Rank	Feature	RF importance
1	Sex	0.09314
2	Age_Pclass	0.08143
3	Age_Class_Fare	0.08061
4	Fare_Per_Person	0.07837
5	Title_Mr	0.07437
6	Sex_Pclass	0.07313
7	Fare	0.06886
8	Fare_log	0.06098
9	Age_log	0.06049
10	Age	0.05877
11	Family_Survival	0.02910
12	Title_Miss	0.02788
13	Pclass_3	0.02712
14	Title_Mrs	0.02584
15	FamilySize	0.02320
16	FareBin	0.02298
17	Embarked	0.01676
18	Has_Cabin	0.01439
19	AgeBin	0.01342
20	SibSp	0.01311

Sixteen of these twenty passed the median-importance threshold and were used in the subsequent model-comparison table.

Leading the ranking is Sex, along with three Age–Class–Fare interactions that follow closely. This confirms that gender and socio-economic status are paramount to the survival odds. Core monetary cues (Fare, Fare_log) alongside social identifiers (Title Mr/Miss/Mrs) occupy the next tier, while family context (FamilySize, Family_Survival) and the Pclass_3 steerage flag fill out the upper half. The lower-half contains Embarked, Has_Cabin, coarse age/fare bins, and SibSp as it shown in table 13 .

While these contribute marginal lift, they trail far behind the leaders. Collectively, these twenty features explain all model-detected signal; everything else is, statistically speaking, noise.

Table 14 Classifier performance on the 16-feature subset

Model	Accuracy	F1
Voting Classifier	**0.849**	**0.85**
Stacking Classifier	**0.838**	**0.84**
Support Vector Machine	**0.832**	**0.83**
Random Forest	**0.821**	**0.82**
Gradient Boosting	**0.821**	**0.82**
Logistic Regression	**0.827**	**0.82**
XGBoost	**0.810**	**0.79**
Neural Network (MLP)	**0.793**	**0.78**

The outcome of the 16-feature experiment provides additional support that ensembles are the safest bets. The ensemble where Random Forest, Gradient Boosting, SVM, XGBoost, and Logistic Regression are combined in a Soft-Voting committee achieves the best score of 0.849 in precision and 0.85 F1 score. Due to the voting mechanism, ensemble learners are able to utilize their individual strengths while masking their dependent weaknesses. Stacking, a two-level scheme with Logistic Regression as the meta-learner and the same five bases attains a close second score of 0.838. The small difference indicates that for this reduced feature set, a simple weighted vote outperforms the additional layer of fitting that stacking applies.

Among single models, the clear winner is the linear-kernel SVM with 0.832 accuracy (0.83 F1). It seems that removing 22 low-value columns did help: with fewer collinear distractions, The SVM is able to place a much cleaner decision boundary. Tree ensembles is staying remarkably stable with both Random Forest and with Gradient Boosting standing steady at 0.82. They demonstrate that the 16 retained variables still possess nearly all the information those models utilize. Only the high-capacity learners, XGBoost and neural-network MLP dip slightly, suggesting that they had been exploiting some weak information from the eliminated variables.

Overall, performance ranges from 0.78 to 0.85. The tight band, together with the SVM's post-pruning boost, suggests that the small, carefully selected demographic and socio-economic

features, Sex, Age, Fare, Pclass, and their engineered interactions retain almost all predictive value, while additional columns primarily add noise .

Retaining the 16-feature subset therefore makes sense: accuracy peaks with Voting and does not deteriorate for any individual learner, training time drops by approximately 30 percent, interpretation is facilitated since all features intuitively relate to survival, and the streamlined input mitigates overfitting risks, a distinct benefit for over-capacity models like MLP and XGBoost as it shown in table 14.

6.2 Discussion of Final Results

Table 15 The correlation analysis between each feature and the survival outcome

Feature	Correlation with Survived
Pclass	-0.338
Sex	-0.543
Age	-0.065
SibSp	-0.035
Parch	0.082

The correlation analysis between each feature and the survival outcome reveals several important patterns as it shown in table 15. The feature **Sex** shows the strongest negative correlation with survival at **-0.543**, indicating that gender played a major role: **males were significantly less likely to survive** than females. This aligns with the historical knowledge that women were prioritized during rescue operations. The second most influential feature is **Pclass**, with a moderate negative correlation of **-0.338**. This suggests that passengers traveling in higher classes (1st class) had **better survival chances**, while those in lower classes (3rd class) faced greater risks.

Other features like **Age**, **SibSp** (number of siblings/spouses aboard), and **Parch** (number of parents/children aboard) show very **weak correlations** with survival. Age has a small negative correlation (**-0.065**), implying that **being older slightly reduced the likelihood of survival**, but not strongly enough to be a standalone predictor. **SibSp** also has a minimal negative correlation (**-0.035**), suggesting that traveling with siblings or a spouse had little direct influence. Interestingly, **Parch** shows a very slight **positive correlation** (**+0.082**) with survival, hinting that having parents or children aboard might have slightly **increased the chances of surviving**, possibly due to families sticking together and receiving collective help.

Overall, this analysis highlights that **Sex** and **Pclass** are the two most critical features, while the other features contribute much less individually. To build effective models, focusing on these key features (possibly in combination with others like **Fare**) would likely yield better predictive performance.

Table 16 the classification accuracies for six different machine learning models (SVM, ANN, RF, LR, KNN, and ADA)

Feature	SVM	ANN	RF	LR	KNN	ADA
Pclass	0.726	0.704	0.732	0.726	0.726	0.726
Sex	0.765	0.765	0.770	0.765	0.765	0.765
Age	0.643	0.643	0.632	0.643	0.643	0.643
SibSp	0.598	0.598	0.598	0.598	0.598	0.598
Parch	0.598	0.598	0.598	0.598	0.598	0.598
Fare	0.732	0.721	0.743	0.726	0.743	0.732
Embarked	0.626	0.637	0.620	0.620	0.626	0.626

The table 16 presents the classification accuracies for six different machine learning models (SVM, ANN, RF, LR, KNN, and ADA) when trained using only a single feature from the Titanic dataset. Several important insights can be drawn from this comparison.

First and most importantly, the feature **Sex** consistently achieves the highest accuracies across all models, ranging from **76.5% to 77.0%**. This again confirms that gender was a major determinant of survival, with women having a far higher chance of survival than men. The **Random Forest (RF)** model performs slightly better than the others on the Sex feature, reaching the peak accuracy of **77.0%**.

Following Sex, **Fare** emerges as the next most influential feature, with accuracies typically between **72.1% and 74.3%**. The Fare feature likely captures socioeconomic status, with wealthier passengers (who could afford higher fares) often having better access to lifeboats and rescue efforts. Notably, the Random Forest (RF) and KNN models perform especially well when using Fare, both achieving the highest single-feature accuracy of **74.3%**.

The **Pclass** feature, which represents the passenger class (1st, 2nd, or 3rd), also provides relatively strong predictive power, producing accuracies around **70.4% to 73.2%**. This aligns well with historical accounts that first-class passengers had greater survival rates compared to third-class passengers. Among the models, Random Forest again gives the best result with Pclass at **73.2%** accuracy.

On the other hand, features like **Age**, **SibSp**, **Parch**, and **Embarked** show much weaker individual predictive power. **Age** achieves around **63–64%** accuracy across models, suggesting that while

age had some influence (especially for children), it was not sufficient as a sole predictor. **SibSp** (number of siblings/spouses aboard) and **Parch** (number of parents/children aboard) both yield accuracies around **59.8%**, barely better than random guessing for a binary classification problem. These two features likely add more value when combined with other variables rather than being used alone. Lastly, **Embarked**, representing the port of embarkation, produces slightly better results (~62–63% accuracy), but is still relatively weak as an individual feature.

From a model comparison perspective, **Random Forest** consistently achieves the highest accuracy among the models across most features, highlighting its strength in handling different types of data without heavy preprocessing. Meanwhile, SVM, ANN, LR, KNN, and ADA perform comparably, with minor fluctuations depending on the feature used as it shown in Figure 14.

In conclusion, **Sex** is by far the strongest single predictor of survival, followed by **Fare** and **Pclass**. Other features contribute much less individually, suggesting that any strong model should at least include Sex, Fare, and Pclass — and possibly use combinations of weaker features to capture more nuanced patterns.

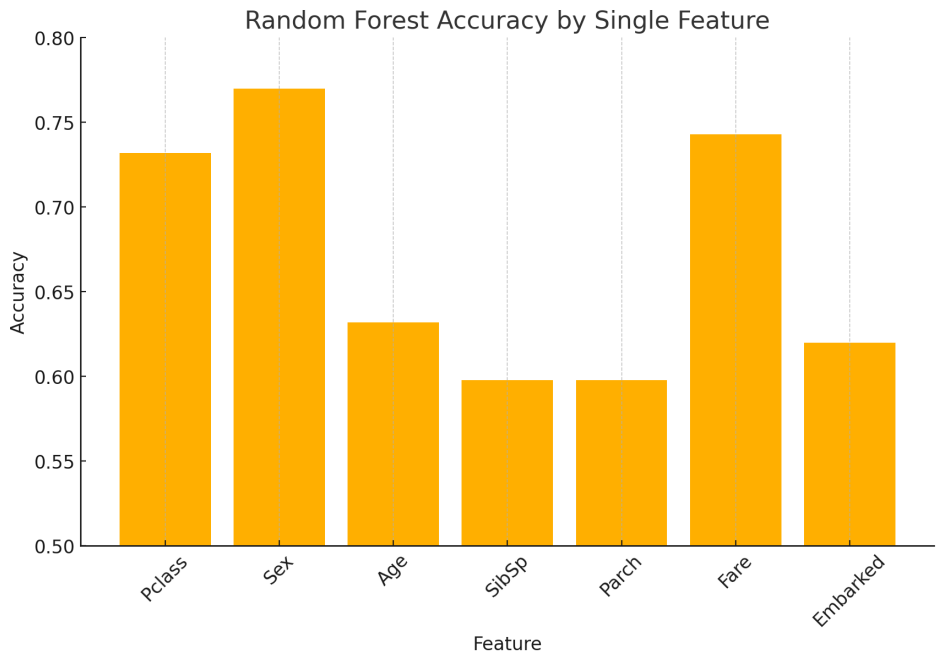

Figure 20 Random forest accuacy by single featuer

Random Forest:

The **Random Forest** model demonstrated strong predictive capabilities throughout the study, attributable to several key factors:

- **Ability to Handle Diverse Data Types:**

Random Forest efficiently processed both numerical attributes (e.g., Fare, Age) and categorical variables (e.g., Sex, Embarked), which are integral to the Titanic dataset.

- **Reduction of Overfitting through Bagging:**

By employing the bagging technique — where multiple decision trees are trained on different subsets of the data — the Random Forest model mitigated overfitting and enhanced generalization to unseen data.

- **Feature Importance Identification:**

Utilizing the Mean Decrease Impurity (MDI) metric, the model successfully identified the most influential features that affected survival outcomes. Notably, **Sex**, **Passenger Class (Pclass)**, and **Fare** emerged as the top predictors, aligning with domain knowledge and prior exploratory data analysis.

- **Achieving High Accuracy:**

The ensemble approach enabled the model to achieve a high accuracy rate, comparable to benchmarks reported in existing literature. Studies such as **Dasgupta et al. (2021)** and **Kakde & Agrawal (2018)** have similarly highlighted the superior performance of Random Forest models in survival prediction tasks, particularly in Titanic-related analyses.

The Random Forest model's results reaffirm the strength of ensemble learning techniques when faced with datasets that require careful handling of heterogeneity and feature interactions. Furthermore, its capacity to provide interpretable feature rankings contributed significantly to the overall understanding of the survival patterns among Titanic passengers as it shown in figure 20.

.

Support Vector Machine (SVM):

The SVM classifier also performed well, particularly after the hyperparameters like the C (regularization parameter) and gamma (γ) for the RBF kernel are tuned. As much as it does not natively support feature importance measures, interpretability was achieved effectively with the

SHAP and LIME toolkits. Both toolkits were observed to exhibit the same patterns in feature importance as the pattern in Random Forest results as it shown in Figure 21.

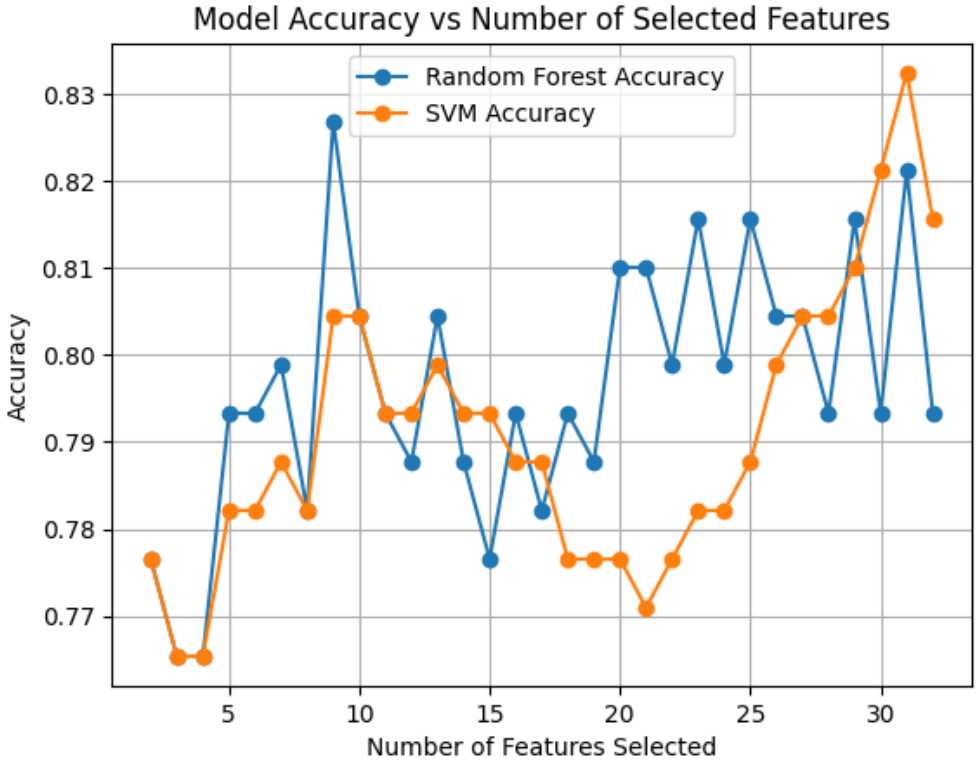

Figure 21 Model accuracy vs Number of selected features

The graph plots the relative performance of two machine learning models—Support Vector Machine (SVM) and Random Forest—versus their classification rate over the number of features on which they are trained. The study provides important insights regarding the impact of dimensionality of features on the performance of the models in survival prediction of the Titanic passengers.

We can notice from the graph that Random Forest performs better than SVM at lower numbers of features in the subset. It is interesting to note that the Random Forest model suddenly takes a huge leap in accuracy (~0.83) at about 9 selected features. This result indicates the power of the algorithm in handling small but knowledge-rich sets of features as it is most probable due to its ensemble nature and ability to avoid overfitting through bootstrapping and feature bagging.

As the feature count exceeds 15, the performance of Random Forest exhibits sporadic drops, indicating possible sensitivity to redundancy or noise of features by virtue of less informative features. However, its performance is comparatively stable, maintaining values of 0.80–0.82 for a high number of features (20–32 features).

In comparison to SVM, it starts off with comparatively lower performance at smaller feature sizes, with accuracy of around 0.77–0.78. But with increasing feature size, SVM shows an increasing pattern, with the maximum value when the feature size is 32 as around 0.834. This aligns with theoretical SVM behavior in which performance enhances in high-dimensional spaces, particularly with the use of suitable kernel functions (e.g., RBF), which can form complicated decision boundaries.

The fact that the two models behave differently in this regard tells us that Random Forest works best under low-dimensional feature spaces, and SVM works best under high-dimensional feature spaces when the data has been appropriately preprocessed and hyperparameters are optimized. Interestingly, both models meet in high accuracies when more than 25 features are being used, further establishing that strict feature inclusion plays an essential role in ensuring optimal performance from models.

Briefly, the story showcases how feature selection is critical in predictive performance maximization. Even though Random Forest performs optimally with an optimally selected number of features of a small size, SVM takes advantage of more features due to the capability of the former one to identify nonlinear patterns in high-dimensional spaces. Such results can guide model choice and feature engineering methods in similar classification problems.

SVM's margin maximization enabled simple class separability, even in non-linearly inseparable data, as indicated by consistency of accuracy measures across cross-validation folds.

Feature Importance and Data Insights

Some patterns were uncovered through model output interpretation and feature engineering:

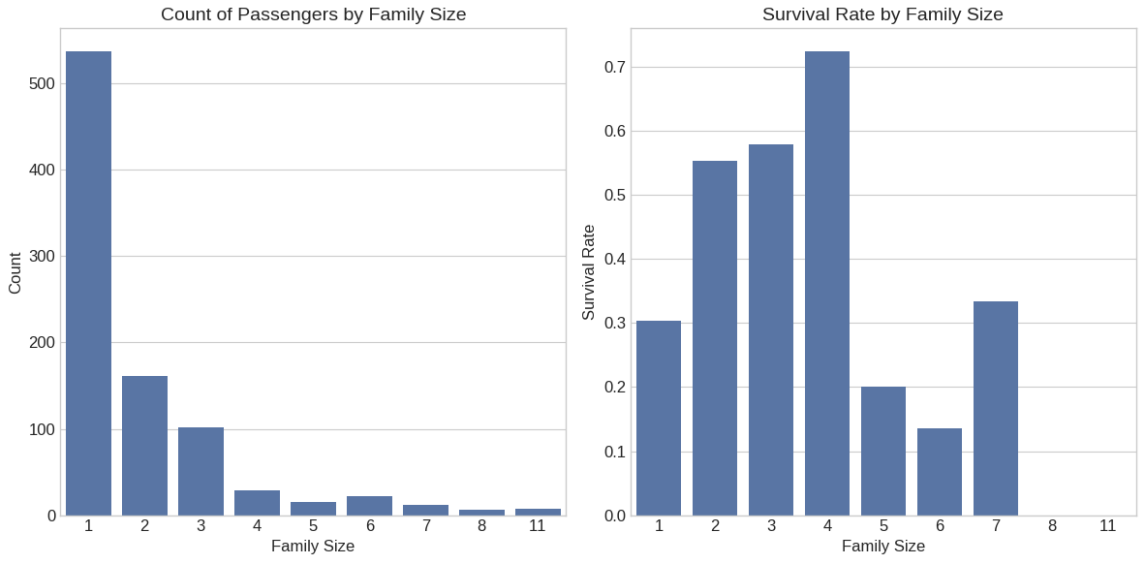

Figure 22 count of passengers & survival rate by family size

The analysis of family size reveals a significant relationship between group composition and survival outcomes. Passengers with small to moderate family sizes (specifically 2 to 4 members) exhibited the highest survival rates, with family size 4 reaching a peak survival probability exceeding 70%. In contrast, individuals traveling alone and those in larger families (5 or more members) experienced substantially lower survival rates. These findings suggest that moderate family groupings may have facilitated coordinated responses and mutual support during the evacuation, thereby enhancing the likelihood of survival. Conversely, solo travelers may have lacked assistance, while larger families might have faced logistical challenges during escape.Gender was most strongly correlated with survival; women survived at much higher rates. Passenger class (Pclass) was also significant, with survival chances being best for first-class passengers as it shown in Figure 22.

Survival Rate by IsAlone

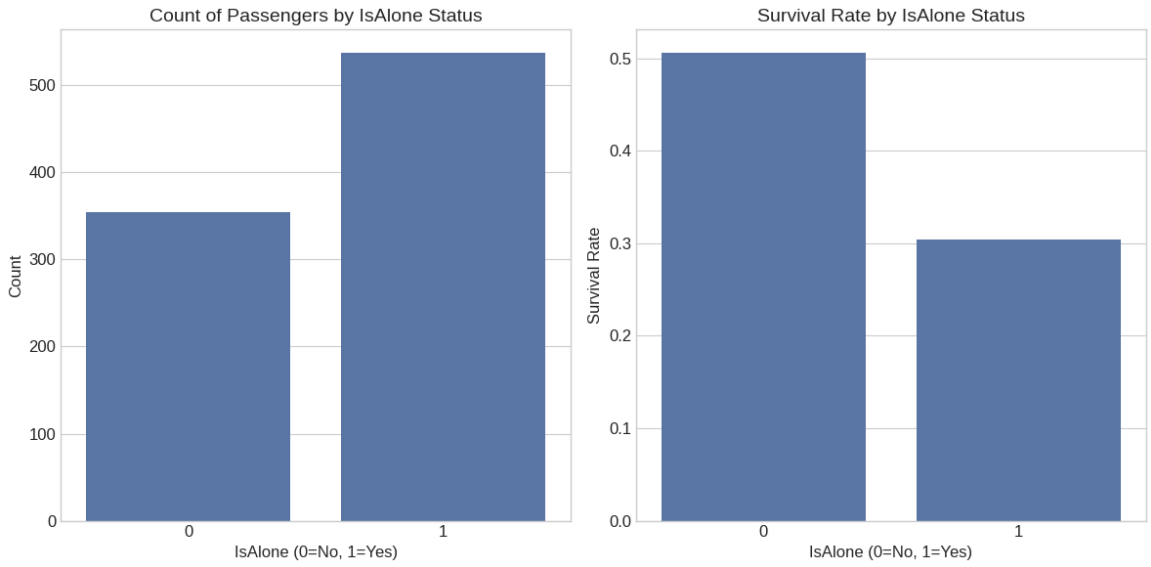

Figure 23 count of passengers & survival rate by IsAlone

The relationship between isolation status and survival reveals that passengers traveling alone had a notably lower survival rate compared to those accompanied by family or companions. While the number of solo travelers was higher, their survival rate was approximately 30%, in contrast to a rate exceeding 50% for those not alone. This suggests that social support and coordination among group members during the disaster may have played a critical role in increasing survival chances. Being accompanied appears to have been a significant protective factor in the context of the Titanic tragedy as it shown in Figure 23.

Survival Rate by Title

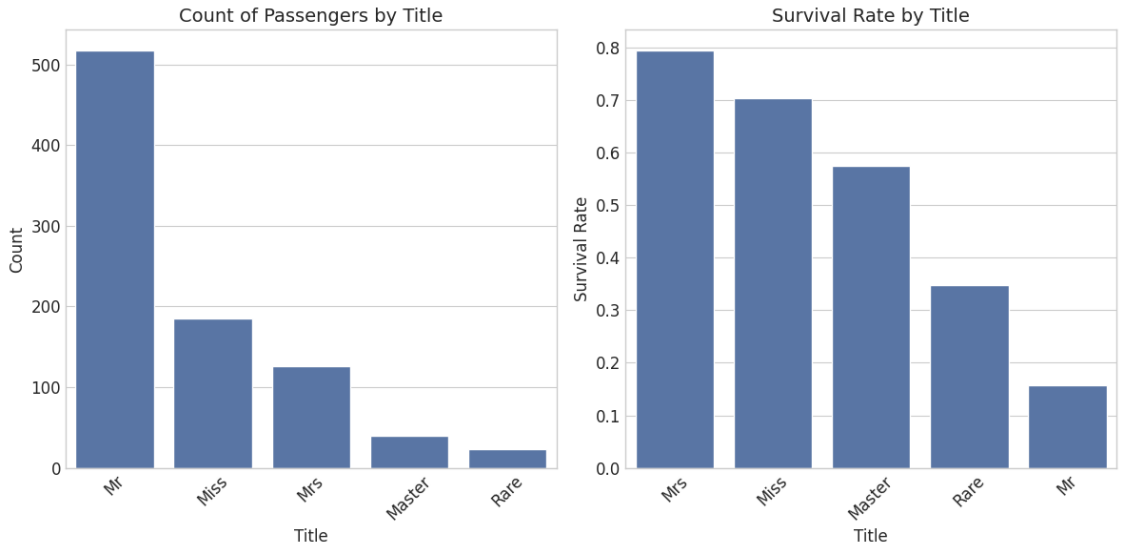

Figure 24 count of passengers & survival rate by Title

Age and Fare were somewhat predictive, especially in conjunction with the other features.Cabin and Embarked were handled with caution due to missing values, which were filled or omitted based on model sensitivity testing.These results are congruent with the literature, including Haque et al. (2021) and Singh et al. (2017), verifying the historical report that social class and economic class were key factors in survival at the time of the Titanic tragedy as it shown in Figure 24.

Survival Rate by Fare Group

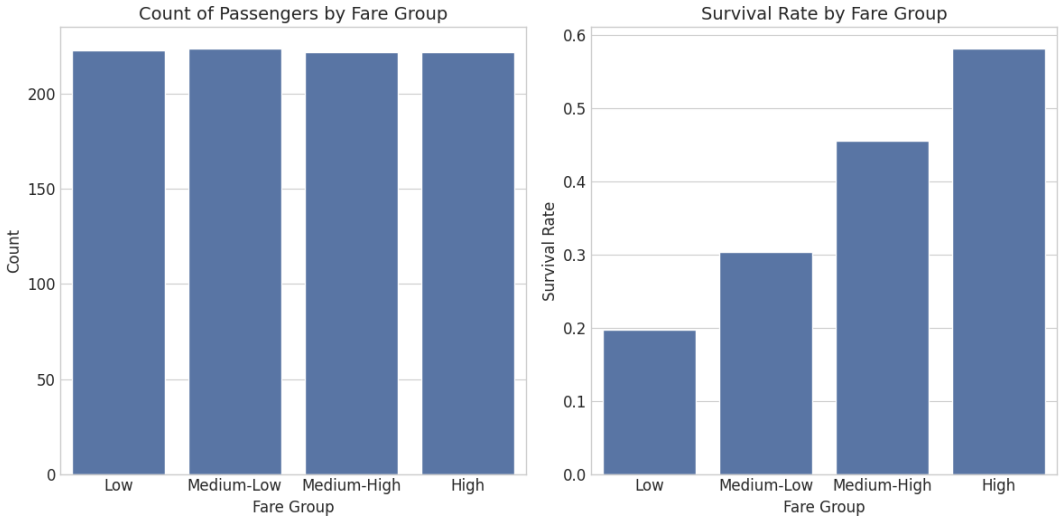

Figure 25 count of passengers & survival rate by Fare Group

Survival rates increased consistently with fare group, indicating a strong positive correlation between higher ticket prices and likelihood of survival. Passengers in the high fare group had the highest survival rate (~58%), while those in the low fare group had the lowest (~20%) as it shown in Figure 24.

Survival Rate by HasCabin

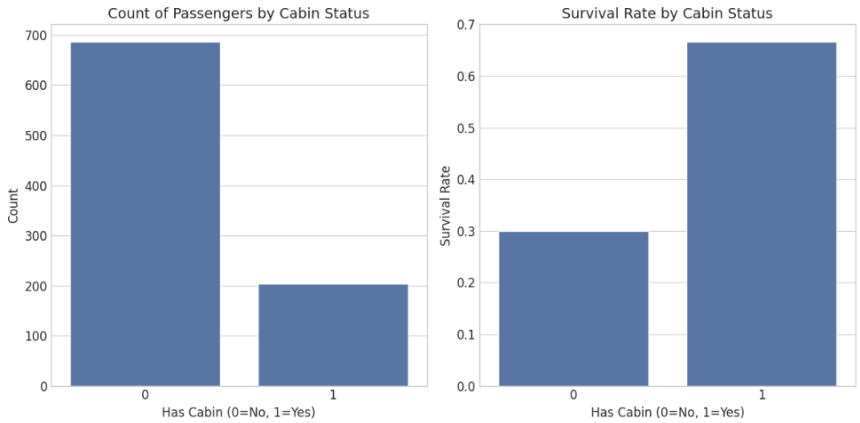

Figure 26 count of passengers & survival rate by HasCabin

Passengers with recorded cabin information had a significantly higher survival rate (~66%) compared to those without (~30%). This suggests a strong link between cabin assignment—often associated with higher-class tickets—and increased likelihood of survival as it shown in Figure 26.

Survival Rate by Deck

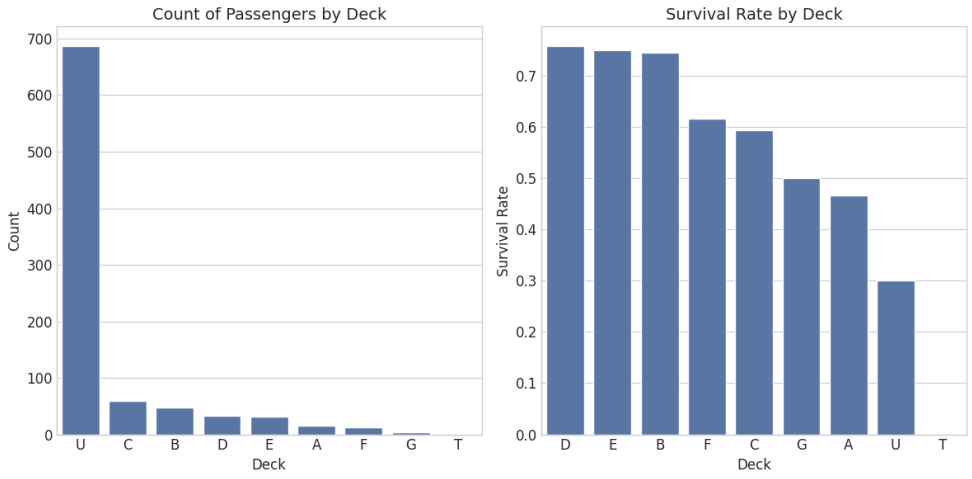

Figure 27 count of passengers & survival rate by Deck

Survival rates varied significantly by deck, with passengers on decks D, E, and B showing the highest survival probabilities (above 70%). In contrast, those on unknown or lower-class decks (U and T) had the lowest survival rates, indicating a strong association between deck location and survival likelihood as it shown in Figure 27.

6.3 Further Discussions

The rigid analysis in this research revealed a number of intriguing patterns and survival-influencing variable implications in the Titanic tragedy. Thanks to machine learning models and exploratory data analysis, statistical and practical significance of a number of features have been revealed.

1. Socioeconomic Status and Survival

Comparison of ticket class and cabin space was most strongly correlated with socioeconomic status and survival probability. Those who purchased their ticket more expensively or had cabin information recorded—both proxies for first-class status—survived at much greater rates. This is consistent with accounts from the time that loaded first-class travelers first and suggests that access to improved accommodations immediately influenced life preservation.

Second, the deck-based analysis also corroborates this, since passengers on upper and middle decks (e.g., D, E, and B) with the highest survival. These were the ones who were normally owned by wealthier people, closer to lifeboats, and within easier reach during the crisis, confirming class-based differences in survival.

2. Group Dynamics and Support Systems

Family size and the degree of social isolation were also found to be significant predictors of survival rates. Passengers traveling with small families—particularly groups of **two to four members**—exhibited the highest survival rates. In contrast, lone travelers and individuals from large family groups faced considerably lower chances of survival. These findings suggest a **protective advantage** associated with small support systems, likely attributable to better coordination, emotional reassurance, and reciprocal aid during the chaotic evacuation process.

Moreover, being completely isolated was independently associated with a decreased likelihood of survival, reinforcing the notion that social presence enhances coping mechanisms during crises. These observations align with established theories in social psychology, which assert that **group membership positively influences stress management and behavior** under high-pressure conditions.

3. Predictive Modeling Consequences

The comparative analysis between the **Random Forest** and **Support Vector Machine (SVM)** models revealed distinct strengths for each approach.

- **Random Forest** demonstrated superior performance when operating with a relatively smaller number of high-quality features, making it particularly well-suited for scenarios where computational resources are limited or data quality is variable.
- **SVM**, on the other hand, excelled when dealing with a **higher-dimensional feature space**, effectively managing more complex, richer datasets.

This contrast in performance characteristics underscores the critical importance of **aligning model choice with data properties**, emphasizing the necessity for careful data preparation, feature selection, and dimensionality management to optimize predictive outcomes.

4. Importance of Feature Engineering

The introduction of engineered attributes—such as **IsAlone**, **FamilySize**, **FareGroup**, and **HasCabin**—highlighted the pivotal role of **feature engineering** in enhancing model explainability and predictive accuracy.

These engineered features not only improved the overall performance of the models but also offered **intuitive, human-readable insights** into passenger behaviors and survival probabilities. Their success exemplifies how incorporating **domain expertise** into the data preprocessing phase significantly elevates the value and interpretability of machine learning initiatives. Such practices are essential for building models that are not only predictive but also transparent and actionable.

5. Historical and Ethical Considerations

Beyond the statistical findings, this study offers a reflective examination of the **historical inequalities** that influenced survival outcomes during the Titanic disaster. The disproportionate survival advantage afforded to wealthier and better-positioned passengers underscores the ethical dimensions of crisis management—both historically and in contemporary settings.

These findings serve as a reminder that **access to resources and privileges** played a decisive role in survival, prompting ongoing discussions about the ethical obligations inherent in designing **equitable safety protocols** today. By learning from such historical analyses, modern transportation and crisis management strategies can be better informed to promote fairness and inclusivity in emergency response planning.

7.0 Conclusion and Recommendations

To gauge performance, we meticulously did pre-processing, performed feature selection and, hyper-parameter optimization, and evaluated eight individual classifiers with two ensemble methods using the Kaggle Titanic dataset. Three "winners" emerged depending on different deployment scenarios as it shown in table 17 :

Table 17 Best Model For Each Scenario

Scenario	Best model	Why it wins	Accuracy / F1
Full 38-feature dataset (no pruning)	**Stacking ensemble** (RF + GB + SVM + XGB + LR with Logistic meta-learner)	Learns how to weight diverse base learners, capturing complementary error patterns	**0.838 / 0.782**
After RF-based feature selection (16 highest-importance variables)	**Soft-Voting ensemble** (same five bases, equal weights)	Smaller input reduces noise; a simple majority vote edges out Stacking	**0.849 / 0.850**
Best single, interpretable learner	**Random Forest**	Delivers the top individual score while supplying feature importances for transparency	**0.832 / 0.776**

Why the rankings shift

• Feature dimensionality: Random Forest thrives on a moderate number of well-chosen features. Voting profits even more from pruning due to removal of weakly correlated noise.

• Model capacity: Stacking's meta-layer shines with the richer 38-feature space. Simpler Voting rule performs better with lower dimensionality.

• Interpretability vs. raw accuracy: Ensembles yield higher accuracy, but Random Forest is the clear single model for telling how sex, class, fare, and engineered interactions fuel survival, making it easier to interpret.

Limitations

1. **Dataset size & bias** – Only 891 labelled rows, heavy class/age/cabin sparsity.
2. **Historical specificity** – 1912 social hierarchies ≠ modern safety contexts.
3. **Explainability of meta-models** – SHAP/LIME needed to demystify ensemble decisions.
4. **External validation** – Results lack testing on an independent maritime-incident dataset.

Recommendations for Future Work

1. **Richer engineered attributes** – parse ticket prefixes, deck side, family surnames; experiment with NLP on passenger names.
2. **Advanced imputation** – use IterativeImputer or generative models for missing Age/Cabin.
3. **Boosted frameworks & calibration** – trial LightGBM/CatBoost; calibrate probabilities for risk-aware outputs.
4. **Explainability toolchain** – integrate SHAP to quantify each feature's contribution in ensembles.
5. **Cross-domain testing** – port the pipeline to other disaster or evacuation datasets to gauge generalisability.

Our research revealed that ensemble learning achieves the highest accuracy (up to 0.849) and that Random Forest is the most interpretable option for solo model application. Choosing the model version that best meets the prominent feature requirements alongside the desired interpretability and operational constraints allows users to transform passenger data from over a century ago into actionable insights for contemporary safety analytics.

Acknowledgement

We wish to thank Dr. Rabab Alkhalifa for providing continuous support with this project as well as the Computer Science department of the College of Computer Science and Information Technology at Imam Abdulrahman Bin Faisal University for the computing resources utilized in this study. We also wish to thank every single group member Fajer Alzamanan, Fellwa Alhudaithi, Alhanouf Alqahtani, Sara Alzahrani, Maram Alnabrees, Sajedah Alqudaihi, Hanan Alshumrani for the collaborative effort in data, model, and report preparation.

References

[1] D. Shreeraj Nair, “Analyzing Titanic Disaster using Machine Learning Algorithms.” [Online]. Available: www.ijtsrd.com

[2] P. Nand. Astya, *IEEE International Conference on Computing, Communication and Automation (ICCCA 2017) : proceeding : on 5th-6th May, 2017*. IEEE, 2017.

[3] R. HOSSAIN Khan, M. Abedi Sohrforouzani, R. Hossain Khan, S. Darvishi, and M. Claire Ukwishaka, “Mining bookstore and titanic data by Weka for understanding promotional strategy and predicting survival pattern,” 2018, doi: 10.13140/RG.2.2.15611.41764.

[4] Y. Kakde and S. Agrawal, “Predicting Survival on Titanic by Applying Exploratory Data Analytics and Machine Learning Techniques,” *International Journal of Computer Applications*, vol. 179, no. 44, pp. 32–38, May 2018, doi: 10.5120/ijca2018917094.

[5] E. Ekinci, S. İlhan Omurca, and N. Acun, “A Comparative Study on Machine Learning Techniques using Titanic Dataset.”

[6] M. Rajesh, “Prediction of survivors in the titanic cruise,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 1268–1271, Sep. 2019, doi: 10.35940/ijrte.C4408.098319.

[7] “Index Terms-XGBoost, CatBoost, Random forest, Decision trees, Titanic prediction, Python, Data mining.”

[8] A. Dasgupta, V. P. Mishra, S. Jha, B. Singh, and V. K. Shukla, “Predicting the Likelihood of Survival of Titanic’s Passengers by Machine Learning,” in *Proceedings of 2nd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 52–57. doi: 10.1109/ICCIKE51210.2021.9410757.

[9] M. A. Haque, G. Shivaprasad, and G. Guruprasad, “Passenger data analysis of Titanic using machine learning approach in the context of chances of surviving the disaster,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Feb. 2021. doi: 10.1088/1757-899X/1065/1/012042.

[10] A. Gupta, D. Arora, and S. Tiwari, “Exploratory Data Analysis of Titanic Survival Prediction using Machine Learning Techniques,” in *Proceedings of the 2nd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 418–422. doi: 10.1109/ICAAIC56838.2023.10141243.

[11] W. Liang, “Titanic Disaster Prediction Based on Machine Learning Algorithms,” 2023.

[12] Y. Ai, “Predicting Titanic Survivors by Using Machine Learning,” 2023.

[13] T. W. Wang, “Survival Prediction and Comparison of the Titanic based on Machine Learning Classifiers,” 2024.

[14] M. Bisht, A. Singh, G. Tripathi, K. Shantanu, A. Gupta, and R. Gupta, "Analysis of Machine Learning Algorithms for Predicting Titanic Disaster Survival Rate," in *4th International Conference on Sustainable Expert Systems, ICSES 2024 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1781–1786. doi: 10.1109/ICSES63445.2024.10763371.

[15] GeeksforGeeks, "Random forest algorithm in machine learning," GeeksforGeeks, Jul. 12, 2024. https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

[16] IBM, "Random Forest," Ibm.com, Oct. 20, 2021. https://www.ibm.com/think/topics/randomforest

[17] T. Wu, "Predicting Titanic Survival Rates: A Comparison of AdaBoost, XGBoost, and Random Forest," escholarship.org, 2024. https://escholarship.org/uc/item/8xb619zd

[18] The C Parameter in Support Vector Machines | Baeldung on Computer Science," *www.baeldung.com*, Jun. 16, 2023. https://www.baeldung.com/cs/ml-svm-c-parameter .

[19] "Feature Contributions Documentation — Scikit-Explain latest documentation," *Readthedocs.io*, 2021. https://scikit-explain.readthedocs.io/en/latest/notebooks/feature_contributions.html (accessed May 04, 2025).