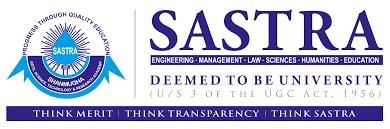
****

**TITANIC SURVIVAL PREDICTION USING MACHINE LEARNING TECHNIQUES**

**July - November 2025**

**Submitted By**

**Swathi Srinivasan - 125018069**

**B. Tech Computer Science and Business Systems**

**Submitted To**

**DR. Swetha Varadarajan**

**TABLE OF CONTENT**

**1. ABSTRACT**

**2. INTRODUCTION**

**3. BACKGROUND**

**4. METHODOLOGY**

**5. DISCUSSION**

**6. LEARNING OUTCOMES**

**7. REFERENCES**

**8. RESULT**

**Abstract**

This project leverages machine learning models to predict the survival of Titanic passengers based on demographic and ticket information from the well-known Titanic dataset. The dataset includes features such as age, gender, passenger class, fare, and family size. Data pre-processing involves handling missing values (mean imputation for age, dropping the Cabin feature), creating new features like family size and extracting titles from names, and label encoding categorical variables like gender and embarkation point. Three machine learning models—Logistic Regression, Random Forest, and Decision Tree—are trained to classify whether a passenger survived or not.

The Logistic Regression model achieved an accuracy of 81%, while the Decision Tree model reached 80.45% after hyperparameter tuning using GridSearchCV. The Random Forest model outperformed the others, with an accuracy of 83.8%, and further hyperparameter tuning through RandomizedSearchCV improved its generalizability, achieving a cross-validated accuracy of 83.16%. The Random Forest model was found to be the most effective at leveraging key features like gender, fare, and family size to predict survival outcomes.

Results are evaluated using accuracy, confusion matrix, and classification report, which highlight the strong predictive power of the Random Forest model. Cross-validation is used to ensure the models generalize well to unseen data. The project demonstrates that machine learning techniques can provide valuable insights into survival patterns. Future improvements could include using more advanced models like XGBoost, expanding feature engineering, and performing more detailed outlier analysis to further enhance prediction accuracy.

**Introduction**

The Titanic disaster remains one of the most tragic events in history, and its associated dataset has become a classic resource for machine learning applications. The goal of this project is to predict the survival of Titanic passengers based on various demographic and ticket-related features, such as age, gender, class, fare, and family size. Using machine learning models, we aim to uncover patterns that distinguish survivors from non-survivors. The Titanic dataset is ideal for this task due to its diverse range of features, which include both numerical and categorical data. By leveraging models like Logistic Regression, Random Forest, and Decision Tree, this project aims to build a robust classification system that predicts survival with high accuracy.

The project involves data pre-processing, feature engineering, model building, and evaluation. After handling missing data, creating new features, and encoding categorical variables, the dataset is split into training and testing sets. Multiple models are trained and evaluated, with a particular focus on optimizing Random Forest and Decision Tree through hyperparameter tuning. The project's results are presented in terms of accuracy, confusion matrix, and classification reports, providing insights into the key factors influencing survival.

**Background**

Models Used in the Project:

Three machine learning models—**Logistic Regression**, **Random Forest**, and **Decision Tree**—are used to classify passengers as survivors or non-survivors. Logistic Regression is a simple, interpretable model, while Random Forest and Decision Tree offer more flexibility and the ability to handle non-linear relationships in the data. Hyperparameter tuning is applied to Random Forest and Decision Tree models to improve their performance.

*Pre-processing Techniques:*

Pre-processing includes imputing missing values for the age column using mean imputation, dropping the cabin column due to excessive missing data, and creating new features such as family size and title extraction from names. Categorical variables like gender and embarkation point are label encoded. Fare values above 300 are capped to mitigate the impact of outliers. The dataset is then split into training and testing sets for model evaluation

**Methodology**

*Experimental Design:*

The project involves using the Titanic dataset to predict passenger survival using Logistic Regression, Random Forest, and Decision Tree models. Data preprocessing includes handling missing values, creating new features (e.g., family size), and label encoding. The models are trained and evaluated based on accuracy, confusion matrix, and classification report. Hyperparameter tuning is conducted for Random Forest and Decision Tree models to optimize performance.

*Environment and Tools:*

The project is implemented in Google Colab using Python. Key libraries include pandas and numpy for data handling, scikit-learn for model building, and matplotlib for visualization. RandomizedSearchCV and GridSearchCV are used for hyperparameter tuning.

*Code Location:*

The code and project documentation are available on GitHub at [Titanic Comparison repository] (https://github.com/Swathi2004/TitanicComparision.git).

*Pre-processing Steps:*

The dataset contains features such as age, gender, fare, and class. Missing values in the age column are imputed using the mean, and the Cabin column is dropped due to excessive missing values. The dataset is transformed by engineering features like family size and titles. Label encoding is used for categorical variables such as gender, embarked location, and title. The processed dataset is split into training and test sets with an 80-20 ratio.

**Discussion**

*Overall Results:*

The Random Forest model achieved the highest accuracy (83.8%) in predicting Titanic survival, outperforming the Logistic Regression and Decision Tree models. The results indicate that features like gender, fare, and family size significantly impact survival chances.

*Overfitting and Under fitting:*

Random Forest showed robust performance, but there was a potential risk of overfitting in the Decision Tree model. Hyperparameter tuning, such as adjusting the depth of the tree and minimum samples for splitting, helped control overfitting. Cross-validation was used to mitigate underfitting and ensure model generalization.

*Hyperparameter Tuning:*

RandomizedSearchCV was used to tune Random Forest parameters such as the number of estimators, max depth, and minimum samples required for splits. GridSearchCV was applied to fine-tune the Decision Tree, optimizing for max depth and minimum samples per leaf. Both methods helped improve model performance by identifying the best combination of hyperparameters.

*Model Comparison and Selection:*

Among the three models, Random Forest provided the best trade-off between accuracy and generalizability, with a cross-validation accuracy of 83.16%. Logistic Regression, although simpler, achieved a reliable 81% accuracy, while Decision Tree tuning resulted in an 80.45% test accuracy. Random Forest was ultimately selected due to its higher accuracy and stability across different parameter settings.

**LEARNING OUTCOMES**

#### *Link to GitHub Repository:*

The complete code and related files for this project are available in the **GitHub repository** at https://github.com/Swathi2004/TitanicComparision.git

#### *Skills and Tools Used:*

In this project, I utilized a range of skills, including data pre-processing, feature engineering, machine learning model building, and hyperparameter tuning. The primary tools used were **Google Colab** for interactive code execution and model training, and **Python** as the programming language. Data manipulation and pre-processing were handled with **pandas** and **numpy**, while **scikit-learn** was used for implementing models such as Logistic Regression, Random Forest, and Decision Tree, as well as for performing cross-validation and hyperparameter tuning. Visualization of data and model performance was done using **matplotlib**, allowing for clear interpretation of results. These tools combined provided a comprehensive environment for building, evaluating, and optimizing machine learning models.

#### *Dataset Used:*

The dataset used for this project is the Titanic survival dataset, which can be accessed at the following link: <https://github.com/datasciencedojo/datasets/blob/master/titanic.csv> .The dataset contains features such as age, gender, class, fare, and family size of passengers.

#### *Learning from this project*

Through this project, I learned how to apply multiple machine learning models (Logistic Regression, Random Forest, and Decision Tree) to a classification problem. I gained hands-on experience with **data pre-processing** techniques like handling missing values and feature engineering. I also improved my skills in **hyperparameter tuning** using GridSearchCV and RandomizedSearchCV to optimize model performance. Most importantly, I learned the importance of model evaluation using metrics such as accuracy, precision, recall, and cross-validation to ensure the model's generalizability across different datasets. Additionally, I explored the impact of different features (e.g., gender, fare) on survival predictions, providing insights into their significance in real-world scenarios.

**REFERENCE**

1. ChatGPT
2. <https://www.kaggle.com/c/titanic>
3. <https://www.analyticsvidhya.com/blog/2021/07/titanic-survival-prediction-using-machine-learning/>
4. https://medium.com/@melodyyip\_/titanic-survival-prediction-using-machine-learning-89a779656113

**RESULT**

*Logistic Regression*

The Logistic Regression model achieved an accuracy of 81% on the test set. The confusion matrix shows that the model correctly predicted 90 survivors and 55 non-survivors out of 179 passengers. While it provided a decent level of precision (83% for non-survivors and 79% for survivors), it struggled slightly with recall for survivors (74%), indicating that it missed a few passengers who survived. Overall, Logistic Regression performed well given its simplicity but did not capture all the non-linear patterns in the data.

*Random Forest*

The Random Forest model performed the best among the models with an accuracy of 83.8%. The confusion matrix revealed that the model correctly classified 91 survivors and 59 non-survivors, showing better recall and precision across the board. The Random Forest model was particularly effective at identifying both survivors and non-survivors due to its ability to handle complex interactions between features like age, fare, and gender. After hyperparameter tuning using RandomizedSearchCV, the model's best accuracy reached 83.16% during cross-validation, making it the most robust and reliable model for this task.

*Decision Tree*

The Decision Tree model achieved an accuracy of 80.45% on the test set. Hyperparameter tuning using GridSearchCV improved its performance by optimizing parameters like max depth and minimum samples per leaf. While its accuracy was slightly lower than Random Forest, it provided meaningful insights into the data's structure, showing clear decision rules based on features like gender and fare. However, the Decision Tree model tended to overfit slightly due to its sensitivity to training data structure, especially with deeper trees.

