Savitribai Phule Pune University
Modern Education Society's
Wadia College of Engineering, Pune
19, Bund Garden, V.K. Joag Path, Pune 411001.

ACCREDITED BY NAAC WITH A++ GRADE (CGPA 3.54)

DEPARTMENT OF COMPUTER ENGINEERING

A
REPORT
ON

"Mini Project"

"Survival of Passengers on Titanic using Machine Learning"

B.E. (COMPUTER)

SUBMITTED BY
Ashish A. Shisal (F21111032)
Saurabh K. Butale(F21111033)
Ayush S. Acharya(F21111036)

INTRODUCTION

This mini-project aimed to predict the survival of passengers on the Titanic using machine learning techniques. The project utilized the Titanic dataset, available from the famous Kaggle competition, to classify passengers as survivors or non-survivors based on various features such as age, gender, fare, and class.

OBJECTIVE

The primary goal was to build a machine learning model that could predict whether a passenger survived or not, based on the data available in the Titanic dataset.

DATASET OVERVIEW

- **Source**: Titanic dataset (train.csv and test.csv files)
- **Number of Records**:
  - Training dataset: 891 entries
  - Test dataset: Not specified
- **Number of Features**: 12 features including Passenger ID, Survival status, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked.

DATA PREPROCESSING

A. Handling Missing Values:

- **Age**: 177 missing values were replaced with the median values based on the passenger's title.
- **Cabin**: This column had 687 missing values, which were dropped due to the high number of missing values.
- **Embarked**: 2 missing values were filled with the most common port of embarkation ("S").
- **Fare**: Missing values in the test set were replaced by the average fare based on passenger class.

B. Feature Engineering:

- **Age Grouping**: Age was grouped into categories like 'Baby', 'Child', 'Teenager', 'Young Adult', 'Adult', and 'Senior' using binning techniques.
- **Title Extraction**: Titles like "Mr", "Mrs", "Miss", and others were extracted from the name to further enhance the predictive power of the model.
- **Cabin Information**: A binary feature called CabinBool was created to denote whether a passenger had cabin information or not.

C. Feature Mapping:

- **Sex**: Converted into numerical form, where male = 0 and female = 1.
- **Embarked**: The embarkation ports were mapped to numerical values: S = 1, C = 2, and Q = 3.
- **Fare**: The fare column was grouped into four bands for easier analysis.

DATA VISUALIZATION

Several visualizations were generated to explore the relationships between features and survival rates:

- **Survival Rates**: A pie chart and count plot showed the proportion of survivors and non-survivors.
- **Sex vs Survival**: Bar plots and count plots were used to compare survival rates between males and females.
- **Title vs Survival**: A crosstab showed how different titles (e.g., Mr, Mrs, Miss) affected the likelihood of survival.

MODEL SELECTION

A. Introduction

The Random Forest algorithm is one of the most widely used and effective ensemble learning techniques in machine learning. It is capable of performing both classification and regression tasks by aggregating predictions from multiple decision trees to improve accuracy and robustness. Random Forest reduces the tendency of overfitting often associated with individual decision trees, making it a highly popular algorithm for tasks in areas such as fraud detection, medical diagnostics, and image classification.

B. Working of Random Forest

The Random Forest algorithm builds a collection of decision trees and combines their outputs to provide the final prediction. Here's an overview of how it works:

Step 1: Bootstrapping

Random Forest creates multiple decision trees, each trained on a random subset of the training data. This is done using a technique called bootstrapping, where samples are drawn with replacement from the dataset to create each tree's training dataset. As a result, some samples may appear more than once, while others may not appear at all.

Step 2: Building Decision Trees

For each bootstrapped dataset, a decision tree is built. At each node of the tree, Random Forest randomly selects a subset of features from the total set of features and chooses the best feature from this subset for splitting the node. This randomness ensures that each tree is different from the others, promoting diversity.

Step 3: Aggregation of Predictions

- For classification tasks, each decision tree outputs a class, and the Random Forest selects the class with the most votes (majority voting).
- For regression tasks, each decision tree outputs a numerical value, and the Random Forest takes the average of these values to produce the final prediction.

C. Key Features of Random Forest

- Ensemble Learning: Random Forest combines multiple decision trees to create an ensemble model, enhancing accuracy and reducing the risk of overfitting.
- Randomization: The algorithm introduces two layers of randomness — random selection of training samples (bootstrapping) and random selection of features at each split — which ensures tree diversity and decreases overfitting.
- Feature Importance: Random Forest provides insights into which features are most influential in making predictions, allowing users to interpret the model to some extent.

- Handles Missing Data: The algorithm can handle missing values in the dataset and still produce reliable predictions.
- Works Well with High-Dimensional Data: Random Forest is capable of handling large datasets with many features and can capture non-linear patterns effectively.

D. Advantages of Random Forest

- High Accuracy: Due to its ensemble approach, Random Forest is known for high accuracy, especially when compared to single decision trees.
- Robustness to Overfitting: Random Forest reduces overfitting by averaging the predictions from multiple trees, which minimizes the impact of outliers or noise in the dataset.
- Scalability: Random Forest can efficiently handle large datasets with many features.
- Automatic Feature Selection: Random Forest selects relevant features during training, which can simplify feature engineering.

E. Disadvantages of Random Forest

- Slower Predictions: Because Random Forest relies on a large number of trees, making predictions can be slower compared to simpler algorithms, especially with a high number of trees (n_estimators).
- Less Interpretability: While Random Forest provides feature importance metrics, it is still less interpretable than a single decision tree, making it harder to understand the exact reasoning behind its predictions.
- Memory Intensive: Random Forest models can consume significant memory due to the need to store multiple decision trees.

MODEL PLANNING

A **Random Forest Classifier** was chosen for the prediction task due to its robustness and ability to handle both categorical and numerical data.

A. Feature Selection:

- The target variable was Survived.
- Predictor variables included Pclass, Sex, AgeGroup, SibSp, Parch, Embarked, CabinBool, Title, and FareBand.

B. Model Training:

- The dataset was split into training and validation sets using an 80-20 ratio.
- The model was trained using the Random Forest algorithm.

C. Model Evaluation:

- **Accuracy**: The model achieved an accuracy of **83.24%** on the validation set.

PREDICTIONS

The trained Random Forest model was used to predict the survival of passengers in the test set. The predictions were saved to a CSV file (titanic_submission1.csv).

CONCLUSION

The Random Forest model proved to be an effective algorithm for predicting Titanic survival, achieving an accuracy of over 83%. By leveraging feature engineering, missing data handling, and a well-chosen classification model, the project successfully met its objectives.