**Titanic - Machine Learning from Disaster**

**Submitted for**

**CSET211 - Statistical Machine Learning**

Submitted by:

**(E23CSEU0035) SHAIK SHASHA VALI**

Submitted to:

**DR. PRASHANT KAPIL**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**Abstract:**

The Titanic Survival Prediction project employs machine learning to predict passenger survival based on demographic and ticket data from the Titanic dataset. The study emphasizes effective preprocessing techniques, including handling missing values, feature engineering, and scaling, followed by evaluating models such as Logistic Regression, Random Forest, and Gradient Boosting. Gradient Boosting achieved the best performance with 80.7% accuracy and a high ROC-AUC score, underscoring the importance of ensemble methods in predictive tasks. Key features like Sex, P\_ class, and Age significantly influenced survival predictions. This project highlights the critical role of preprocessing, feature engineering, and model selection, providing a solid foundation for future improvements with deep learning and additional data sources. This complete project is uploaded on my git hub.

[**https://github.com/shashavali-8524/ML-Project**](https://github.com/shashavali-8524/ML-Project)

1. **Introduction:**

The Titanic Survival Prediction project is an exploratory data analysis and machine learning endeavour aimed at predicting the survival outcomes of passengers aboard the Titanic based on their demographic and ticket information. This project is motivated by the historical significance of the Titanic disaster and the challenge of deriving insights from structured data. The primary objectives are:

* To explore and preprocess the Titanic dataset.
* To apply machine learning models for survival prediction.
* To evaluate the effectiveness of different models using performance metrics.

Key contributions include providing a streamlined methodology for preprocessing, feature engineering, and machine learning application on the Titanic dataset.

1. **Related Survey:**

1. Title: Tackling the Titanic Dataset with Machine Learning

Techniques/Tools: Python, scikit-learn.

Preprocessing: Data loading from Kaggle, data cleaning, feature engineering.

Models Used: Decision Tree, hyperparameter tuning with grid search.

Performance: Achieved validation accuracy of 0.82 after tuning; best parameters included n\_ estimators = 200.

Reference: [Medium Article]

**(**[**https://medium.com/@sanjay\_dutta/tackling-the-titanic-dataset-with-machine-learning-5ece2f77b4f2**](https://medium.com/@sanjay_dutta/tackling-the-titanic-dataset-with-machine-learning-5ece2f77b4f2)**)**

2. Title: Predicting Titanic Survivors Using Machine Learning Techniques

Techniques/Tools: Python, various ML libraries.

Preprocessing: Handling missing values, converting features to numerical representations.

Models Used: Naïve Bayes, Support Vector Machine (SVM), Decision Tree.

Performance: Compared multiple models to assess predictive accuracy, utilizing a dataset of 891 individuals for training.

Reference: [Stanford Project]

**(**[**https://cs229.stanford.edu/proj2012/LamTang-TitanicMachineLearningFromDisaster.pdf**](https://cs229.stanford.edu/proj2012/LamTang-TitanicMachineLearningFromDisaster.pdf)**)**

3. Title: Titanic Disaster Prediction Based on Machine Learning Algorithms

Techniques/Tools: Python, Decision Tree, Random Forest for model building, feature importance visualization.

Preprocessing: Cleaned data, analyzed age, sex, and ticket class as significant features.

Models: Decision Tree and Random Forest used for survival prediction.

Performance: Decision Tree outperformed Random Forest in accuracy for predicting survival.

Reference: [Titanic Prediction Research]

**(**[**https://www.researchgate.net/publication/370569058\_Titanic\_Disaster\_Prediction\_Based\_on\_Machine\_Learning\_Algorithms**](https://www.researchgate.net/publication/370569058_Titanic_Disaster_Prediction_Based_on_Machine_Learning_Algorithms)**)**

4. Title: Machine Learning Approaches for Titanic Survival Prediction

Techniques/Tools: Python, pandas, scikit-learn.

Preprocessing: Data cleaning, feature normalization by replacing missing values with means.

Models Used: Naïve Bayes, SVM, Decision Tree analysis.

Performance: Evaluation of different techniques to establish a baseline for survival prediction.

Reference: [Stanford Project]

**(**[**https://cs229.stanford.edu/proj2012/LamTang-TitanicMachineLearningFromDisaster.pdf**](https://cs229.stanford.edu/proj2012/LamTang-TitanicMachineLearningFromDisaster.pdf)**)**

5. Title: Predicting Titanic Survivors Using Machine Learning Techniques

Techniques/Tools: Python, various ML libraries.

Preprocessing: Data cleaning, feature engineering.

Models Used: Multiple ML techniques including Decision Trees.

Performance: Utilized a dataset of 891 passengers to predict survival outcomes effectively.

Reference: [Medium Article]

**(**[**https://medium.com/@sanjay\_dutta/tackling-the-titanic-dataset-with-machine-learning-5ece2f77b4f2**](https://medium.com/@sanjay_dutta/tackling-the-titanic-dataset-with-machine-learning-5ece2f77b4f2)**)**

1. **Datasets**

The data we used for our project was provided on the **Kaggle website.** We were given **891** passenger samples for our training set and their associated labels of whether or not the passenger survived. For each passenger, we were given **his/her passenger** **class, name, sex, age,** number of siblings/spouses aboard, number of parents/children aboard, ticket number, fare, cabin embarked, and port of embarkation. For the test data, we had **418** samples in the same format.

The dataset is not complete, meaning that for several samples, one or many of fields were not available and marked empty (especially in the latter fields **– age, fare, cabin, and port**). However, all sample points contained at least information about gender and passenger class. To normalize the data, we replace missing values with the mean of the remaining data set or other values.

**3.1 Data Preprocessing:**

 Handling Missing Values: Imputing missing values in Age, Embarked, and Cabin columns using mean, mode, or categorical imputation.

 Feature Engineering: Creating new features such as Family Size and encoding categorical variables (Sex, Embarked) using one-hot encoding.

 Normalization: Scaling numerical features like Age and Fare for models sensitive to feature magnitudes.

 Outlier Detection: Identifying and addressing outliers in numerical features.

**Visualization:**

**Exploratory Data Analysis (EDA)**:

* + **Survival Distribution**: A bar plot was used to show the overall survival rate, revealing an imbalanced dataset with more passengers not surviving.
  + **Feature Relationships**:
    - **Sex vs. Survival**: A grouped bar chart showed that women had a significantly higher survival rate than men.
    - **P class vs. Survival**: A stacked bar chart highlighted that passengers in first class had the highest survival rate.
    - **Age Distribution**: A histogram displayed age distribution, with a kernel density estimate (KDE) showing survival likelihood varying across age groups.
    - **Fare vs. Survival**: A boxplot depicted higher fares correlating with a greater likelihood of survival
    - **Correlation Matrix**: A heatmap showed correlations between numerical features like Fare, Pclass, and survival

1. **Methodology:**

**Training and Testing**

1. **Data Splitting**:
   * The dataset was split into training (80%) and testing (20%) subsets using **Stratified K-Fold Cross-Validation** to ensure balanced class representation in each fold.
2. **Model Training**:
   * **Logistic Regression**:  
     A baseline model with minimal hyperparameter tuning achieved reasonable accuracy, emphasizing interpretability.
   * **Random Forest**:  
     Trained with hyperparameters like the number of estimators (n\_ estimators=100) and maximum depth (max\_ depth=5). Cross-validation optimized these parameters, resulting in improved accuracy and F1-Score.
   * **Gradient Boosting (XG Boost)**:  
     Tuned for learning rate (0.1), number of trees (n\_ estimators=200), and maximum depth (max\_ depth=4). It consistently outperformed other models due to its robust handling of imbalanced data and feature interactions.
3. **Evaluation**:
   * Performance metrics such as **accuracy**, **precision**, **recall**, **F1-Score**, and **ROC-AUC** were computed on both training and testing sets.
   * Confusion matrices were visualized for each model to identify misclassification patterns.
4. **Testing Results**:
   * The final evaluation on the testing set confirmed the high performance of Gradient Boosting, achieving:
     + **Accuracy**: 80.7%
     + **F1-Score (Survived Class)**: 0.80

**4.1 Hardware and Software Requirements:**

* **Hardware:**
  + The hardware required for this project is COMPUTER OR LAPTOP with minimum processer of intel i3.
* **Software:**
  + The software used in this project is Python 3.10
* **Libraries:**

Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, XG Boost, TensorFlow

* 1. **Performance Metrics:**

The following metrics are used to evaluate model performance:

* Accuracy: Measures the overall correctness of predictions.
* Precision, Recall, F1-Score: Provide insights into class-specific performance, particularly for the minority class (survived).
* ROC-AUC Score: Evaluates model performance in terms of distinguishing between survival and non-survival.

1. **Results and Analysis:**
   * Logistic Regression achieved an accuracy of 80.4% with moderate F1-Score for the survival class.
   * Random Forest outperformed simpler models with an accuracy of 80.1% and an ROC-AUC score of 0.88.
   * Gradient Boosting (XG Boost) achieved the highest performance with an accuracy of 80.7% and a balanced F1-Score.

Feature importance analysis highlighted Sex, P class, and Age as the most influential features.

1. **Conclusions and Future Works:**

The Titanic Survival Prediction project successfully demonstrated the power of machine learning in predicting survival outcomes based on structured data. Key findings include:

* The importance of feature engineering and preprocessing.
* The effectiveness of ensemble methods in improving predictive performance.

**Future work may include:**

* Exploring deep learning techniques for feature extraction.
* Incorporating natural language processing for textual features (e.g., passenger names and ticket details).
* Evaluating additional datasets for generalization.