**Comprehensive Report: Titanic Survival Prediction Using Machine Learning**

## **1. Introduction**

The Titanic disaster of 1912 remains one of the most infamous maritime tragedies in history. The goal of this project is to develop a machine learning model to predict whether a passenger survived the disaster. Using key features such as passenger class, age, gender, and fare price, we apply data preprocessing techniques and train a classification model for accurate survival prediction.

Our dataset consists of essential passenger information such as:

* **Pclass** (Ticket class: 1st, 2nd, or 3rd)
* **Sex** (Gender)
* **Age** (Passenger age)
* **SibSp & Parch** (Number of siblings/spouses and parents/children aboard)
* **Fare** (Ticket price)
* **Embarked** (Port of embarkation)
* **Cabin** (Cabin number, often missing)
* **Survived** (Target variable: 1 for survival, 0 for non-survival)

This project follows a structured pipeline involving data preprocessing, feature engineering, model selection, hyperparameter tuning, and evaluation.

## **2. Data Preprocessing**

### **2.1 Handling Missing Values**

The dataset contains missing values, which need to be addressed before training the model. Our approach includes:

* Filling missing **Age** values with the median age.
* Replacing missing **Fare** values with the median fare.
* Filling missing **Embarked** values with the mode (most common value).
* Dropping the **Cabin** column due to excessive missing values.

### **2.2 Feature Engineering**

To improve model performance, we engineer new features:

* **FamilySize**: Created by summing SibSp + Parch + 1.
* **Title**: Extracted from passenger names to capture social status.

### **2.3 Encoding Categorical Variables**

Machine learning models require numerical inputs. We encode categorical variables using **Label Encoding**:

* **Sex** (Male/Female → 0/1)
* **Embarked** (C, Q, S → 0, 1, 2)
* **Title** (Mapped to numerical values)

### **2.4 Normalizing Numerical Data**

Certain features such as **Age, Fare, and FamilySize** have different scales. We normalize these using **StandardScaler** to improve model efficiency and convergence.

## **3. Model Training and Hyperparameter Tuning**

### **3.1 Train-Test Split**

We split the dataset into training (80%) and testing (20%) sets using train\_test\_split().

### **3.2 Model Selection**

We use **Random Forest Classifier**, a robust ensemble learning method, due to its ability to handle categorical and numerical data while reducing overfitting.

### **3.3 Hyperparameter Tuning with GridSearchCV**

To optimize model performance, we tune key hyperparameters:

* **n\_estimators** (100, 200, 300) - Number of trees in the forest.
* **max\_depth** (5, 10, None) - Maximum depth of trees.
* **min\_samples\_split** (2, 5, 10) - Minimum samples required to split a node.

We use **GridSearchCV** with 5-fold cross-validation to find the best combination of parameters.

## **4. Model Evaluation**

### **4.1 Performance Metrics**

After training, we evaluate the model using:

* **Accuracy**: Overall correctness of predictions.
* **Precision**: Correctly predicted survivors vs. total predicted survivors.
* **Recall**: Correctly predicted survivors vs. total actual survivors.
* **F1 Score**: Harmonic mean of precision and recall.
* **Classification Report**: A detailed summary of model performance.

### **4.2 Cross-Validation Results**

To ensure generalizability, we conduct 5-fold cross-validation, achieving an average accuracy of **{cv\_scores.mean():.2f} ± {cv\_scores.std():.2f}**.

## **5. Feature Importance Analysis**

We analyze feature importance from the **Random Forest model**, identifying key predictors of survival:

1. **Sex** (Strongest predictor: Women had a higher survival rate.)
2. **Fare** (Higher fare correlated with better survival odds.)
3. **Title** (Social status played a role in survival.)
4. **Pclass** (First-class passengers had a better chance of survival.)

A feature importance plot visualizes these rankings.

## **6. Future Improvements**

While our model performs well, improvements could include:

* **Exploring Gradient Boosting models** (e.g., XGBoost, LightGBM).
* **Refining Title feature** by grouping similar social titles.
* **Applying Deep Learning models** for advanced feature learning.

## **7. Conclusion**

This project successfully predicts Titanic passenger survival using machine learning. Our **Random Forest model** achieves strong accuracy and generalization, with insights into survival factors such as **gender, fare, and social status**. Further refinements could enhance predictive power and interpretability.

**Final Verdict:** A well-structured and high-performing model for Titanic survival prediction!