Titanic Survival Prediction Project

1 Project Overview

The **Titanic Survival Prediction Project** is a comprehensive data analysis and machine learning endeavor focused on predicting the survival of passengers on the RMS Titanic. The project utilizes the famous Titanic dataset from Kaggle, which includes various features such as passenger age, gender, class, and more. The goal is to build a predictive model that can accurately determine whether a passenger would survive or perish based on these features.

2 Objectives

1. **Data Exploration and Preprocessing**:
   - Load and explore the Titanic dataset to understand its structure and the distribution of different features.
   - Handle missing data, outliers, and any inconsistencies within the dataset.
   - Convert categorical variables into numerical formats suitable for machine learning algorithms.
2. **Feature Engineering**:
   - Identify key features that significantly impact survival rates.
   - Create new features that may improve the predictive power of the model.
3. **Data Visualization**:
   - Use visualizations to gain insights into the relationships between different features and the survival rate.
   - Explore correlations and patterns within the dataset using plots and graphs.
4. **Modeling**:
   - Implement various machine learning algorithms to build predictive models.

  - Evaluate model performance using accuracy, precision, recall, F1-score, and confusion matrices.
  - Fine-tune models using techniques such as cross-validation and hyperparameter tuning.

5. **Model Comparison and Selection**:

  - Compare the performance of different models to select the best-performing one.
  - Discuss the advantages and limitations of each model.

6. **Deployment**:

  - Prepare the final model for deployment.
  - (Optional) Deploy the model as a web application or a script that can be used to make predictions on new data.

3 Project Workflow

3.1 1. Installation of Necessary Libraries

The project begins by installing essential Python libraries that are used throughout the analysis and modeling process. These include:

- `numpy` for numerical operations.
- `pandas` for data manipulation and analysis.
- `matplotlib` for data visualization.
- `scikit-learn` for implementing machine learning models.

3.2 2. Data Loading and Exploration

The Titanic dataset is loaded into a Pandas DataFrame for exploration. This step involves:

- Checking the structure of the dataset.
- Summarizing the data using descriptive statistics.
- Identifying and handling missing values.

3.3 3. Data Preprocessing

Data preprocessing includes:

- Encoding categorical variables (e.g., converting `Sex` from male/female to 0/1).
- Filling in missing values (e.g., using median or mean for numerical features, or the most frequent category for categorical features).
- Normalizing or scaling numerical features if necessary.

3.4 4. Feature Engineering

In this step, new features may be created, or existing features may be transformed to improve model performance. For example:

- Combining `SibSp` (number of siblings/spouses aboard) and `Parch` (number of parents/children aboard) into a `FamilySize` feature.
- Extracting titles from the `Name` feature to categorize passengers (e.g., Mr., Mrs., Miss, etc.).

3.5 5. Data Visualization

Visualization is crucial for understanding the relationships between features and the target variable (`Survived`). Common visualizations include:

- Bar plots to show survival rates by gender, class, and embarkation port.
- Histograms to show age distributions of survivors and non-survivors.
- Heatmaps to show correlations between features.

3.6 6. Model Building

Several machine learning models were implemented and evaluated, including:

- **Logistic Regression**: A simple and interpretable model for binary classification.
- **Decision Trees**: A model that captures non-linear relationships between features.
- **Random Forest**: An ensemble method that improves prediction accuracy by combining multiple decision trees.
- **Support Vector Machines (SVM)**: A robust model for classification tasks.
- **K-Nearest Neighbors (KNN)**: A non-parametric method based on proximity between data points.

3.7 7. Model Evaluation

The models were evaluated based on various metrics:

- **Accuracy**: The proportion of correctly predicted instances.
- **Precision and Recall**: Measures of model performance on the positive class.
- **F1-Score**: The harmonic mean of precision and recall.
- **Confusion Matrix**: A table showing the model's true positive, false positive, true negative, and false negative predictions.

Among the models used, Random Forest showed the highest accuracy and was therefore selected for further processes.

3.8 8. Model Selection and Tuning

The best-performing model, Random Forest, was selected based on the evaluation metrics. Hyperparameter tuning was conducted using methods like `GridSearchCV` to optimize the model's performance.

3.9 9. Conclusion and Next Steps

The project concludes with a summary of findings and potential next steps, such as deploying the model or exploring more advanced techniques like deep learning.