1. TITANIC SURVIVAL MODEL

The Titanic survival prediction model aims to predict whether a passenger survived or not based on various features such as age, sex, passenger class, and embarkation point. This documentation provides a detailed overview of the model development process, including data preprocessing, model selection, evaluation, and hyperparameter tuning.

DATA PREPROCESSING

- Loading Data: Two datasets, train (1).csv and test.csv , were loaded into separate DataFrames using Pandas.
- EDA: The training dataset's structure, data types, summary statistics, and missing values were examined to gain insights into the data
- Handling Missing Values: Removed the 'Cabin' column due to many missing values and deemed irrelevant, Filled missing values in the 'Age' column with the average age of passengers, Filled missing values in the 'Embarked' column with the most common port of embarkation.
- Categorical Variables: variables 'Sex' and 'Embarked' were encoded using LabelEncoder to convert them into numerical format for modelling.
- Train-Test Spilt: The training dataset was split into features (independent variables) and the target variable ('Survived').

MODEL SELECTION

- Classification Models: Several classification algorithms were considered for evaluation, including Logistic Regression, Random Forest, Decision Tree, K Nearest Neighbors, Support Vector Machine, and Gaussian Naive Bayes.

MODEL EVALUATION

- Cross-Validation: Each model's performance was evaluated using 5-fold cross-validation
- Performance Metrics: Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess each model's performance on the test set.

HYPERPARAMETER TUNING

- GridSearch: Hyperparameters for the Random Forest classifier were fine-tuned using GridSearchCV, an exhaustive search technique to find the best combination of hyperparameters.

RESULTS

- Best Model: Random Forest
- Test Accuracy of Best Model: 80%
- Best Cross-Validation Accuracy: 83%
- Test Accuracy of the Tuned Model:82%

2. BREAST CANCER MODEL

The breast cancer classification model aims to predict whether a breast tumour is malignant (cancerous) or benign (non-cancerous) based on various features extracted from digitized images of breast mass. This documentation provides an overview of the model development process, including data preprocessing, model selection, evaluation, and hyperparameter tuning.

DATA PREPROCESSING

- Loading data: The dataset was loaded from a CSV file named "data.csv" using the Pandas library.
- Data Cleaning: An unnamed column was identified in the dataset and subsequently dropped as it seemed to be irrelevant to the analysis
- EDA: Exploring the dataset was conducted to understand its structure, data types, summary statistics, and identify any missing values.
- Target variable: target variable 'diagnosis' was encoded as binary values ('M' for malignant, 'B' for benign) to prepare it for modelling.
- Train-Test Spilt: dataset was split into training and testing sets using the train_test_split function from the scikit-learn library.

MODEL SELECTION

- Classification model: Several classification algorithms were considered for evaluation, including Logistic Regression, Random Forest, Decision Tree, K Nearest Neighbors, and Support Vector Machine. Each model has its strengths and weaknesses, making them suitable for different types of data and problem domains

MODEL EVALUATION

- Cross- Validation: To access models performance I use 5-fold cross-validation. This technique splits the data into five equal parts, trains the model on four-fifths of the data, and evaluates it on the remaining one-fifth, repeating this process five times with different subsets.

- Performance Metrics: Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess each model's performance on the test set.

HYPERPARAMETER TUNING

- GridSearch: Hyperparameters for the Random Forest classifier were fine-tuned using GridSearchCV, an exhaustive search technique to find the best combination of hyperparameters.

RESULTS

- Best Model: Random Forest
- Test Accuracy of Best Model: 97%
- Best Cross-Validation Accuracy: 94%
- Test Accuracy of the Tuned Model:98%