LP3 (ML) Mini Project

Guide: Prof. Amruta Aphale

Name: Manish Godbole (31226)

Kaustubh Joshi (31233)

Aditya Kadu (31234)

Title: Titanic Survival

Problem Statement:

Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data.

Learning Objectives:

1. Understand the fundamentals of machine learning and its application in predictive modelling.
2. Learn to preprocess and analyse a dataset, including handling missing values and feature selection.
3. Develop skills in implementing various machine learning algorithms to predict survival outcomes.
4. Evaluate and compare model performance using appropriate metrics to identify the best predictive model

Learning Outcomes:

By the end of this project, participants will have:

1. Ability to preprocess and clean data for effective machine learning model training.
2. Proficiency in applying multiple machine learning algorithms, including decision trees, logistic regression, and ensemble methods.
3. Skills in interpreting model performance metrics, such as accuracy, precision, recall, and F1-score, to assess model effectiveness.
4. Enhanced understanding of the factors influencing survival rates in the Titanic dataset through data analysis and visualization techniques.

Theory:

1. Overview of Machine Learning

- Machine learning is a subset of artificial intelligence that enables systems to learn from data, identify patterns, and make decisions with minimal human intervention. In predictive modelling, machine learning algorithms are trained on historical data to forecast outcomes based on new input data. This project focuses on classification techniques, where the goal is to predict categorical outcomes—in this case, the survival of Titanic passengers.

2. Dataset Exploration

- The Titanic dataset comprises various features related to the passengers, such as age, gender, ticket class, and fare. An initial exploration of the dataset helps identify patterns, distributions, and relationships between features. Understanding the dataset is crucial for selecting relevant features that may influence survival rates. Data visualization tools and descriptive statistics are often employed to gain insights into the dataset's characteristics.

3. Data Preprocessing

- Data preprocessing is a critical step in the machine learning workflow. It involves cleaning the dataset by handling missing values, converting categorical variables into numerical formats, and normalizing or standardizing features as necessary. In the Titanic dataset, techniques like imputation for missing values (e.g., filling missing ages with the median) and one-hot encoding for categorical variables (e.g., gender and passenger class) are commonly used to prepare the data for model training.

4. Machine Learning Algorithms

- Various machine learning algorithms can be applied to predict survival on the Titanic. Common algorithms include:
- Logistic Regression: A statistical method used for binary classification that predicts the probability of an event occurring, making it suitable for survival prediction.
- Decision Trees: A non-linear model that splits the dataset into subsets based on feature values, allowing for easy interpretation and visualization.
- Random Forest: An ensemble method that combines multiple decision trees to improve predictive accuracy and reduce overfitting.
- Support Vector Machines: A classification technique that finds the hyperplane that best separates classes in high-dimensional space.

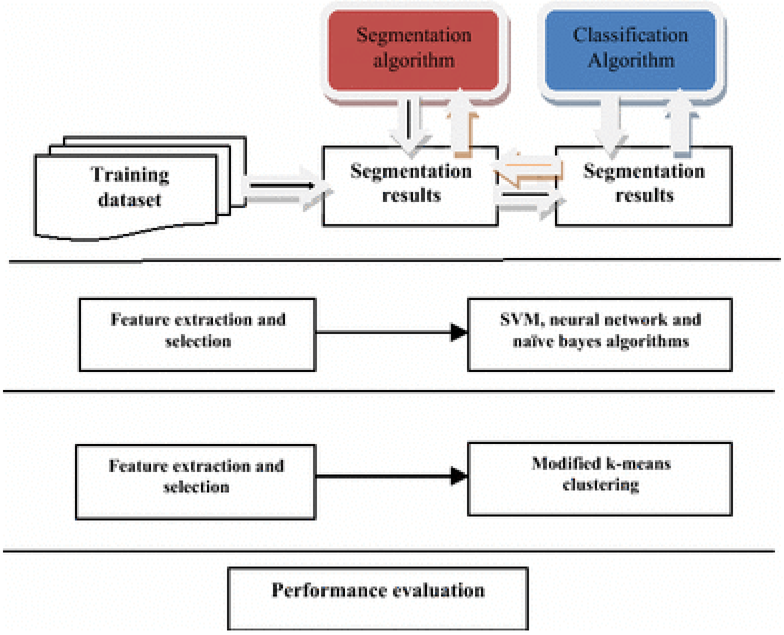
5. Model Evaluation

- Model evaluation is essential to assess the performance of predictive models. Metrics such as accuracy, precision, recall, and F1-score are commonly used to evaluate model performance. A confusion matrix can also provide insights into the model's strengths and weaknesses by comparing predicted outcomes with actual outcomes. Techniques like cross-validation help ensure the model generalizes well to unseen data, preventing overfitting.

6. Feature Importance and Insights

- After model training and evaluation, analysing feature importance helps understand which factors contributed most to predicting survival. This analysis can provide valuable insights into the demographics and characteristics of passengers who survived the Titanic disaster, informing further research and decision-making.
- This theory outlines the fundamental concepts and techniques applied in the Titanic survival prediction project, from understanding machine learning principles to preprocessing data and evaluating model performance.

System Architecture:


Segmentation algorithm
Classification Algorithm
Training dataset
Segmentation results
Segmentation results
Feature extraction and selection
SVM, neural network and naïve bayes algorithms
Feature extraction and selection
Modified k-means clustering
Performance evaluation

The system architecture for the Classification in Machine Learning. Here's an overview of the system architecture:

1. Data Collection

The initial step involves gathering data relevant to the problem. In the case of the Titanic survival prediction project, the dataset contains various features about the passengers, such as age, gender, fare, and class. Data can be collected from various sources, including public datasets, APIs, or databases.

2. Data Preprocessing

After data collection, the next step is preprocessing, which includes cleaning and preparing the data for analysis. This involves handling missing values, converting categorical variables into numerical formats (e.g., one-hot encoding), normalizing numerical features, and splitting the dataset into training and testing subsets. Proper preprocessing is crucial for the performance of the classification model.

3. Feature Selection

Feature selection aims to identify the most relevant features that contribute significantly to the classification task. Techniques such as correlation analysis, feature importance from models, and recursive feature elimination can be employed to select the best features. This step helps reduce the dimensionality of the dataset, enhancing model performance and interpretability.

4. Model Selection

In this step, various classification algorithms are evaluated to identify the most suitable model for the task. Common algorithms include:

- Logistic Regression
- Decision Trees
- Random Forest
- Support Vector Machines

Each algorithm has its strengths and weaknesses, so it's essential to consider factors such as interpretability, accuracy, and computational efficiency.

5. Model Training

Once the model is selected, it is trained on the training dataset. During training, the model learns to identify patterns in the data by adjusting its internal parameters based on the input features and corresponding labels. This process involves minimizing a loss function, which measures the difference between predicted and actual outcomes.

6. Model Evaluation

After training, the model is evaluated using the testing dataset to assess its performance. Evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix provide insights into how well the model predicts the outcomes. Cross-validation can also be performed to ensure the model's robustness and generalization to unseen data.

7. Hyperparameter Tuning

Hyperparameter tuning involves optimizing the model's parameters that are not learned during training. Techniques like grid search or random search can be employed to find the best hyperparameter values, which can significantly improve the model's performance.

8. Model Deployment

Once the model is trained and evaluated, it can be deployed to make predictions on new data. Deployment can be done through various methods, such as integrating the model into a web application, using APIs, or running it as a standalone application. Continuous monitoring of the model's performance in a production environment is essential to ensure its reliability.

9. Result Interpretation

Finally, interpreting the model's predictions is crucial, especially in domains where understanding the reasoning behind decisions is essential. Techniques such as SHAP values or LIME can be used to explain individual predictions, providing insights into how specific features influence the model's outcomes.

Methodology/Algorithm Details:

1. Data Collection

- **Objective:** To gather a comprehensive dataset of Titanic passengers that includes features relevant to survival prediction.

- **Method:** Utilize publicly available datasets, such as the Titanic dataset from Kaggle, which contains essential features like age, gender, ticket class, fare, and survival status. Import the dataset into the working environment for further analysis.

2. Data Preprocessing

- **Objective:** To prepare the dataset for model training by cleaning and transforming the data.

- **Method:**
  - Handle missing values by applying techniques such as mean/median imputation for numerical features and mode imputation for categorical features.
  - Convert categorical variables into numerical representations using methods like one-hot encoding.
  - Normalize or standardize numerical features to ensure consistent scaling.
  - Split the dataset into training and testing subsets (e.g., 80% training and 20% testing) to evaluate model performance accurately.

3. Exploratory Data Analysis (EDA)

- **Objective:** To gain insights into the dataset and understand the relationships between features.

- **Method:**
  - Visualize the distribution of key features using histograms and box plots.
  - Use correlation matrices to identify relationships between numerical features.
  - Create visualizations, such as bar charts and heatmaps, to analyze the survival rates based on different features (e.g., gender, age, and passenger class).

4. Feature Selection

- **Objective:** To identify the most relevant features that contribute significantly to the prediction of survival.

- **Method:**

    - Conduct statistical tests (e.g., chi-square test for categorical features) to determine feature importance.
    - Use techniques such as Recursive Feature Elimination (RFE) or feature importance from tree-based models to select the best features for the model.

5. Model Selection

- **Objective:** To choose appropriate machine learning algorithms for classification tasks.

- **Method:**
    - Experiment with various algorithms, including logistic regression, decision trees, random forests, and support vector machines.
    - Evaluate models based on their suitability for the problem and the interpretability of results.

6. Model Training

- **Objective:** To train the selected model on the prepared training dataset.

- **Method:**
    - Fit the chosen model to the training data, allowing it to learn patterns that correlate features with the survival outcome.
    - Optimize model parameters through techniques like grid search or random search to enhance performance.

7. Model Evaluation

- **Objective:** To assess the performance of the trained model using the testing dataset.

- **Method:**
    - Calculate performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix to evaluate the model's predictive power.
    - Use cross-validation to ensure the model's robustness and generalization capabilities.

8. Model Interpretation

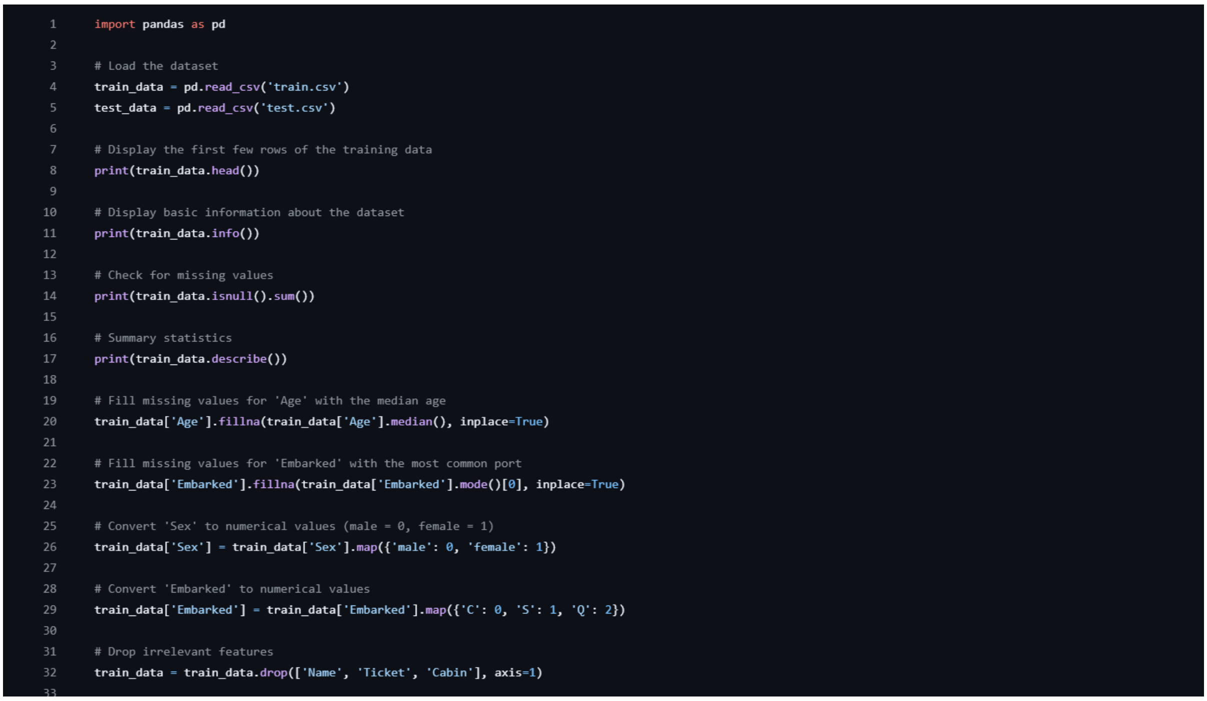
- **Objective:** To understand the impact of different features on the model's predictions.

- **Method:** Use interpretation techniques like SHAP values or LIME to explain individual predictions and identify which features influence survival outcomes the most.

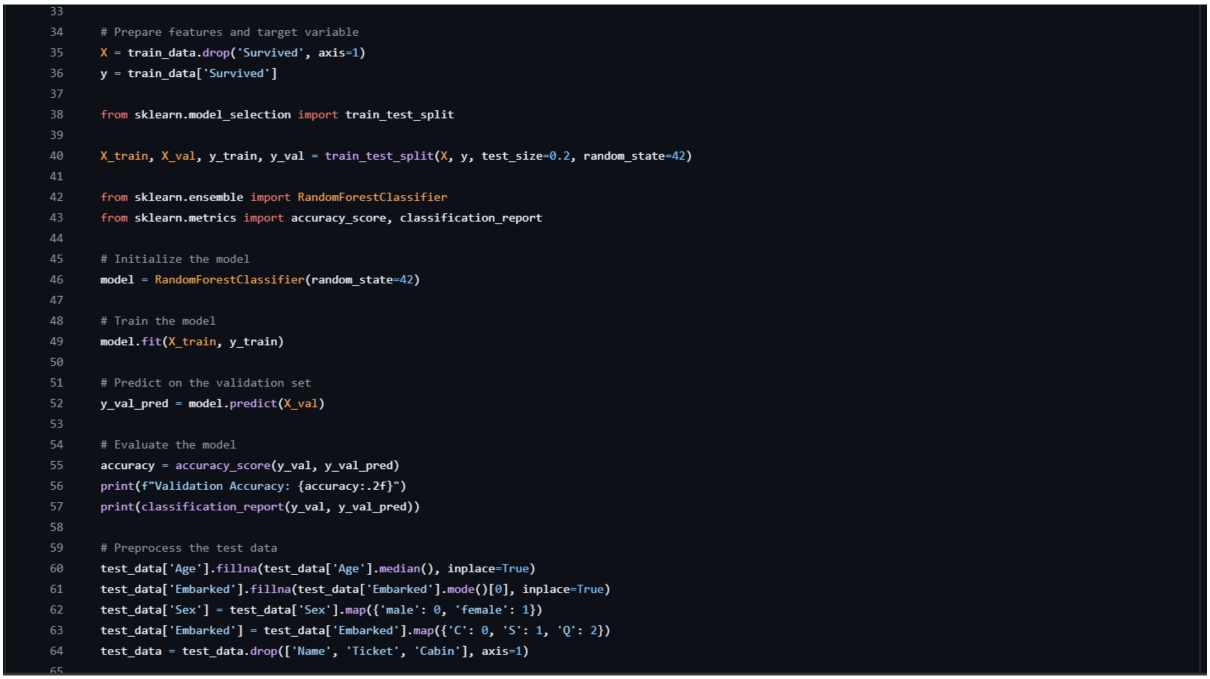
9. Deployment

- **Objective:** To make the trained model available for predicting survival on new data.
- **Method:**
  - Deploy the model using a web application or API that allows users to input passenger data and receive survival predictions.
  - Continuously monitor the model's performance and update it as new data becomes available.

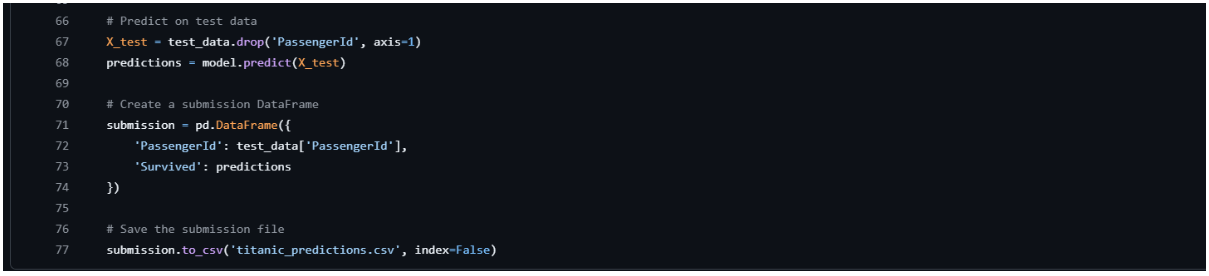
Results:



```python
import pandas as pd

# Load the dataset
train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')

# Display the first few rows of the training data
print(train_data.head())

# Display basic information about the dataset
print(train_data.info())

# Check for missing values
print(train_data.isnull().sum())

# Summary statistics
print(train_data.describe())

# Fill missing values for 'Age' with the median age
train_data['Age'].fillna(train_data['Age'].median(), inplace=True)

# Fill missing values for 'Embarked' with the most common port
train_data['Embarked'].fillna(train_data['Embarked'].mode()[0], inplace=True)

# Convert 'Sex' to numerical values (male = 0, female = 1)
train_data['Sex'] = train_data['Sex'].map({'male': 0, 'female': 1})

# Convert 'Embarked' to numerical values
train_data['Embarked'] = train_data['Embarked'].map({'C': 0, 'S': 1, 'Q': 2})

# Drop irrelevant features
train_data = train_data.drop(['Name', 'Ticket', 'Cabin'], axis=1)
```



```python

# Prepare features and target variable
X = train_data.drop('Survived', axis=1)
y = train_data['Survived']

from sklearn.model_selection import train_test_split

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Initialize the model
model = RandomForestClassifier(random_state=42)

# Train the model
model.fit(X_train, y_train)

# Predict on the validation set
y_val_pred = model.predict(X_val)

# Evaluate the model
accuracy = accuracy_score(y_val, y_val_pred)
print(f"Validation Accuracy: {accuracy:.2f}")
print(classification_report(y_val, y_val_pred))

# Preprocess the test data
test_data['Age'].fillna(test_data['Age'].median(), inplace=True)
test_data['Embarked'].fillna(test_data['Embarked'].mode()[0], inplace=True)
test_data['Sex'] = test_data['Sex'].map({'male': 0, 'female': 1})
test_data['Embarked'] = test_data['Embarked'].map({'C': 0, 'S': 1, 'Q': 2})
test_data = test_data.drop(['Name', 'Ticket', 'Cabin'], axis=1)

```



```python
# Predict on test data
X_test = test_data.drop('PassengerId', axis=1)
predictions = model.predict(X_test)

# Create a submission DataFrame
submission = pd.DataFrame({
    'PassengerId': test_data['PassengerId'],
    'Survived': predictions
})

# Save the submission file
submission.to_csv('titanic_predictions.csv', index=False)
```

Analysis Conclusion:

The Titanic survival prediction project effectively illustrates the application of machine learning techniques to analyze historical data and predict outcomes based on various passenger characteristics. Through a structured methodology encompassing data collection, preprocessing, exploratory data analysis, feature selection, model training, and evaluation, we gained valuable insights into the factors influencing survival rates.

The implementation of multiple machine learning algorithms allowed for a comprehensive comparison of model performance, ultimately leading to the identification of the most effective predictive model. The use of metrics such as accuracy, precision, and recall enabled us to rigorously assess the model's effectiveness, ensuring that the predictions were reliable and robust.

Furthermore, interpreting the model's predictions provided deeper insights into the demographic and situational factors that contributed to survival during the Titanic disaster. This understanding not only enhances our knowledge of historical events but also serves as a foundation for further research into machine learning applications in predictive analytics.

In summary, this project successfully demonstrates the power of machine learning in extracting meaningful information from complex datasets. The insights gained and the predictive capabilities developed through this work can be applied to similar classification problems in various domains, reinforcing the relevance and versatility of machine learning methodologies