Machine Learning Model Report:

Titanic Dataset

1. Introduction

In this project, I explored the **Titanic dataset**, which is a well-known dataset in the field of machine learning and data science. The dataset contains information about the passengers on the Titanic, including their age, gender, class, and whether they survived or not. This dataset is particularly relevant as it allows us to apply classification techniques to predict survival outcomes based on various features. Understanding the factors that influenced survival can provide insights into historical events and improve our predictive modelling skills.

2. Model Selection

For this analysis, I chose to implement a **Decision Tree Classifier**. The Decision Tree algorithm is intuitive and easy to interpret, making it an excellent choice for beginners and for datasets with categorical features. Additionally, it works well with smaller datasets like the Titanic dataset, allowing for clear visualizations of decision paths. The ability to handle both numerical and categorical data without extensive preprocessing further justified my choice.

3. Methodology

1. Data Preprocessing:

1. **Loading the Data**: I began by loading the Titanic dataset from a CSV file.
2. **Exploratory Data Analysis**: I examined the dataset using `data.head()` and `data.info()` to understand its structure and identify any missing values.
3. **Handling Missing Values**:
    - For the **Embarked** column, I filled missing values with the most frequent embarkation point.
    - For the **Age** column, I used a probabilistic approach to fill missing values based on the distribution of ages among passengers.
4. **Encoding Categorical Variables**: I converted categorical variables such as `Sex` and `Embarked` into numerical format using one-hot encoding.
5. **Feature Selection**: After preprocessing, I separated the features (X) from the target variable (y), where `Survived` was my target.

2. Model Training:

After preprocessing the data, I split it into training and test sets using an 80-20 split. This allowed me to train my model on a majority of the data while reserving some for evaluation. I initialized the Decision Tree Classifier and fit it to the training data.

3. Model Evaluation:

To evaluate model performance, I used several metrics:

- **Accuracy**: The proportion of correctly predicted instances.
- **Precision**: The ratio of true positive predictions to all positive predictions.
- **Recall**: The ratio of true positive predictions to all actual positives.
- **F1 Score**: The harmonic mean of precision and recall.

I also performed cross-validation to assess model reliability by splitting the data into five subsets for training and testing.

4. Results

After training and evaluating my model, I obtained the following performance metrics:

- **Accuracy**: 0.77
- **Precision**: 0.77
- **Recall**: 0.77
- **F1 Score**: 0.77

The cross-validation scores were as follows:

- **Cross-Validation Scores:** [0.7430, 0.7921, 0.8146, 0.7472, 0.7809]
- **Mean CV Score:** 0.78

These metrics indicate that my model performs reasonably well in predicting survival on the Titanic, with a balanced accuracy across different evaluation criteria.

5. Conclusion

In conclusion, this project allowed me to gain valuable insights into machine learning model development using a real-world dataset. The Decision Tree Classifier proved to be an effective choice for this classification task due to its interpretability and ease of use with categorical data. While my model achieved satisfactory performance metrics, there is always room for improvement. Future work could involve tuning hyperparameters, exploring other classification algorithms (such as Random Forest or Support Vector Machines), or incorporating additional features to enhance predictive accuracy.

Overall, this experience has deepened my understanding of machine learning processes from data preprocessing through model evaluation, setting a solid foundation for future projects in this domain.