Decision Tree Report

Introduction

A Decision Tree is a widely used machine learning model for classification and regression tasks. It operates by splitting the data into subsets based on feature conditions, forming a tree-like structure. This report evaluates the performance of a Decision Tree model applied to a given dataset.

Methodology

A Decision Tree follows a recursive partitioning approach based on the following principles:

- It selects the best feature for splitting using criteria such as Gini Impurity or Information Gain.
- The dataset is split iteratively until a stopping condition is met (e.g., maximum depth or minimum samples per leaf).
- Predictions are made by traversing the tree from the root to a leaf node.

Steps Taken:

1. **Data Preprocessing:**
   - Checked for missing values.
   - Conducted exploratory data analysis (EDA) to examine feature distributions and correlations.
   - Encoded categorical variables using one-hot encoding.
   - Scaled numerical features if necessary.
   - Split data into training (70%) and testing (30%) sets.
2. **Model Training:**
   - Implemented Decision Tree using sklearn.tree.DecisionTreeClassifier.
   - Experimented with different criterion functions (Gini and Entropy).
   - Tuned hyperparameters such as max_depth, min_samples_split, and min_samples_leaf.

3. **Performance Evaluation:**

   - Measured using Accuracy, Precision, Recall, and F1-Score.
   - Evaluated feature importance to understand the key contributors to predictions.

Decision Tree Model Analysis

No	Model Type	Parameters	Results
01	Default Configuration	-	0.9333
02	Limited Depth for Preventing Overfitting	criterion="squared_error", max_depth=5	0.9271
03	Minimum Samples per Split	min_samples_split=10	0.9112
04	Using the "friedman_mse" Criterion	criterion="friedman_mse"	0.9154
05	Restricting the Number of Features Considered for Splitting	max_features="sqrt"	-0.7754
06	Random State for Reproducibility	random_state=42	0.9122
07	Reducing Overfitting	max_leaf_nodes=20	0.9057
08	Criterion for Count Data	criterion="poisson"	0.9159

Conclusion

The Decision Tree Model Analysis highlights the impact of different hyperparameters on model performance. The **Default Configuration** achieved the highest result (**0.9333**), indicating that the dataset may not require excessive tuning for optimal performance. However, specific modifications provided valuable insights:

- **Limited Depth (max_depth=5)** slightly reduced accuracy (**0.9271**) but improved generalization, reducing overfitting risks.
- **Minimum Samples per Split (min_samples_split=10)** and **Random State (random_state=42)** maintained stable performance (**0.9112 - 0.9122**).
- **Using the "friedman_mse" Criterion** (**0.9154**) and **Poisson Criterion** (**0.9159**) showed potential but did not outperform the default.
- **Restricting Features (max_features="sqrt")** resulted in poor performance (**-0.7754**), indicating that using all features might be necessary for this dataset.
- **Reducing Overfitting (max_leaf_nodes=20)** slightly lowered accuracy (**0.9057**), suggesting that further fine-tuning is required.

Recommendations

1. **Use Default Configuration** as the primary model since it performed best (**0.9333**).
2. **Limit tree depth (max_depth=5)** for better generalization, especially if overfitting is a concern.
3. **Avoid limiting features (max_features="sqrt")**, as it negatively impacted performance.
4. **Consider using "friedman_mse" or "poisson" criteria** for specialized cases but validate improvements with additional testing.
5. **Further tune min_samples_split, max_leaf_nodes, and max_depth** to balance accuracy and generalization.

Final Recommendation

The **Default Configuration** remains the best choice for optimal performance. However, if overfitting is a concern, **limiting depth (max_depth=5)** and fine-tuning **min_samples_split & max_leaf_nodes** could enhance model stability without significantly sacrificing accuracy.