Experiment/Practical 6 Decision Tree

Title: Implementation of Decision Tree for Classification

Aim: To implement and understand the Decision Tree algorithm for classification tasks and evaluate its performance on a given dataset.

Objective: Students will learn

- Understand the decision tree classification mechanism.
- Implement the decision tree algorithm on a dataset for classification.
- Evaluate the performance using relevant metrics such as accuracy, precision, recall, and F1-score.
- Visualize the decision tree structure and its decision boundaries.

Problem Statement

Use the given dataset(s) to demonstrate the application of the Decision Tree algorithm for classification. The task is to classify the data points into different classes based on the features and to understand how the decision tree algorithm splits the data at each node.

Stepwise Procedure / Algorithm

1. Decision Tree Overview

A Decision Tree is a supervised learning algorithm used for classification and regression. It recursively splits the dataset based on feature values, forming a tree structure where each node represents a decision based on an attribute.

2. Mathematical Formulation

Entropy

H(S)=−∑i=1cpilog⁡2piH(S) = -\sum_{i=1}^{c} p_i \log_2 p_iH(S)=−i=1∑cpilog2pi

where pip_ipi is the probability of class iii. Entropy measures impurity—lower entropy means purer nodes.

Information Gain

IG(S,A)=H(S)−∑v∈Values(A)|Sv||S|H(Sv)IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)IG(S,A)=H(S)−v∈Values(A)∑|S||Sv|H(Sv)

where AAA is the attribute used for splitting.

Gini Index

Gini(S)=1−∑i=1cpi2Gini(S) = 1 - \sum_{i=1}^{c} p_i^2Gini(S)=1−i=1∑cpi2

Lower Gini values indicate purer nodes.

3. Importance of Decision Trees

- Simple to understand and interpret
- Requires minimal data preprocessing
- Handles both numerical and categorical data
- Non-linear relationships between parameters do not affect performance

4. Applications of Decision Trees

- Medical diagnosis (predicting diseases)
- Financial risk analysis
- Customer segmentation in marketing
- Fraud detection in banking

5. Performance Metrics

- **Accuracy** = TP+TNTP+TN+FP+FN\frac{TP + TN}{TP + TN + FP + FN}TP+TN+FP+FNTP+TN
- **Precision** = TPTP+FP\frac{TP}{TP + FP}TP+FPTP
- **Recall** = TPTP+FN\frac{TP}{TP + FN}TP+FNTP
- **F1-score** = 2×Precision×RecallPrecision+Recall2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}2×Precision+RecallPrecision×Recall
- **Confusion Matrix**: Evaluates true positives, false positives, true negatives, and false negatives.
- **Cross-validation**: Splits data into multiple training/testing sets to prevent overfitting.

Input & Output

Input:

- **Dataset:**
  - **Iris Dataset** (from `dct.pdf`) with features like petal length, petal width, etc.
  - **Diabetes Dataset** (from `dct2.pdf`) with features like glucose, insulin, age, etc.
- **User Input:**
  - Parameters like `max_depth`, `min_samples_split`

Output:

- **Predictions:** Class labels for test data
- **Model Evaluation:** Accuracy, precision, recall, F1-score, confusion matrix
- **Visualizations:**
  - Decision Tree structure
  - Decision boundaries
  - Confusion matrix

Example Results from the Diabetes Dataset (dct2.pdf)

- **Confusion Matrix**

```
[[76, 33],[ 9, 36]]
```

- **Accuracy:** 72.72%

Conclusion

The Decision Tree model successfully classifies data by splitting nodes based on entropy or Gini index. Hyperparameters like `max_depth`, `min_samples_split`, and `min_samples_leaf` impact model performance by controlling overfitting and generalization.