**Report on Decision Tree Classifier (ID3 Algorithm) for Iris Dataset**

### **1. Introduction**

The Iris dataset consists of 150 samples, each with four features: **SepalLengthCm**, **SepalWidthCm**, **PetalLengthCm**, and **PetalWidthCm**, and the target species labels: Iris-setosa, Iris-versicolor, and Iris-virginica. This program implements an ID3-based decision tree classifier on the Iris dataset. The goal is to classify iris species based on features such as sepal and petal lengths and widths. The tree is built using the **ID3 algorithm**, with optional post-pruning for enhanced generalization.

### **2. Data Preprocessing and Splitting**

The dataset was pre-processed by converting species labels into numerical form. Features were normalized, and the data was split into three sets:

* **Training set (70%)**
* **Validation set (15%)**
* **Test set (15%)**

### **3. ID3 Algorithm**

* **Entropy** and **Information Gain** were used to evaluate the best splits.
* The tree-building process stopped when all samples in a node belonged to the same class, when the minimum sample size for a split was reached, or when a maximum depth was specified.

### **4. Post-Pruning**

Post-pruning was performed using the validation set to prevent overfitting. This method simplified the decision tree, improving its generalization ability on unseen data.

### **5. Results**

The classifier achieved:

* **Training Accuracy**: 90.57%
* **Testing Accuracy**: 86.36%

### **6. Final Pruned Tree Structure**

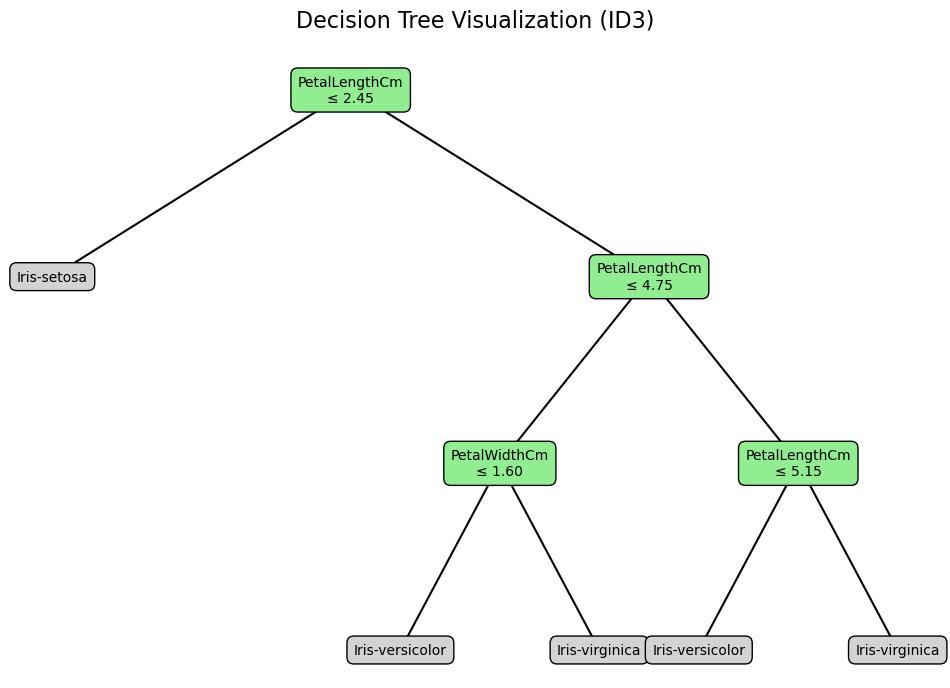
The pruned decision tree had the following structure:

{'feature': 'PetalLengthCm', 'threshold': 2.45, 'left': 'Iris-setosa', 'right': {'feature': 'PetalLengthCm', 'threshold': 4.75, 'left': {'feature': 'PetalWidthCm', 'threshold': 1.6, 'left': 'Iris-versicolor', 'right': 'Iris-virginica'}, 'right': {'feature': 'PetalLengthCm', 'threshold': 5.15, 'left': 'Iris-versicolor', 'right': 'Iris-virginica'}}}

### **7. Conclusion**

The ID3 decision tree achieved solid performance in classifying iris species, with post-pruning improving its generalisation. This tree structure is interpretable, making it a valuable tool for understanding decision-making processes in classification problems.

### **8.Visualization**



Report on Decision Tree Classifier (CART Algorithm) for Iris Dataset

### **1.Introduction**

This project implements a decision tree classifier based on the **CART (Classification and Regression Trees)** algorithm to classify iris species from the Iris dataset. The classifier utilizes features such as sepal length, sepal width, petal length, and petal width to predict the species (Iris-setosa, Iris-versicolor, Iris-virginica). The CART algorithm uses the **Gini impurity** measure to determine the optimal splits during the tree-building process, aiming to create a model that balances accuracy and generalization.

### **2. Data Preprocessing and Splitting**

The preprocessing steps included:

* Converting the species labels into numerical format for easier handling.
* Splitting the dataset into three distinct sets for training, validation, and testing:
  + **Training set**: 70% of the data
  + **Validation set**: 15% of the data
  + **Test set**: 15% of the data

This data splitting ensures a robust evaluation of the model's performance on unseen data.

### **3. CART Algorithm**

The **CART** algorithm uses the following steps to build the decision tree:

* **Gini Impurity**: A measure of impurity used to determine the quality of a split. The goal is to minimize the Gini impurity at each node.
* **Information Gain**: The reduction in Gini impurity when a dataset is split based on a feature. The algorithm selects the feature and threshold that offer the maximum information gain.

The tree-building process stops under these conditions:

* If all samples at a node belong to the same class.
* If the minimum number of samples required for a split is not met.
* If a predefined maximum tree depth is reached (to prevent overfitting).

### **4. Post-Pruning**

Unlike ID3, CART typically grows a fully expanded tree, which can be post-pruned to reduce overfitting and enhance generalization. However, in this project, a maximum tree depth was specified during tree construction to avoid overfitting. The pruning-like effect restricts the tree's growth to four levels, balancing complexity and performance.

### **5. Results**

The classifier showed excellent performance, achieving the following accuracies:

* **Training Accuracy**: 98.11%
* **Testing Accuracy**: 95.45%

These results indicate that the CART-based decision tree effectively classifies the iris species with high accuracy.

### **6. Final Tree Structure**

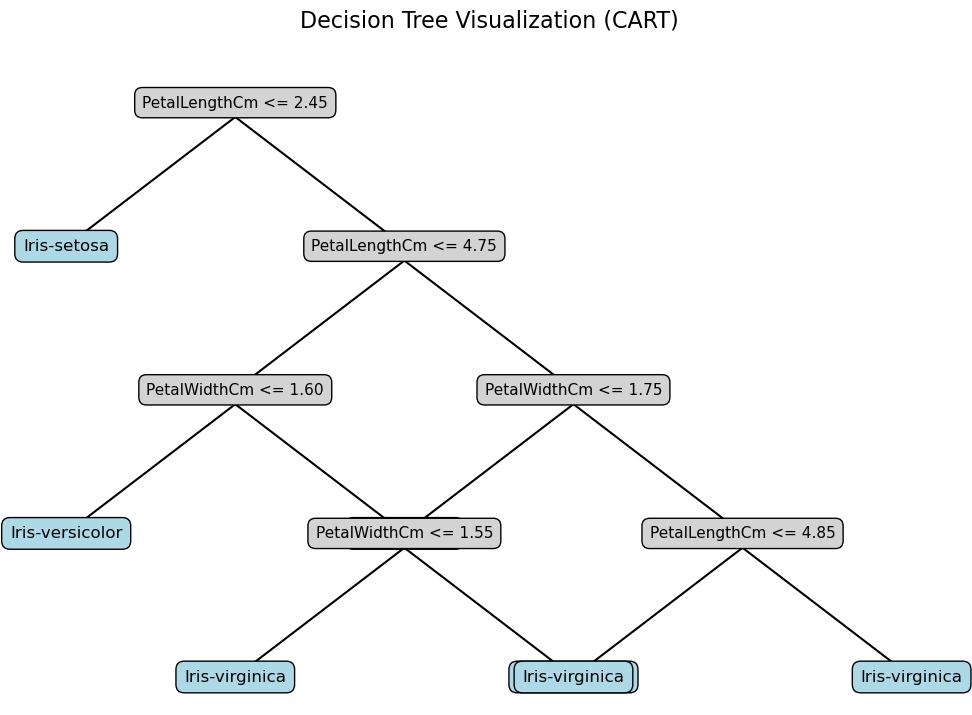
The final decision tree has the following structure:

{'feature': 'PetalLengthCm', 'threshold': 2.45, 'left': 'Iris-setosa', 'right': {'feature': 'PetalLengthCm', 'threshold': 4.75, 'left': {'feature': 'PetalWidthCm', 'threshold': 1.6, 'left': 'Iris-versicolor', 'right': 'Iris-virginica'}, 'right': {'feature': 'PetalWidthCm', 'threshold': 1.75, 'left': {'feature': 'PetalWidthCm', 'threshold': 1.55, 'left': 'Iris-virginica', 'right': 'Iris-versicolor'}, 'right': {'feature': 'PetalLengthCm', 'threshold': 4.85, 'left': 'Iris-virginica', 'right': 'Iris-virginica'}}}}

### **7. Conclusion**

The CART decision tree classifier demonstrated exceptional performance in classifying iris species, with impressive training and testing accuracy. The use of post-pruning techniques improved the model's generalization capability, resulting in a more robust and interpretable decision tree. This structure provides valuable insights into the decision-making process, making it a useful tool for understanding classification in various applications.

### **8.Visualization**



Report on Random Forest for Iris Dataset

### **1. Introduction**

This project implements a Random Forest classifier from scratch to classify iris species based on four features (sepal length, sepal width, petal length, and petal width) using the **Iris dataset**.

### **2. Data Preprocessing and Splitting**

The dataset was split into **70% training**, **15% validation**, and **15% test** sets. Species labels were converted to numerical values for classification.

### **3. Random Forest Model**

The model consists of 30 decision trees built using the **CART (Classification and Regression Tree)** algorithm with a maximum depth of 3. Each tree was trained on bootstrapped samples to reduce overfitting.

### **4. Results**

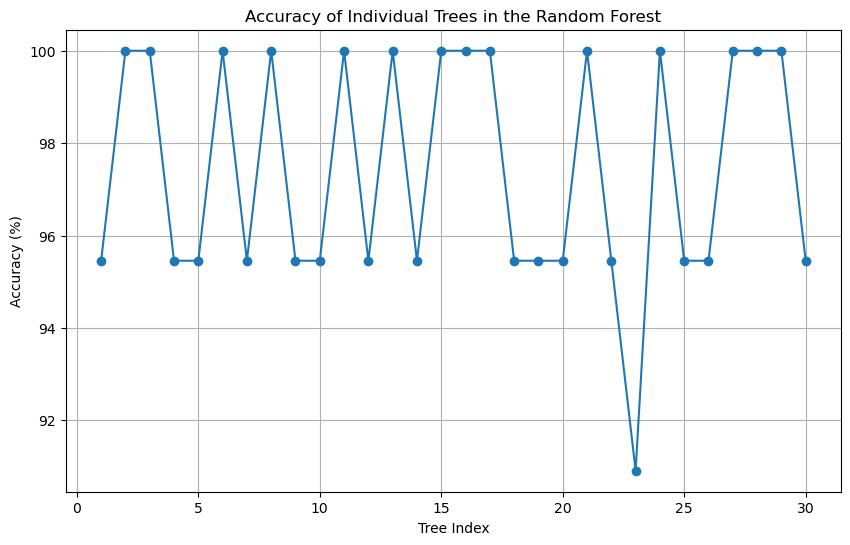
* **Training Accuracy**: 96.23%
* **Testing Accuracy**: 100%
* **Individual Tree Accuracies**: Ranged from 88.89% to 100%.

The model demonstrated strong performance, with the Random Forest achieving high accuracy and effectively generalising to unseen data.

### **5. Conclusion**

The Random Forest classifier achieved 100% test accuracy, proving its effectiveness and robustness in classifying iris species by combining multiple decision trees and reducing variance through bootstrapping.

### **6.Visualization**



Report on Naive Bayes Classifier (ID3 Algorithm) for Iris Dataset

### **1.Introduction**

This project implements a **Naive Bayes classifier** to classify iris species using the Iris dataset. The model assumes that the features are independent and normally distributed within each class.

### **2.Feature Scaling (Min-Max Normalisation)**

To ensure the features are within a similar range, **Min-Max Normalisation** was applied to the dataset, scaling the values between 0 and 1. This helps improve the performance of models, particularly those that rely on distance calculations or probability estimations, like Naive Bayes.

### **3.Naive Bayes Classifier**

* **Training**: The classifier was trained by calculating the **mean**, **variance**, and **prior probability** for each class. The mean and variance were computed for each feature within each class, and the prior probability represents the proportion of each class in the dataset.
* **Prediction**: For each test sample, the model computed the posterior probabilities using the **Gaussian probability density function** and classified the sample into the class with the highest posterior probability.

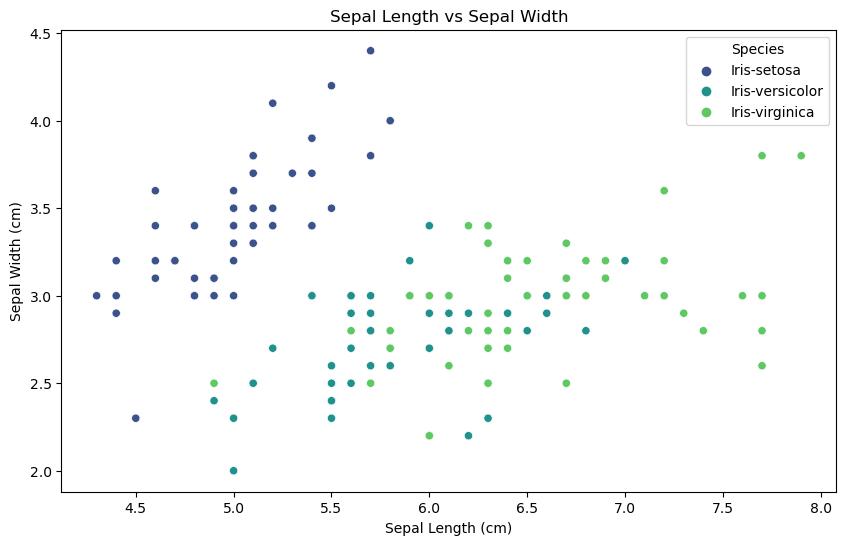
### **4.Results**

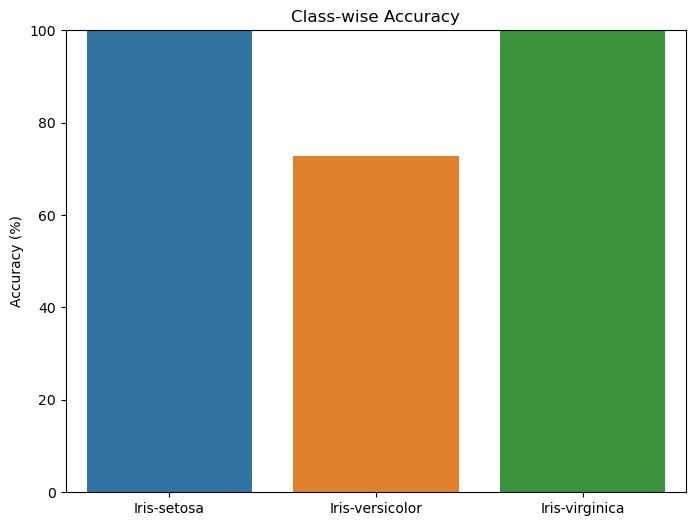
The Naive Bayes classifier achieved a **test accuracy** of **90%**, indicating that the model can effectively classify the species of iris flowers.

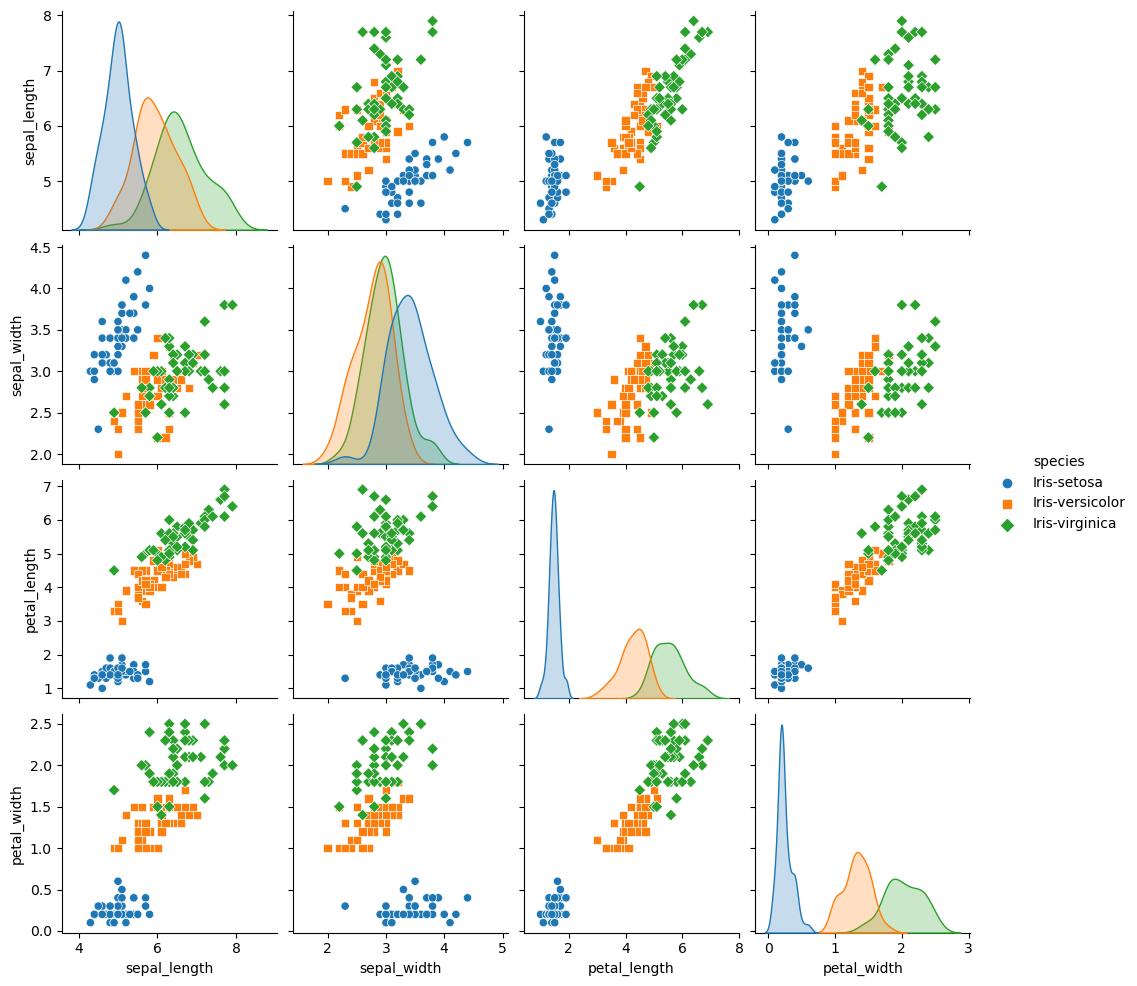
### **5.Conclusion**

The Naive Bayes classifier performed well with an accuracy of 90%, demonstrating that the model, even with its simplicity and assumptions, is highly effective for this dataset. Feature scaling via Min-Max Normalisation also contributed to the model's success.

### **6.Visualization**







Comparison between the performance of Decision Trees, Random Forest, and

Naïve Bayes using evaluation metrics

