IRIS CASE STUDY

Frilles, Roljohn C. rjthehokage@gmail.com	Macaraeg, Paul Angelo O. paulmacaraeg24@gmail.com	Manansala, Roan P. manansalaroan11@gmail.com	Manucom, Precious Grace Deborah deborahgrace0118@gmail.com

Executive Summary

This case study explores the application of three classic machine learning algorithms: Decision Trees, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) in classifying the Iris flower based on four physical characteristics. Using the standard Iris dataset, the goal was to compare the performance of these models in terms of their accuracy and generalization capability. Both KNN and Linear SVM achieved perfect test accuracy at 100%, confirming the dataset's strong linear separability, while pruned Decision Trees (C5.0) demonstrated the highest cross-validation stability and the simplest, most interpretable model structure. The results underscore that algorithm selection should be guided not only by raw accuracy but also by practical considerations such as model complexity, stability across data splits, and the need for interpretability depending on the specific application context.

1 Introduction

In the field of machine learning, one of the most supervised learning tasks is the ability to identify or classify data into distinct categories (Hastie et al., 2009). The main goal of classification is to predict the categorical label of a new input data based on the patterns learned from a labeled training dataset (Bishop, 2006). These types of problems are widely used in real-world applications, from medical diagnosis, spam detection, and even image recognition.

A classic example of a classification problem is the prediction of Iris flower species based on measurable characteristics such as petal and sepal dimensions. Each flower must be assigned to one of several predefined categories based on its features. This case study focuses on building models that can perform this classification, highlighting the challenges and considerations when choosing which of them is the best for each case.

The dataset used for this study is the famous Iris dataset. It is one of the oldest and most referenced datasets in machine learning and statistics, meaning you can see it in almost every machine learning example. The Iris dataset contains 150 records, each representing an individual flower sample. Every sample is described by four numerical features: sepal length, sepal width, petal length, and petal width. The flowers are categorized into three species: Iris Setosa, Iris Versicolor, and Iris Virginica.

One of the reasons the Iris dataset remains a popular benchmark is its simplicity combined with the presence of slight overlaps between classes, which makes the classification task easy to understand but still slightly challenging. The dataset is also balanced, with 50 samples for each class, and contains no missing values, making it ideal for initial experimentation with classification algorithms. Because of these characteristics, the Iris dataset provides a controlled environment to explore different machine learning techniques and understand their strengths, limitations, and behavior when facing a multi-class classification task.

In this study, three machine learning models were selected for implementation and comparison: Decision Trees, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). Each model represents a different approach to classification, offering insights into how various algorithms make decisions based on data structures and feature distributions.

Decision Trees

A decision tree is a supervised learning algorithm that has a tree-like structure where the internal nodes represent the attribute tests, the branches are attribute values and the leaf nodes are the final decisions or predictions. It is able to classify an instance by traversing the tree starting from the root node and going through each of the branch nodes until it can come to a decision on the leaf nodes. They are easy to interpret and visualize, which makes them intuitive, especially for beginners. However, without proper pruning, they can easily overfit the training data or create unnecessary branches (Hastie et al., 2009).

K-Nearest Neighbor

K-Nearest Neighbors (KNN) is a simple way to classify things by looking at what's nearby (Cover & Hart, 1967). There is no explicit model trained; instead, the training data is stored and directly compared during prediction. KNN is simple and effective for datasets where similar instances naturally group together, but it can become computationally expensive as the size of the dataset grows. In addition, the KNN's performance is highly sensitive to the choice of k (the number of neighbors) and the distance metric used (Hastie et al., 2009).

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that is used in classification tasks for high-dimensional spaces (Cortes & Vapnik, 1995). SVMs aim to find the optimal decision boundary (also called a hyperplane) that separates the data points of different classes. SVMs maximize the margin between the closest points of different classes (called support vectors) and the decision boundary (Hastie et al., 2009).

While each of the models has its strengths, comparing them is important because they each approach the classification problem from a different perspective; it's like seeing the problem through different lenses (James et al., 2021). Decision trees try to break down the problem into a series of simple, understandable rules, but they can get caught in small details if not controlled properly. K-Nearest Neighbors make decisions based purely on how near they are to each other, but it can be confusing if the neighbors are too noisy or crowded. Meanwhile, the SVM creates a definite partition between the vectors, but it is hard to consider the hyperparameters for tuning.

By comparing these three algorithms, we can see how different modeling strategies handle the same classification challenge, and we can learn when simplicity, closeness, or separation gives the better solution, depending on the data's structure. This comparison not only highlights their individual capabilities but also teaches valuable lessons about model selection in real-world machine learning problems.

1.1 Objectives

The main goal of this study is to explore the different machine learning algorithms: Decision Trees, KNN, and SVM. Specifically, the study aims to:

- To implement Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) models on the Iris dataset.
- To evaluate and compare the models based on accuracy and classification performance.
- To analyze how each algorithm approaches the classification task differently.
- To identify the strengths and limitations of each model.
- To draw insights on the importance of model selection in supervised learning problems.

1.2 Scope and Limitations

This study focuses on the classification of Iris flower species using three machine learning algorithms: Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The models are trained and evaluated exclusively on the standard Iris dataset, which consists of four numerical features and three target classes. The scope includes implementing, training, testing, and comparing these models based on accuracy and classification behavior.

However, the study is limited to the Iris dataset, which is relatively small, balanced, and clean compared to real-world datasets. As a result, the model performance observed here may not be directly generalized to more complex or noisy datasets. In addition, only default or basic tuning strategies are applied, and more advanced optimization techniques such as ensemble methods, feature engineering, or deep learning are beyond the scope of this work.

2 Case Context and Data Description

2.1 Problem Statement

This study addresses the need to understand and compare three popular algorithms: Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) to solve the same classification problem using the Iris dataset. The core problem is not just to achieve high accuracy, but to observe and analyze how each model interprets the data differently, handles decision boundaries, and balances simplicity with predictive power. Through this comparison, the study aims to build a deeper understanding of model behavior and guide more informed algorithm selection in future classification tasks.

2.2 Dataset Overview

The dataset used in this study is the Iris dataset, one of the most famous and widely studied datasets in the field of machine learning and statistics. It was first introduced by Ronald

A. Fisher in 1936 as part of his research in discriminant analysis (Fisher, 1936). The dataset has since become a classic benchmark for evaluating classification algorithms due to its simplicity, balanced class distribution, and the intuitive nature of its features.

The Iris dataset consists of 150 samples, each representing an individual Iris flower. Each flower is described using four numerical features: sepal length, sepal width, petal length, petal width.

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	target	species
0	5.1	3.5	1.4	0.2	Iris-setosa	0	setosa
1	4.9	3.0	1.4	0.2	Iris-setosa	0	setosa
2	4.7	3.2	1.3	0.2	Iris-setosa	0	setosa
3	4.6	3.1	1.5	0.2	Iris-setosa	0	setosa
4	5.0	3.6	1.4	0.2	Iris-setosa	0	setosa

Image 1. Describes the important features of the Iris Dataset

The target variable or (label) classifies each flower into one of three species: Iris Setosa, Iris Versicolor, Iris Virginica. Each class contains exactly **50 samples**, making the dataset perfectly balanced across categories. This balance ensures that no class is overrepresented or underrepresented, making evaluation straightforward without requiring additional techniques like resampling.

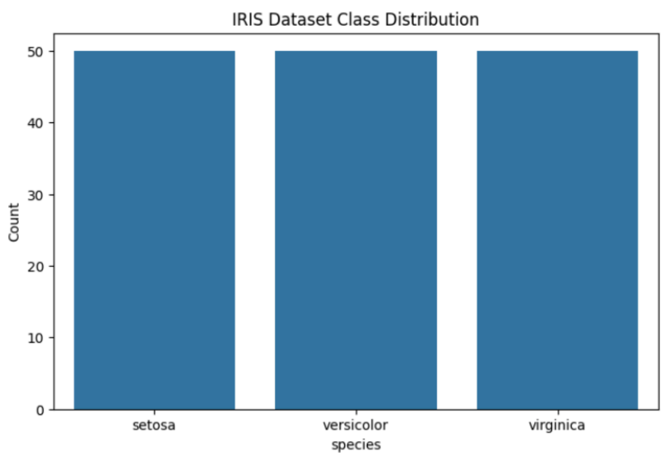


Image 2. Frequency distribution of each class of the dataset.

In terms of feature distribution, one notable aspect is that the Setosa class is linearly separable from the other two species when plotted in feature space, particularly using petal length and width. However, Versicolor and Virginica show some overlap, making their classification slightly more challenging. This slight complexity makes the dataset an excellent playground for testing how different algorithms handle clean separations versus ambiguous cases.

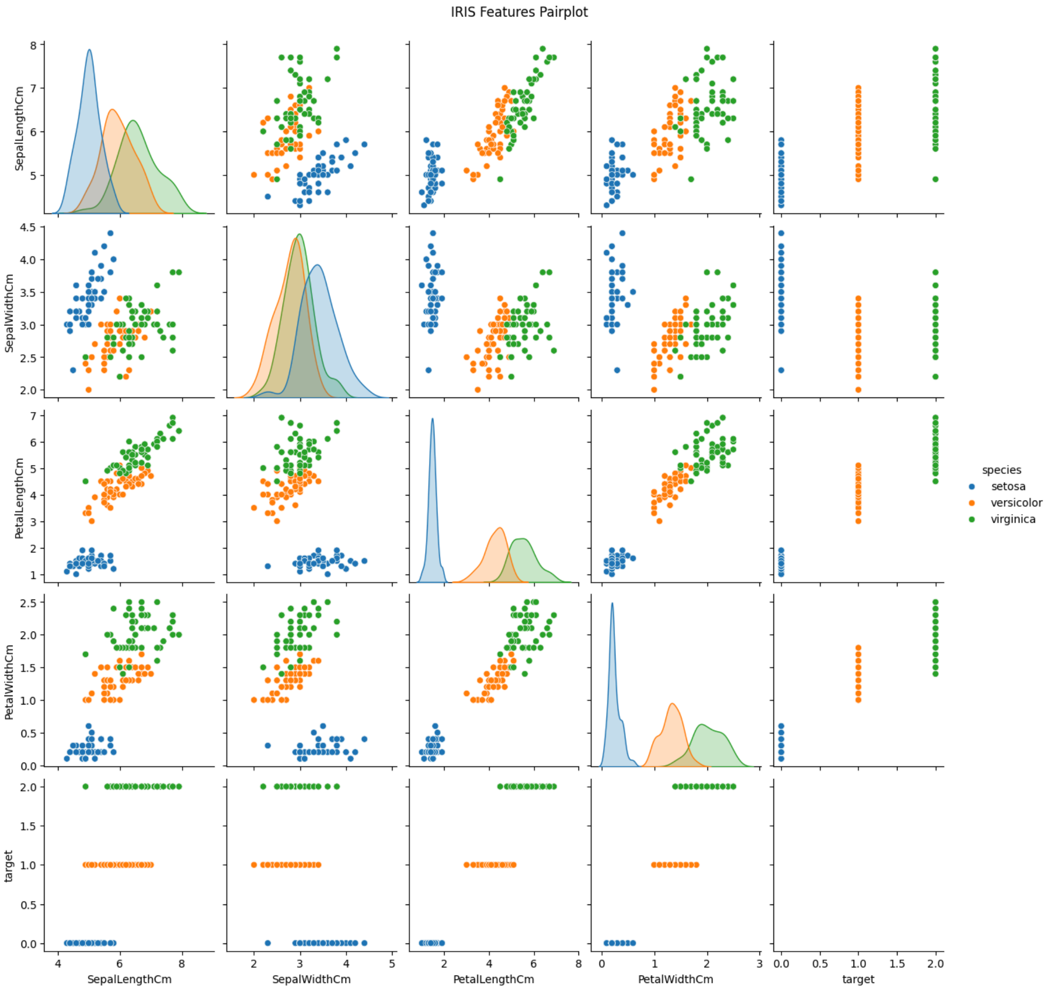


Image 3. Feature Plot that shows the overlap between the petal width and petal length between Versicolor and Virginica

Another notable feature of this dataset is that, because it is widely used, the dataset is already complete and clean. There are no missing values, all features are continuous and numerical, and the features are on a similar scale because the sizes are not far off from each other. Given the size, structure, and characteristics, the Iris dataset provides an ideal setting for comparing these machine learning models. It is simple enough to allow experimentation, but it is rich enough to reveal the difference between the models that will be tested.

3 METHODOLOGY

3.1 Methodology

In order to analyze the Iris dataset and classify them into their respective species, this study followed a typical data mining workflow that included data loading, data preprocessing, model training, model evaluation, and performance comparison and analysis.

1. **Data Loading**

The first step done is to load the dataset in order to start the process. It was loaded directly from the scikit-learn's dataset module. The data consists of the feature variables and the target variables.

2. **Data Preprocessing**

The next step is to prepare the dataset through preprocessing steps, in which the features and target labels are separated. Then, the dataset was split into training and testing data to evaluate the model. In this case, the data was separated using the 70-30 split for all implementation purposes.

3. **Model Training**

Following this is the training phase, where the three different machine learning models were trained separately in different notebooks.

- Decision Tree Classifier was trained by finding the optimal splits in the feature space to create a hierarchical decision model. For the decision tree, three models were used to compare: CART (Breiman et al., 1984), ID3 (Quinlan, 1986), and C5.0 (based on Quinlan, 1993).
- K-Nearest Neighbors (KNN) stored the training data and prepared to classify new instances based on proximity.
- Support Vector Machine (SVM) was trained to find the best hyperplane that maximizes the margin between classes.

Each model learned from the training set using their own internal learning strategy.

4. **Model Evaluation**

After the training, the models were evaluated on the unseen test set. They were measured by finding the accuracy, precision, recall, F1-score, and support. In addition, the confusion matrices were generated to observe the performance of each class.

5. **Performance Comparison**

Finally, the results of all three models were compared to each other in order to assess the accuracies, determine the strengths and weaknesses of each model based on how they approach the classification, and find insights into which of the models was most effective for the dataset and why.

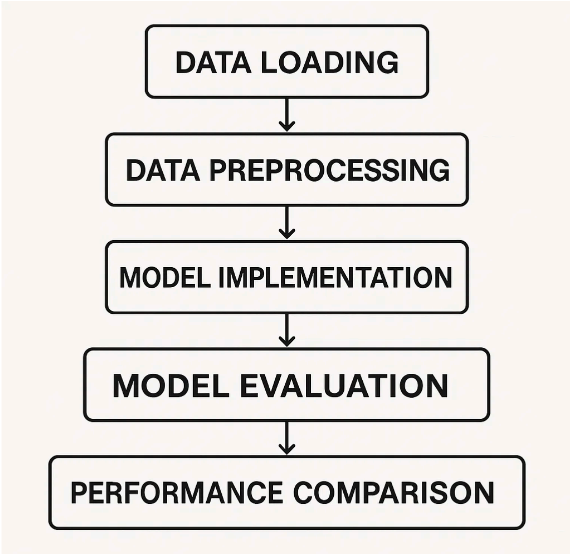


Image 4. Overview of the Methodological Steps Followed

3.2 Tools and Technologies

This study was implemented using Jupyter Notebook, which is an interactive development environment that allows writing, testing, and documenting code in one place. The notebook is primarily encoded in Python with markdown cells to further explain each step of the process.

The following Python libraries were used throughout the project:

- **Pandas:** pandas is used for data manipulation and preprocessing tasks such as loading the datasets into the DataFrames, selecting feature columns, and organizing target labels. It was mainly used in the Decision Tree and SVM notebooks.
- **Numpy:** used for numerical operations and array handling, especially for model evaluation and graph plotting.
- **Sci-kit Learn** (sklearn) serves as the main machine learning library across all notebooks. It provides the functionalities for:
  - Building Models: `DecisionTreeClassifier`, `SupportVectorClassifier (SVC)`
  - Splitting the data into training and testing sets `(train_test_split)`
  - Performing cross-validation with cross_val_score
  - Evaluating models using metrics like `accuracy_score`, `confusion_matrix and classification_report.`
  - Visualizing the model performance through `confusion_matrix`
- **Matplotlib** is used for generating various visualizations, such as plotting confusion matrices, learning curves, and hyperparameter analysis graphs. It is present in all notebooks to improve the interpretability of results.

3.3 Process Description

This study follows the CRISP-DM (Cross Industry Standard Process for Data Mining) model, which provides a structured approach to solving data-focused problems. This methodology follows six main phases, each of which was applied throughout the study as follows:

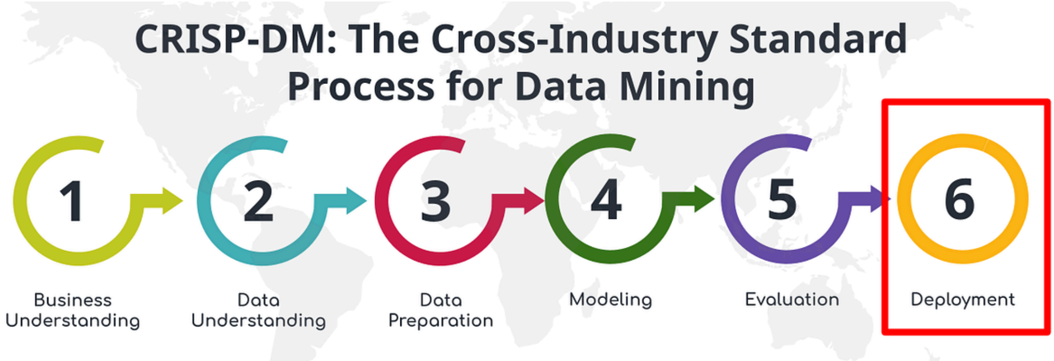


Image 5. Summary of CRISP-DM Implementation

Business Understanding

The first step is to define the main objective of the project, which is to evaluate and compare different machine learning models — specifically Decision Trees, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) — on the Iris dataset. The goal is to determine the relative strengths, weaknesses, and ideal use cases of each model in terms of prediction accuracy, stability, and interpretability.

Data Understanding

In this phase, the Iris dataset was examined. The dataset contains 150 samples evenly distributed across three species: Iris-setosa, Iris-versicolor, and Iris-virginica. Each sample includes four numeric features: sepal length, sepal width, petal length, and petal width. Initial exploration confirmed that the dataset is clean, complete, and well-balanced, making it an ideal candidate for benchmarking basic classification algorithms. Visualizations and basic statistical analyses were used to familiarize oneself with the feature distributions and relationships.

Data Preparation

Next is the data preparation, in this phase, the dataset was organized in order to be suitable for modeling. Features and labels were separated, and the data was split into training and testing sets using a 70-30 ratio. There is no further cleaning or imputation necessary because the Iris dataset was already prepared to be suitable for machine learning models.

Modeling

In this phase, the three types of machine learning were developed. Decision Trees were implemented using different splitting strategies — CART (Gini impurity), ID3 (Entropy), and

C5.0 (Entropy with pruning). The Support Vector Machine was configured with a linear kernel to separate the classes based on maximizing margins. For K-Nearest Neighbors, a straightforward distance-based classification approach was used. Each model was tuned using basic hyperparameters, and cross-validation was incorporated where appropriate to assess initial performance during training.

Evaluation

This phase was focused on assessing the models' predictive capabilities and analyzing their comparative performance. Evaluation metrics such as accuracy, confusion matrices, precision, recall, and F1-score were prepared to measure each model's strengths and weaknesses thoroughly. Cross-validation was applied particularly to decision tree models to examine their stability across different data splits.

Deployment (Performance Comparison)

Although this study was not deployed in a real setting, the results of each machine algorithm model were compared in order to analyze their performance. In order to finalize this study, it has a final discussion that summarizes key takeaways, limitations, and recommendations for future work.

4 Results and Analysis

This section details the performance evaluation of the three primary machine learning algorithms implemented for Iris flower classification: Decision Trees, k-Nearest Neighbors (KNN), and Support Vector Machines (SVM). Within the Decision Tree category, three specific variants were also compared: CART (Gini), ID3 (Entropy), and an approximation of C5.0 (Entropy with pruning parameters). The Iris dataset was divided into training and testing subsets to evaluate the models, with stratified sampling employed to ensure representative class distributions in both. Key performance metrics, including accuracy, precision, recall, F1-score, and confusion matrices, were analyzed for each model based on its predictions on the test set. Additionally, cross-validation techniques were utilized where appropriate to further assess model stability and generalization capabilities across different data partitions.

I. Decision Trees

A. CART (Classification and Regression Trees - Gini Impurity)

The CART model, utilizing the Gini impurity criterion for splitting, was trained and evaluated.

- **Performance Metrics:** The model achieved a **test accuracy of 93.33%**, correctly classifying 42 out of the 45 test samples. The detailed classification report is shown in Table 1.

Classification Report - CART (Gini)

	Setosa	Versicolor	Virginica	Macro Average
Precision	1.0	1.0	0.83	0.94
Recall	1.0	0.8	1.0	0.93
F1-Score	1.0	0.89	0.91	0.93

Table 1. Classification Report for CART (Gini) Model

The CART model demonstrates perfect precision (1.00) and recall (1.00) for the Setosa class, indicating it perfectly identified and classified all Setosa samples. It also achieved perfect recall (1.00) for Virginica, meaning all true Virginica samples were correctly identified, although its precision was lower (0.83), suggesting some other flowers were incorrectly classified as Virginica. For Versicolor, precision was perfect (1.00), but recall was lower (0.80), indicating that while all flowers predicted as Versicolor were Versicolor, 20% of the actual Versicolor samples were missed. The F1-scores reflect this, being perfect for Setosa (1.00) and high for Versicolor (0.89) and Virginica (0.91).

- **Confusion Matrix**

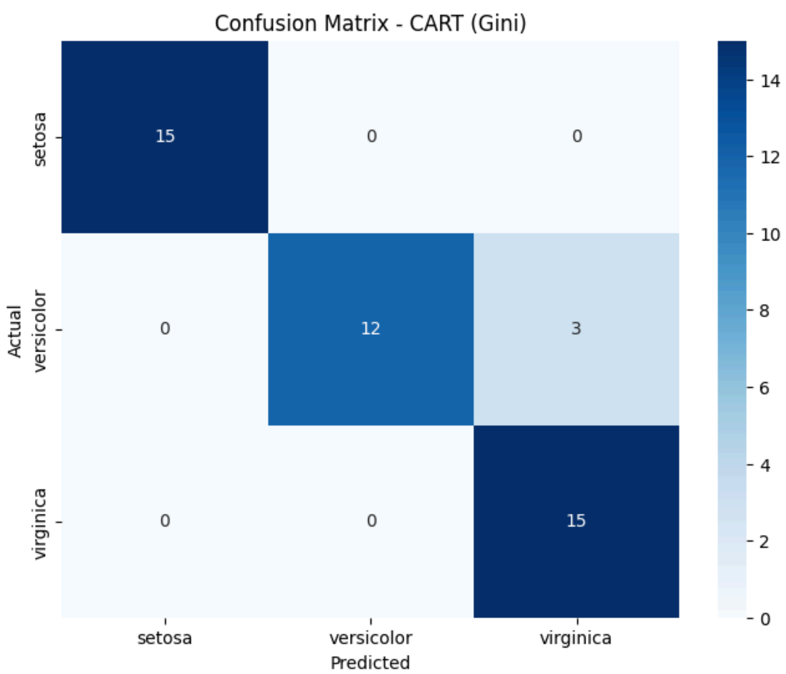


Figure 1. Confusion Matrix for CART (Gini) Model on the Test Set

Figure 1 confirms the classification report findings. All 15 *Setosa* samples were correctly classified. For *Versicolor*, 12 out of 15 samples were correctly classified, while 3 were misclassified as *Virginica*. All 15 *Virginica* samples were correctly classified. The errors made by the CART model were solely in misclassifying *Versicolor* as *Virginica*.

- **Model Structure:** As shown in Figure 2, the trained CART tree had a maximum depth of 5, consisting 15 nodes and 8 leaf nodes. The initial split was based on `PetalLengthCm <= 2.45`, effectively isolating the *Setosa* class.

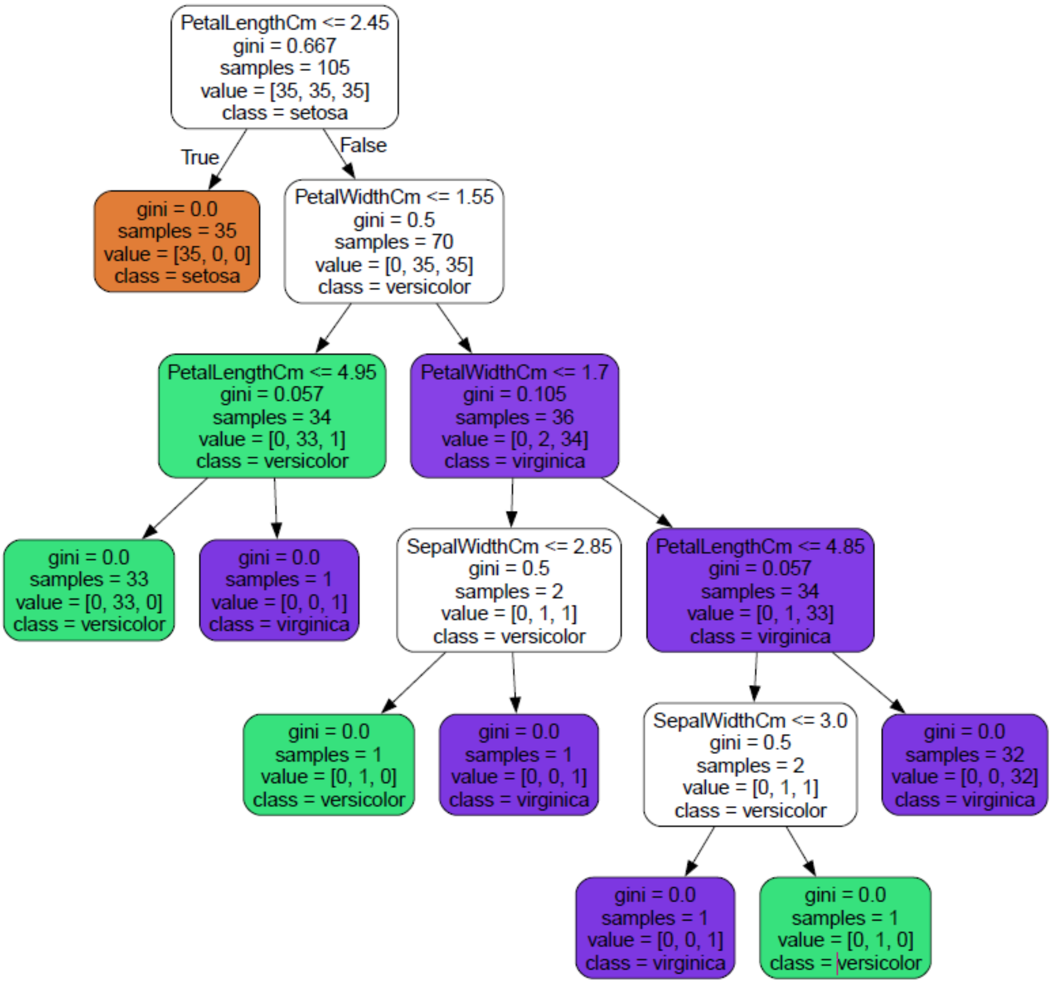


Figure 2. CART Decision Tree Structure

B. ID3 (Iterative Dichotomiser 3 - Entropy)

The ID3 model, using entropy and information gain as the splitting criterion, was evaluated next.

- **Performance Metrics:** This model achieved a lower **test accuracy of 88.89%**, correctly classifying 40 out of 45 test samples. The classification report is presented in Table 2.

Classification Report - ID3 (Entropy)

	Setosa	Versicolor	Virginica	Macro Average
Precision	1.0	0.81	0.86	0.89
Recall	1.0	0.87	0.8	0.89
F1-Score	1.0	0.84	0.83	0.89

Table 2. Classification Report for ID3 (Entropy) Model

Similar to CART, ID3 perfectly classified the *Setosa* class (1.00 precision and recall). Performance for *Versicolor* (0.81 precision, 0.87 recall, 0.84 F1) and *Virginica* (0.86 precision, 0.80 recall, 0.83 F1) was reasonably good but slightly lower and more balanced in terms of precision/recall trade-offs compared to CART. The overall accuracy and weighted average F1-score (0.89) are lower than CART's.

- **Confusion Matrix**

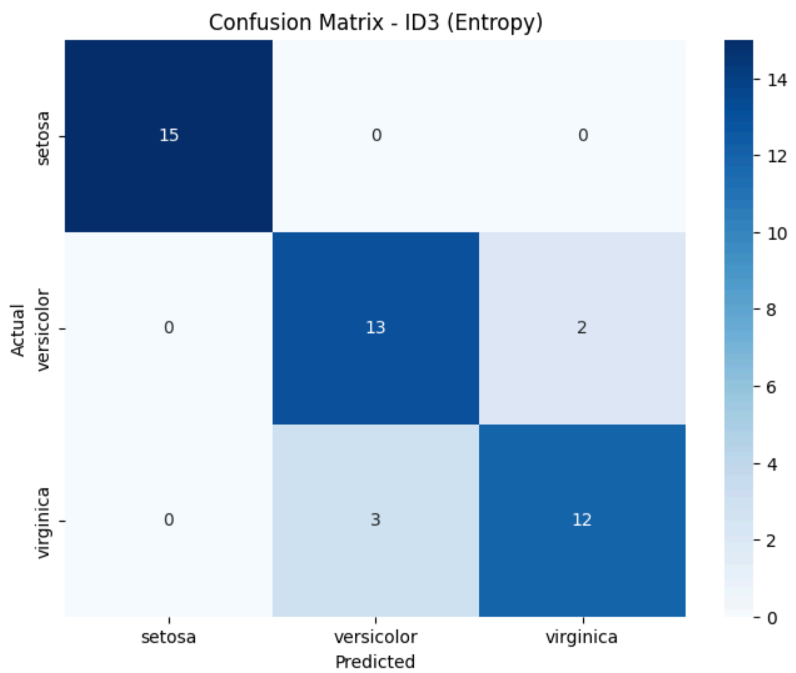


Figure 3. Confusion Matrix for ID3 (Entropy) Model on the Test Set

Figure 3 shows perfect classification for *Setosa* (15/15). Unlike CART, ID3 made errors in both directions between the other two classes: 2 *Versicolor* samples were misclassified as

Virginica, and 3 *Virginica* samples were misclassified as *Versicolor*. This indicates slightly more confusion between these two classes compared to the Gini-based model.

- **Model Structure:** As shown in Figure 4, the ID3 tree was the deepest, reaching a depth of 6. It contained the same number of nodes (15) and leaves (8) as the CART model, suggesting a slightly more complex structure to achieve slightly lower test accuracy.

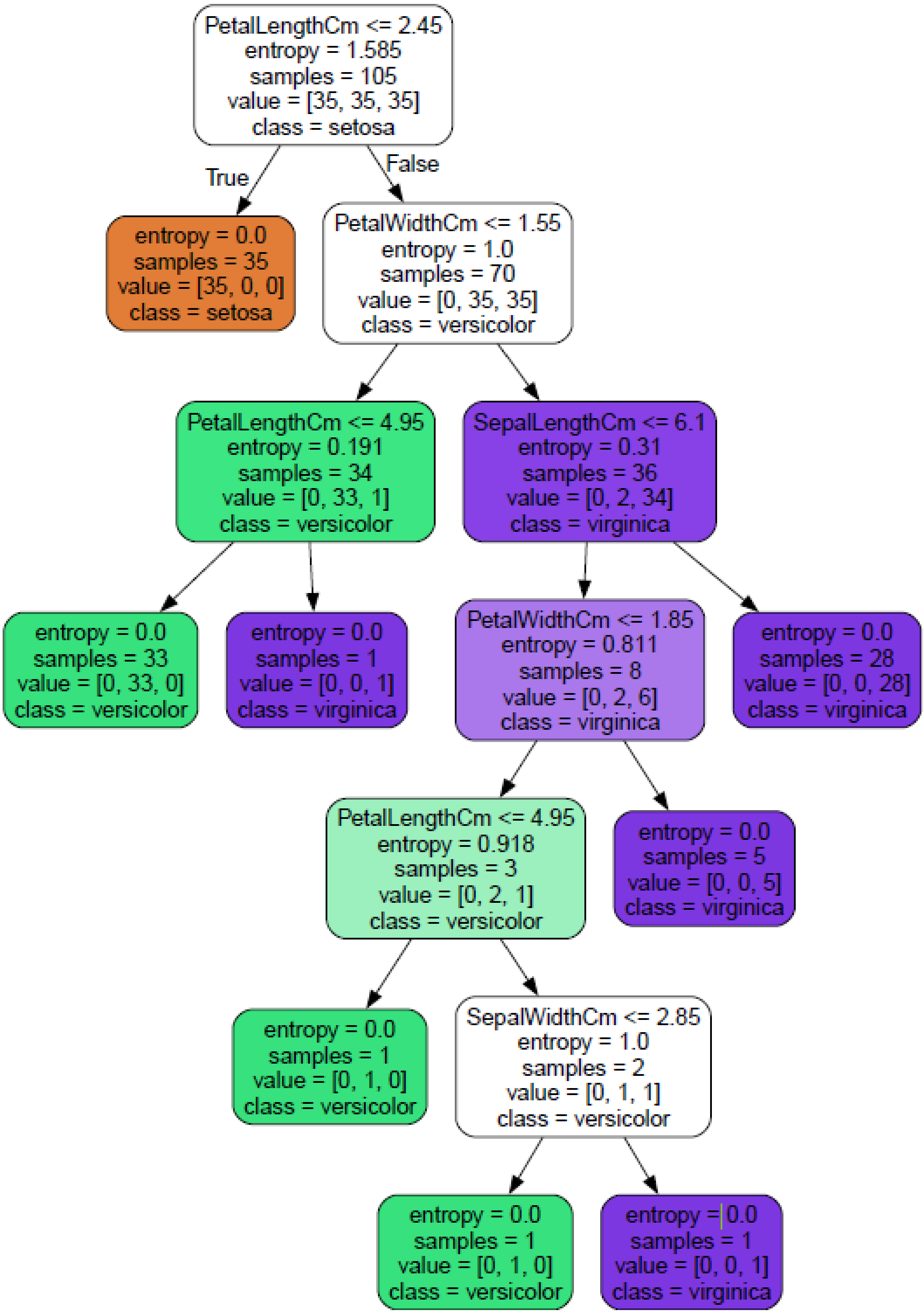


Figure 4. ID3 Decision Tree Structure

C. C5.0 (Approximation - Entropy with Pruning)

An approximation of the C5.0 algorithm was implemented using entropy and setting pruning-related parameters (`min_samples_split=5`, `min_samples_leaf=2`).

- **Performance Metrics**: This pruned model yielded the lowest **test accuracy of 84.44%**, correctly classifying 38 out of 45 samples. Table 3 shows the classification report.

Classification Report - C5.0 (Approximation)

	Setosa	Versicolor	Virginica	Macro Average
Precision	1.0	0.72	0.83	0.85
Recall	1.0	0.87	0.67	0.84
F1-Score	1.0	0.79	0.74	0.84

Table 3. Classification Report for C5.0 (Approximation) Model

The *Setosa* class was again perfectly classified. However, performance on the other classes degraded further. *Versicolor* had lower precision (0.72) but maintained high recall (0.87). *Virginica* suffered significantly in recall (0.67), meaning one-third of the actual *Virginica* samples were misclassified, although its precision remained relatively high (0.83). The overall accuracy and F1-score (0.84) were the lowest among the three models.

- **Confusion Matrix**

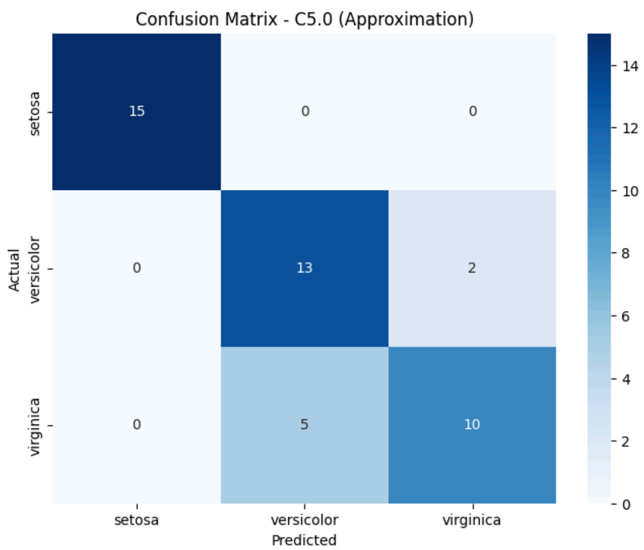


Figure 5. Confusion Matrix for C5.0 (Approximation) Model on the Test Set

Figure 5 confirms the perfect *Setosa* classification. The model misclassified 2 *Versicolor* samples as *Virginica*. Notably, it misclassified 5 *Virginica* samples as *Versicolor*, showing the highest error rate, particularly struggling with correctly identifying Virginica. This model made 7 misclassifications in total.

- **Model Structure:** As shown in Figure 6, reflecting the pruning parameters, the C5.0 approximation produced the simplest tree: depth 4, 11 nodes, and 6 leaves.

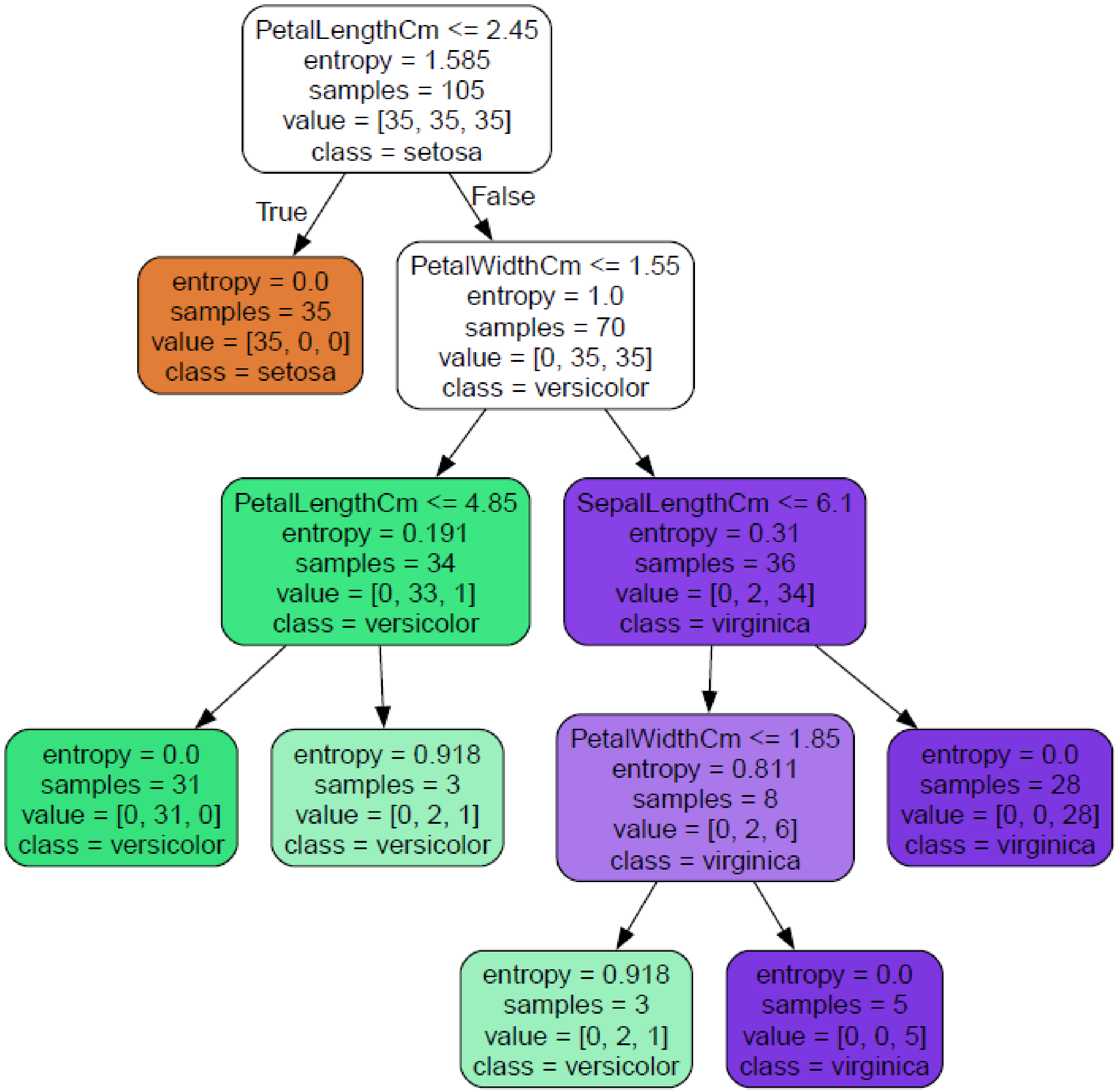


Figure 6. C5.0 Decision Tree Structure

Comparing the three decision tree models reveals important trade-offs between accuracy, complexity, and stability.

While CART achieved the highest accuracy on the single test split (93.3%), 10-fold cross-validation provided a more accurate estimate of generalization performance.

Cross-validation results (10-fold):

	CART (Gini)	ID3 (Entropy)	C5.0 (Approximation)
Fold 1	1.000000	1.000000	1.000000
Fold 2	0.933333	0.933333	0.933333
Fold 3	1.000000	1.000000	1.000000
Fold 4	0.933333	0.933333	0.933333
Fold 5	0.933333	0.933333	0.933333
Fold 6	0.866667	0.866667	0.933333
Fold 7	0.933333	0.933333	0.933333
Fold 8	0.933333	0.933333	0.933333
Fold 9	1.000000	1.000000	1.000000
Fold 10	1.000000	1.000000	1.000000
Mean	0.953333	0.953333	0.960000
Std	0.042687	0.042687	0.032660

Image 6. Comparative Performance Metrics

As shown in Image 6, cross-validation revealed that the C5.0 approximation, despite having the lowest test set accuracy, achieved the highest average accuracy (96.0%) across the 10 folds and exhibited the lowest standard deviation (0.033), indicating the most stable and consistent performance across different subsets of the data. CART and ID3 had identical mean CV accuracy (95.3%) and stability. This suggests that C5.0's simpler, pruned structure might generalize slightly better on average, even if it performed worse on the specific 70/30 split used for initial testing.

The structural complexity varied significantly, as seen in Figure 7.

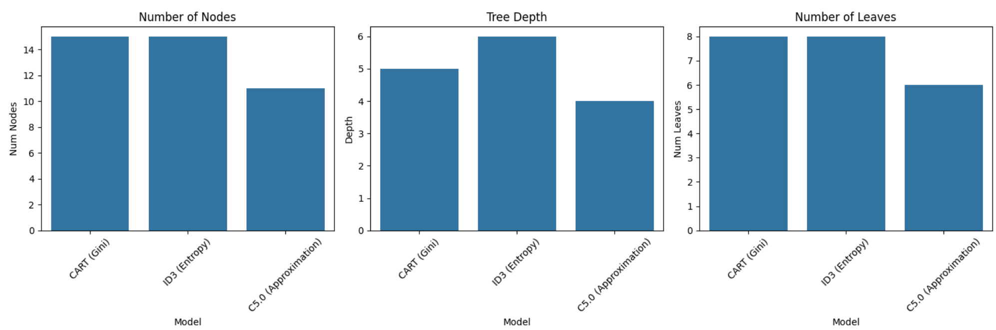


Figure 7. Comparison of Tree Complexity Metrics

C5.0 produced the simplest tree (shallowest, fewest nodes/leaves) due to pruning. ID3 produced the deepest tree, while CART was intermediate in depth but matched ID3 in node/leaf count. The simplicity of C5.0 likely contributes to its cross-validation stability but potentially leads to underfitting on specific data splits (like the test set), whereas ID3's depth might risk overfitting. CART strikes a balance.

All models agreed on the relative importance of the features, primarily relying on petal dimensions.

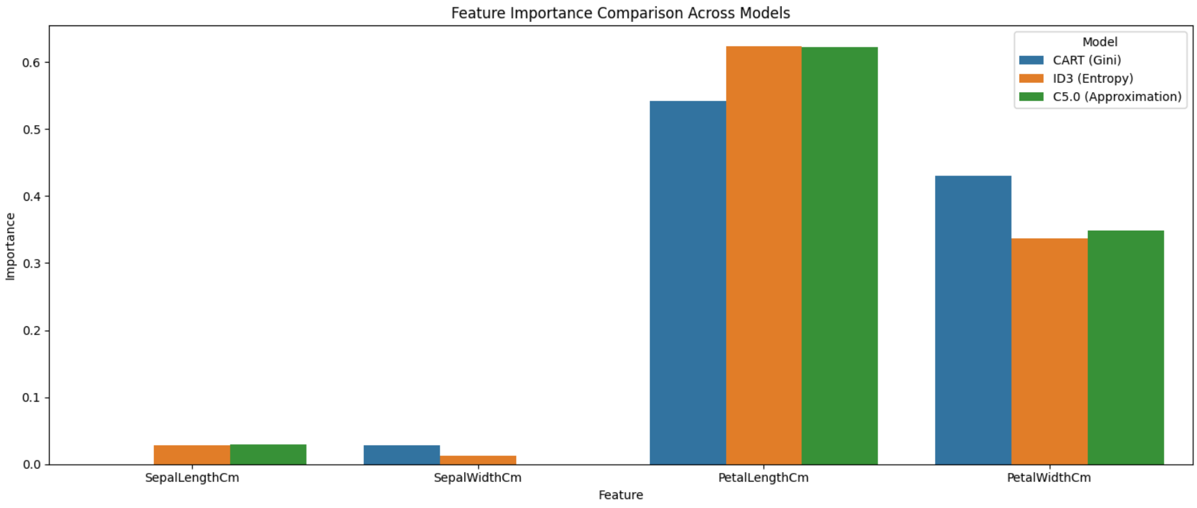


Figure 8. Comparison of Feature Importances Across Models

In Figure 8, `PetalLengthCm` and `PetalWidthCm` were consistently ranked as the two most important features by all algorithms, accounting for over 95% of the importance combined. Sepal dimensions (`SepalLengthCm`, `SepalWidthCm`) had minimal impact (<4% each). There were subtle differences: CART showed a more balanced importance between Petal Length (~54%) and Petal Width (~43%), while ID3 (~60% / ~36%) and C5.0 (~62% / ~34%) placed a stronger emphasis on Petal Length.

Visualizing the decision boundaries in the Petal Length vs. Petal Width space highlights the models' partitioning strategies (see Figure 9).

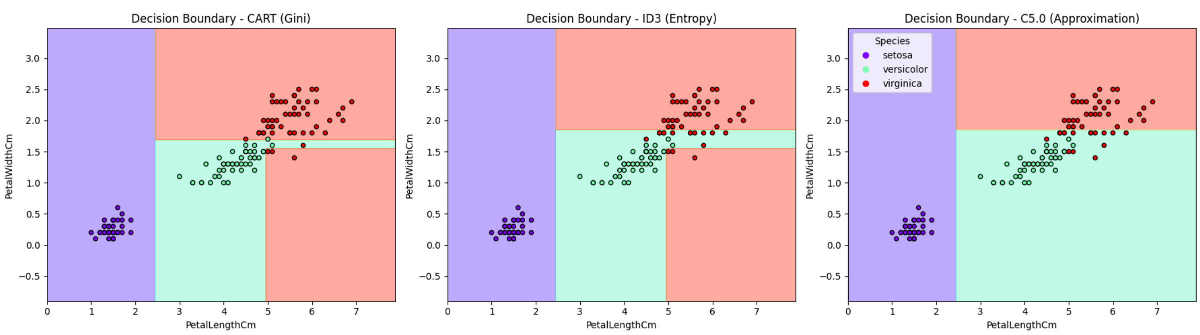


Figure 9. Decision Boundaries for CART, ID3, and C5.0 Models (Petal Features)

All models created the characteristic orthogonal (right-angled) boundaries typical of decision trees. They all successfully isolated the *Setosa* class with a vertical split around `PetalLengthCm = 2.5`. The main differences lay in the complexity of the boundary separating *Versicolor* and *Virginica*. CART and ID3 produced similar, more complex boundaries involving both horizontal (`PetalWidthCm ≈ 1.6`) and subsequent vertical splits. The C5.0 boundary was noticeably simpler, reflecting its pruned structure and potentially explaining its higher rate of misclassification between these two overlapping classes in the test set.

II. K- Nearest Neighbors

The K-Nearest Neighbors (KNN) model achieved a perfect accuracy of 100% on the test set, correctly classifying all 30 samples across the three Iris species: Setosa, Versicolor, and Virginica. The confusion matrix showed that every sample was correctly placed in its true class, with no misclassifications.

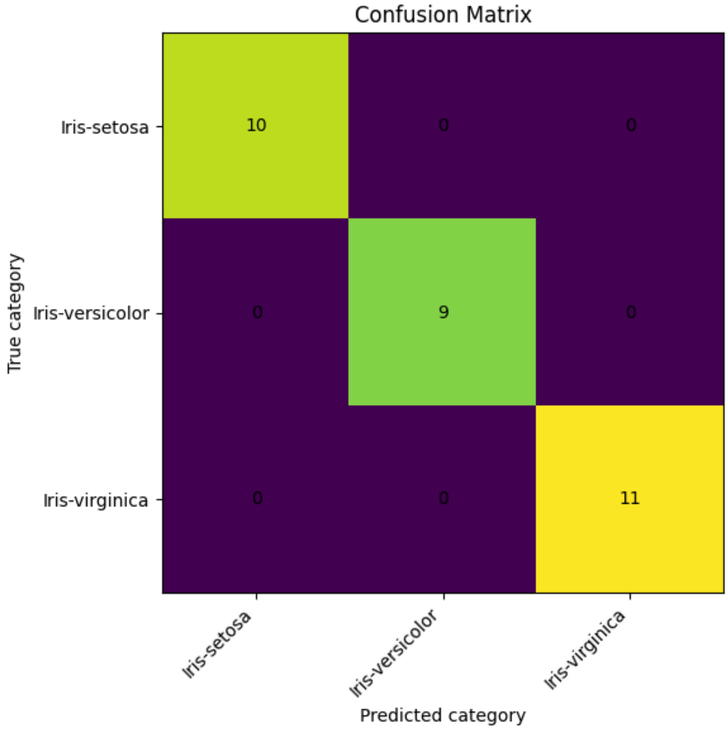


Figure 10. Confusion matrix of KNN where k=5 with test size 30

Furthermore, the classification report indicates perfect scores across all classes, achieving a precision, recall, and F1-score of 1.00.

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	10
versicolor	1.00	1.00	1.00	9
virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Figure 11. Classification Report of KNN where k=5 with test size 30

This outstanding performance is consistent with insights from the exploratory data analysis (EDA), where petal-related features were shown to strongly differentiate the species, particularly Setosa. Given that KNN relies on distance-based decision-making, and the Iris dataset has well-separated clusters, especially for Setosa and with sufficient separation for Versicolor and Virginica in the sampled test data, the model was able to achieve this ideal result.

However, the evaluation was based on a relatively small test with only 30 samples. Thus, further validation using cross-validation techniques is necessary to confirm whether this perfect performance generalizes across different splits of the data.

To validate the robustness of the KNN model, a 5-Fold Cross-Validation was performed with k=5. The results demonstrated high and consistent accuracies across all folds, with individual fold accuracies ranging from 93.33% to 100%. The average cross-validation accuracy was 97.33%, confirming that the model generalizes well to unseen data and that the earlier perfect test accuracy was not merely due to a favorable test split.

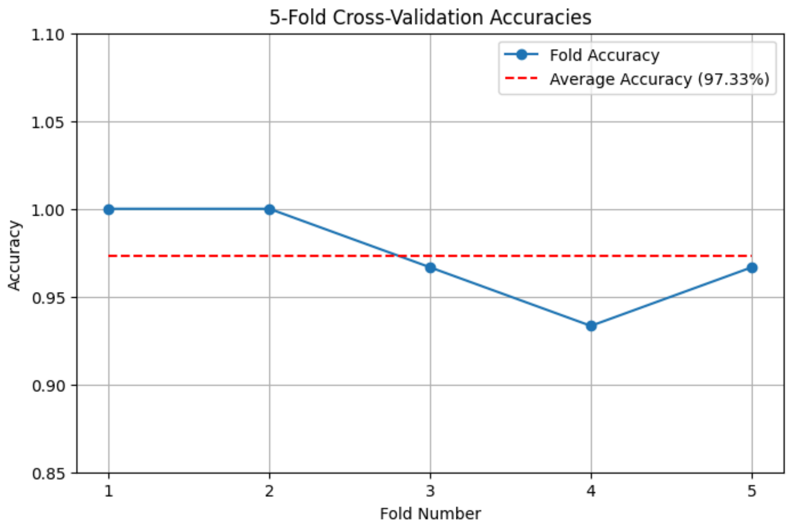


Figure 12. Accuracy plot over 5-fold Cross-Validation

```
Fold 1 Accuracy: 1.0000
Fold 2 Accuracy: 1.0000
Fold 3 Accuracy: 0.9667
Fold 4 Accuracy: 0.9333
Fold 5 Accuracy: 0.9667

Cross-Validation Accuracies: ['1.00', '1.00', '0.97', '0.93', '0.97']
Average Cross-Validation Accuracy: 0.9733
```

Figure 13. Accuracies over 5-fold Cross-Validation

The averaged confusion matrix analysis further revealed that Iris-Setosa was perfectly classified in all folds, with no misclassifications. Meanwhile, minor misclassifications occurred between Iris-Versicolor and Iris-Virginica, where a small portion of Versicolor samples were incorrectly predicted as Virginica, and vice versa. This pattern is consistent with the natural overlap between these two species observed during exploratory data analysis.

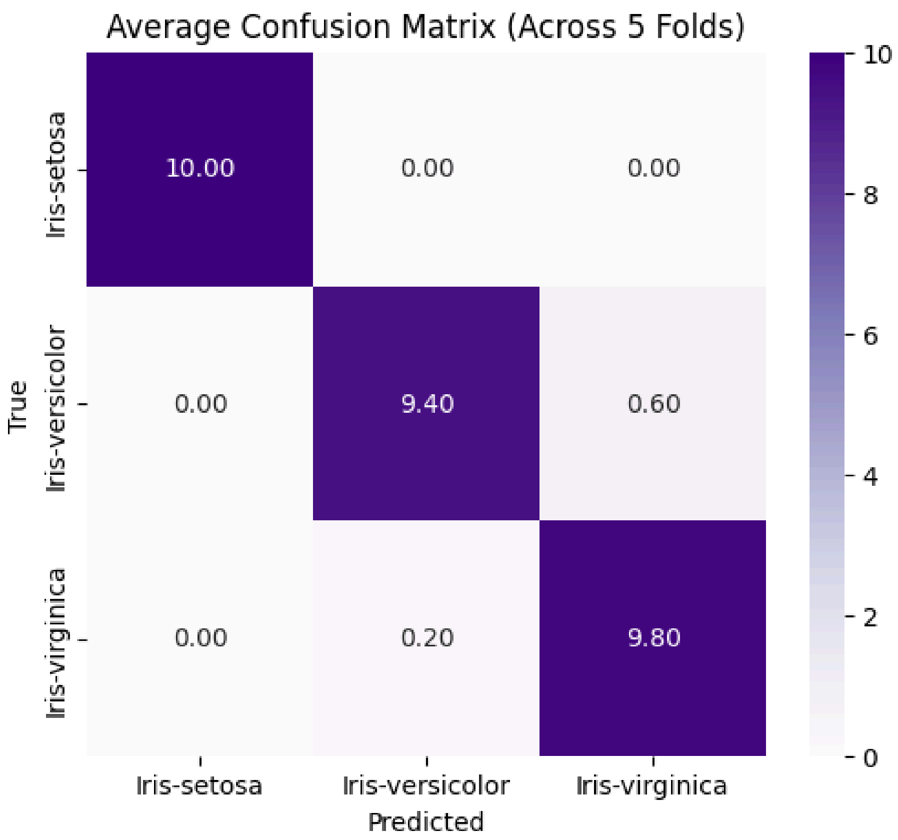


Figure 14. Average Confusion Matrix over 5-fold Cross-Validation

Furthermore, the average classification report across the folds showed that the model achieved a precision of 1.00 for Setosa, 0.98 for Versicolor, and 0.95 for Virginica. Similarly, the recall values were 1.00 for Setosa, 0.94 for Versicolor, and 0.98 for Virginica, while the F1-scores were consistently high across all classes, averaging around 0.97 overall. These results reinforce that the model not only performs accurately but also maintains a strong balance between precision and recall, particularly excelling in the classification of Iris-Setosa.

Average Classification Report (5-Fold Cross-Validation)

	Setosa	Versicolor	Virginica	Macro Average
Precision	1.0	0.98	0.95	0.98
Recall	1.0	0.94	0.98	0.97
F1-Score	1.0	0.96	0.96	0.97

Table 4. Classification Report of kNN Model

Overall, the model's performance remains excellent, showcasing that KNN is a highly effective classifier for the Iris dataset, particularly when distinguishing Setosa. The minor classification errors between Versicolor and Virginica show some of the challenges in differentiating closely related classes but do not significantly impact the overall model reliability.

III. Support Vector Machines

The Support Vector Machine (SVM) model also achieved a perfect accuracy of **100%** on the test set, successfully classifying all 30 samples across the three Iris species — Setosa, Versicolor, and Virginica. The confusion matrix (see Figure 15) confirmed that each sample was correctly assigned to its respective class, with no instances of misclassification observed.

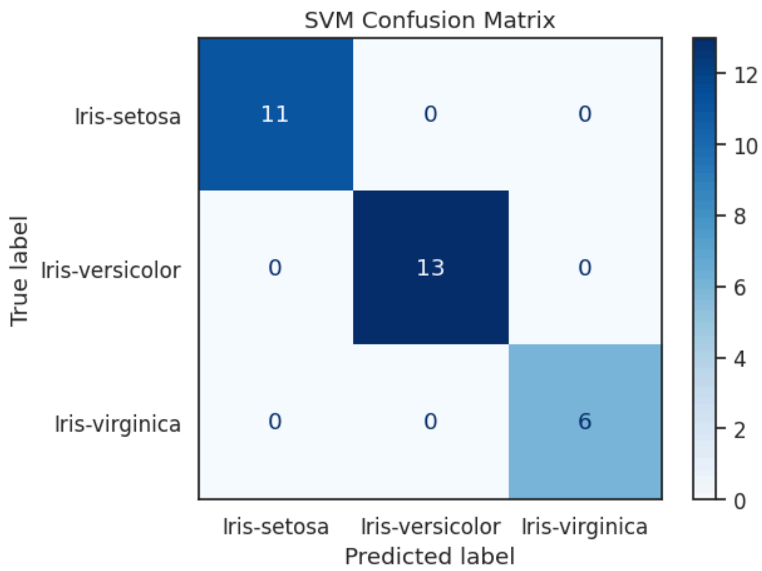


Figure 15. Confusion Matrix of Linear SVM Model

This confusion matrix for the Linear SVM model shows excellent classification performance across all three Iris species. The model correctly predicted all 11 instances of Iris-setosa, all 13 instances of Iris-versicolor, and 6 instances of Iris-virginica, with zero misclassifications for any class. This indicates that the Linear SVM was highly effective in distinguishing among the three classes, achieving perfect or near-perfect accuracy for each. The clear diagonal dominance in the matrix and absence of off-diagonal values confirm that the model was able to separate the classes without confusion, demonstrating strong generalization on the dataset.

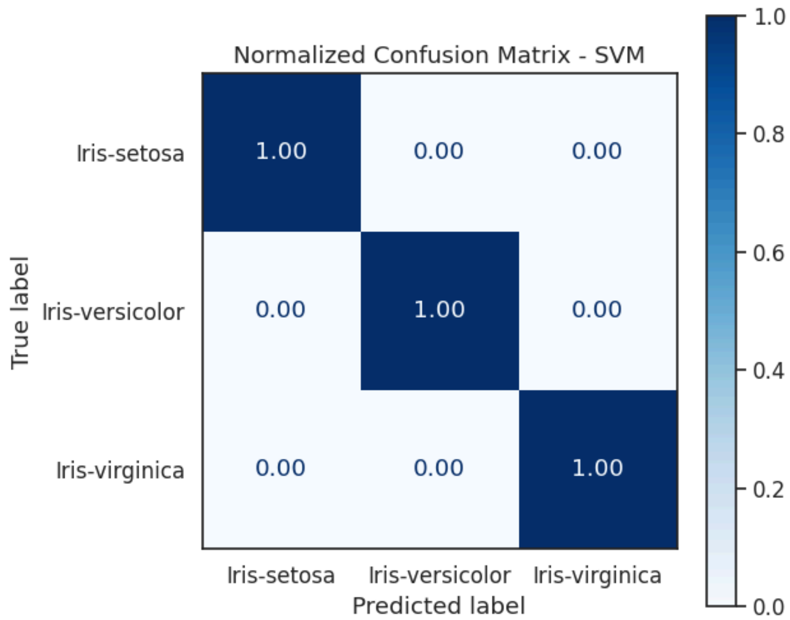


Figure 15.1. Confusion Matrix of Linear SVM Model

The normalized confusion matrix of this Linear SVM model shows perfect classification across all three Iris species. Each class, Iris setosa, Iris versicolor, and Iris virginica, achieved a true positive rate of 1.00 without misclassifications. This confirms the earlier visual results, providing strong numerical evidence that the Linear SVM model accurately separated the classes.

```
Kernel: linear
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00         6

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

Figure 16. Classification Report of Linear SVM Model

Moreover, this classification report of the Linear SVM model, as seen in Figure 16, shows outstanding results, achieving a perfect precision, recall, and F1-score of 1.00 across all three classes: Iris-setosa, Iris-versicolor, and Iris-virginica. This indicates that the model made no false positives or false negatives and classified every instance in the test set correctly, resulting in an overall accuracy of 100%. Both the macro and weighted averages also reached 1.00, suggesting consistent performance despite the slight imbalance in class support. These results imply that the dataset is likely well-suited for linear separation and that the Linear SVM model is highly effective for this classification task.

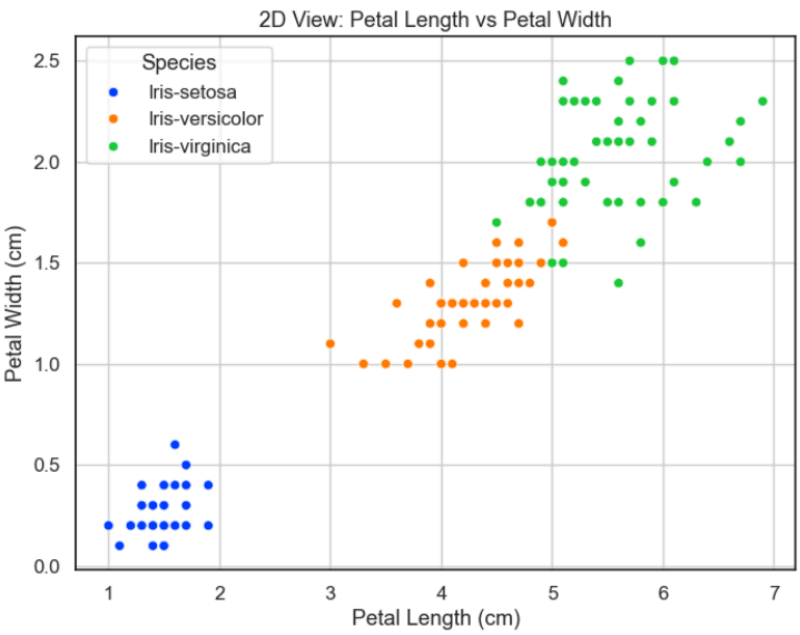


Figure 17. 2D Plot of Petal Length vs. Petal Width

This clearly shows distinct clusters for each Iris species, with Iris-setosa (blue) separated from Iris-versicolor (orange) and Iris-virginica (green). Iris-setosa forms a compact group with small petal measurements, while Iris-versicolor and Iris-virginica occupy progressively higher ranges of petal size. The minimal overlap between classes indicates strong linear separability, supporting the perfect classification performance achieved by the Linear SVM model.

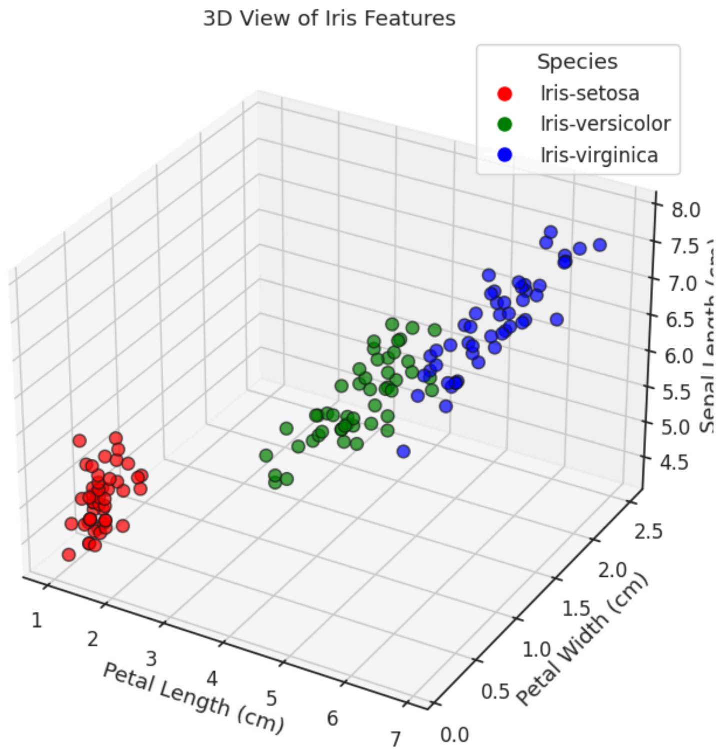


Figure 18. 3D Plot of IRIS Features

In the final 3D plot analysis shown in Figure 18, the Linear SVM model's capability is further validated by the clear spatial separation among the three Iris species when considering Petal Length, Petal Width, and Sepal Length simultaneously. Compared to the earlier 2D plot, the addition of a third feature enhances the distinction, especially between Iris-versicolor and Iris-virginica, which showed slight overlap in 2D.

Lastly, this 3D visualization confirms that the dataset benefits from linear separability across multiple dimensions, allowing the SVM model to maintain perfect classification performance. This comparative analysis highlights that while 2D features already provide strong separability, incorporating a third feature further strengthens the model's robustness and justifies the consistent 100% accuracy achieved.

4.2 Interpretation of Findings

Across all the algorithms, a clear theme can be seen; the Iris dataset is inherently well-structured and largely linearly separable, especially when described by their petal measurements.

Decision Tree

CART (Gini) yielded a strong test accuracy of 93.33%, with perfect identification of Setosa and Virginica but some confusion in Versicolor. ID3 (Entropy) dropped to 88.89%, trading a slightly deeper, more complex tree (depth 6) for only marginal gains in balancing precision and recall. The pruned C5.0 approximation achieved the simplest tree (depth 4) and excelled in cross-validation—averaging 96.0% accuracy with the lowest variance—yet underfit the specific hold-out split (84.44% test accuracy). In practice, this suggests that pruning (as in C5.0) promotes the most consistent performance across different subsamples, while deeper trees (ID3) risk over-fitting to nuances of a single split, and intermediate complexity (CART) often strikes the best balance of interpretability and test-set accuracy.

K-Nearest Neighbors

With a k=5, KNN achieved 100% accuracy on the 30-sample test set and gathered a high cross-validation mean (97.33%), though the individual folds ranged from 93.33% to 100%. The few mistakes between the Versicolor and Virginica mirror their natural overlap in petal dimensions. This shows that distance-based methods can deliver good results on clean, clustered data, but can be sensitive to how the test set is created.

Support Vector Machine

The linear SVM also has a 100% test accuracy, with perfect precision, recall, and F1 score across all classes. Both 2D and 3D visualizations show that the petal length, petal width and the sepal length create linearly separable clusters. As such, a simple linear boundary is enough to distinguish species, validating the SVM's suitability for this task. Its performance is deterministic—unlike KNN's slight variability—and it requires far fewer hyperparameter decisions when the data align with linear separability.

In summary, the Iris dataset's clear clustering structure allows all three algorithm families to perform exceptionally. The key distinction lies in how each handles complexity and generalization: pruning yields stability, depth can overfit, and distance or margin criteria require clean feature scaling but reward with high accuracy. These insights will guide model selection for similar classification tasks, balancing the need for interpretability against the desire for absolute predictive performance.

5 Conclusions and Recommendations

5.1 Summary of Key Findings

This study successfully evaluated and compared the performance of three fundamental classification algorithms — Decision Trees (CART, ID3, and C5.0 variants), Support Vector

Machine (SVM), and K-Nearest Neighbors (KNN) — using the Iris dataset. Each model demonstrated strong predictive capabilities, confirming their suitability for structured and well-separated classification tasks.

Based on the comparative analysis, K-Nearest Neighbors (KNN) and Linear Support Vector Machine (SVM) are highly recommended if the primary goal is maximizing predictive accuracy on the Iris dataset or similar well-separated data, as both achieved perfect test set accuracy and KNN demonstrated excellent cross-validation stability (97.33% mean). However, if interpretability and understanding the decision-making process are crucial, Decision Trees offer a superior alternative. Among the trees, CART (Gini) is recommended for providing the best balance between high test accuracy (93.33%) and moderate complexity, while the pruned C5.0 approximation is the preferred choice when model simplicity, ease of explanation, and consistent generalization across different data samples (evidenced by its top cross-validation score of 96.0% and lowest variance) are prioritized, even at the cost of slightly lower accuracy on a specific test split. Ultimately, the optimal choice hinges on the specific project requirements: prioritize KNN or SVM for pure accuracy, CART for balanced accuracy and interpretability, or C5.0 for maximum simplicity and stable generalization.

5.2 Recommendations

Based on the evaluation of Decision Trees, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) classifiers on the Iris dataset, several recommendations can be made for selecting and applying models to similar classification problems. For applications where both high accuracy and model interpretability are important, a pruned Decision Tree (such as the C5.0 approximation with pruning parameters) is the preferred choice, offering a simple, stable, and easily understandable model structure with strong generalization performance. If maximizing predictive accuracy is the sole priority, the Linear SVM model is recommended, having achieved perfect classification on the test set with minimal tuning requirements.

The KNN model (with k=5) also presents an excellent alternative for small, clean, and well-separated datasets, achieving near-perfect cross-validation results and intuitive classification based on proximity to existing instances. However, it is crucial to apply feature scaling techniques, such as StandardScaler, when using KNN or SVM, as their performance is highly sensitive to the range and distribution of input features.

Ultimately, the choice of model should be guided by the specific goals of the application—whether prioritizing maximum predictive accuracy, maintaining model interpretability, or ensuring consistency across different data partitions. Additionally, incorporating cross-validation as a standard practice is recommended to better estimate model generalization performance and prevent overfitting to a single data split.

5.3 Limitations

Although the results are positive, there are still limitations within this dataset. The analysis was confined to the well-balanced and relatively simple Iris dataset, which may not fully represent the challenges posed by more complex, high-dimensional, or noisy real-world data.

Additionally, the performance evaluations were based on a single train-test split without employing multiple random seeds or nested cross-validation, which could introduce some variability in the reported results.

In addition, the scope of algorithms explored was limited to basic classifiers (Decision Trees, SVM, and KNN) without incorporating other more complex algorithms such as Random Forests, Gradient Boosting, or more sophisticated kernel-based SVMs. Advanced feature engineering techniques, ensemble learning, or deep learning approaches were not explored in this comparative study. Therefore, while the results are meaningful within the scope of fundamental algorithms, they do not capture the broader range of available machine learning strategies.

5.4 Suggestions for Future Studies

Future research could extend this comparative analysis by including more complex models such as Random Forests, Gradient Boosting Machines, or Support Vector Machines with non-linear kernels (e.g., RBF, polynomial). Investigating the effects of feature scaling methods, such as Min-Max Scaling or Robust Scaling, could also offer valuable insights, particularly when applying distance-based algorithms to less clean datasets.

Additionally, applying these classification models to larger, imbalanced, or more complex datasets would help evaluate their generalizability and robustness under more challenging conditions. Employing nested cross-validation or repeated train-test splits would provide more reliable estimates of model performance and help mitigate the effects of random data partitioning.

References

Bishop, C. M. (2016). *Pattern recognition and machine learning*. Springer.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. In Routledge eBooks. https://doi.org/10.1201/9781315139470

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27. https://doi.org/10.1109/tit.1967.1053964

Fisher, R. (1936). Iris [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. In Springer series in statistics. https://doi.org/10.1007/978-0-387-84858-7

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning*. In Springer texts in statistics. https://doi.org/10.1007/978-1-0716-1418-1

Macaraeg, P. (2025). *IRIS-Dataset-Comparative-Analysis* [Source code repository]. GitHub. https://github.com/PaullyMac/IRIS-Dataset-Comparative-Analysis

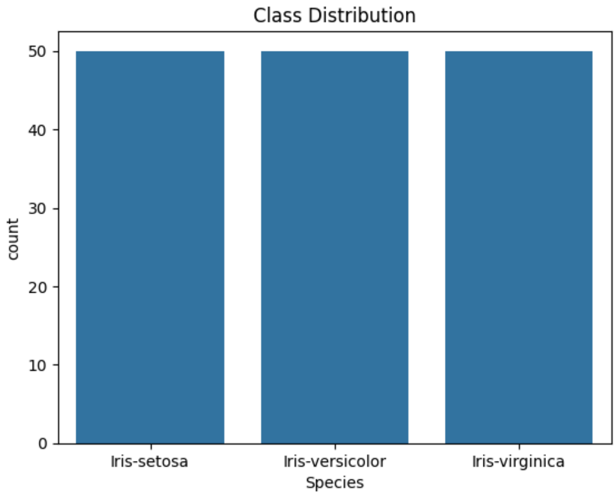
Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1007/bf00116251

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Appendices

Class Distribution:

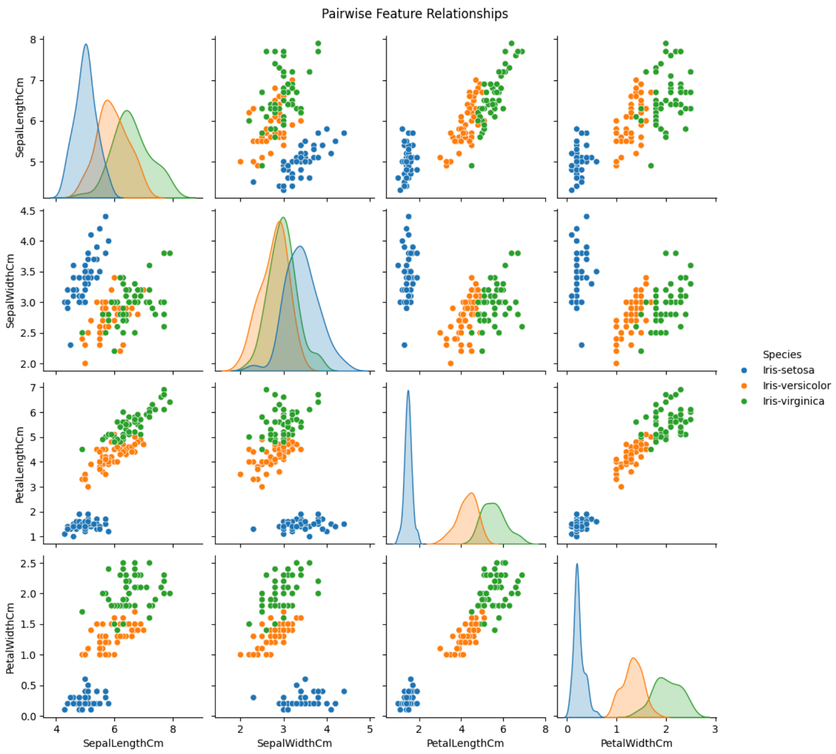
The bar plot shows an equal distribution of the three species, indicating a balanced classification problem without class imbalance issues.



Pairwise Feature Relationships:

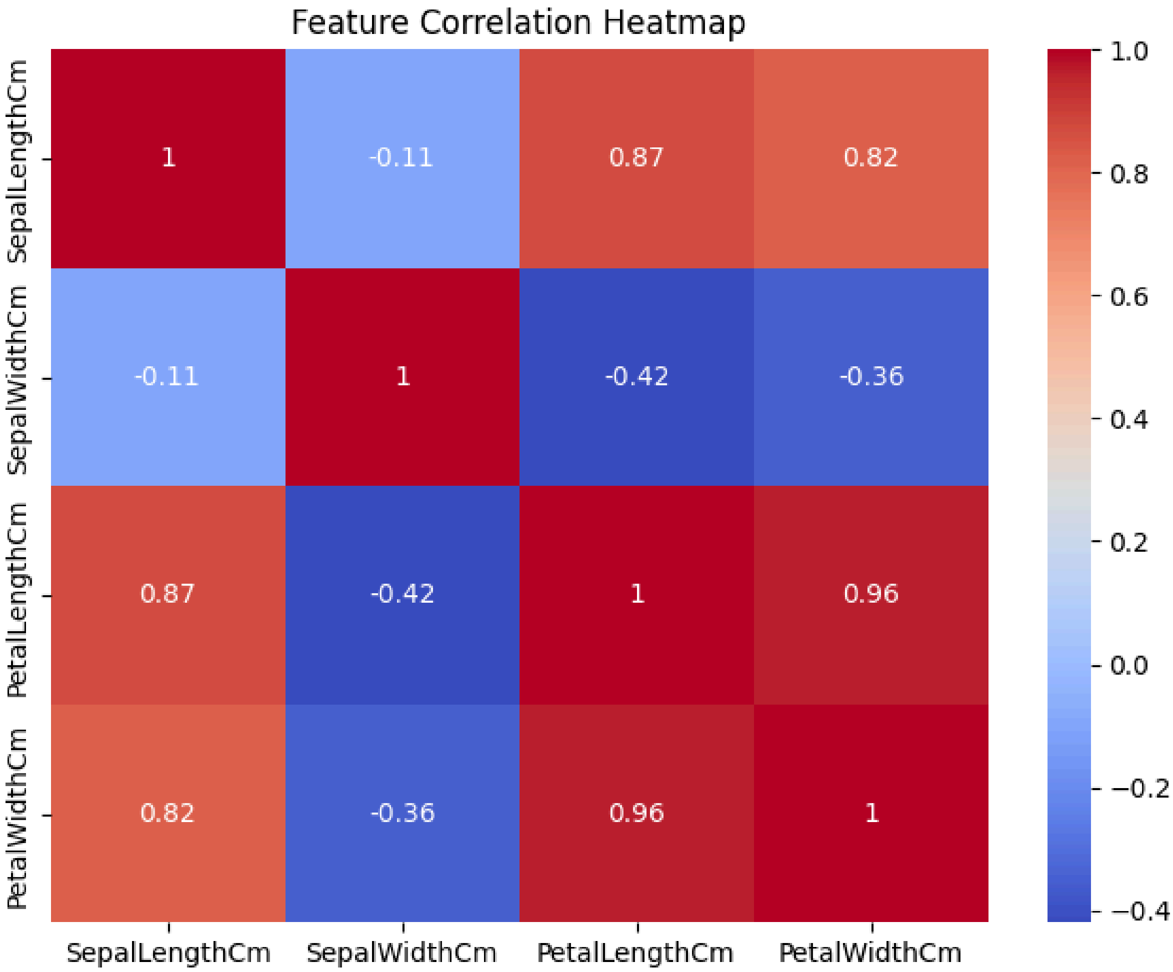
The pairplot reveals:

- Iris-setosa is clearly separable from the other two species across almost all feature pairs.
- Iris-versicolor and Iris-virginica show partial overlap, especially in Sepal Width and Sepal Length features.
- Petal Length and Petal Width provide the clearest separation among all three species.



Feature Correlations:

- Petal Length and Petal Width have an extremely strong positive correlation (r = 0.96).
- Sepal Length is moderately correlated with Petal Length (r = 0.87) and Petal Width (r = 0.82).
- Sepal Width is weakly correlated with all other features and negatively correlated with Petal measurements.



Feature Distributions:

- Petal Width and Petal Length:
  - Setosa has significantly smaller values compared to Versicolor and Virginica.
  - Virginica shows higher petal measurements than Versicolor.
- Sepal Width:
  - Setosa shows slightly higher median sepal width, but Versicolor and Virginica overlap considerably.
- Sepal Length:
  - Virginica has the largest sepal lengths, followed by Versicolor and Setosa.