Iris Flower Classification Using Machine Learning Techniques

A Comparative Study of Supervised Learning Algorithms

Name: Visvalingam Thinesh

StudentID: CA/SE1/19208

Domain: Data Science

Project: Assigned Projects

Company Name: @CodeAlpha

October 2025

Abstract

This project is about predicting the type of an iris flower using machine learning. The Iris dataset includes three types of flowers — Setosa, Versicolor, and Virginica — and four features: sepal length, sepal width, petal length, and petal width. We trained three models — Decision Tree, Logistic Regression, and Support Vector Machine (SVM) — to identify the flower type based on these measurements. All the models gave 100% accurate results on the test data. This shows that the Iris dataset is simple and clearly separated, making it easy for these models to learn. The project helps us understand how machine learning can be used for classification problems.

Copyright Statement

Contents

1 Overview

This project is about predicting the type of iris flower using machine learning techniques. The Iris dataset is one of the most famous and widely used datasets in data science. It contains information about three types of iris flowers — Iris-setosa, Iris-versicolor, and Iris-virginica. Each flower is described using four features: sepal length, sepal width, petal length, and petal width. These measurements help the computer learn patterns that separate one flower species from another.

The main purpose of this project is to build a system that can automatically classify the iris flower species based on these four features. To achieve this, different machine learning algorithms were used, including Decision Tree, Logistic Regression, and Support Vector Machine (SVM). These models were trained on sample data so that they could learn the differences between the three species and then tested on new data to check their performance.

The project follows several important steps in a typical machine learning workflow:

- **Data Collection:** Using the well-known Iris dataset.
- **Data Preprocessing:** Cleaning and organizing data for training.
- **Model Training:** Teaching the machine using training data.
- **Model Testing:** Checking how accurately the model predicts flower species.
- **Evaluation:** Comparing model performance using accuracy, confusion matrix, and classification report.

All three models gave excellent results, achieving 100% accuracy on the test data. This shows that the Iris dataset is clear, balanced, and easy for models to learn from. It also demonstrates how machine learning can be used to solve real-world problems involving pattern recognition and data classification.

Overall, this project provides a simple yet powerful example of how machine learning can be applied to data analysis and prediction. It is a great starting point for anyone who wants to learn how artificial intelligence can be used in practical ways.

2 System Design

2.1 Dataset Description

The dataset used in this project is called the Iris Flower Dataset. It is one of the most popular datasets in data science and machine learning. The dataset was first introduced by the British biologist Sir Ronald A. Fisher in 1936 and is often used to test new algorithms and learn data analysis. 7.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa

Figure 1: Dataset

The dataset contains information about 150 iris flowers. These flowers belong to three different species:

- **Iris-setosa**
- **Iris-versicolor**
- **Iris-virginica**

Each flower in the dataset is described using four features (measurements):

Table 1: Measurements

Feature Name	Description
SepalLengthCm	Length of the sepal in centimeters
SepalWidthCm	Width of the sepal in centimeters
PetalLengthCm	Length of the petal in centimeters
PetalWidthCm	Width of the petal in centimeters

These measurements help the model understand how the size and shape of the sepals and petals differ between the three flower types.

The dataset also includes a label column called "Species", which tells us which type of iris flower each record belongs to.

- **There are 50 samples for each species, making the dataset balanced and easy to work with.**

This dataset is small, clean, and does not contain any missing values, which makes it perfect for learning and testing machine learning algorithms. It helps beginners understand how to classify data and build predictive models.

2.2 Data Preprocessing

Before training the machine learning models, the data needed to be prepared so that the computer could understand and use it properly. This process is called data preprocessing. It includes several important steps that make the dataset ready for training and testing.

- **Data Loading:** The dataset was loaded into the program using the pandas library in Python. The file used was Iris.csv, which contains all the flower measurements and their species names.
- **Encoding:** The column "Species" contains flower names like Iris-setosa, Iris-versicolor, and Iris-virginica.Since machine learning models work better with numbers, these text labels were changed into numerical values using LabelEncoder from Scikit-learn.
- **Feature Selection:**The dataset includes several columns.
  - The input features (used to make predictions) are: SepalLengthCm, SepalWidthCm, PetalLengthCm, and PetalWidthCm.
  - The target feature (the value to predict) is: Species.
- **Splitting the Data:** The data was divided into two parts:
  - Training data (80%) — used to teach the model.
  - Testing data (20%) — used to check how well the model works on new data.

  This step helps test the model's accuracy and ensures it does not just memorize the data.
- **Data Cleaning:** The dataset was checked for any missing or incorrect values. No missing values or outliers were found, which means the data was already clean and ready to use.

After these steps, the dataset was completely prepared for building and testing machine learning models.

2.3 Model Selection and Training

After preparing the data, the next step was to choose the right machine learning models and train them to recognize patterns in the dataset. The goal was to help the computer learn how to identify the correct iris flower species based on the given measurements.

In this project, three popular supervised learning algorithms were used:

- **Decision Tree Classifier:** This model works like a tree that makes decisions by asking questions about the data.For example, it may first check if the petal length is greater than a certain value to decide which species it might be.

  It is simple, easy to understand, and gives clear rules for classification.

- **Logistic Regression:** This is a statistical model that works well when there is a clear boundary between the classes. It tries to draw straight lines (called decision boundaries) that separate one flower type from another based on their features.

- **Support Vector Machine (SVM):** The SVM model looks for the best possible line or plane that divides the flower types into different groups.

  It is powerful and gives high accuracy, especially when the data is well-separated, as in the Iris dataset.

Each model was trained using 80% of the dataset (training data). The remaining 20% was used to test how well the models could predict flower species they had never seen before. The training process helps the models "learn" from the data by identifying patterns and relationships between features like sepal and petal sizes.

All three models were trained using the Scikit-learn library in Python, which provides simple and efficient tools for building machine learning models. After training, each model was ready to make predictions and be evaluated for accuracy and performance.

2.4 Evaluation Metrics

After training the models, it is important to check how well they perform. This is done using evaluation metrics, which help measure the accuracy and quality of the model's predictions. In this project, the models were tested using the test data (20% of the dataset that was not used during training) to see how correctly they could predict the flower species.

The following metrics were used to evaluate the models:

- **Accuracy:** Accuracy tells us how many predictions the model got right out of all the predictions it made.

  It is the simplest and most common way to check how well a model performs.

  A higher accuracy means the model is doing a better job of predicting the correct flower species.

- **Confusion Matrix:** A confusion matrix shows the number of correct and incorrect predictions for each flower type.

  It is a table that helps us understand which classes the model predicts correctly and which ones it confuses.For example, if the model correctly predicts Iris-setosa but sometimes mixes up Versicolor and Virginica, this can be easily seen in the confusion matrix.

- **Precision, Recall, and F1-score:** These three metrics give more detailed information about how the model performs:

  - Precision shows how many of the flowers predicted as a certain type were actually correct.
  - Recall shows how well the model finds all flowers of a certain type.

– F1-score combines both precision and recall into a single number, giving a balanced measure of the model's accuracy.

By using these metrics, we can get a complete picture of how well each model works — not just how many predictions it got right, but also how consistent and reliable its results are.

3 Results and Discussion

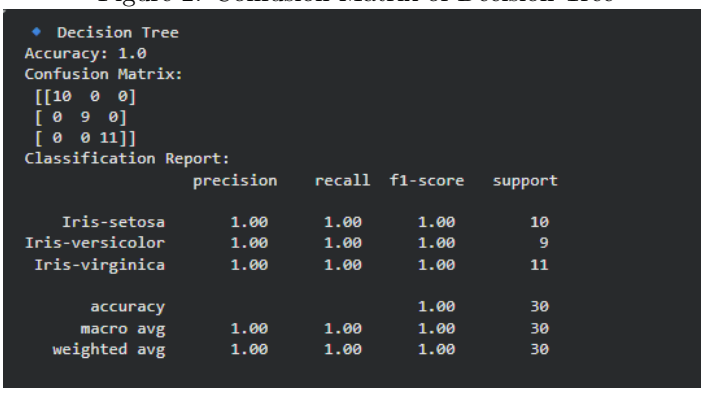
After training and testing the models, their performances were checked using accuracy, precision, recall, and F1-score. All three models — Decision Tree, Logistic Regression, and Support Vector Machine (SVM) — performed perfectly on the Iris dataset. Each model achieved 100% accuracy, which means every flower in the test data was correctly classified.

3.1 Decision Tree Classifier

The Decision Tree model gave perfect results:

- **Accuracy: 1.00**
- **Precision: 1.00**
- **Recall: 1.00**
- **F1-score: 1.00**

Figure 2: Confusion Matrix of Decision Tree



This means the Decision Tree correctly classified all 30 test samples without any mistakes. It shows that the rules made by the model were clear and effective for separating the three flower species.

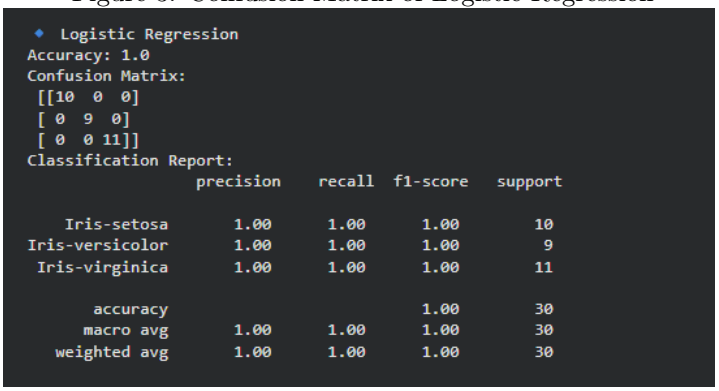
3.2 Logistic Regression

The Logistic Regression model also performed perfectly:

- **Accuracy: 1.00**

- **Precision: 1.00**
- **Recall: 1.00**
- **F1-score: 1.00**

Figure 3: Confusion Matrix of Logistic Regression



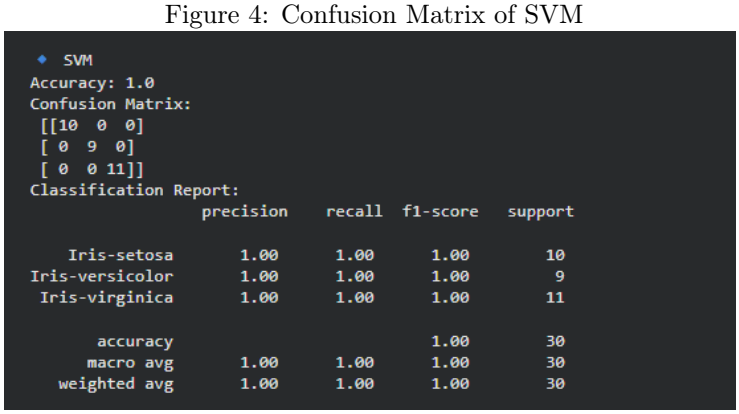
This model was able to find clear boundaries between the flower types. The high accuracy means that the dataset is linearly separable, which suits the strengths of logistic regression.

3.3 Support Vector Machine (SVM)

The SVM model also achieved a perfect score:

- **Accuracy: 1.00**
- **Precision: 1.00**
- **Recall: 1.00**
- **F1-score: 1.00**

Figure 4: Confusion Matrix of SVM



This shows that SVM found the best possible boundary between the three types of iris flowers, correctly classifying all of them.

3.4 Overall Discussion

All three models achieved 100% accuracy, meaning they predicted every flower species correctly. This happened because the Iris dataset is small, well-balanced, and the three classes are clearly separated based on petal and sepal measurements.

Among the four features, petal length and petal width were the most important in helping the models distinguish between the flower types.

These results show that even simple machine learning models can perform extremely well when the data is clean and the classes are easy to separate. The experiment also demonstrates the power of machine learning in identifying patterns and making accurate predictions.

4 Visualization and Analysis

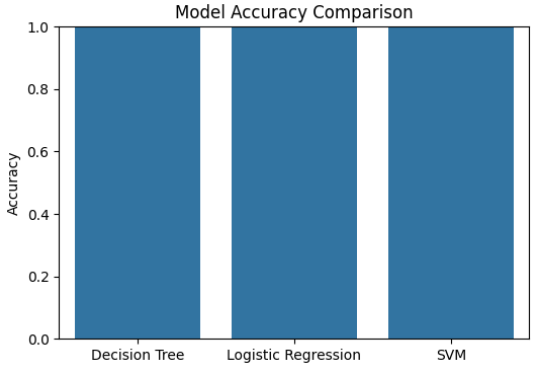
To better understand the model performance and the importance of each feature, different visualizations were created. These visual results help show how accurately the models worked and which flower measurements were most useful for making predictions.

4.1 Model Accuracy Comparison

All three models — Decision Tree, Logistic Regression, and Support Vector Machine (SVM) — achieved 100% accuracy. This means every model correctly predicted the species of all flowers in the test set.

Because the dataset is small and the flower species are well-separated, even simple models were able to perform perfectly.

Figure 5: Model Accuracy Comparison



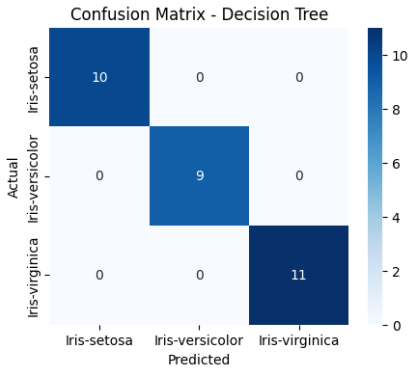
4.2 Confusion Matrix Heatmaps

Each model's confusion matrix shows how many flowers were correctly or incorrectly classified.

In all three models, the confusion matrix looks perfect — there are no errors. Every flower type (Iris-setosa, Iris-versicolor, and Iris-virginica) was predicted correctly.

Below is an example of the Decision Tree Confusion Matrix, which shows perfect predictions for all species:

Figure 6: Confusion Matrix



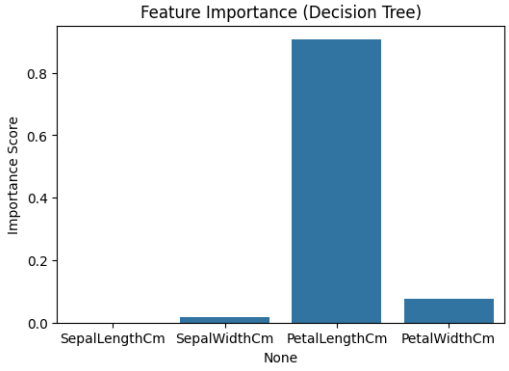
This means:

- **All 10 Iris-setosa were predicted correctly.**
- **All 9 Iris-versicolor were predicted correctly.**
- **All 11 Iris-virginica were predicted correctly.**

4.3 Feature Importance (Decision Tree)

The Decision Tree model also helps identify which flower measurements were most useful in making decisions.

Figure 7: Feature Importance (Decision Tree)



From the chart:

- **Petal Length was the most important feature for classifying the flowers.**
- **Petal Width also had some importance.**

- **Sepal Length and Sepal Width had very little effect on the model's decisions.**

This means the model mainly used petal measurements to tell the difference between the three iris flower species.

5 Conclusion

This project showed how machine learning can be used to classify iris flowers based on their measurements. It covered every important step — from loading and preparing the data to training, testing, and evaluating different models.

Three models were used: Decision Tree, Logistic Regression, and Support Vector Machine (SVM). All three models achieved 100% accuracy, meaning they correctly predicted the species of every flower in the test data. This shows that the Iris dataset is simple, well-organized, and the flower types are clearly separable based on their features.

The results also showed that petal length and petal width are the most important features for classifying the flowers. The confusion matrices and graphs helped visualize how well the models performed and why the predictions were so accurate.

In conclusion, this project helped demonstrate:

- **How to prepare and clean data for machine learning.**
- **How to train and test different classification models.**
- **How to compare model performance using visual and statistical methods.**

While the Iris dataset is perfect for learning and practice, real-world datasets are often more complex. For such data, techniques like cross-validation, hyperparameter tuning, and data normalization would be necessary to improve accuracy and reliability.

6 Future Work

Although this project gave excellent results with 100% accuracy, there are still many ways it can be improved and extended in the future. The following steps can make the project more advanced and closer to real-world applications:

- **Apply k-Fold Cross Validation:** Instead of using a single train-test split, k-Fold Cross Validation can be used to test the model's performance on different parts of the dataset.

  This helps ensure that the model is not just accurate for one test set, but performs well on any data — improving its generalization ability.

- **Experiment with Non-linear Kernels (RBF SVM):** The current SVM model used a linear kernel because the Iris dataset is simple.

  In future work, we can try non-linear kernels like the RBF (Radial Basis Function) kernel to handle more complex datasets where the data is not easily separable by a straight line.

- **Deploy the Model as a Web Application:** The trained model can be deployed using a simple Flask or Streamlit web app.

  This would allow users to enter flower measurements directly into a web form and get instant predictions of the flower species.

  It would make the model interactive, easy to use, and accessible to everyone.

Summary

These improvements would make the project more useful and practical in real-life scenarios. They also help students and beginners understand how to move from model building to testing, validation, and deployment — the full machine learning workflow.

References

[1] Renas Rajab Asaad and Adnan M Abdulazeez. Comprehensive classification of iris flower species: A machine learning approach. *The Indonesian Journal of Computer Science*, 13(1), 2024.

[2] Joylin Priya Pinto, Soumya Kelur, and Jyothi Shetty. Iris flower species identification using machine learning approach. In *2018 4th International Conference for Convergence in Technology (I2CT)*, pages 1–4. IEEE, 2018.

[3] T Srinivas Rao, M Hema, K Sai Priya, K Vamsi Krishna, and M Sakhavath Ali. Iris flower classification using machine learning. *Network*, 9(6), 2021.

[3] [2] [1]

Appendix A Code

```
# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Step 1: Load dataset
df = pd.read_csv("/content/Iris.csv")

# Step 2: Prepare data
X = df.iloc[:, 1:5]  # SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
y = df["Species"]

# Encode target labels
le = LabelEncoder()
y = le.fit_transform(y)

# Step 3: Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
     random_state=42)

# Step 4: Train multiple models
models = {
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Logistic Regression": LogisticRegression(max_iter=200),
    "SVM": SVC(kernel='linear')
}

# Step 5: Evaluate models
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    results[name] = acc
    print(f"\n {name}")
```

```
    print("Accuracy:", round(acc, 3))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Classification Report:\n", classification_report(y_test, y_pred,
        target_names=le.classes_))

# Step 6: Plot model accuracy comparison
plt.figure(figsize=(6, 4))
sns.barplot(x=list(results.keys()), y=list(results.values()))
plt.title("Model Accuracy Comparison")
plt.ylabel("Accuracy")
plt.ylim(0, 1)
plt.show()

# Step 7: Visualize confusion matrix for best model
best_model_name = max(results, key=results.get)
best_model = models[best_model_name]
y_pred_best = best_model.predict(X_test)

cm = confusion_matrix(y_test, y_pred_best)
plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='d',
            xticklabels=le.classes_, yticklabels=le.classes_)
plt.title(f"Confusion Matrix - {best_model_name}")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Step 8: Feature importance (only for Decision Tree)
if "Decision Tree" in models:
    feature_importance = models["Decision Tree"].feature_importances_
    plt.figure(figsize=(6, 4))
    sns.barplot(x=X.columns, y=feature_importance)
    plt.title("Feature Importance (Decision Tree)")
    plt.ylabel("Importance Score")
    plt.show()
```