# Iris Classification Project Report

## 1. Introduction

The Iris dataset is one of the most famous datasets in machine learning. It contains 150 iris flowers across three species (Setosa, Versicolor, Virginica) described by four features: sepal length, sepal width, petal length, and petal width. The goal is to classify flowers into the correct species.

## 2. Tools and Technologies

- Python (VS Code, Jupyter Notebook)

- Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn

## 3. Dataset Exploration

- 150 samples (50 per species)

- Balanced dataset, no missing values

- Features: SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm

## 4. Exploratory Data Analysis (EDA)

- Petal features (length, width) provide strong class separation.

- Sepal features overlap between species.

- Pairplots, heatmaps, and boxplots confirm correlations and feature importance.

## 5. Data Preprocessing

- Split into features (X) and labels (y)

- Train-test split (80%-20%)

- StandardScaler applied to normalize features

## 6. Model Training

- KNN: Simple, effective, 100% accuracy with k=3

- Decision Tree: Easy to interpret, ~97–100% accuracy

- Random Forest: Robust, ~98–100% accuracy

- SVM: Strong generalization, ~97–99% accuracy

- Logistic Regression: ~92–95% accuracy

## 7. Model Evaluation

Metrics used: Accuracy, Precision, Recall, F1-score, Confusion Matrix

Example (KNN, k=3):

- Accuracy: 100%

- Precision, Recall, F1-score: 1.00 for all classes

## 8. Hyperparameter Tuning

- GridSearchCV applied to KNN

- Best parameter: k=3

## 9. Model Insights

- Feature importance (Random Forest): Petal length > Petal width > Sepal features

- Decision boundaries: Petal features separate species clearly

## 10. Conclusion

- KNN, Random Forest, and SVM performed best.

- KNN achieved 100% accuracy, but Random Forest and SVM are more robust.

- Logistic Regression less effective due to non-linear separability.

## 11. Future Improvements

- Deploy model as a web app (Flask/Django)

- Use PCA for dimensionality reduction and visualization

- Experiment with neural networks

This project demonstrates the complete ML workflow: data preprocessing, visualization, training, evaluation, tuning, interpretation, and reporting.