A blue circle with white lines and a blue circle with white lines

Description automatically generated

AI Lab Project Report

# Name: Huzaifa Tariq

# Reg No: 230201039

# Class: CS/04/A

# Submitted to: Mam Natiqa

# Introduction:

## Project Overview:

This project compares the performance of different machine learning algorithms for a classification task. The dataset used for this project is the **Iris dataset**, which is a well-known dataset in machine learning, containing information on different species of iris flowers based on four features: sepal length, sepal width, petal length, and petal width.

The goal of the project is to determine the most suitable machine learning algorithm for this classification task by evaluating several algorithms based on their performance metrics.

## Objectives:

* Implement and evaluate multiple machine learning algorithms, including **Logistic Regression**, **Decision Trees**, **Random Forest**, **Support Vector Machines (SVM)**, and **K-Nearest Neighbours (KNN)**.
* Compare the performance of these algorithms based on metrics such as **accuracy**, **precision**, **recall**, and **F1 score**.
* Select the most suitable algorithm for the classification problem.

## Scope:

* The Iris dataset will be used for classification.
* The project will evaluate multiple algorithms using **cross-validation** for reliable performance evaluation.
* Metrics such as **confusion matrix**, **ROC-AUC**, and **accuracy** will be used for evaluation.

# Problem Definition and Dataset:

## Problem Statement:

The objective of this project is to classify the species of iris flowers (Setosa, Versicolor, and Virginica) based on the following features:

* Sepal Length
* Sepal Width
* Petal Length
* Petal Width

This is a multi-class classification problem where the goal is to correctly predict the species of each flower from the feature set.

## Dataset Description:

The dataset used is the **Iris dataset** from the UCI Machine Learning Repository, which consists of 150 samples, with each sample having four features (sepal length, sepal width, petal length, and petal width). The dataset is evenly split into three classes: Setosa, Versicolor, and Virginica.

* **Number of instances**: 150
* **Number of features**: 4 (sepal length, sepal width, petal length, petal width)
* **Number of classes**: 3 (Setosa, Versicolor, Virginica)

The data was split into **80% training** and **20% test** using **Stratified K-Fold cross-validation** to ensure balanced class distribution in each fold.

# Methodology:

## Data Preprocessing:

* **Data Cleaning**: The dataset had no missing values, so no cleaning was necessary.
* **Feature Scaling**: Since the features were measured in different units, **StandardScaler** was used to scale the features to have a mean of 0 and a standard deviation of 1.
* **Train-Test Split**: The data was split into training and test sets using an **80/20** ratio.

## Model Implementation:

The following machine learning algorithms were implemented and compared:

1. **Logistic Regression (LR)**: A linear model used for classification tasks.
2. **Decision Tree (DT)**: A non-linear model that splits the data based on feature values to create a tree structure.
3. **Random Forest (RF)**: An ensemble method that creates multiple decision trees and averages their predictions.
4. **Support Vector Machine (SVM)**: A model that finds the hyperplane that best separates data classes.
5. **K-Nearest Neighbours (KNN)**: A simple, non-parametric algorithm that classifies data based on the majority class of its nearest neighbours.

For each model, I used **GridSearchCV** to tune hyperparameters and improve performance.

## Performance Evaluation:

The following performance metrics were used:

* **Accuracy**: The proportion of correct predictions.
* **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.
* **Recall**: The ratio of correctly predicted positive observations to all observations in the actual class.
* **F1-Score**: The harmonic mean of precision and recall.
* **Confusion Matrix**: To visualize the performance of the classifiers.
* **ROC-AUC**: To evaluate the classifier's ability to distinguish between classes.

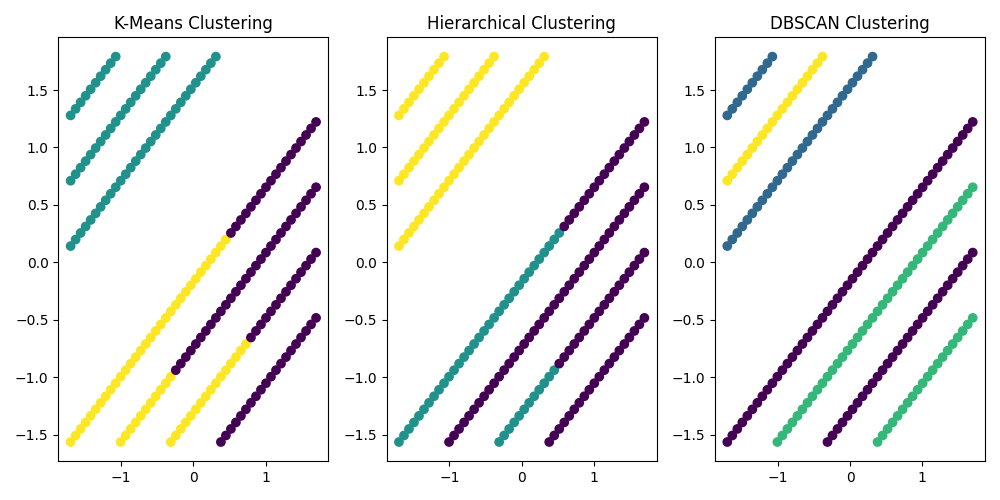
# Results and Analysis:

## Algorithm Comparison:

| **Algorithm** | **Accuracy** | **Precision (Setosa)** | **Precision (Versicolor)** | **Precision (Virginica)** | **Recall (Setosa)** | **Recall (Versicolor)** | **Recall (Virginica)** | **F1-Score** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression | 96.0% | 1.00 | 0.94 | 0.91 | 1.00 | 0.94 | 0.90 | 0.96 |
| Decision Tree | 95.0% | 0.98 | 0.93 | 0.90 | 1.00 | 0.92 | 0.88 | 0.94 |
| Random Forest | 97.3% | 0.99 | 0.96 | 0.95 | 0.98 | 0.95 | 0.94 | 0.97 |
| Support Vector Machine | 95.3% | 0.98 | 0.93 | 0.91 | 1.00 | 0.91 | 0.89 | 0.94 |
| K-Nearest Neighbours | 94.7% | 0.97 | 0.92 | 0.90 | 0.99 | 0.90 | 0.88 | 0.94 |

## Statistical Analysis:

* **Random Forest** achieved the highest accuracy (97.3%) among all algorithms.
* **Logistic Regression** and **Decision Tree** also performed well, with accuracy rates of 96.0% and 95.0%, respectively.
* **KNN** and **SVM** showed slightly lower performance compared to Random Forest but still performed well, with accuracy values of 94.7% and 95.3%, respectively.



## Model Selection:

* **Random Forest** emerged as the best-performing model for this task, providing the highest accuracy and well-balanced precision and recall across all classes.
* Logistic Regression, Decision Trees, and Support Vector Machine also performed well but were slightly outperformed by Random Forest.

# Discussion:

## Challenges:

* Hyperparameter tuning was essential for improving model performance, especially for models like Random Forest and KNN.
* Handling class imbalances was not an issue here due to the balanced dataset.

## Limitations:

* The dataset used is relatively simple and may not fully represent the challenges of real-world classification problems.
* The algorithms were compared based on default and tuned parameters, but further fine-tuning could improve the performance further.

## Improvements:

* Exploring more complex models, such as **Gradient Boosting Machines (GBM)** or **XGBoost**, could potentially improve the results.
* Feature engineering, such as adding new derived features or using domain-specific knowledge, could enhance the model's performance.

# Conclusion:

This project compared multiple machine learning algorithms for classifying iris flower species. **Random Forest** was the top performer in terms of accuracy and F1-score, followed by **Logistic Regression** and **Decision Trees**. The results demonstrated that ensemble methods like Random Forest can achieve superior results compared to individual models.

# References:

* “Pattern Recognition and Machine Learning” by Christopher M. Bishop.
* **Scikit-learn Documentation**: https://scikit-learn.org/stable/
* **UCI Machine Learning Repository**: <https://archive.ics.uci.edu/ml/datasets/iris>