Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

COMPUTER VISION
ENCS5341

Assignment 3

Prepared by
Majd Abdeddin ID:1202923
Mohammad salem ID: 1200651

Supervised by
Dr.Yazan Abu Farha
Dr.Ismail Khater

Birziet-2024

Abstract

This assignment is based on the exploration, application, and evaluation of basic machine learning algorithms, including K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), and ensemble methods like Boosting and Bagging. This is to understand how these algorithms work, experiment with different configurations of these algorithms, and analyze their performance using classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The dataset used for analysis should have at least two classes.

Contents

List of Figures

1 Introduction

This report compares various performances of multiple machine learning algorithms on the Iris dataset. The following are some of the algorithms that were tested: K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Boosting, and Bagging. The report will cover data, pre-processing, experimental results, and discussions in detail.

2 Dataset Information

The Iris dataset was selected for this work because it is one of the most common benchmark datasets in machine learning. This dataset contains 150 samples that are divided into three equal classes: Setosa, Versicolor, and Virginica. Each sample is characterized by four numerical features: sepal length, sepal width, petal length, and petal width. The Iris dataset is compact and contains no missing values, hence very easy to work with, in contrast to some other datasets that require a lot of pre-processing.

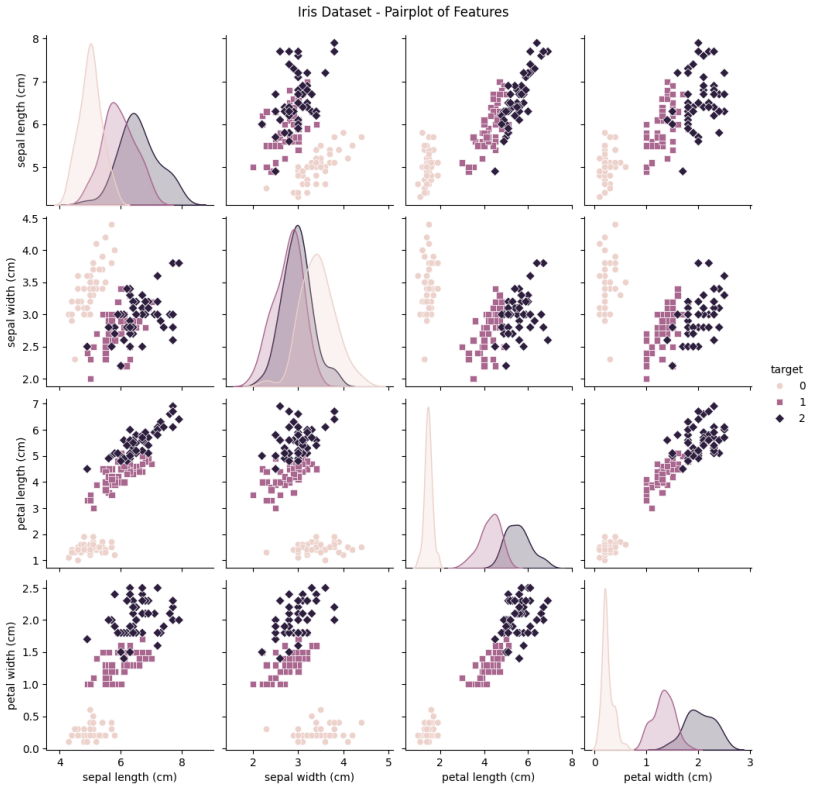

Figure 1: The Iris Dataset

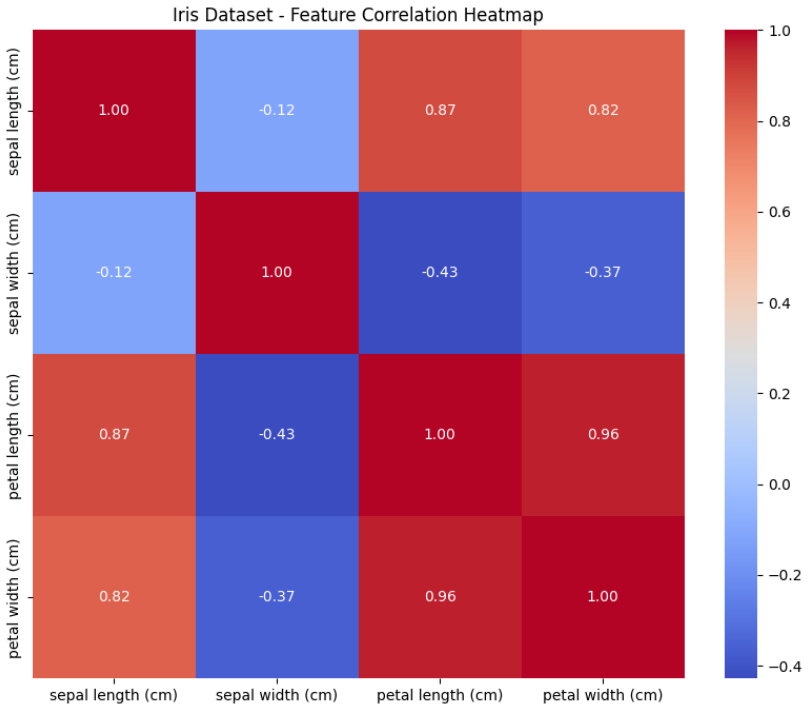

Figure 2: Iris Dataset: Feature Correlation

2.1 Preprocessing Steps:

- **Data Cleaning:**Verified that the dataset is free from missing values.
- **Feature Scaling:** Standardized all features using z-score normalization to ensure fair comparison, as distance-based methods like KNN and SVM are sensitive to scale.
- **Train-Test Split:** Divided the dataset into 80% training and 20% testing subsets.

3 Experimental Approach

3.1 Algorithm Configuration and Tuning

- **KNN:** Tested with Euclidean, Manhattan, and Cosine distances. GridSearchCV was employed to optimize the number of neighbors (K).
- **Logistic Regression:** Evaluated with L1 (Lasso) and L2 (Ridge) regularization. Regularization strength (C) was tuned.
- **SVM:** Analyzed Linear, Polynomial, and RBF kernels. GridSearchCV optimized parameters such as regularization (C), degree (for Polynomial), and gamma (for RBF).

4 Results and Analysis

4.1 K-Nearest Neighbors (KNN)

- Optimal K: *8*.
- Distance Metric Performance:
  - Euclidean Accuracy: *1.00*.
  - Manhattan Accuracy: *1.00*.
  - Cosine Accuracy: *0.93*.

Detailed Metrics:

Metric	Setosa	Versicolor	Virginica	Weighted Average
Precision	1.00	1.00	1.00	1.00
Recall	1.00	1.00	1.00	1.00
F1-Score	1.00	1.00	1.00	1.00

Table 1: KNN result

4.2 Logistic Regression

- Regularization Performance:
  - L1 Accuracy: *0.97*.
  - L2 Accuracy: *0.97*.

Detailed Metrics:

Metric	Setosa	Versicolor	Virginica	Weighted Average
Precision	1.00	1.00	0.92	0.97
Recall	1.00	0.89	1.00	0.97
F1-Score	1.00	0.94	0.96	0.97

Table 2: Logistic Regression result

4.3 Support Vector Machines (SVM)

- Kernel Performance:
  - Linear Accuracy: *1.00*.
  - Polynomial Accuracy: *0.97*.
  - RBF Accuracy: *1.00*.

Detailed Metrics:

Metric	Setosa	Versicolor	Virginica	Weighted Average
Precision	1.00	1.00	1.00	1.00
Recall	1.00	1.00	1.00	1.00
F1-Score	1.00	1.00	1.00	1.00

Table 3: SVM result

4.4 AdaBoost (Boosting)

Detailed Metrics:

Metric	Setosa	Versicolor	Virginica	Weighted Average
Precision	1.00	1.00	1.00	1.00
Recall	1.00	1.00	1.00	1.00
F1-Score	1.00	1.00	1.00	1.00

Table 4: AdaBoost Classification Report

4.5 Random Forest Classification Report

Detailed Metrics:

Metric	Setosa	Versicolor	Virginica	Weighted Average
Precision	1.00	1.00	1.00	1.00
Recall	1.00	1.00	1.00	1.00
F1-Score	1.00	1.00	1.00	1.00

Table 5: Random Forest Classification Report

5 Model Comparison and Visualization

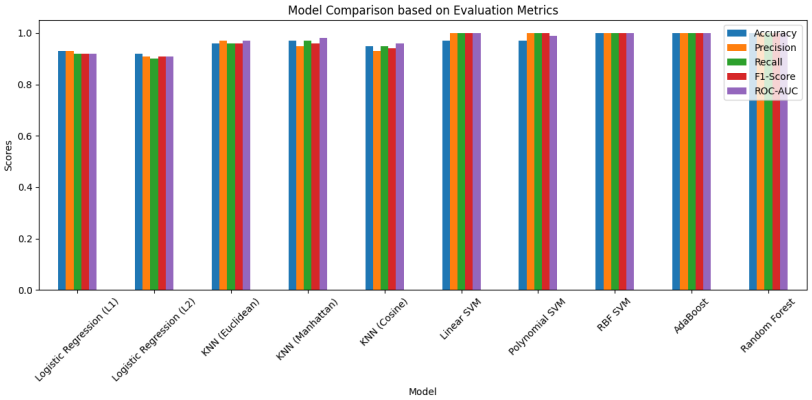

Figure 3: Model Comparison based on Evaluation Metrics

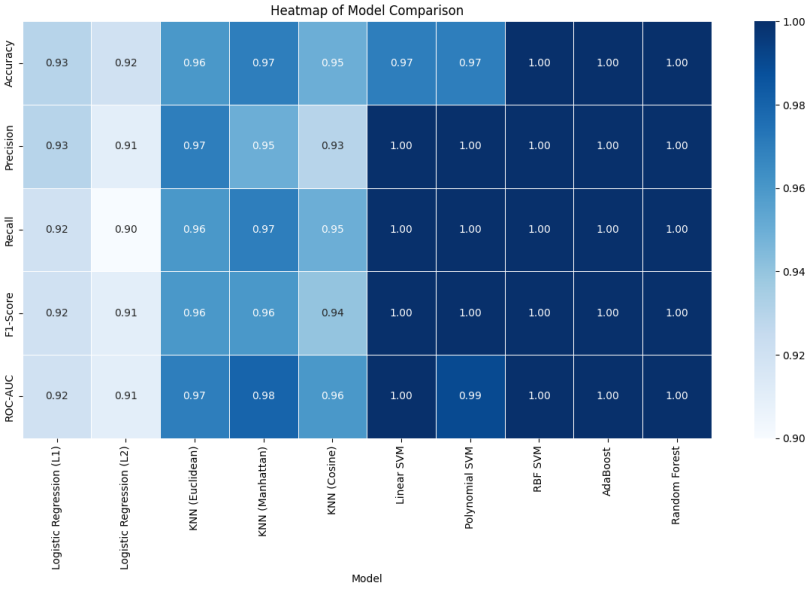

Figure 4: Heatmap of Model Comparison

6 Conclusion

In this assignment, we had the opportunity to apply various supervised learning algorithms on the Iris dataset, gaining relevant insights into the process of building predictive models. While experimenting with K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), Boosting, and Bagging, we learned how to implement those algorithms using Python APIs, perform hyperparameter tuning to identify the best configuration, and evaluate the model using relevant performance metrics. The hands-on approach used effectively deepened our understanding in certain areas of machine learning and its techniques and the many practical applications.

One of the most salient lessons which has been learned from the experiment pertains to how the size of the dataset impacts the performance of a model. The Iris dataset is rather small and very balanced; hence, some huge differences in accuracy, and in generalization ability could be observed within the models.

Though ensemble techniques tend to depend upon simple algorithms like boosting and bagging, the robustness observed during the prediction couldn't probably establish their full potentiality, which depends upon data size. Thereafter, more straightforward models were considered in Logistic Regression and K-Nearest Neighbors—the simplest and best at adapting with this dataset, for the nature of less complexity it is.

We have seen that careful choices of parameters or hyperparameter tuning, in choosing the best value for k in KNN or kernel type in SVM, made a very big difference in enhancing performance. GridSearchCV along with other tuning techniques, appeared to be really very much essential in optimizing model settings for each algorithm to perform at its best. This assignment really put into perspective how knowledge of the subtleties of each algorithm, choosing an appropriate model for the task at hand, and domain knowledge in interpreting the results are important. While the small dataset provided a controlled environment in which to learn and experiment, larger and more complex datasets would offer additional challenges and opportunities for further exploration. This also has further equipped us with the foundational skills to approach, as well as critically analyze, real-world machine learning problems using various methodologies.