REPORT

**Introduction-**

Understanding and using several techniques available in the Python sci-kit-learn module on the built-in digits and Iris datasets is the main goal. Six distinct classifiers are compared, their performance is assessed, their hyperparameters are adjusted, and their efficacy is examined regarding accuracy and runtime performance.

**Datasets Used-**

This analysis made use of two datasets:

1. The Digits Dataset is an integrated scikit-learn dataset. We utilize this dataset to assess each of the six classifiers.

2. The Iris Dataset is a dataset, that contains measurements for 150 iris blooms belonging to three distinct species. This dataset was used to assess a subset of classifiers.

**Classifiers-**

This is the following classifiers that are used:

1. Perceptron

2. Logistic Regression

3. Linear Support Vector Machine (SVM)

4. Non-linear SVM using Radial Basis Function (RBF) kernel

5. Decision Tree

6. K-Nearest Neighbors (KNN)

**Tuning using Hyperparameter-**

For every classifier—except the Decision Tree and KNN, which had no hyperparameter adjustments made in this analysis—GridSearchCV was utilized to fine-tune the parameters. Among the adjusted hyperparameters were:

1. C for Logistic Regression and SVMs

2. Alpha and Eta0 for Perceptron

**Results-**

These are the values we get when we change for two models in the hyperparameters:

Evaluating for Digits Dataset

Perceptron:

This is Training Accuracy: 0.8197

This is Testing Accuracy: 0.7312

This is Training Time: 0.7822 sec.

This is Best Parameter: {'model\_\_alpha': 0.01, 'model\_\_eta0': 1}

Logistic Regression:

This is Training Accuracy: 0.7365

This is Testing Accuracy: 0.8500

This is Training Time: 6.7588 sec

This is Best Parameter: {'model\_\_C': 0.001}

Linear SVM:

This is Training Accuracy: 0.7182

This is Testing Accuracy: 0.8565

This is Training Time: 0.2425 sec.

This is Best Parameter: {'model': 100}

Non-linear SVM (RBF):

This is Training Accuracy:0 .8881

This is Testing Accuracy: 0.8136

This is Training Time: 1.0905 sec

This is Best Parameter: {'model\_\_C': 0.01}

Decision Tree:

This is Training Accuracy: 0.7900

This is Testing Accuracy: 0.7563

This is Training Time: 0.2007 sec

This is Best Parameter: {'model\_\_C': 1}

KNN:

This is Training Accuracy: 0.8688

This is Testing Accuracy: 0.7501

This is Training Time: 0.5033 sec

This is Best Parameter: {'model\_\_C': 0.001}

Evaluating for Iris Dataset

Logistic Regression:

This is Training Accuracy: 0.7267

This is Testing Accuracy: 0.6000

This is Training Time: 0.0989 sec

This is Best Parameter: {'model\_\_C': 1}

Non-linear SVM (RBF):

This is Training Accuracy: 0.9031

This is Testing Accuracy: 0.8323

This is Training Time: 0.0656 sec

This is Best Parameter: {'model\_\_C': 100}

Decision Tree:

This is Training Accuracy: 0.8910

This is Testing Accuracy: 0.7000

This is Training Time: 0.1114 sec

This is Best Parameter: 'model\_\_C': 0.0001}

These are the values of the Best hyperparameters and the final output:

Evaluating for Digits Dataset

Perceptron:

This is Training Accuracy: 0.9687

This is Testing Accuracy: 0.9500

This is Training Time: 0.7822 sec

This is Best Parameter: {'model\_\_alpha': 100, 'model\_\_eta0': 100}

Logistic Regression:

This is Training Accuracy: 0.9889

This is Testing Accuracy: 0.9667

This is Training Time: 7.0028 sec

This is Best Parameter: {'model\_\_C': 1}

Linear SVM:

This is Training Accuracy: 1.0000

This is Testing Accuracy: 0.9750

This is Training Time: 0.2425 sec

This is Best Parameter: {'model\_\_C': 1000}

Non-linear SVM (RBF):

This is Training Accuracy: 1.0000

This is Testing Accuracy: 0.9806

This is Training Time: 1.8585 sec

This is Best Parameter: {'model\_\_C': 10}

Decision Tree:

This is Training Accuracy: 1.0000

This is Testing Accuracy: 0.8583

This is Training Time: 0.0207 sec

This is Best Parameter: None

KNN:

This is Training Accuracy: 0.9868

This is Testing Accuracy: 0.9750

This is Training Time: 0.0033 sec

This is Best Parameter: None

Evaluating for Iris Dataset

Logistic Regression:

This is Training Accuracy: 0.9832

This is Testing Accuracy: 0.9000

This is Training Time: 0.0898 sec

This is Best Parameter: {'model\_\_C': 100}

Non-linear SVM (RBF):

This is Training Accuracy: 0.9832

This is Testing Accuracy: 0.9333

This is Training Time: 0.0657 sec

This is Best Parameter: {'model\_\_C': 1}

Decision Tree:

This is Training Accuracy: 1.0000

This is Testing Accuracy: 0.9000

This is Training Time: 0.0014 sec

This is Best Parameter: None

**Conclusion-**

Specific parameter settings that maximized performance for every classifier were found through the hyperparameter tuning process. For example, to prevent overfitting, a lower C value for Logistic Regression was used as a preference for a simpler model; in contrast, higher C values helped the SVM classifiers by providing greater flexibility in the decision border. So, it is important to select the correct classifier and adjust hyperparameters depending on the dataset and the task at hand, as this comparative analysis of scikit-learn classification methods highlights. Between accuracy and generalization, the SVM with RBF kernel proved to be an especially effective tool for both datasets. Yet, based on the characteristics of the data and the needs of the application, each classifier's efficacy can differ.