MINI PROJECT - 2

COMP472

Comparison Between the two Classifiers:

On the Iris dataset, both Classifiers perform very well with perfect precision, recall, and F1 scores. The Decision Tree Classifier achieves an accuracy score of 1.0, while the MLP Classifier also achieves an accuracy of 1.0. There were no misclassified instances in the test set for both the MLP and the Decision Tree Classifier. Same results on many different runs here indicate that it's easy for both the Classifiers to learn the Iris dataset.

On the other hand, the difference between the performance of the both Classifiers increase in MNIST digits dataset. The MLP Classifier outperforms the Decision Tree Classifier reaching 0.98 accuracy while the decision tree reaches 0.86. This difference is indicative of the fact that MLP Classifier can handle complex and non-linear relationships better than the Decision tree Classifier.

Additionally, the Decision Tree Classifier's graphical representation offers insights into the decision-making process, whereas the MLP Classifier is a closed system and does not offer this level of transparency. The MLP Classifier, however, provides greater flexibility for managing intricate non-linear connections between input features and desired outcomes.

A closer inspection of the error cases in the MNIST dataset reveals a systematic difference between the two Classifiers. The Decision Tree Classifier tends to confuse similar-looking digits, such as 2, 3, 9 and 8, resulting in relatively high false-positive and false-negative rates. This can be observed by looking at the classification report and confusion matrix of the Decision Tree Classifier where the recall score is 0.69, which means that 31% of instances that are actually class 8 are misclassified as something else and the confusion matrix shows that the Classifier is confusing 2, 3 with 8. Inspecting the error cases can reveal further information. In contrast, the MLP Classifier demonstrates a better ability to distinguish between these digits, resulting in significantly lower false-positive and false-negative rates. This effect can be observed after a number of runs.

In conclusion, the complexity and distribution of the input characteristics and target outputs in the dataset have a significant impact on the performance of the Classifiers. On the Iris dataset, both the decision tree and MLP Classifiers perform perfectly, however on the more complex MNIST dataset, the MLP Classifier outperforms the Decision Tree Classifier. MLP Classifier is better suited for image classification tasks, where the input features exhibit complex non-linear relationships as suggested by the systematic difference observed in the error cases between the two Classifiers.

Description of each run:

Decision Tree on Iris Dataset

For this run, we trained a Decision Tree Classifier on the popular Iris dataset. The Iris dataset consists of 150 instances with 4 features: sepal length, sepal width, petal length, and petal width. The dataset is commonly used in machine learning research, and it has 3 classes representing different species of Iris flowers.

The goal of this run was to evaluate the performance of the Decision Tree Classifier on this dataset. During training, we split the dataset into a training set and a test set, with 80% of the instances used for training and 20% used for testing.

The Decision Tree Classifier was trained with default parameters using scikit-learn, the trained model was used to make predictions on the test set. The results of this run were excellent, with the Decision Tree Classifier achieving perfect precision, recall, and F1 scores. The accuracy score was also perfect, with no misclassified instances in the test set. These results indicate that the Decision Tree Classifier is able to learn the Iris dataset very easily and accurately.

MLP Classifier on Iris Dataset

In this run, MLP Classifier was trained on the same Iris dataset used. The goal was to compare the performance of the MLP Classifier with that of the Decision Tree Classifier on this dataset. Same training set and a test set that were used in the previous run were used here, with 80% of the instances used for training and 20% used for testing.

The MLP Classifier was trained with 1 layer containing 100 neurons, 1000 max iterations, Alpha (L2 Regularization) of 0.0001, solver as Adam, and initial learning rate at 0.001 using scikit-learn. The trained model was used to make predictions on the test set. The results of this run were also excellent, with the MLP Classifier achieving perfect precision, recall, and F1 scores. The accuracy score was also perfect, with no misclassified instances in the test set. These results indicate that the MLP Classifier is also able to learn the Iris dataset very easily and accurately, and it performs similarly to the Decision Tree Classifier on this dataset.

Decision Tree Classifier on MNIST Dataset

For this run, The Decision Tree Classifier was trained on the popular MNIST handwritten digits dataset. The dataset was loaded using the sklearn's sklearn.datasets.load_digits method. The MNIST dataset in sklearn consists of 1797 grayscale images of handwritten digits, with each image being 8x8 pixels. The goal of this run was to evaluate the performance of the Decision Tree Classifier on this more complex dataset.

During training, the dataset was split into a training set and a test set, with 80% of the images used for training and 20% used for testing. The dataset had flattened images making it 64 features for each instance. The Decision Tree Classifier was trained with default parameters using scikit-learn, and the trained model was used to make predictions on the test set. The results of this run were not as good as the previous runs, with the Decision Tree Classifier achieving an accuracy score of 0.86. This indicated that the Decision Tree Classifier struggles to predict more complex dataset.

MLP Classifier on the MNIST Dataset

In this run, we used an MLP Classifier to classify handwritten digits from the MNIST dataset. The goal was to compare the performance of MLP Classifier with the Decision Tree Classifier on the MNIST dataset. The same training and test split, 80% training data and 20% testing data, from the previous run was used to train and test the Classifier.

The MLP Classifier was trained with 1 Hidden Layer with 100 neurons, 1000 Max iterations, Alpha (L2 Regularization) of 0.0001, solver as Adam, and initial learning rate at 0.001, and the trained model was then used to make predications on the test data. The results were better than the Decision Tree Classifier. It achieved the accuracy score of 0.975 with good macro precision, recall, and F1 scores, indicating that the MLP Classifier can learn the complex relationship better than the Decision Tree Classifier.

Evaluation of each run:

Decision Tree Classifier on Iris Dataset

The Decision Tree Classifier achieved perfect accuracy, precision, recall, and F1 scores on the Iris dataset. The model also did not encounter any misclassified instances in the test set. The Training time of the Decision Tree Classifier was also low. This high level of accuracy suggests that the Decision Tree Classifier is an excellent choice for this type of dataset. However, it is worth noting that the Iris dataset is relatively small, and the Decision Tree Classifier may not perform as well on larger and more complex datasets.

MLP Classifier on Iris Dataset

In this run, MLP Classifier was used to classify flowers in the Iris dataset. The MLP Classifier achieved perfect accuracy on the test set which is same as the Decision Tree Classifier. However, The MLP Classifier required more computing resources compared to the Decision Tree Classifier and took relatively more time to train. Although the model performed perfecting on the test dataset indicating that it’s able to learn well on the dataset, the time required was higher than the Decision Tree Classifier hence the Decision Tree Classifier is the best choice when going with simple and small dataset.

Decision Tree Classifier on MNIST Dataset

In this run, Decision Tree Classifier was used to classify handwritten digits in the MNIST dataset. The Decision Tree Classifier was much faster than the MLP Classifier and achieved an accuracy of 0.86 on the test set. However, this accuracy is significantly lower than that of the MLP Classifier. One advantage of the Decision Tree Classifier is that it is easy to interpret and can provide insight into the decision-making process of the model. However, it may not be suitable for more complex datasets or those with a large number of features as indicated by the accuracy score. The Decision Tree model also struggled to learn the differences between similar looking digits such as 3 and 8.

MLP Classifier on MNIST Dataset

In this run, we used an MLP Classifier to classify handwritten digits in the MNIST dataset. The MLP Classifier was able to achieve an accuracy of 0.975 on the test set, which is quite impressive. However, the training process took a considerable amount of time due to the size of the dataset and the complexity of the MLP Classifier. The high accuracy on this dataset achieved by the MLP Classifier suggests that the Classifier is able to learn complex data and relationships quite effectively and it will perform better than the Decision Tree Classifier on complex and large datasets.

Variation from the Lab Solution:

The optimizer for the MLP Classifier was changed from SGD (Stochastic Gradient Descent) to Adam. Both SGD optimizer and Adam optimizer were tested, Adam optimizer’s loss was significantly lower than that of SGD optimizer and it converged faster than SGD optimizer making it the optimal choice for both of the dataset. Accuracy score also improved with the

Adam optimizer. For Iris Dataset Adam converged at only 613 iterations while SGD didn’t converge even after 1000. Increasing the max iteration to 2000 revealed that SGD converges after 1157 iterations. For the MNIST Dataset, Adam converges after just 140 iterations while SGD converges after 240 keeping everything else constant. The Accuracy score using SGD for Iris dataset was the same as the Adam optimizer but in the case of MNIST Dataset, the accuracy score for SGD was 0.9667 while for Adam it was 0.975.