Machine Learning

Olivetti Face Data – Decision Trees

Download the Olivetti faces dataset.
Visit https://scikit-learn.org/0.19/datasets/olivetti_faces.html
There are 40 classes (corresponding to 40 people), each class having 10 faces of the individual; so there are a total of 400 images.
Here each face is viewed as an image of size 64 × 64 (= 4096) pixels; each pixel having values 0 to 255 which are ultimately converted into floating numbers in the range [0,1].

Split the dataset into train
and test parts. Do this splitting randomly 10 times and report the average accuracy.
You may vary the test and train dataset sizes.

Build a decision tree using the training data. Tune the parameters
corresponding to pruning the decision tree. Use the best decision tree to classify
the test dataset and obtain the accuracy. Use misclassification impurity also.

(a) SOLUTION

CODE:

Please find the code for Decision Tree Classifier committed as *DecisionTrees_OlivettiFaceData_Impl.py*

- **DecisionTreeClassifier** from sklearn is used on a train size of 320 samples (80%) of the olivetti dataset that contains a total of 400 samples.
- Using a library function called **GridSearchCV** of sklearn's model selection package, best parameters from the listed hyper parameters of the model are found.
- Now using the best estimator, the **DecisionTreeClassifier** is run 10 times on randomly split data using randomly selected test and train sizes from range 0.2 to 0.35.
- Accuracy is found after fitting this best decision tree and average accuracy is computed for 10 iterations.

RESULT:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=50,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')

Test Size Ratio: 0.25
Accuracy : 0.56
Test Size Ratio: 0.35
Accuracy : 0.5428571428571428
Test Size Ratio: 0.25
Accuracy : 0.61
Test Size Ratio: 0.25
Accuracy : 0.49
Test Size Ratio: 0.3
```

```
Accuracy : 0.625
Test Size Ratio: 0.35
Accuracy : 0.6
Test Size Ratio: 0.3
Accuracy : 0.575
Test Size Ratio: 0.25
Accuracy : 0.58
Test Size Ratio: 0.2
Accuracy : 0.6125
Test Size Ratio: 0.2
Accuracy : 0.575
avg accuracy  =  0.5770357142857143
```

INFERENCE/ANALYSIS:

- For tuning the parameters of the decision tree, max_depth, min_samples_leaf and Gini and Entropy criterion were considered.

- **Pruning was tried with following parameters: 'max_depth': [5, 10, 20, 50, 100], 'min_samples_leaf': [1, 2, 5, 10, 20], 'criterion': ["gini", "entropy"]**

- **Despite using the best fit of Max_depth = 50, Criterion = Gini, Min_samples_leaf = 1, Min_samples_split = 2, decision tree generally shows a lower accuracy for this Olivetti dataset.**

- Decision Trees in general tries to overfit the samples. Literature mentions that trees are prone to inaccuracies as they are greedy and deterministic.

- **In this dataset the number of features are too high i.e – 4096 features. It is a high dimensional dataset and a single decision tree despite pruning with hyperparameters and using a cross validator like GridSearchCV , is only able to achieve an avg accuracy of 0.577 for 10 random trials.**

- **For such datasets, other classifers like Random Forest and XGboost provide better accuracy as can be seen in following sections.**

Decision tree did not give good accuracy for high dimensional featured dataset like an olivetti dataset with 4096 features due to risk of overfitting.
In order to solve this limitation of decision tree, classifiers like Random forest are generally preferred that use multiple decision tree and XGboost classifier which also uses multiple decision trees and a host of other parameters outperforms the simple decision tree despite best parameterization.

RESOURCES USED FOR THE ASSIGNMENT:

- **Environment:**
  Anaconda, Jupyter notebook
- **Software :**
  Python
  Python libraries/modules: Pandas, Numpy, SkLearn ,XGboost etc