Assignment 4 Summary : Classifiers

Name : Harika Malapaka **Student ID :** hsmalapa

High Level Summary of tools used:

PyCharm, and IDE of Python was used to implement this project. There are many classifier packages available in Python. These are the ones I've used :

pandas – To load the data file as csv

svm and DecisionTreeClassifier from sklearn – > To use these two classifiers.

accuracy_score –> to calculate accuracy of the model.

Precision_recall_fscore_support --> To calculate F1 score

cross_val-score –> to perform 10K fold cross validation as mentioned in description.

test_train_split –> to split the data into training data, training labels, testing data and testing labels.

NOTE : Structuring the data like converting Strings to Float were done on Excel and it's bought as a csv file to pyCharm interface. Thus cleaning/structuring was not part of the code.

Steps taken while coding :

1. Install all the required packages
2. Load the data into the tool as csv file.
3. Define feature set and label set
4. Feature set :
    a. Unique Step Id
    b. Grade
    c. Student Id
    d. Total No of observations in the session

    e. Students Activity (Integers were assigned to the activity name. Ex : Wholecarpet as 1, Wholedesks as 2.
    f. Number of task transitions
    g. Number of activities
    h. Number of format changes required
    i. Number of formats
    j. Duration
    k. Total time
5. Now select the Labels : ONTASK value.
6. Now split the data into training data, training labels, testing data and testing labels using train_test_split function.
7. Define 2 classifier functions by giving the parameters required.
8. Fit the training data and training labels into these classifiers using fit function.
9. Predict your model using predict function, passing test data as input.
10. Calculate the accuracy of the model by comparing predictions in the above step with test labels.
11. Cross validation :
    - Send the classifier function along with training data, training labels and value of cv (no of K folds in cross validation) as input for the function cross_val_score.
    - Now we can see the accuracy in each of the 10 iterations (As we took cv=10).

Results of Analysis :

The predictions of the 2 classifiers can be seen. We can also give our own record similar to training data to get a prediction.

The accuracy of Decision Tree Classifier is 63.27%

The accuracy of SVM classifier is 67.50%

The accuracy of Decision Tree classifier (when 10k-fold) is 60.95%

The accuracy of SVM classifier (when 10k-fold) is 67.07%

The F1 score of Decision Tree classifier is 0.533

The F1 score of SVM classifier is 0.403

Quesstion and Answers :

1. *Which classifier performed better?*
   Ans :
   The SVM Classifier performed better.

2. *How did the performance change from static sampling to cross-validation?*
   Ans : In Decision Trees, the accuracy got reduced by 3% when Cross Validation was implemented.
   In SVM Classifier, the accuracy was reduced by 0.36 % when Cross Validation was implemented.

   This means, to avoid overfitting, cross-validation is a better option.

3. *What feature of the algorithm's inductive bias may explain its performance?*
   Ans : In Decision Tree, the length of the tree can explain the inductive bias. Short trees are preferred than longer ones. However, because of heuristic search, we cannot exactly determine where the bias will happen.

   In SVM, distinct classes are separated by large margins. We can see the distance through decision_function().

4. *How did you control for overfitting in the dataset?*

- *In Decision tree classifier, I reduced the number of parameters, max_depth was given as 5 and max_leaf_nodes was given as 3 and class_weight ='balanced' in the constructor function of Decision Tree classifier.*
- *The number of features is less than the data size, so SVM can restrain from overfitting. Anyway the kernel parameter by default is 'rbf' which prevents from overfitting.*

And since dataset is huge enough, it's ensured that overfitting doesn't happen that easily.