Week 8 - DT and RF

DS3010 - Introduction to Machine Learning

Instructions

1. Provide commented, indented code. Variables should have meaningful names.
2. Submit one .ipynb file containing all answers. The name should be [student name][roll_number] assignment[number].ipynb
3. Read the questions carefully before answering. If a question asks to follow a particular approach or to use a specific data structure, then it must be followed.
4. Write questions in separate text blocks in Jupyter Notebook before the code blocks containing answers.
5. All plots should have appropriate axis labels, titles, and legends.
6. **The student who finds the best accuracy using hyperparameter tuning will receive an additional mark.**

Tasks for the Lab

1. Preprocessing (5)

A. Load the given student_performace_data_.csv.
B. Check the duplicate rows and nan values in the datasets.
C. Drop the unnecessary columns (mention the reasons in comment).
D. Split the data into train and test in an 8:2 ratio.
E. Visualize the distribution for the target class “GradeClass”.

2. Decision Tree (5)

A. Create an instance of a decision tree classifier and fit the model.
B. Predict the labels for train and test data and print the classification report.
C. Print precision, recall, f1-score for each class with the help of ‘average’ parameter for both train and test data.
D. Store the predictions of test data in y_test with a column name ‘DT_test_predicted’.
E. print the ‘depth of the tree’, ‘number of leaves’ for the above learned decision tree.

3. Random Forest (4)

A. Create an instance of a Random Forest classifier and fit the model.
B. Predict the labels for train and test data and print the classification report.
C. Print precision, recall, f1-score for each class with the help of ‘average’ parameter for both train and test data
D. Store the predictions of test data in y_test with a column name ‘RT_test_predicted’.

4. Hyperparameter Tuning With GridSearchCV (11)

I. Decision Tree

A. Define a param_grid dictionary with the list of permissible values of your choice for the hyper-parameters "criterion”, "splitter", "max_depth", "min_samples_split", "min_samples_leaf", "max_leaf_nodes", "max _features"
B. Print the best parameters and train the classifier with best parameters.
C. Predict the labels for test data and Store the predictions of test data in the above CSV file with a column name ‘Tuned_DF_test_predicted’.
D. Print the classification report for test data.
E. Print precision, recall, f1-score for each class with the help of the ‘average’ parameter for test data.

II. Random Forest Classifier

A. Follow the same steps as above to tune the hyper parameters for Random Forest but use RandomisedSearchCV method with hyper parameters “n_estimators”, “max_features”, “max_depth”, “min_samples_split”, “min_samples_leaf”, “bootstrap” and Store the predictions in the above csv file with column name ‘Tuned_RF_test_predicted’
B. Download the final test prediction file and submit it along with the .ipynb file.