COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH

Machine Learning I

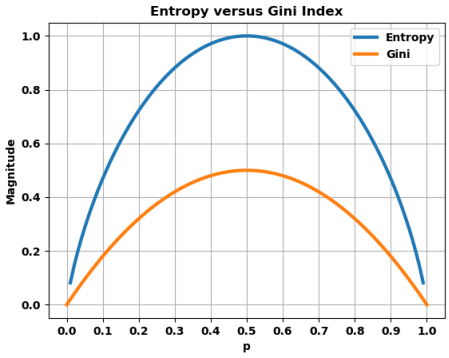
CS 5805

Homework # 7

In this LAB, you will learn how to develop Decision Tree classifier manually and how to train the decision tree classifier for real datasets using the pre-pruning and post pruning technique. The LAB is divided into two phases. Set the random_state = 5805 for all problems. Display numbers with 2-digit decimal precision.

Phase I:

1. Plot the entropy and Gini impurity (decision tree classifier) versus probability in one graph. Linewidth = 3. The final figure should be like below figure: [10pts]



2. Without use of python, develop a decision tree model that classifies the Play Tennis target in the following dataset. Graph the final tree and show all your work. Use the Entropy approach.
    a. What is the prediction for the test observation [Outlook=sunny, Temp=cool, Humidity=high, Wind=strong]? Show your work. [20pts]

3. Without use of python, develop a decision tree model that classifies the Play Tennis target in the following dataset. Graph the final tree and show all your work. Use the Gini Impurity approach.
    a. What is the prediction for the test observation [Outlook=sunny, Temp=cool, Humidity=high, Wind=strong]? Show your work. [20pts]

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Data set for problem three & four

Phase II:

In this phase of the LAB, you will work with the real dataset 'titanic.csv' which is available on the seaborn package repository. The objective is to compare the performance of two classifiers (Decision tree and Logistic regression) on the target variable 'survival' and pick the best classifier for this dataset. You must use python software to answer the questions in this LAB. Drop the missing observations as shown below.

```
import seaborn as sns
df = sns.load_dataset('titanic')
df.dropna(how='any', inplace=True)
```

4. No-Pruning Select numerical features and split the dataset to train-test using 80-20% method with random state of 5805. Keep the ratio of the target variable the same in the test and trains set [stratify]. Develop a decision tree-based model [without pruning] and display the score accuracy of the train and test set. Write your comments about the score accuracy of the train and test set. Display the parameter values of the tree after training. Display the graph of the tree using the sklearn package. [10pts]

   ccp_alpha', 'class_weight', 'criterion', 'max_depth', 'max_features', 'max_leaf_nodes', 'min_impurity_decrease', 'min_samples_leaf', 'min_samples_split', 'min_weight_fraction_leaf', 'monotonic_cst', 'random_state', 'splitter'

5. Pre-Pruning: Using a Grid Search with cross validation perform a search on the following parameters. Pick the range for the search in the hyperspace using grid search with 5-fold cross validation. Plot the final tree [after finding the optimum parameters for the tree] and the score accuracy of ye train and test on pruned tree. Write your comments about the score accuracy of the train and test set. Did the pre-pruning improve the performance of this classification? Explain your answer. The tuned parameters are: [10pts]

```
tuned_parameters = [{'max_depth': [1, 2, 3, 4, 5],
                     'min_samples_split': [20,30,40],
                     'min_samples_leaf': [10,20,30],
                     'criterion':['gini','entropy','log_loss'],
                     'splitter':['best','random'],
                     'max_features':['sqrt','log2']}]
```

6. Post-Pruning: Using python program, find the optimum alpha in the cost complexity function. Plot the Accuracy score of the train and test set. Display the optimum alfa parameter on the console. With the optimum alpha, develop the pruned decision tree. Display the final tree on the console. Write your comment about the effect of post pruning on the performance of the tree comparing with the pre-pruned tree and no pruned tree.[10pts]

7. Final classifier selection Develop a Logistic Regression [LR] classifier for the titanic dataset using the same train and test set created in step 4. Hint: No need to hyperspace search. Use the default parameters for the LR classifier. Display the score of accuracy on the test and train test. [5pts]

8. Final classifier selection Create a table and compare the accuracy, confusion matrix, recall, AUC, and ROC [display the curve with the AUC as the legend for three cases: DT pre-pruned, DT post-pruned, and LR] of the pre-pruned, post-pruned tree and logistic regression classifier. Which classifier works better for this dataset? Justify your answer. [15pts]

Upload a formal **report (as a single pdf**) plus **the .py file** through BB by the due date.