Subject: 19CSE305

Lab Session: 04

Notes:

1. Please read the assignment notes carefully and comply to the guidelines provided.
2. Code should be checked into the GitHub and the report to TurnItIn. Once done, please submit your assignments in Teams.
3. Code non-availability in GitHub shall be marked as zero.
4. Any content copy (statements, figures, codes etc.) from anywhere shall attract a penalty of 10 marks. If you obtain content from anywhere for illustration purposes, please cite the source to avoid penalty.
5. Snapshot / screenshot of code and results not allowed in the report. You may copy content from your own code & results and add to the report.
6. Provide data, code snippets or illustrations to support your answer, as applicable.

Please use the data associated with your own project.

Refer:

1. **https://scikit-learn.org/stable/modules/tree.html**
2. **https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html**

Main Section (Mandatory):

A1. For the data table provided below, calculate the entropy associated with each attribute / feature at the root node. Using this information, identify the first feature that you'd select for constructing the decision tree. Use Information Gain as the impurity measure to identify the root node. 'buys_computer' is the class label.

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

A2. Create a Decision Tree for the above data. Get the depth of the constructed tree.

```
model = DecisionTreeClassifier()
model = ml_model.fit(Tr_X,Tr_y)
model.score(Tr_X, Tr_y)#Training Set accuracy
print(model.get_depth())#print the tree depth
```

A3. Visualize the constructed tree with plot_tree() command. Following code snippet for help.

```
import matplotlib.pyplot as plt
from sklearn import tree
plt.figure(figsize=(70,20))
plot_tree(model, filled=True)
plt.show()
```

A4. Create a Decision Tree classifier on your project data. Study the accuracy for training and test data and infer the accuracy of tree construction. Plot the Decision Tree obtained above. Below code for help.

```
model = DecisionTreeClassifier()
model = ml_model.fit(Tr_X,Tr_y)
model.score(Tr_X, Tr_y)#Training Set accuracy
model.score(Te_X,Te_y)#Test Set Accuracy
```

A5. Impose a max_depth constraint on the tree construction. Construct the tree again and check the accuracies. Visualize the tree constructed with max_depth constraint.

```
model = DecisionTreeClassifier(max_depth=5)
```

A6. Study the criterion of the DT in the above model. Change the criterion to “Entropy” and study the model & graph. Find the differences between the default criterion and entropy criterion. Refer code below for criterion.

```
DecisionTreeClassifier(criterion="entropy")
```

A7. Construct a random forest classifier on your project data. Find the differences between the decision tree & random forest classifiers with the help of the performance metrics.

A8. Study the various parameters and model attributes of random forest classifier. Understand their significance in the behavior of the model.

Optional Section:

O1. Perform hyper-parameter tuning on the decision tree. After hyper-parameter tuning, check the hyper-parameter values for criterion, depth etc. Following code for help.

```
from sklearn.model_selection import GridSearchCV
```

```
from sklearn.cross_validation import  cross_val_score

tree_para =
{'criterion':['gini','entropy'],'max_depth':[4,5,6,7,8,9,10]}

model = GridSearchCV(DecisionTreeClassifier, tree_para, cv=5)

model.fit(X, Y)
```

O2. Study the various ensemble classifiers available under the package ***sklearn.ensemble***. Use them to compare the performance of these on your dataset.

Report Assignment:

1. Update your IEEE format report with the following.
    a. Update the Introduction section as per your revised understanding of the project and study of research papers.
    b. The implemented methods of DT and RF classifiers with their parameters and hyperparameters should be described in methodology section.
    c. Update the results section with various obtained results on your dataset for experiments in A4, A6 & A7. Discuss the model performance and state of training based on observation of performance metrics such as Precision, Recall, Accuracy, F-score, AUROC. Discuss on the overfit / underfit scenario for your classifiers.
    d. Discuss on the usage of pruning to avoid overfitting in decision trees. Relate that to your data and performance from experiment in A5.
    e. Update the conclusion section as appropriate.
2. Search to identify and download more relevant papers for your project. Study them and update the Literature survey section as appropriate.