# CO 544 Machine Learning and Data Mining Lab 05

E/17/297

**Task 1:** Build two decision tree classifiers with Gini index and entropy criteria for the given Wine.csv data set. More information on the dataset is available on UCI Machine Learning Repository (source: [https://archive.ics.uci.edu/ml/datasets/Wine](https://archive.ics.uci.edu/ml/datasets/Wine)).

First we have to preprocess the data

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Decision tree classifier
from sklearn.model_selection import train_test_split
from sklearn import metrics # scikit-learn metrics module for computing accuracy
```

```
[4] folder = '/content/drive/MyDrive/CO544/Lab 5/'

    import warnings
    warnings.filterwarnings('ignore') # Ignore warning messages
```

```
wine_df = pd.read_csv(folder + 'wine.csv', header=None, sep='\n')
wine_df.head() # Preview the dataset
```

```
wine_df = wine_df[0].str.split(',', expand=True)
wine_df = wine_df.drop(wine_df.columns[14], axis=1)
wine_df = wine_df.drop(index=0)
wine_df.head()
```

```
[11] col_names = ['Alcohol','Malic acid','Ash','Alcalinity','Magnesium','Total phenols',
                  'Flavanoids','Nonflavanoid phenols','Proanthocyanins','Color intensity','Hue',
                  'OD280/OD315 of diluted wines','Proline','Class']# Define new column names
    # Rename column names
    wine_df.columns = col_names
```

Next we have to define feature vector and target variable

```
[18] # Defining feature vector and target variable
     X = wine_df.drop(['Class'], axis=1) # Drop the target variable
     y = wine_df['Class']
```

Then split the data set into training set and test set.

```
# Splitting data
# 75% training and 25% test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)
X_train.shape, X_test.shape # Shapes of X_train and X_test
```

classifiers with Gini index

```
# Create decision tree classifier object
clf_gini = DecisionTreeClassifier(criterion='gini',
                                  max_depth=4,
                                  random_state=0)
```

classifiers with entropy

```
# Create decision tree classifier object
clf_entropy = DecisionTreeClassifier(criterion='entropy',
                                     max_depth=4,
                                     random_state=0)
```

**Task 2**: Demonstrate how decision trees deal with missing values.
If the dataset is large missing values can be ignored but not for dataset with smaller number of data. We can Divide the instances into pieces that have missing values.

**Task 3**: Evaluate the classifiers with suitable performance metrices.

Accuracy and confusion matrix for Gini index

```
[ ] # Train the classifier
    clf_gini.fit(X_train, y_train)

    DecisionTreeClassifier(max_depth=4, random_state=0)

[ ] # Predicting results for the test set
    y_pred = clf_gini.predict(X_test)

[ ] # Evaluating model
    print('Accuracy: ', metrics.accuracy_score(y_test, y_pred))

    Accuracy:  0.9555555555555556

[ ] # Confusion matrix
    from sklearn.metrics import confusion_matrix
    conf_mat = confusion_matrix(y_test,y_pred)
    print(conf_mat)

    [[18  0  0]
     [ 1 15  1]
     [ 0  0 10]]
```

Accuracy and confusion matrix for entropy

```
# Predicting results for the test set
y_pred = clf_entropy.predict(X_test)

# Evaluating model
print('Accuracy: ', metrics.accuracy_score(y_test, y_pred))

Accuracy:  0.9555555555555556

# Confusion matrix
from sklearn.metrics import confusion_matrix
conf_mat = confusion_matrix(y_test,y_pred)
print(conf_mat)

[[17  1  0]
 [ 1 16  0]
 [ 0  0 10]]
```

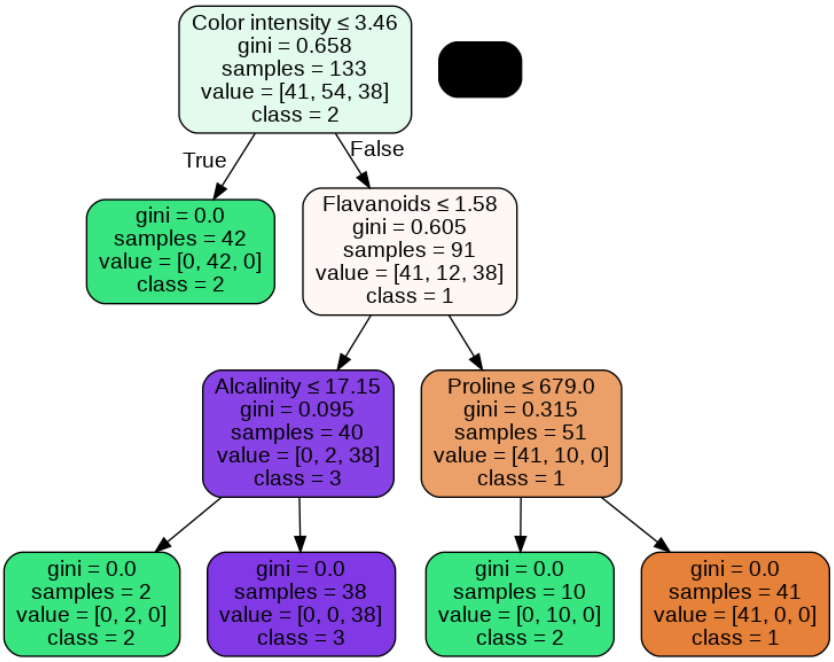
**Task 4**: Demonstrate how pruning can be applied to overcome overfitting of decision tree classifiers.

The pre-pruning process is used to stop a tree's growth early in order to keep the model from becoming overfitted. Prior to the training pipeline, it fine-tunes the hyperparameters. The decision tree is allowed to grow to its full depth after pruning.

Following that, the tree branches will be removed to prevent the model from becoming overfitted.

**Task 5**: Visualize decision trees.

Decision tree for Gini index




Decision tree for entropy