Practice 2.1 - Decision Trees

Machine Learning – LECD, LEEC

Informatics Engineering Department

Ex1

1.1 Import libraries:

- pandas, numpy
- sklearn.datasets <- load_iris
- sklearn.tree <- DecisionTreeClassifier
- sklearn.model_selection <- train_test_split
- sklearn.metrics <- accuracy_score
- sklearn.metrics <- confusion_matrix

1.2 Load IRIS dataset

Goals: Use Decision trees to create models;

```
#Start with imports
#To do: type your code here:

#Loading the iris data
#To do: type your code here:


print('Classes to predict: ', data.target_names)
#Result should be:
#Classes to predict:  ['setosa' 'versicolor' 'virginica']
```

```
    Classes to predict:  ['setosa' 'versicolor' 'virginica']
```

Ex2: Extracting data attributes (X) and target/ class labels (y)

```
#To do: type your code here:

#To do: type your code here:
#Extracting data attributes
#X =

### Extracting target/ class labels
#y =

X[:4] #First 4 iris features
#results should be:
#array([[5.1, 3.5, 1.4, 0.2],
#       [4.9, 3. , 1.4, 0.2],
#       [4.7, 3.2, 1.3, 0.2],
#       [4.6, 3.1, 1.5, 0.2]])

print('Number of examples in the data:', y.shape[0]) ## result should be 150
```

Ex3: Create train and test sets (use train_test_split), 25% for testing

```
#To do: type your code here:
```

Ex4: Create a Decision Tree (DT) classifier based on entropy (use DecisionTreeClassifier)

```
#To do: type your code here:
#clf =
```

Ex5: Train the DT classifier

```
#To do: type your code here:
#clf.
```

Ex6: Predict labels on the test set.

```
#To do: type your code here:

#y_pred =

print(y_pred.size) ## result should be 38
```

Ex7: Evaluate results (accuracy on train and test set with accuracy_score)

```
#To do: type your code here (fill the ...):

#print('Train data accuracy: ', ...)
#print('Test data accuracy: ', ...)

#Typical Output:
#Train data accuracy:  1.0
#Test data accuracy:  0.9736842105263158
```

Ex8: Evaluate results (confusion matrix)

```
#To do: type your code here (fill the ...):
#cf_matrix = ....
```

Ex9: Comment on results

```
#This code generates a graphical confusion matrix
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.heatmap(cf_matrix, annot=True, cmap='Blues')
ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Iris Category')
ax.set_ylabel('Actual Iris Category ');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['Setosa','Versicolor', 'Virginia'])
ax.yaxis.set_ticklabels(['Setosa','Versicolor', 'Virginia'])

## Display the visualization of the Confusion Matrix.
plt.show()
```

Generate the confusion matrix using the code above and comment on the results:

Ex 10: Visualize the tree

Use plot_tree(clf) importing tree from sklearn

```
#To do: type your code here (fill the ...):

#from ... import ...
#tree....
```

Ex11: Try different trees

def **init**(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)

```
#To do: type your code here
```

Comment on results with differeny trees.

Practice 2.2 - KNN

Machine Learning – LECD, LEEC

Informatics Engineering Department

Ex1: Generate a KNN Classifier and compare results.

```
#To do: type your code here (fill the ...):

from sklearn.neighbors import KNeighborsClassifier as KNN

#create classifier
#knn = ...
#train the model
#knn.fit(...)
#y_pred = knn.predict(...)

#print('Train data accuracy: ', ...)
#print('Test data accuracy: ', ...)
```

Ex2: Try different datasets (examples in Pratice 1)

```
#To do: type your code
from sklearn.datasets import load_breast_cancer
breastCancer = load_breast_cancer()

X = breastCancer.data
y = breastCancer.target
```