Data Science and Analytics (IT3080)

B.Sc (Hons) in IT (IT Specialization)

Year 3

SLIIT
Discover Your Future

Lab Sheet 04

Classification using Decision Tree

Activity 1

Open Jupyter Notebook and create a folder named lab 05. Download the data sets for lab 05 from the courseweb.

Explore the content of the zoo.csv file. The file contains names of set of animals, their features and a category they belongs to. Note that there are 7 categories to which each animal falls in to. The seven categories and the animals falling into those categories are as follows:

1. aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby, wolf
2. chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
3. pitviper, seasnake, slowworm, tortoise, tuatara
4. bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
5. frog, frog, newt, toad
6. flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
7. clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

Now create a new Notebook under Jupyter Notebook. Similar to the previous lab sheets you need to import the pandas library first. Additionally, you will need the sklearn library from which you would require two very important functions. Add the following statements to do this.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

Import the data in the Zoo.csv file to a data frame. Explore the content using methods familiar to you. Verify whether there are seven different categories in the class_type column.

Now split your dataset into two sets using the *iloc* method (lets name them as x and y). x should contain the features of the animals. Y should contain the class types.

For our exercise we need a training data set to train the classification model as well as some test data to check the accuracy of the model build. *train_test_split* function in *sklearn.model_selection* could be used for this purpose. The function accepts set of arrays as an input, split arrays or matrices input into random train and test subsets and returns them. Use the following statement to split the features and their corresponding class types in to two sets.

```
X_train,X_test,y_train,y_test = train_test_split(x,y, test_size=0.25)
```

Observe the content of the resulted data.

Building the classifier

Now we need to build our classifier. Use the following statement to develop the classifier using the decision tree. Note that the default method measure the quality of the split for the classifier is gini index.

```
<classifier name> = DecisionTreeClassifier(random_state=0)
```

Next we train the classifier using the training data set we have.

```
<classifier name>.fit(X_train,y_train)
```

Testing the model

Let's test the accuracy of the classifier now. This is done using the *DecisionTreeClassifier .score(x,y)* method. The method returns the mean accuracy on the given test data(x) and their corresponding class labels(y).

Making prediction

We will next try to make a prediction based using our model. If you want you can input a totally new test data and observe the output. In this case, lets select few rows from our test data itself. For making the predictions *predict(x)* returns the predicted class for the set of features x. Try,

```
<classifier name>.predict(X_test[10:15])
```

Compare the results you get with the class labels corresponding to the features you have in your test data.

Activity 2

Open the car.csv file available in the data sets. The file content how decisions have been made on accepting a car based on some of its features. The descriptions of the columns is given below.

Column	Values	Description
buying	v-high, high, med, low	buying price
maint	v-high, high, med, low	price of the maintenance
doors	2, 3, 4, 5-more	number of doors
persons	2, 4, more	capacity in terms of persons to carry
lug_boot	small, med, big	the size of luggage boot
safety	low, med, high	estimated safety of the car
Car	Unacc,acc, good, v-good	Car acceptability

Develop a classification model based on the data given.

Important: Note that the values in columns are categorical rather than numerical like the previous exercise. Most classification algorithm requires data in numerical format rather than categorical data. This could be done using pandas.factorize(*values*) method where values is an array of values. The method returns two series as output. First array is the numerical values for the input data. The second array is has the unique values in the input.

In this exercise, you could use a statement as follows to convert each categorical column in the dataframe to convert the data to numerical format.

<Dataframe name>[<column_name>],_ =pd.factorize(<datadframe name>[<column name>])

Additional Information: Visualizing the decision tree

Visualizing the created tree could be done using the graphviz library. Use the following code to do so.

```
from sklearn import tree
import graphviz
feature_names = all_x.columns
class_names=str(data['class_type'].unique())
dot_data = tree.export_graphviz(clf, out_file=None, filled=True, rounded=True,
                    feature_names=feature_names,  class_names=class_names)
graph = graphviz.Source(dot_data)
```