Step 1

```
import pandas as pd
animal_data = pd.read_csv('animal.csv')
```

Step 2

```
animal_data
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
animal_data["Label"] = label_encoder.fit_transform(animal_data["species"])
animal_data
Categories = list(label_encoder.inverse_transform([0,1]))
Categories
X = animal_data.drop(['Label', 'species'], axis=)
Y = animal_data.Label
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
                                                    random_state=0)
clf = DecisionTreeClassifier(criterion='entropy', max_depth=4,
                             random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_pred
print("Accuracy of train:", clf.score(X_train, y_train))
print("Accuracy of test:", clf.score(X_test, y_test))
print('\n')
print(classification_report(y_test, y_pred))
```

## Lab8. Animal Classification using Decision Trees

**Objectives**
In this lab, you will build ID3 and CART Decision Tree to classify whether an animal is a Mammal or Reptile.

**Learning Outcomes**
After completing this lab, you will be able to

- Create and import training dataset and test dataset
- Create ID3 Decision Tree using Entropy metric
- Create CART Decision Tree using Gini metric
- Visualize graph using graphviz

**Step1. [Create Dataset]**
Create the following dataset using Excel and save it as CSV file.

| Toothed | hair | breathes | legs | species |
|---|---|---|---|---|
| TRUE | TRUE | TRUE | TRUE | Mammal |
| TRUE | TRUE | TRUE | TRUE | Mammal |
| TRUE | FALSE | TRUE | FALSE | Reptile |
| FALSE | TRUE | TRUE | TRUE | Mammal |
| TRUE | TRUE | TRUE | TRUE | Mammal |
| TRUE | TRUE | TRUE | TRUE | Mammal |
| TRUE | FALSE | FALSE | FALSE | Reptile |
| TRUE | FALSE | TRUE | FALSE | Reptile |
| TRUE | TRUE | TRUE | TRUE | Mammal |
| FALSE | FALSE | TRUE | TRUE | Reptile |

**Step2. [Model building using ID3]**

- Import your data set
- Create DT model using 'entropy' criterion
- Perform training and testing
- Print accuracy and classification report.
- Interpret your results
- Visualize your DT model using graphviz

Install necessary package for visualization. The following sample code will help you to understand visualization.

```
with open("tree1.dot", 'w') as f:
     f = tree.export_graphviz(dtc_entropy1,
                              out_file=f,
                              max_depth = 4,
                              impurity = False,
                              feature_names = X.columns.values,
                              class_names = ['Reptile', 'Mammal'],
                              filled= True )
```

Note: Once dot file is created, then you can visualize online at http://www.webgraphviz.com/. Open and copy dot file contents and paste it here. You can view your graph.

**Step3. [Create a Test Set]**
Create a testing csv file with 3 samples as below. (columns 2, 3, 4, 5 only)

```
from sklearn import tree
from sklearn.tree import export_graphviz
with open("tree.dot", 'w') as f:
    f = tree.export_graphviz(clf, out_file=f, max_depth=4,
                             impurity=False, feature_names=)
```

Another Way to Visualize:

```
from sklearn import tree
%matplotlib inline
tree.plot_tree(clf)
```

Step:3

```
import csv
fields = ['Name', 'Teethed', 'Hair', 'Breathes', 'Legs', 'Species']
data rows of csv file
rows = [ ['Turtle', 'FALSE', 'FALSE', 'TRUE', 'FALSE', 'Reptile',
          'Blue Whales', 'FALSE', 'TRUE', 'TRUE', 'TRUE', 'Mammal'
          'Crocodile', 'TRUE', 'FALSE', 'TRUE', 'TRUE', 'Reptile'] ]
filename = "testing.csv"
with open(filename, 'w') as file:
    writer = csv.writer(file)
    writer = writerow(fields)
    writer.writerows(rows)
test_set = pd.read_csv('testing.csv')
test_set
test_set["label"] = label_encoder.fit_transform
                    (test_set["species"])
test_set.
```

| | | | | | |
|---|---|---|---|---|---|
| Turtile | FALSE | FALSE | TRUE | FALSE | Reptile |
| Blue whales | FALSE | TRUE | TRUE | TRUE | Mammal |
| Crocodile | TRUE | FALSE | TRUE | TRUE | Reptile |

**Step4. [Perform prediction]**
Use your ID3 DT model that you created before and predict labels for this test set.
Check your predictions. Correct?

**Step5. [Build CART Decision Tree Model]**

- Now, you are going to build a new CART decision tree using criterion='gini'.
- Train you model with full training data (No, train test split, this time)
- Predict samples for the test file
- Visualize your CART DT using graphviz

**Step6. [Buid DT with Zoo dataset]**

- Download full animal dataset at https://archive.ics.uci.edu/ml/datasets/Zoo
- Import, build model using ID3 and CART, train and test accuracy. Print classification report. Visualize your trees.

Step:4

```
stp4 = test_set.drop(['Name', 'species', 'Label'], axis=1)
y_pred = clf.predict(stp4)
y_pred
accuracy_score(test_set.Label, y_pred)
```

Step:5

```
clf_1 = DecisionTreeClassifier(criterion = 'gini', max_depth=4,
                               random_state=42)
clf_1.fit(x,y)

DecisionTreeClassifier(max_depth=4, random_state=42)

clf_1.predict(stp4)

with open("tree2.txt", 'w') as f:
    f = tree.export_graphviz(clf_1, out_file=f, max_depth=4,
                             impurity=False, feature_names
```

Another way to visualize:

```
from sklearn import tree
tree.plot_tree(clf_1)
```

step: 6

```
animal_2 = pd.read_csv("zoo.csv")
animal_2.head()
animal_2.shape
animal_2.info()
X1 = animal_2.drop(['name','type'], axis=1)
y1 = animal_2.type
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size=
clf_2 = DecisionTreeClassifier(criterion='entropy', max_depth=3,
                               random_state=42)
clf_2.fit(x_train, y_train)
clf_2.predict(x_test)
clf_3 = DecisionTreeClassifier(criterion='gini', max_depth=4,
                               random_state=42)
clf_3.fit(X_train, y_train)
y_pred = clf_3.predict(X_test)
y_pred
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
SS = scale.fit_transform(X_train)
SS1 = scale.transform(X_test)
print("model accuracy:", accuracy_score(y_test, y_pred))
print("train accuracy", clf_3.score(SS, y_train))
print("test accuracy:", clf_3.score(SS1, y_test))
animal_2.type.value_counts(dropna=False)
```

NOTES

```
with open("trees.txt", 'w') as f:
f = tree.export_graphviz(clf, out_file=f, max_depth=16,
                         impurity=False.

print(classification_report(y_test, y_pred))

from sklearn import tree

tree.plot_tree(clf, 3)
```