

Lab8. Animal Classification using Decision Trees

Name : ARUL KUMAR ARK
(225229103)

STEP 1

```
In [1]: import pandas as pd
```

```
In [3]: data=pd.read_csv('animals.csv')  
data.head()
```

```
Out [3]:
```

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal

```
In [4]: data.shape
```

```
Out [4]: (10, 5)
```

```
In [5]: data.columns
```

```
Out [5]: Index(['toothed', 'hair', 'breathes', 'legs', 'species'], dtype='object')
```

```
In [6]: data.size
```

```
Out [6]: 50
```

step 2

```
In [7]: from sklearn.tree import DecisionTreeClassifier
```

```
In [8]: dc=DecisionTreeClassifier(criterion='entropy')
```

```
In [9]: from sklearn.model_selection import train_test_split
```

```
In [10]: X=data.drop("species",axis=1)
y=data['species']
```

```
In [11]: x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.33)
```

```
In [12]: dc.fit(x_train,y_train)
```

```
Out[12]: DecisionTreeClassifier(criterion='entropy')
```

```
In [13]: y_pred=dc.predict(x_test)
```

```
In [14]: from sklearn.metrics import accuracy_score
```

```
In [15]: a_score=accuracy_score(y_test,y_pred)
a_score
```

```
Out[15]: 1.0
```

```
In [16]: from sklearn.metrics import classification_report
cr=classification_report(y_pred,y_test)
print(cr)
```

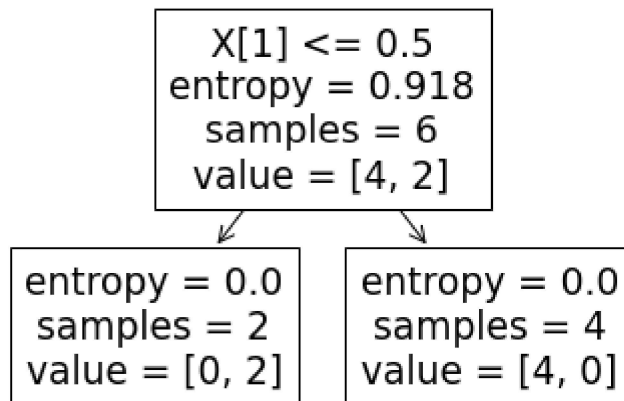
	precision	recall	f1-score	support
Mammal	1.00	1.00	1.00	2
Reptile	1.00	1.00	1.00	2
accuracy			1.00	4
macro avg	1.00	1.00	1.00	4
weighted avg	1.00	1.00	1.00	4

```
In [17]: from sklearn.tree import export_graphviz
from sklearn import tree
```

```
In [18]: with open ("tree1.dot",'w') as f:
f = tree.export_graphviz(dc,
                        out_file=f,
                        max_depth=4,
                        impurity=False,
                        feature_names=X.columns.values,
                        class_names=['Reptile','Mammal'],
                        filled=True)
```

```
In [20]: tree.plot_tree(dc)
```

```
Out[20]: [Text(0.5, 0.75, 'X[1] <= 0.5\nentropy = 0.918\nsamples = 6\nvalue = [4, 2]'),
Text(0.25, 0.25, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.75, 0.25, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]')]
```



STEP 3

```
In [21]: x_test1=pd.read_csv("animals_test.csv")
x_test1
```

```
Out[21]:
```

	toothed	hair	breathes	legs	species
0	False	False	True	False	Reptile
1	False	True	True	True	Mammal
2	True	False	True	True	Reptile

```
In [22]: dc
```

```
Out[22]: DecisionTreeClassifier(criterion='entropy')
```

STEP 4

```
In [23]: tx=x_test1.drop("species",axis=1)
```

```
In [24]: y_pred=dc.predict(tx)
```

```
In [25]: y_pred
```

```
Out[25]: array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

STEP 5

```
In [26]: DTC=DecisionTreeClassifier(criterion='gini')
```

```
In [27]: DTC.fit(X,y)
y_pred1=DTC.predict(tx)
```

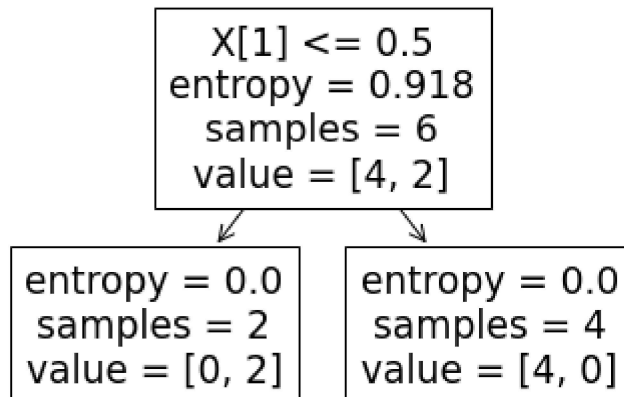
```
In [28]: y_pred1
```

```
Out[28]: array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

```
In [29]: with open ("tree1.dot2",'w') as f:
          f = tree.export_graphviz(dc,
                                   out_file=f,
                                   max_depth=4,
                                   impurity=False,
                                   feature_names=X.columns.values,
                                   class_names=['Reptile','Mammal'],
                                   filled=True)
```

```
In [38]: tree.plot_tree(dc)
```

```
Out[38]: [Text(0.5, 0.75, 'X[1] <= 0.5\nentropy = 0.918\nsamples = 6\nvalue = [4, 2]'),
          Text(0.25, 0.25, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
          Text(0.75, 0.25, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]')]
```



STEP 6

```
In [39]: z_data=pd.read_csv('zoo.csv')
z_data
```

Out[39]:

	animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	L
0	aardvark	1	0	0	1	0	0	1	1	1	
1	antelope	1	0	0	1	0	0	0	1	1	
2	bass	0	0	1	0	0	1	1	1	1	
3	bear	1	0	0	1	0	0	1	1	1	
4	boar	1	0	0	1	0	0	1	1	1	
96	wallaby	1	0	0	1	0	0	0	1	1	
97	wasp	1	0	1	0	1	0	0	0	0	
98	wolf	1	0	0	1	0	0	1	1	1	
99	worm	0	0	1	0	0	0	0	0	0	
100	wren	0	1	1	0	1	0	0	0	1	

101 rows • 18 columns

```
In [40]: x=z_data.drop(['animal_name','class_type'],axis=1)
y=z_data.class_type
```

```
In [41]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_statc
```

```
In [42]: ID3=DecisionTreeClassifier(criterion='entropy',max_depth=3)
```

```
In [43]: ID3.fit(x_train,y_train)
y_pred2=ID3.predict(x_test)
y_pred2
```

Out[43]: array([1, 1, 1, 1, 1, 7, 1, 1, 1, 1, 4, 7, 7, 2, 7, 1, 1, 2, 4, 1, 7, 7,
7, 7, 1, 7, 7, 7, 1, 1, 2, 7, 1, 1], dtype=int64)

```
In [44]: print("model accuracy: ",accuracy_score(y_test,y_pred2))
print("Train accuracy: ",ID3.score(x_train,y_train))
print("Test accuracy: ",ID3.score(x_test,y_test))
```

model accuracy: 0.7352941176470589
Train accuracy: 0.8805970149253731
Test accuracy: 0.7352941176470589

```
In [45]: cr=classification_report(y_pred2,y_test)
print(cr)
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	3
3	0.00	0.00	0.00	0
4	1.00	1.00	1.00	2
5	0.00	.	0.00	0
6	0.00	8.88	0.00	0
7	1.00	0.25	0.40	12
accuracy			0.74	34
macro avg	0.57	0.46	0.49	34
weighted avg	1.00	0.74	0.79	34

C:\Users\joshua\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use zero_division' parameter to control this behavior.

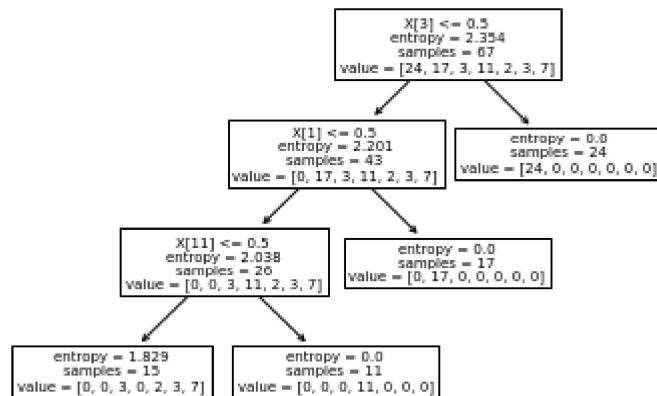
_warn_prf(average, modifier, msg_start, len(result))
 C:\Users\joshua\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))
 C:\Users\joshua\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use zero_division parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

```
In [46]: from sklearn import tree
tree.plot_tree(ID3)
```

```
Out[46]: [Text(0.6666666666666666, 0.875, 'X[3] <= 0.5\nentropy = 2.354\nsamples = 67\nvalue = [24, 17, 3, 11, 2, 3, 7]'),
Text(0.5, 0.625, 'X[1] <= 0.5\nentropy = 2.201\nsamples = 43\nvalue = [0, 17, 3, 11, 2, 3, 7] '),
Text(0.3333333333333333, 0.375, 'X[11] <= 0.5\nentropy = 2.038\nsamples = 26\nvalue = [0, 0, 3, 11, 2, 3, 7]'),
Text(0.16666666666666666, 0.125, 'entropy = 1.829\nsamples = 15\nvalue = [0, 0, 3, 0, 2, 3, 7] '),
Text(0.5, 0.125, 'entropy = 0.0\nsamples = 11\nvalue = [0, 0, 0, 11, 0, 0, 0]'),
Text(0.6666666666666666, 0.375, 'entropy = 0.0\nsamples = 17\nvalue = [0, 17, 0, 0, 0, 0, 0]'),
Text(0.8333333333333334, 0.625, 'entropy = 0.0\nsamples = 24\nvalue = [24, 0, 0, 0, 0, 0, 0]')]
```



```
In [ ]:
```