HARI PRASATH S

225229110

LAB8:ANIMAL CLASSIFICATION USING DECISION TREES

Step1

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

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

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
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

In [8]:
```
df['species'].value_counts()
```

Out[8]:
```
Mammal     6
Reptile    4
Name: species, dtype: int64
```

Step2-Model building using ID3

In [10]:
```
df.shape
```

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

In [4]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [5]:
```
DT=DecisionTreeClassifier(criterion='entropy')
```

In [6]:
```
from sklearn.model_selection import train_test_split
```

In [7]:
```
x=df.drop("species",axis=1)
y=df['species']
```

In [9]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33)
```

In [11]:
```
DT.fit(x_train,y_train)
```

Out[11]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [30]:
```
y_pred=DT.predict(x_test)
```

In [31]:
```
from sklearn.metrics import accuracy_score
```

In [32]:
```
acc=accuracy_score(y_test,y_pred)
acc
```

Out[32]:
```
1.0
```

In [33]:
```
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         3
    Reptile       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00         4
```

In [62]:
```
!pip install graphviz
```

```
Collecting graphviz
  Using cached https://files.pythonhosted.org/packages/9d/fb/886e8ec7862989afc0c35d15813b6c665fe134cc6027cdde2fa300abe9a2/graph
viz-0.19.1-py3-none-any.whl (https://files.pythonhosted.org/packages/9d/fb/886e8ec7862989afc0c35d15813b6c665fe134cc6027cdde2fa3
00abe9a2/graphviz-0.19.1-py3-none-any.whl)
Installing collected packages: graphviz

Exception:
Traceback (most recent call last):
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\basecommand.py", line 215, in
main
    status = self.run(options, args)
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\commands\install.py", line 34
2, in run
    prefix=options.prefix_path,
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\req\req_set.py", line 784, in
install
    **kwargs
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\req\req_install.py", line 851,
in install
    self.move_wheel_files(self.source_dir, root=root, prefix=prefix)
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\req\req_install.py", line 106
4, in move_wheel_files
    isolated=self.isolated,
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\wheel.py", line 345, in move_w
heel_files
    clobber(source, lib_dir, True)
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\wheel.py", line 316, in clobbe
r
    ensure_dir(destdir)
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\pip\utils\__init__.py", line 83, i
n ensure_dir
    os.makedirs(path)
  File "C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\os.py", line 220, in makedirs
    mkdir(name, mode)
PermissionError: [WinError 5] Access is denied: 'C:\\Program Files (x86)\\Microsoft Visual Studio\\Shared\\Anaconda3_64\\Lib\\s
ite-packages\\graphviz'
You are using pip version 9.0.1, however version 23.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [63]:
```
from sklearn.tree import export_graphviz
from sklearn import tree
```

In [64]:
```
with open("tree1.dot",'w') as f:
    f=tree.export_graphviz(DT,
                           out_file=f,
                           max_depth=4,
                           impurity=False,
                           feature_names=x.columns.values,
                           class_names=['Reptile','Mammal'],
                           filled=True)
```

In [70]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
tree.plot_tree(DT)
plt.show()
```

Step3

In [44]:
```
test_data=pd.read_csv('testing.csv')
test_data
```

Out[44]:

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

In [49]:
```
x_test1=test_data.drop('species',axis=1)
x_test1
```

Out[49]:

	Toothed	hair	breathes	legs
0	False	False	True	False
1	False	True	True	True
2	True	False	True	True

Step4

In [50]:
```
DT
```

Out[50]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                       splitter='best')
```

In [51]:
```
y_pred=DT.predict(x_test1)
```

In [52]:
```
y_pred
```

Out[52]:
```
array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

Step5

In [53]:
```
DTC=DecisionTreeClassifier(criterion='gini')
```

In [54]:
```
DTC.fit(x,y)
```

Out[54]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                       splitter='best')
```

In [56]:
```
y_pred1=DTC.predict(x_test1)
```

In [57]:
```
y_pred1
```

Out[57]:
```
array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

In [74]:
```
with open("tree2.dot",'w') as f:
    f=tree.export_graphviz(DTC,
                           out_file=f,
                           max_depth=4,
                           impurity=False,
                           feature_names=x.columns.values,
                           class_names=['Reptile','Mammal'],
                           filled=True)
```

In [ ]:
```
from sklearn import tree
tree.plot_tree(DTC)
plt.show()
```

Step6

In [77]:
```
zoo_df = pd.read_csv('zoo.csv')
zoo_df
```

Out[77]:

	animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	class_ty
0	aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	
1	antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	
2	bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	
3	bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	
4	boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	
5	buffalo	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	
6	calf	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	
7	carp	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	
8	catfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	
9	cavy	1	0	0	1	0	0	0	1	1	1	0	0	4	0	1	0	
10	cheetah	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	
11	chicken	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	
12	chub	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	
13	clam	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	
14	crab	0	0	1	0	0	1	1	0	0	0	0	0	4	0	0	0	
15	crayfish	0	0	1	0	0	1	1	0	0	0	0	0	6	0	0	0	
16	crow	0	1	1	0	1	0	1	0	1	1	0	0	2	1	0	0	
17	deer	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	
18	dogfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	1	
19	dolphin	0	0	0	1	0	1	1	1	1	1	0	1	0	1	0	1	
20	dove	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	
21	duck	0	1	1	0	1	1	0	0	1	1	0	0	2	1	0	0	
22	elephant	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	
23	flamingo	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	1	
24	flea	0	0	1	0	0	0	0	0	0	1	0	0	6	0	0	0	
25	frog	0	0	1	0	0	1	1	1	1	1	0	0	4	0	0	0	
26	frog	0	0	1	0	0	1	1	1	1	1	1	0	4	0	0	0	
27	fruitbat	1	0	0	1	1	0	0	1	1	1	0	0	2	1	0	0	
28	giraffe	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	
29	girl	1	0	0	1	0	0	1	1	1	1	0	0	2	0	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
71	rhea	0	1	1	0	0	0	1	0	1	1	0	0	2	1	0	1	
72	scorpion	0	0	0	0	0	0	1	0	0	1	1	0	8	1	0	0	
73	seahorse	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	0	
74	seal	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	
75	sealion	1	0	0	1	0	1	1	1	1	1	0	1	2	1	0	1	
76	seasnake	0	0	0	0	0	1	1	1	1	0	1	0	0	1	0	0	
77	seawasp	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	
78	skimmer	0	1	1	0	1	1	1	0	1	1	0	0	2	1	0	0	
79	skua	0	1	1	0	1	1	1	0	1	1	0	0	2	1	0	0	
80	slowworm	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	
81	slug	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	
82	sole	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	0	
83	sparrow	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	
84	squirrel	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	0	
85	starfish	0	0	1	0	0	1	1	0	0	0	0	0	5	0	0	0	
86	stingray	0	0	1	0	0	1	1	1	1	0	1	1	0	1	0	1	
87	swan	0	1	1	0	1	1	0	0	1	1	0	0	2	1	0	1	
88	termite	0	0	1	0	0	0	0	0	0	1	0	0	6	0	0	0	
89	toad	0	0	1	0	0	1	0	1	1	1	0	0	4	0	0	0	
90	tortoise	0	0	1	0	0	0	0	0	1	1	0	0	4	1	0	1	
91	tuatara	0	0	1	0	0	0	1	1	1	1	0	0	4	1	0	0	
92	tuna	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	1	
93	vampire	1	0	0	1	1	0	0	1	1	1	0	0	2	1	0	0	

	animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	class_ty
94	vole	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	0	
95	vulture	0	1	1	0	1	0	1	0	1	1	0	0	2	1	0	1	
96	wallaby	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	1	
97	wasp	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	0	
98	wolf	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	
99	worm	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	
100	wren	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	

101 rows × 18 columns

In [79]:
```
zoo_df.shape
```

Out[79]:
```
(101, 18)
```

In [81]:
```
zoo_df.head()
```

Out[81]:

	animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	class_type
0	aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
1	antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
2	bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
3	bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
4	boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1

In [87]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [88]:
```
ID3=DecisionTreeClassifier(criterion='entropy',max_depth=3)
```

In [89]:
```
ID3.fit(x_train,y_train)
y_pred2=ID3.predict(x_test)
y_pred2
```

Out[89]:
```
array(['Mammal', 'Mammal', 'Mammal', 'Mammal'], dtype=object)
```

In [90]:
```
print("model accuracy:",accuracy_score(y_test,y_pred2))
print("Train accuracy:",ID3.score(x_train,y_train))
print("Text accuracy:",ID3.score(x_test,y_test))
```

```
model accuracy: 1.0
Train accuracy: 1.0
Text accuracy: 1.0
```

In [91]:
```
cr=classification_report(y_pred2,y_test)
print(cr)
```

```
             precision    recall  f1-score   support

     Mammal       1.00      1.00      1.00         4

avg / total       1.00      1.00      1.00         4
```

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

In [93]:
```
DTC=DecisionTreeClassifier(criterion='gini',max_depth=4)
```

In [94]:
```
DTC.fit(x_train,y_train)
y_pred3=DTC.predict(x_test)
y_pred3
```

Out[94]:
```
array(['Mammal', 'Mammal', 'Mammal', 'Mammal'], dtype=object)
```

In [95]:
```
print("model accuracy:",accuracy_score(y_pred3,y_test))
print("Train accuracy:",DTC.score(x_train,y_train))
print("Test accuracy:",DTC.score(x_test,y_test))
```

```
model accuracy: 1.0
Train accuracy: 1.0
Test accuracy: 1.0
```

In [96]:
```
print(classification_report(y_pred3,y_test))
```

```
             precision    recall  f1-score   support

     Mammal       1.00      1.00      1.00         4

avg / total       1.00      1.00      1.00         4
```

In [ ]:
```
from sklearn import tree
tree.plot_tree(DTC)
plt.show()
```