Name: Ezhilarasan C

Roll No:225229151

Labsheet 8 : Animal prediction using decision tree

Step :1

created the given data set using excel and saved it as csv file

Step 2

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

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

In [42]:
```
df
```

Out[42]:

	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 [ ]:
```
df.shape
```

Out[11]:
```
(10, 5)
```

In [ ]:
```
df.columns
```

Out[12]:
```
Index(['toothed', 'hair', 'breathes', 'legs', 'species'], dtype='objec
t')
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   toothed   10 non-null     bool
 1   hair      10 non-null     bool
 2   breathes  10 non-null     bool
 3   legs      10 non-null     bool
 4   species   10 non-null     object
dtypes: bool(4), object(1)
memory usage: 248.0+ bytes
```

In [ ]:
```
X =  df[['toothed', 'hair', 'breathes','legs' ]]
```

In [ ]:
```
y = df.species
```

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

In [20]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
```

In [21]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [22]:
```
dt= DecisionTreeClassifier(criterion ="entropy")
dt.fit(X_train,y_train)
```

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

In [24]:
```
y_pred = dt.predict(X_test)
```

In [26]:
```
from sklearn.metrics import accuracy_score,classification_report
```

In [27]:
```
acc = accuracy_score(y_test, y_pred)
print("Accuracy score :",acc)
```

```
Accuracy score : 1.0
```

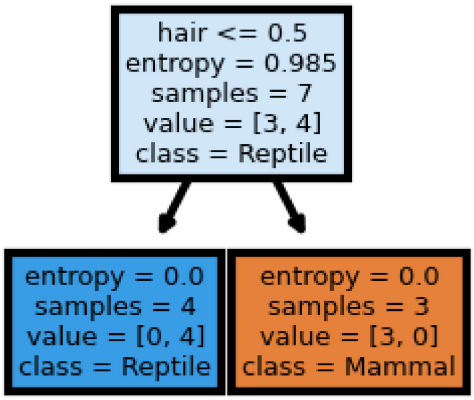
In [28]:
```
clf_report= classification_report(y_test, y_pred)
print("Classification report: ",clf_report)
```

```
Classification report:                precision    recall  f1-score   su
pport

      Mammal       1.00      1.00      1.00         3

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3
```

In [43]:
```
import matplotlib.pyplot as plt
```

In [44]:
```
from sklearn import tree
```

In [46]:
```
fn=['toothed', 'hair', 'breathes','legs']
cn=['Mammal', 'Reptile']
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (1,1), dpi=300)
tree.plot_tree(dt,
               feature_names = fn,
               class_names=cn,
               filled = True);
fig.savefig('imagename.png')
```



Step 3

In [52]:
```
df1=pd.read_csv("animals_test.csv")
```

In [55]:
```
df1
```

Out[55]:

	toothed	hair	breathes	legs	Unnamed: 4	Unnamed: 5
0	False	False	True	False	NaN	NaN
1	False	True	True	True	NaN	NaN
2	True	False	True	True	NaN	NaN

In [56]:
```
df1 = df1[['toothed', 'hair', 'breathes','legs']]
```

In [57]:
```
y_pred_test=dt.predict(df1)
```

Step 4

In [58]:
```
y_pred_test
```

Out[58]:
```
array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

Step 5

In [60]:
```
dt_cart= DecisionTreeClassifier(criterion ="gini")
dt_cart.fit(X,y)
```

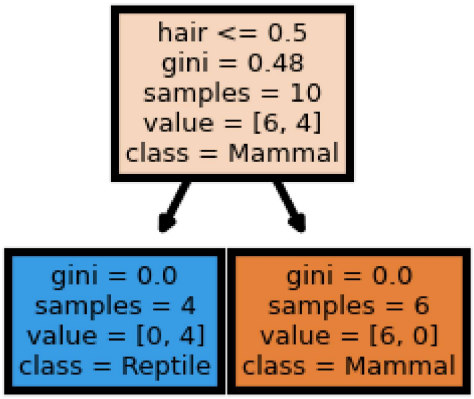
Out[60]:
```
DecisionTreeClassifier()
```

In [61]:
```
y_pred_ctest=dt_cart.predict(df1)
```

In [62]:
```
y_pred_ctest
```

Out[62]:
```
array(['Reptile', 'Mammal', 'Reptile'], dtype=object)
```

In [63]:
```
fn=['toothed', 'hair', 'breathes','legs']
cn=['Mammal', 'Reptile']
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (1,1), dpi=300)
tree.plot_tree(dt_cart,
               feature_names = fn,
               class_names=cn,
               filled = True);
fig.savefig('image1.png')
```



Step 6

In [65]:
```
zoo = pd.read_csv('zoo.data')
```

In [67]:
```
zoo.head()
```

Out[67]:

	aardvark	1	0	0.1	1.1	0.2	0.3	1.2	1.3	1.4	1.5	0.4	0.5	4	0.6	0.7	1.6	1.7
0	antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
1	bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
2	bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
3	boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
4	buffalo	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1

In [72]:
```
zoo.shape
```

Out[72]: (100, 18)

In [68]:
```
X=zoo.drop(['aardvark','1.7'],axis=1)
```

In [71]:
```
y = zoo[['1.7']]
```

In [73]:
```
x_train, x_test, y1_train, y1_test = train_test_split(X, y, test_size=0.25
zoo_entropy = DecisionTreeClassifier(criterion ="entropy")
zoo_entropy.fit(x_train,y1_train)
```

Out[73]:
```
DecisionTreeClassifier(criterion='entropy')
```

In [75]:
```
y1_pred = zoo_entropy.predict(x_test)
```

In [76]:
```
train_pred=zoo_entropy.predict(x_train)
train_pred
```

Out[76]:
```
array([2, 2, 6, 1, 4, 4, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 2, 4,
       5, 1, 7, 5, 1, 4, 6, 1, 2, 4, 1, 2, 3, 2, 2, 1, 1, 6, 1, 1, 1, 3,
       1, 3, 6, 2, 1, 7, 2, 3, 2, 2, 4, 1, 7, 4, 6, 4, 7, 1, 1, 6, 6, 4,
       1, 2, 2, 2, 4, 7, 7, 1, 5])
```

In [77]:
```
test_pred=zoo_entropy.predict(x_test)
test_pred
```

Out[77]:
```
array([1, 1, 2, 7, 1, 6, 2, 7, 2, 1, 1, 1, 1, 4, 5, 1, 7, 2, 7, 1, 2, 5,
       1, 2, 1])
```

In [78]:
```
print("Train Accuracy:", accuracy_score(y1_train, zoo_entropy.predict(x_tr
print("Test Accuracy:", accuracy_score(y1_test, zoo_entropy.predict(x_test
```

```
Train Accuracy: 1.0
Test Accuracy: 0.96
```

In [79]:
```
acc = accuracy_score(y1_test, y1_pred)
print("Accuracy score :",acc)
```

```
Accuracy score : 0.96
```

In [ ]:
```

```