GitHub Assignment Animal classification using Decision Trees The primary task of this notebook is to categorize animals as either mammals or non-mammals

- Repeat the same steps used in the vertebrate.csv decision tree for the animal.csv dataset
- Then upload it to GitHub and share your GitHub repository link on Blackboard

In [1…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2…
```
animals = pd.read_csv('/content/drive/MyDrive/animals.csv')
```

In [2…
```
animals
```

Out[22]:

	Sr	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Brea
0	1	1	0	0	1	0	0	1	1	1	
1	2	1	0	0	1	0	0	0	1	1	
2	3	0	0	1	0	0	1	1	1	1	
3	4	1	0	0	1	0	0	1	1	1	
4	5	1	0	0	1	0	0	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	
85	86	0	0	1	0	0	1	1	0	0	
86	87	0	0	1	0	0	1	1	1	1	
87	88	0	1	1	0	1	1	0	0	1	
88	89	0	0	1	0	0	0	0	0	0	
89	90	0	0	1	0	0	1	0	1	1	

90 rows × 18 columns

In […
```
animals.head()
```

Out[ ]:

	Sr	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Breat
0	1	1	0	0	1	0	0	1	1	1	
1	2	1	0	0	1	0	0	0	1	1	
2	3	0	0	1	0	0	1	1	1	1	
3	4	1	0	0	1	0	0	1	1	1	
4	5	1	0	0	1	0	0	1	1	1	

In […
```
animals.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 18 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Sr         90 non-null     int64
 1    Hair      90 non-null     int64
 2    Feathers  90 non-null     int64
 3    Eggs      90 non-null     int64
 4    Milk      90 non-null     int64
 5    Airborne  90 non-null     int64
 6    Aquatic   90 non-null     int64
 7    Predator  90 non-null     int64
 8    Teeth     90 non-null     int64
 9    Backbone  90 non-null     int64
 10   Breathes  90 non-null     int64
 11   Venemous  90 non-null     int64
 12   Fins      90 non-null     int64
 13   Legs      90 non-null     int64
 14   Tails     90 non-null     int64
 15   Domestic  90 non-null     int64
 16   Catsize   90 non-null     int64
 17  Class      90 non-null     int64
dtypes: int64(18)
memory usage: 12.8 KB
```

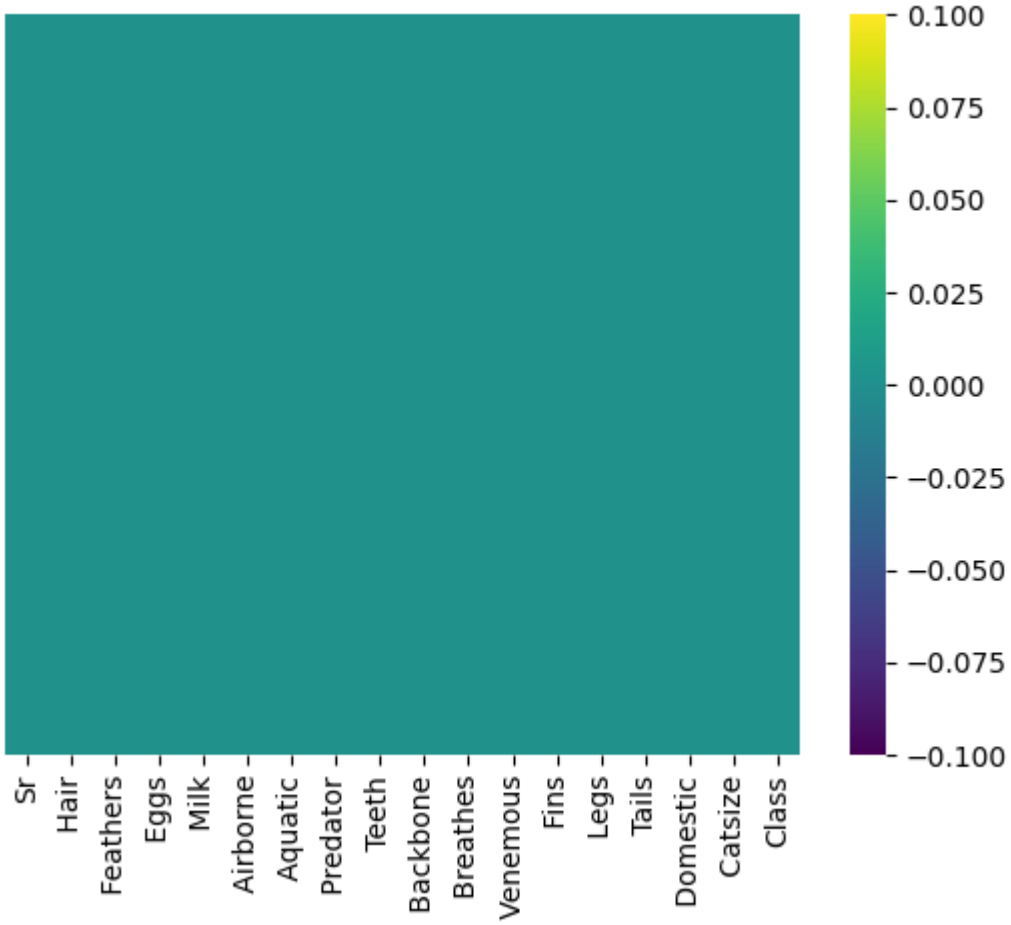
Missing values

In […]:
```
animals.isnull().sum()
```

Out[ ]:
```
Sr           0
 Hair        0
 Feathers    0
 Eggs        0
 Milk        0
 Airborne    0
 Aquatic     0
 Predator    0
 Teeth       0
 Backbone    0
 Breathes    0
 Venemous    0
 Fins        0
 Legs        0
 Tails       0
 Domestic    0
 Catsize     0
Class        0
dtype: int64
```

In […]:
```
sns.heatmap(animals.isnull(),yticklabels=False,cbar=True,cmap='viric
```

Out[ ]:
```
<Axes: >
```



In [2…

```
animals['Class']=animals['Class'].replace([1],'Mammals')
animals['Class']=animals['Class'].replace([2],'Birds')
animals['Class']=animals['Class'].replace([3],'Reptiles')
animals['Class']=animals['Class'].replace([4],'Fish')
animals['Class']=animals['Class'].replace([5],'Amphibians')
animals['Class']=animals['Class'].replace([6],'Insects')
animals['Class']=animals['Class'].replace([7],'Arachnids')
```

In [2…

```
animals
```

Out[24]:

	Sr	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Brea
0	1	1	0	0	1	0	0	1	1	1	
1	2	1	0	0	1	0	0	0	1	1	
2	3	0	0	1	0	0	1	1	1	1	
3	4	1	0	0	1	0	0	1	1	1	
4	5	1	0	0	1	0	0	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	
85	86	0	0	1	0	0	1	1	0	0	
86	87	0	0	1	0	0	1	1	1	1	
87	88	0	1	1	0	1	1	0	0	1	
88	89	0	0	1	0	0	0	0	0	0	
89	90	0	0	1	0	0	1	0	1	1	

90 rows × 18 columns

In [5…

```
animals.sample(25)
```

Out[54]:

	Sr	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Brea
24	25	0	0	1	0	0	0	0	0	0	
9	10	1	0	0	1	0	0	0	1	1	
57	58	0	1	1	0	1	0	0	0	1	
59	60	0	1	1	0	1	0	0	0	1	
84	85	1	0	0	1	0	0	0	1	1	
11	12	0	1	1	0	1	0	0	0	1	
86	87	0	0	1	0	0	1	1	1	1	
75	76	1	0	0	1	0	1	1	1	1	
47	48	1	0	0	1	0	0	1	1	1	
10	11	1	0	0	1	0	0	1	1	1	
56	57	0	1	1	0	0	0	0	0	1	
71	72	0	1	1	0	0	0	1	0	1	
69	70	1	0	0	1	0	0	1	1	1	
19	20	0	0	0	1	0	1	1	1	1	
87	88	0	1	1	0	1	1	0	0	1	
62	63	0	0	1	0	0	0	1	1	1	
8	9	0	0	1	0	0	1	1	1	1	
26	27	0	0	1	0	0	1	1	1	1	
38	39	0	0	1	0	0	1	1	1	1	
4	5	1	0	0	1	0	0	1	1	1	
34	35	0	0	1	0	0	1	0	1	1	
35	36	1	0	0	1	0	0	0	1	1	
45	46	1	0	0	1	0	0	1	1	1	
2	3	0	0	1	0	0	1	1	1	1	
85	86	0	0	1	0	0	1	1	0	0	

In [2…
```
print(animals.columns)
```

```
Index(['Sr', ' Hair', ' Feathers', ' Eggs', ' Milk', ' Airborne',
' Aquatic',
       ' Predator', ' Teeth', ' Backbone', ' Breathes', '
Venemous', ' Fins',
       ' Legs', ' Tails', ' Domestic', ' Catsize', 'Class'],
      dtype='object')
```

In [2…
```
animals.columns = animals.columns.str.strip()
```

In [3…
```
pd.crosstab([animals['Aquatic'],animals['Fins']],animals['Class'])
```

Out[30]:

Aquatic	Fins	Amphibians	Arachnids	Birds	Fish	Insects	Mammals	Reptiles
0	**0**	0	3	12	0	7	31	2
1	**0**	4	6	6	0	0	2	1
	1	0	0	0	12	0	4	0

In [3…
```
from sklearn import tree
```

In [3…
```
y=animals ['Class']
```

In [3…
```
y
```

Out[33]:
```
0        Mammals
1        Mammals
2           Fish
3        Mammals
4        Mammals
         ...
85     Arachnids
86          Fish
87         Birds
88       Insects
89    Amphibians
Name: Class, Length: 90, dtype: object
```

In [3…
```
X=animals.drop(['Sr','Class'],axis=1)
```

In [3…
```
X
```

Out[36]:

	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Breathes
0	1	0	0	1	0	0	1	1	1	1
1	1	0	0	1	0	0	0	1	1	1
2	0	0	1	0	0	1	1	1	1	0
3	1	0	0	1	0	0	1	1	1	1
4	1	0	0	1	0	0	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...
85	0	0	1	0	0	1	1	0	0	0
86	0	0	1	0	0	1	1	1	1	0
87	0	1	1	0	1	1	0	0	1	1
88	0	0	1	0	0	0	0	0	0	1
89	0	0	1	0	0	1	0	1	1	1

90 rows × 16 columns

In [3…
```
clf = tree.DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In [3…
```
clf
```

Out[38]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In [3…
```
clf=clf.fit(X,y)
```

In [4…
```
clf
```

Out[40]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In [4…
```
import pydotplus
```

In [4…
```
from IPython.display import Image
```

In [4…
```
dot_data=tree.export_graphviz(clf, feature_names=X.columns,
                              class_names=['Mammals','Birds','Reptile
                              filled=True,out_file=None)
```

In [4…
```
dot_data
```

Out[44]:
```
'digraph Tree {\nnode [shape=box, style="filled", color="black",
fontname="helvetica"] ;\nedge [fontname="helvetica"] ;\n0
[label="Milk <= 0.5\\nentropy = 2.361\\nsamples = 90\\nvalue = [4,
9, 18, 12, 7, 37, 3]\\nclass = Arachnids", fillcolor="#eacbf8"] ;
\n1 [label="Teeth <= 0.5\\nentropy = 2.35\\nsamples = 53\\nvalue =
[4, 9, 18, 12, 7, 0, 3]\\nclass = Reptiles",
fillcolor="#e2fbe5"] ;\n0 -> 1 [labeldistance=2.5, labelangle=45,
headlabel="True"] ;\n2 [label="Backbone <= 0.5\\nentropy = 1.463\
\nsamples = 34\\nvalue = [0, 9, 18, 0, 7, 0, 0]\\nclass =
Reptiles", fillcolor="#b8f6bf"] ;\n1 -> 2 ;\n3 [label="entropy =
0.989\\nsamples = 16\\nvalue = [0, 9, 0, 0, 7, 0, 0]\\nclass =
Birds", fillcolor="#eff9d3"] ;\n2 -> 3 ;\n4 [label="entropy = 0.0\
\nsamples = 18\\nvalue = [0, 0, 18, 0, 0, 0, 0]\\nclass =
Reptiles", fillcolor="#39e54d"] ;\n2 -> 4 ;\n5 [label="Fins <=
0.5\\nentropy = 1.312\\nsamples = 19\\nvalue = [4, 0, 0, 12, 0, 0,
3]\\nclass = Fish", fillcolor="#95f1f0"] ;\n1 -> 5 ;\n6
[label="entropy = 0.985\\nsamples = 7\\nvalue = [4, 0, 0, 0, 0, 0,
3]\\nclass = Mammals", fillcolor="#f8e0ce"] ;\n5 -> 6 ;\n7
[label="entropy = 0.0\\nsamples = 12\\nvalue = [0, 0, 0, 12, 0, 0,
0]\\nclass = Fish", fillcolor="#39e5e2"] ;\n5 -> 7 ;\n8
[label="entropy = 0.0\\nsamples = 37\\nvalue = [0, 0, 0, 0, 0, 37,
0]\\nclass = Arachnids", fillcolor="#b139e5"] ;\n0 -> 8
[labeldistance=2.5, labelangle=-45, headlabel="False"] ;\n}'
```

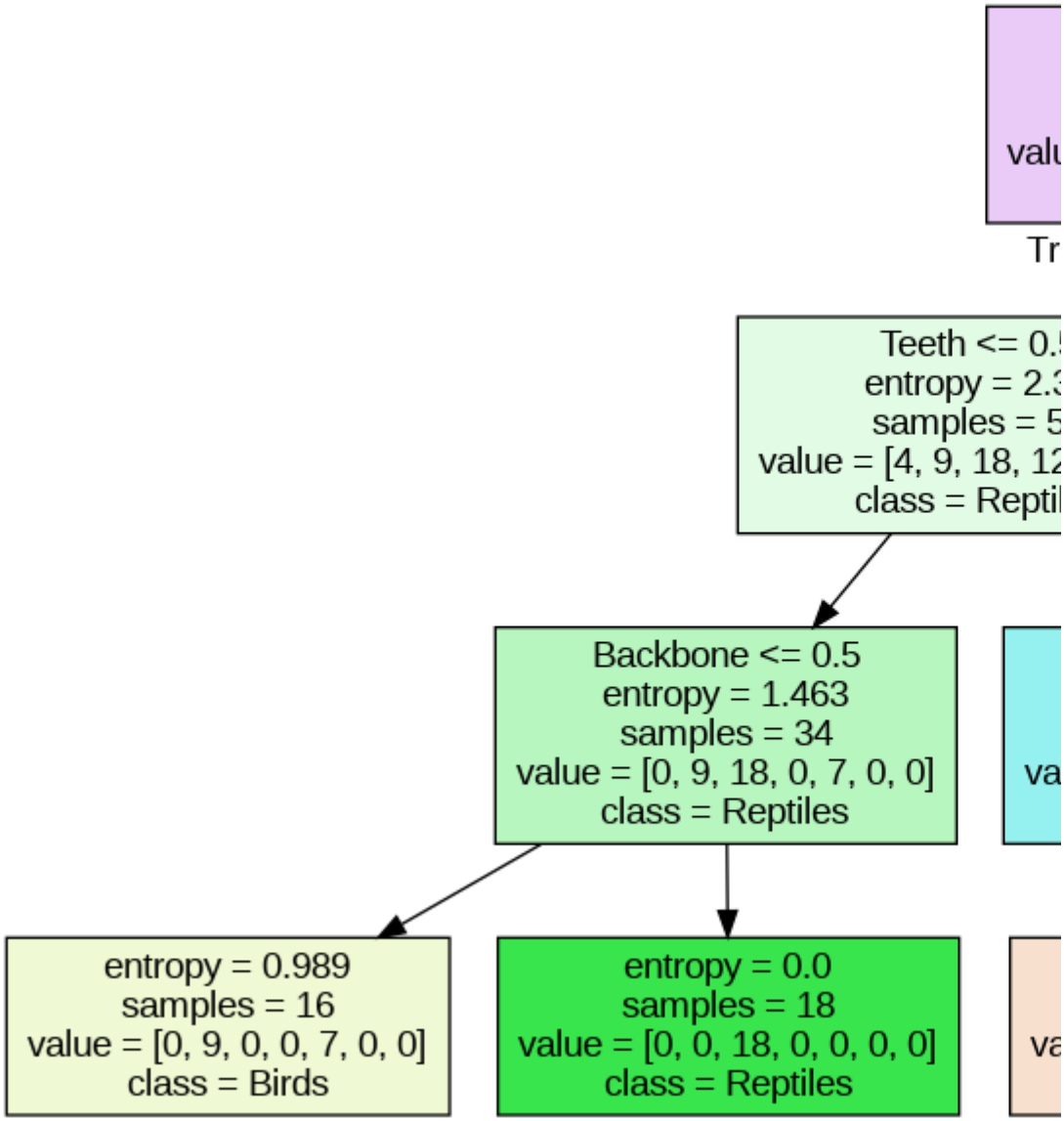
In [4…
```
graph=pydotplus.graph_from_dot_data(dot_data)
```

In [4…
```
graph
```

Out[46]:
```
<pydotplus.graphviz.Dot at 0x786132cf59f0>
```

In [4…
```
Image(graph.create_png())
```

Out[47]:



In [5…
```
test_data = [
    [1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 4, 1, 0, 1, 'Mammals'],
    [0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 5, 0, 0, 0, 'Arachnids'],
    [0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 'Fish'],
    [0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 2, 1, 0, 1, 'Birds'],
    [0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 6, 0, 0, 0, 'Insects'],
    [0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 4, 0, 0, 0, 'Amphibians'],
    [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 'Reptiles']
 ]
```

In [5…
```
test_data
```

Out[56]:
```
[[1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 4, 1, 0, 1, 'Mammals'],
 [0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 5, 0, 0, 0, 'Arachnids'],
 [0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 'Fish'],
 [0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 2, 1, 0, 1, 'Birds'],
 [0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 6, 0, 0, 0, 'Insects'],
 [0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 4, 0, 0, 0, 'Amphibians'],
 [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 'Reptiles']]
```

In [5…
```
animals.columns
```

Out[51]:
```
Index(['Sr', 'Hair', 'Feathers', 'Eggs', 'Milk', 'Airborne',
'Aquatic',
       'Predator', 'Teeth', 'Backbone', 'Breathes', 'Venemous',
'Fins', 'Legs',
       'Tails', 'Domestic', 'Catsize', 'Class'],
      dtype='object')
```

In [5…
```
animals = animals.drop(columns=['Sr'])
```

In [5…
```
animals.columns
```

Out[59]:
```
Index(['Hair', 'Feathers', 'Eggs', 'Milk', 'Airborne', 'Aquatic',
'Predator',
       'Teeth', 'Backbone', 'Breathes', 'Venemous', 'Fins',
'Legs', 'Tails',
       'Domestic', 'Catsize', 'Class'],
      dtype='object')
```

In [6…
```
animals.head()
```

Out[60]:

	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Breathes
0	1	0	0	1	0	0	1	1	1	1
1	1	0	0	1	0	0	0	1	1	1
2	0	0	1	0	0	1	1	1	1	0
3	1	0	0	1	0	0	1	1	1	1
4	1	0	0	1	0	0	1	1	1	1

In [6…
```
test_data=pd.DataFrame(test_data,columns=animals.columns)
```

In [6…
```
test_data
```

Out[62]:

	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Breathes
0	1	0	0	1	0	0	1	1	1	1
1	0	0	1	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1	1	0
3	0	1	1	0	1	1	0	0	1	1
4	0	0	1	0	0	0	0	0	0	1
5	0	0	1	0	0	1	0	1	1	1
6	0	0	1	0	0	0	1	1	1	1

In [6…
```
testy=test_data['Class']
```

In [6…
```
testy
```

Out[64]:
```
0       Mammals
1     Arachnids
2          Fish
3         Birds
4       Insects
5    Amphibians
6      Reptiles
Name: Class, dtype: object
```

In [6…
```
testx=test_data.drop(['Class'],axis=1)
```

In [6…
```
testx
```

Out[66]:

	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Teeth	Backbone	Breathes
0	1	0	0	1	0	0	1	1	1	1
1	0	0	1	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1	1	0
3	0	1	1	0	1	1	0	0	1	1
4	0	0	1	0	0	0	0	0	0	1
5	0	0	1	0	0	1	0	1	1	1
6	0	0	1	0	0	0	1	1	1	1

In [6…
```
predy=clf.predict(testx)
```

In [6…
```
predy
```

Out[68]:
```
array(['Mammals', 'Arachnids', 'Fish', 'Birds', 'Arachnids',
'Amphibians',
       'Amphibians'], dtype=object)
```

In [6…
```
predictions=pd.concat([test_data['Class'],pd.Series(predy,name='Prec
```

In [7…
```
predictions
```

Out[70]:

	Class	Predicted Class
0	Mammals	Mammals
1	Arachnids	Arachnids
2	Fish	Fish
3	Birds	Birds
4	Insects	Arachnids
5	Amphibians	Amphibians
6	Reptiles	Amphibians

In [7…
```
from sklearn.metrics import accuracy_score
```

In [7…
```
accuracy_score(testy,predy)
```

Out[72]: 0.7142857142857143