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

In [2]:
```
df =pd.read_csv('./zoodata.csv')
```

In [3]:
```
df.head()
```

Out[3]:
```
   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 [4]:
```
df.columns =['name', 'hair', 'feathers', 'eggs', 'milk', 'airbone', 'aquatic', 'predator', 'toothed', 'backone', 'breathe
]
```

In [5]:
```
df.head(200)
```

Out[5]:
```
        name  hair  feathers  eggs  milk  airbone  aquatic  predator  toothed  backone  breathes  venomous  fins  legs  tail  domestic  catsiz
 0  antelope     1         0     0     1        0        0         0        1        1         1         0     0     4     1         0
 1      bass     0         0     1     0        0        1         1        1        1         0         0     1     0     1         0
 2      bear     1         0     0     1        0        0         1        1        1         1         0     0     4     0         0
 3      boar     1         0     0     1        0        0         1        1        1         1         0     0     4     1         0
 4   buffalo     1         0     0     1        0        0         0        1        1         1         0     0     4     1         0
...      ...   ...       ...   ...   ...      ...      ...       ...      ...      ...       ...       ...   ...   ...   ...       ...
95   wallaby     1         0     0     1        0        0         0        1        1         1         0     0     2     1         0
```

	name	hair	feathers	eggs	milk	airbone	aquatic	predator	toothed	backone	breathes	venomous	fins	legs	tail	domestic	catsiz
96	wasp	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	
97	wolf	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	
98	worm	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	
99	wren	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	

100 rows × 18 columns

In [6]:

```
print("Shape",df.shape)
# df.type.unique()
```

```
Shape (100, 18)
```

In [7]:

```
df.describe()
```

Out[7]:

	hair	feathers	eggs	milk	airbone	aquatic	predator	toothed	backone	breathes	venomous	
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.00	100.000000	100.000000	100.00000	100.00000	100.000
mean	0.420000	0.200000	0.590000	0.400000	0.240000	0.360000	0.55	0.600000	0.820000	0.79000	0.08000	0.170
std	0.496045	0.402015	0.494311	0.492366	0.429235	0.482418	0.50	0.492366	0.386123	0.40936	0.27266	0.377
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00000	0.00000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	1.000000	1.00000	0.00000	0.000
50%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.00	1.000000	1.000000	1.00000	0.00000	0.000
75%	1.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.00	1.000000	1.000000	1.00000	0.00000	0.000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00	1.000000	1.000000	1.00000	1.00000	1.000

In [8]:

```
x =df.iloc[:, :-1]
y =df.iloc[:, -1]
```

Encode our target variable into a nummerical vairable encode label encode command

In [9]:

```
from sklearn.preprocessing import OneHotEncoder

encoded_x = OneHotEncoder().fit_transform(x).toarray()
print(encoded_x)
```

```
[[1. 0. 0. ... 0. 0. 1.]
 [0. 1. 0. ... 0. 1. 0.]
 [0. 0. 1. ... 0. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 1. 0.]]
```

Split into Traning & Testing Sets

In [10]:

```
 from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(encoded_x, y, test_size=0.30, random_state=123)
```

In [11]:

```
x_train
```

Out[11]:

```
array([[0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 0., 1.]])
```

Create Classifier object

In [12]:

```
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,   hidden_layer_sizes=(5, 2), random_state=1)
```

Train MODEL

In [13]:

```
clf.fit(x_train, y_train)
```

Out[13]:

```
MLPClassifier(alpha=1e-05, hidden_layer_sizes=(5, 2), random_state=1,
              solver='lbfgs')
```

Make Predictions

In [14]:

```
y_pred = clf.predict(x_test)
y_pred
```

Out[14]:

```
array([1, 1, 1, 1, 1, 1, 1, 6, 4, 1, 6, 1, 4, 4, 5, 3, 1, 6, 1, 4, 1, 5,
       1, 6, 1, 4, 4, 1, 1, 6], dtype=int64)
```

In [15]:

```
from sklearn.metrics import confusion_matrix
```

CONFUSION Matrix

In [16]:

```
confusion_matrix(y_test, y_pred)
```

Out[16]:

```
array([[11,  0,  0,  0,  0,  0,  0],
       [ 5,  0,  1,  0,  0,  0,  0],
       [ 0,  0,  0,  1,  0,  0,  0],
       [ 0,  0,  0,  4,  0,  0,  0],
       [ 0,  0,  0,  0,  2,  0,  0],
       [ 0,  0,  0,  0,  0,  5,  0],
       [ 0,  0,  0,  1,  0,  0,  0]], dtype=int64)
```

Classification Report

In [17]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           1       0.69      1.00      0.81        11
           2       0.00      0.00      0.00         6
           3       0.00      0.00      0.00         1
           4       0.67      1.00      0.80         4
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00         5
           7       0.00      0.00      0.00         1

    accuracy                           0.73        30
   macro avg       0.48      0.57      0.52        30
weighted avg       0.57      0.73      0.64        30

S:\Anaconda\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
S:\Anaconda\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
S:\Anaconda\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
In [ ]:
```