In [1]:

```
##### Author : Amir Shokri
##### github link : https://github.com/amirshnll/Abalone
##### dataset link : http://archive.ics.uci.edu/ml/datasets/Abalone
##### email : amirsh.nll@gmail.com
```

In [ ]:

```
import sklearn
import pandas as pd
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

In [ ]:

```
#read file
df = pd.read_csv("D:\\abalone.txt", header=None)
for char in df:
    df = df.replace('M','1')
    df = df.replace('F','-1')
    df = df.replace('I','0')
df

#separate the feature columns from the target column.
features = [0,1,2,3,4,5,6,7]
X = df[features]
y = df[8]
print(X)
print(y)
```

In [ ]:

```
#separate the Training data and Test data
X_train, X_test, y_train, y_test = train_test_split(X,y,random_state=1, test_s
ize=0.2)
# Feature scaling
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

In [ ]:

```
# Finally for the MLP- Multilayer Perceptron
mlp = MLPClassifier(max_iter=1000)
mlp.fit(X_train, y_train.values.ravel())
```

In [ ]:

```
#In the prediction step, the model is used to predict the response for given d
ata.
predictions = mlp.predict(X_test)
print(predictions)
```

In [23]:

```
# Last thing: evaluation of algorithm performance in classifying
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
```

```
[[ 0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  8  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3 10  5  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  5 14 22  9  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  8 36 26  8  1  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3 22 34 38  8  0  0  1  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  1  9 32 54 30  5  0  1  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  8 12 60 28 28  1  6  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  4 31 21 25  4  6  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  1 23 16 14  0  6  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  3  4  9 15  1  6  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  2 10  2  1  6  1  0  1  0  1  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  3  8  5  1  3  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  4  1  0  2  0  0  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  3  3  2  1  1  0  0  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  1  2  3  4  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  3  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  1  0  3  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  1  0  0  0  0  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0]]
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.00      0.00      0.00         2
           4       0.53      0.73      0.62        11
           5       0.56      0.50      0.53        20
           6       0.45      0.26      0.33        53
           7       0.35      0.45      0.39        80
           8       0.28      0.32      0.30       107
           9       0.23      0.41      0.30       132
          10       0.20      0.20      0.20       143
          11       0.24      0.27      0.26        91
          12       0.00      0.00      0.00        63
          13       0.13      0.15      0.14        39
          14       0.50      0.04      0.08        24
          15       0.00      0.00      0.00        21
          16       0.15      0.20      0.17        10
          17       0.00      0.00      0.00        12
          18       0.00      0.00      0.00        10
          19       0.00      0.00      0.00         7
          20       0.00      0.00      0.00         4
          21       0.00      0.00      0.00         3
          22       0.00      0.00      0.00         1
          23       0.00      0.00      0.00         2

    accuracy                           0.26       836
   macro avg       0.16      0.16      0.15       836
weighted avg       0.24      0.26      0.24       836
```

In [ ]: