In [10]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
import sklearn
import warnings
from sklearn.metrics import classification_report, confusion_matrix
warnings.filterwarnings('ignore')

ann = pd.read_csv("D:\iris.csv")
ann.columns = ["sepal.length","sepal.width","petal.length","petal.width","variety"]
ann.head(10)
```

Out[10]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

In [12]:

```
X = df[["sepal.length","sepal.width","petal.length","petal.width"]]
y = df["variety"]

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3)

mlp = MLPClassifier(hidden_layer_sizes=(15,15),max_iter=500)
mlp.fit(X_train,y_train)
```

Out[12]:

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.
9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(15, 15), learning_rate='constant',
              learning_rate_init=0.001, max_iter=500, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=None, shuffle=True, solver='adam', tol=0.0001,
              validation_fraction=0.1, verbose=False, warm_start=False)
```

In [11]:

```
predict_function = mlp.predict(X_test)

print(confusion_matrix(y_test,predict_function))
print(classification_report(y_test,predict_function))
```

```
[[12  0  0]
 [ 0 15  1]
 [ 0  0 17]]
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        12
  versicolor       1.00      0.94      0.97        16
   virginica       0.94      1.00      0.97        17

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

In [ ]:

In [ ]:

ANALYSIS:

- Here,'iris' dataset is used for ANN.
- Here the nth entry in the tuple represents the number of neurons in the nth layer of the MLP model.
- Here i used warning library for removing error of warning.
- I chose 2 layers with the same number of neurons which is 15.
- here,Number of iteration is 500
- This problem is basically works on "confusion matrix".
- In confusion matrix,there are 2 parts:Actual and prection.
- Based on this matrix,i found parameters like precision,recall,f1-score,support etc.

In [ ]:

In [ ]: