In [11]:

```
import pandas as pd
path="C:\\Users\\TANUJA HARISH\\Desktop\\ML and DL Summer
Internship\\diabetes.csv"
dataset=pd.read_csv(path)
dataset.head()
```

Out[11]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [12]:

```
print(dataset.shape)
```

```
(768, 9)
```

In [13]:

```
import numpy as np
x=np.array(dataset.iloc[:,0:8])
y=np.array(dataset[["Outcome"]])
print(x.shape)
print(y.shape)
```

```
(768, 8)
(768, 1)
```

In [14]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_norm=sc.fit_transform(x)
print(x_norm.shape)
```

```
(768, 8)
```

In [15]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_norm,y,test_size=0.2)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(614, 8)
(154, 8)
(614, 1)
(154, 1)
```

In [16]:

```
from sklearn.neural_network import MLPClassifier
mlpclassifier=MLPClassifier(hidden_layer_sizes=(5),max_iter=500,activation=
"tanh",solver="sgd",random_state=42)
mlpclassifier.fit(x_train,y_train)
```

```
C:\Users\TANUJA
HARISH\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptr
on.py:1109: DataConversionWarning: A column-vector y was passed when a 1d
array was expected. Please change the shape of y to (n_samples, ), for example
```

```
using ravel().
  y = column_or_1d(y, warn=True)
```

Out[16]:

```
MLPClassifier(activation='tanh', hidden_layer_sizes=5, max_iter=500,
              random_state=42, solver='sgd')
```

In [17]:

```
 y_pred=mlpclassifier.predict(x_test)
 print(y_pred)
```

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

In [18]:

```
 from sklearn.metrics import
 classification_report,confusion_matrix,accuracy_score
 cm=np.array(confusion_matrix(y_test,y_pred,labels=[0,1]))
 print(cm)
 confusion=pd.DataFrame(cm,index=["Diabetics","No
 Diabetics"],columns=["Diabetics","No Diabetics"])
 print(confusion)
```

```
print(classification_report(y_test,y_pred,labels=[0,1]))
acc=accuracy_score(y_test,y_pred)
print(acc)
```

```
[[87 11]
 [27 29]]
              Diabetics  No Diabetics
Diabetics            87            11
No Diabetics         27            29
              precision    recall  f1-score   support

           0       0.76      0.89      0.82        98
           1       0.72      0.52      0.60        56

    accuracy                           0.75       154
   macro avg       0.74      0.70      0.71       154
weighted avg       0.75      0.75      0.74       154

0.7532467532467533
```

In [ ]: