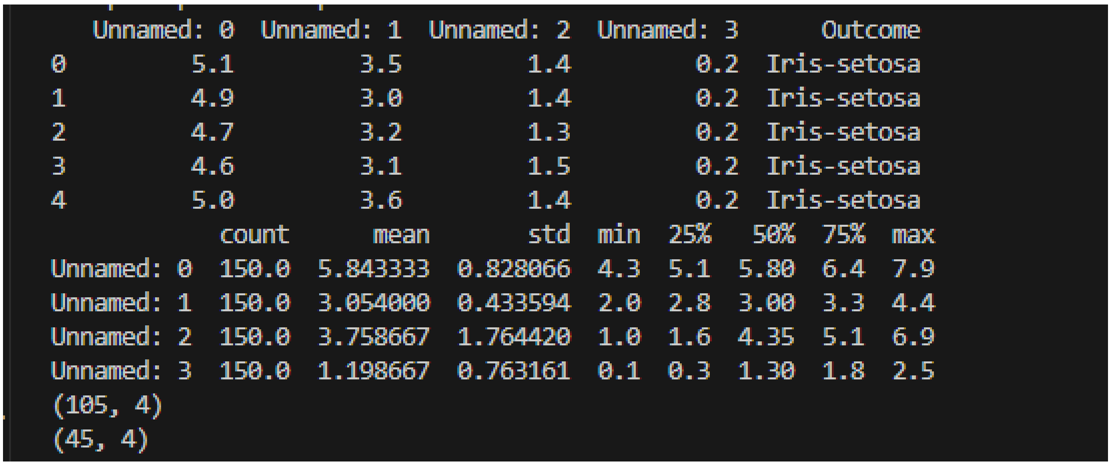
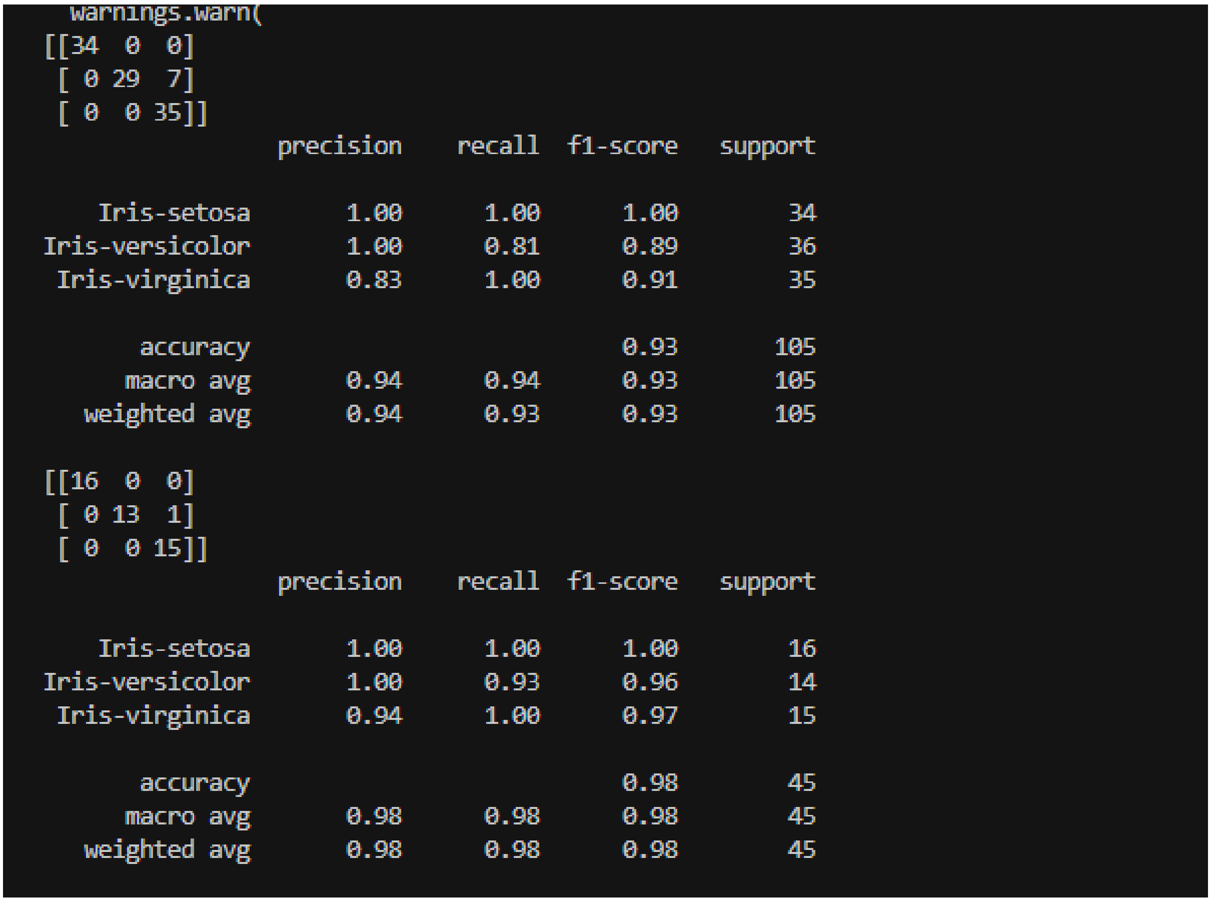
```
from numpy import*
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
df =pd.read_csv('iris.csv')
df.head()
df.describe().transpose()
#Creating Arrays for the Features and the Response Variable
target_column = ['Outcome']
predictors = list(set(list(df.columns))-set(target_column))
df[predictors] = df[predictors]/df[predictors].max()
df.describe().transpose()
#Creating the Training and Test Datasets
X = df[predictors].values
y = df[target_column].values
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.30,random_state=40)
print(X_train.shape); print(X_test.shape)
mlp = MLPClassifier(hidden_layer_sizes=(8,8),
activation='relu', solver='sgd', max_iter=500)
mlp.fit(X_train,y_train)
predict_train = mlp.predict(X_train)
predict_test = mlp.predict(X_test)

print(confusion_matrix(y_train,predict_train))
print(classification_report(y_train,predict_train))
#The next step is to evaluate the performance of the model on the test data that is done with the lines of code below.
print(confusion_matrix(y_test,predict_test))
print(classification_report(y_test,predict_test))
```



```
   Unnamed: 0  Unnamed: 1  Unnamed: 2  Unnamed: 3      Outcome
0         5.1         3.5         1.4         0.2  Iris-setosa
1         4.9         3.0         1.4         0.2  Iris-setosa
2         4.7         3.2         1.3         0.2  Iris-setosa
3         4.6         3.1         1.5         0.2  Iris-setosa
4         5.0         3.6         1.4         0.2  Iris-setosa
            count      mean       std  min  25%   50%  75%  max
Unnamed: 0  150.0  5.843333  0.828066  4.3  5.1  5.80  6.4  7.9
Unnamed: 1  150.0  3.054000  0.433594  2.0  2.8  3.00  3.3  4.4
Unnamed: 2  150.0  3.758667  1.764420  1.0  1.6  4.35  5.1  6.9
Unnamed: 3  150.0  1.198667  0.763161  0.1  0.3  1.30  1.8  2.5
(105, 4)
(45, 4)
```



```
  warnings.warn(
[[34  0  0]
 [ 0 29  7]
 [ 0  0 35]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        34
Iris-versicolor       1.00      0.81      0.89        36
 Iris-virginica       0.83      1.00      0.91        35

       accuracy                           0.93       105
      macro avg       0.94      0.94      0.93       105
   weighted avg       0.94      0.93      0.93       105

[[16  0  0]
 [ 0 13  1]
 [ 0  0 15]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        16
Iris-versicolor       1.00      0.93      0.96        14
 Iris-virginica       0.94      1.00      0.97        15

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