OUTPUT

The following steps have been performed on the dataset of tic tac toe end game obtained from UCI ML repository.

- Read the dataset into dataframe
- Pre-processing of dataset; scaling of attributes
- Splitting the data randomly
- Model Creation
- Evaluation

The following is the configuration of the learner:

```
mlp = MLPClassifier(activation = 'relu', solver = 'lbfgs', hidden_layer_sizes=(9,7,5))
mlp.fit(X_train,y_train)
```

```
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=(9, 7, 5), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='lbfgs', tol=0.0001, validation_fraction=0.1,
       verbose=False, warm_start=False)
```

The parameter for this results were as follows:

Size of Testing Data is 25% of the total dataset.

Hidden Layer Size: (9, 7, 5)

Accuracy on Training Data: 98%

Accuracy on Testing Data: 85%

Below are the results of the model evaluation.

Confusion matrix and Classification Report for Training Data:

In [359]:
```
predictions = mlp.predict(X_train)
print("Confusion Matrix for testing on Training Set:")
print(confusion_matrix(y_train,predictions))
```

```
Confusion Matrix for testing on Training Set:
[[234   9]
 [  8 467]]
```

In [360]:
```
print("Classification Report on training set\n\n")
print(classification_report(y_train,predictions))
```

```
Classification Report on training set


             precision    recall  f1-score   support

          0       0.97      0.96      0.96       243
          1       0.98      0.98      0.98       475

avg / total       0.98      0.98      0.98       718
```

Confusion matrix and Classification Report for Testing Data:

In [361]:
```
predictions = mlp.predict(X_test)
print("Confusion Matrix for testing on Testing Set:")
print(confusion_matrix(y_test,predictions))
```

```
Confusion Matrix for testing on Testing Set:
[[ 68  21]
 [ 14 137]]
```

In [362]:
```
print("Classification Report on testing set\n\n")
print(classification_report(y_test,predictions))
```

```
Classification Report on testing set


             precision    recall  f1-score   support

          0       0.83      0.76      0.80        89
          1       0.87      0.91      0.89       151

avg / total       0.85      0.85      0.85       240
```