In [ ]:
```
The MLPClassifier can be used for "multiclass classification", "binary clas

Multiclass: The outmost layer is the softmax layer

Multilabel or Binary-class: The outmost layer is the logistic/sigmoid.

Regression: The outmost layer is identity
```

In [9]:
```
#Step1: Like always first we will import the modules
from sklearn.datasets import load_iris
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pandas as pd
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
```

In [3]:
```
iris_data = load_iris()
X = pd.DataFrame(iris_data.data, columns=iris_data.feature_names)
y = iris_data.target
```

In [5]:
```
X_train, X_test, y_train, y_test =train_test_split(X,y,random_state=1, test
sc_X = StandardScaler()
X_trainscaled=sc_X.fit_transform(X_train)
X_testscaled=sc_X.transform(X_test)
```

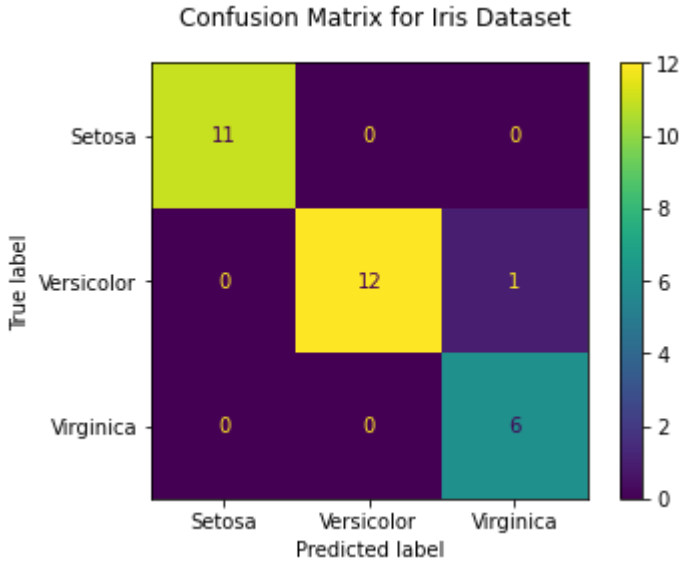
In [6]:
```
clf = MLPClassifier(hidden_layer_sizes=(256,128,64,32),activation="relu"
,random_state=1).fit(X_trainscaled,y_train)
y_pred=clf.predict(X_testscaled)
print(clf.score(X_testscaled, y_test))
```

```
0.9666666666666667
```

In [10]:
```
classification_report(y_test,y_pred )
```

Out[10]:
```
'              precision    recall  f1-score   support\n\n           0
1.00      1.00      1.00        11\n           1       1.00      0.92
0.96        13\n           2       0.86      1.00      0.92         6\n\n
accuracy                           0.97        30\n   macro avg       0.95
0.97      0.96        30\nweighted avg       0.97      0.97      0.97
30\n'
```

In [7]:
```
fig=plot_confusion_matrix(clf, X_testscaled, y_test,display_labels=["Setosa
fig.figure_.suptitle("Confusion Matrix for Iris Dataset")
plt.show()
```



In [ ]:
```

```