```
#load dataset
from sklearn.datasets import load_iris
iris=load_iris()
x=iris.data
y=iris.target
print(x)
print(y)
```

Start coding or generate with AI.

```
#split datasets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
print(x.shape)
print(x_train.shape)
print(x_test.shape)
print(y.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(150, 4)
(105, 4)
(45, 4)
(150,)
(105,)
(45,)
```

```
#Train model
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

```
from sklearn import metrics
metrics.accuracy_score(y_pred,y_test)
```

```
0.9111111111111111
```

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_pred,y_test)
print(cm)
```

```
[[14  0  0]
 [ 0 15  1]
 [ 0  3 12]]
```

```
#pre-processing of data
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)

knn1=KNeighborsClassifier(n_neighbors=5)
knn1.fit(x_train,y_train)
pred1=knn1.predict(x_test)
```

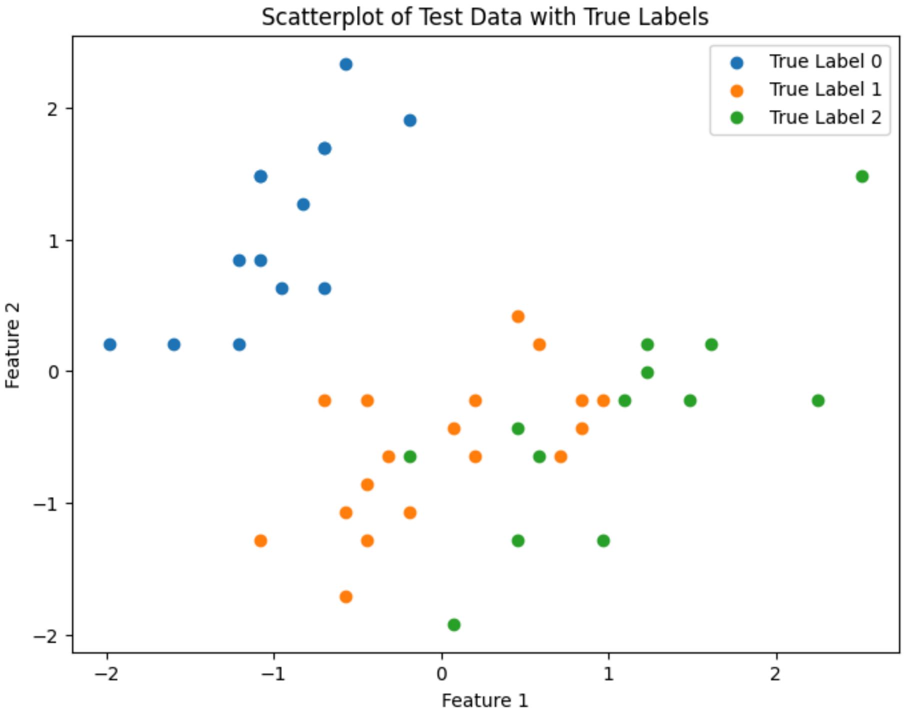
```
from sklearn import metrics
metrics.accuracy_score(pred1,y_test)
```

```
0.9111111111111111
```

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_pred,y_test)
print(cm)
```

```
[[14  0  0]
 [ 0 15  1]
 [ 0  3 12]]
```

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
for i in range(3):
    plt.scatter(x_test[y_test == i, 0], x_test[y_test == i, 1], label=f'True Label {i}')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Scatterplot of Test Data with True Labels')
plt.legend()
plt.show()
```


Scatterplot of Test Data with True Labels
True Label 0
True Label 1
True Label 2
Feature 2
Feature 1