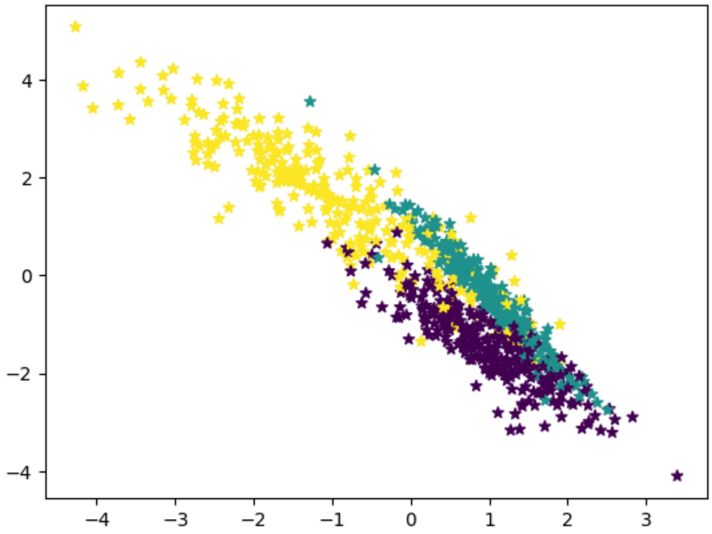
```
from sklearn.datasets import make_classification
X, y = make_classification(
n_features=6,
n_classes=3,
n_samples=800,
n_informative=2,
random_state=1,
n_clusters_per_class=1,)
import matplotlib.pyplot as plt
plt.scatter(X[:, 0], X[:, 1], c=y, marker="*");
```



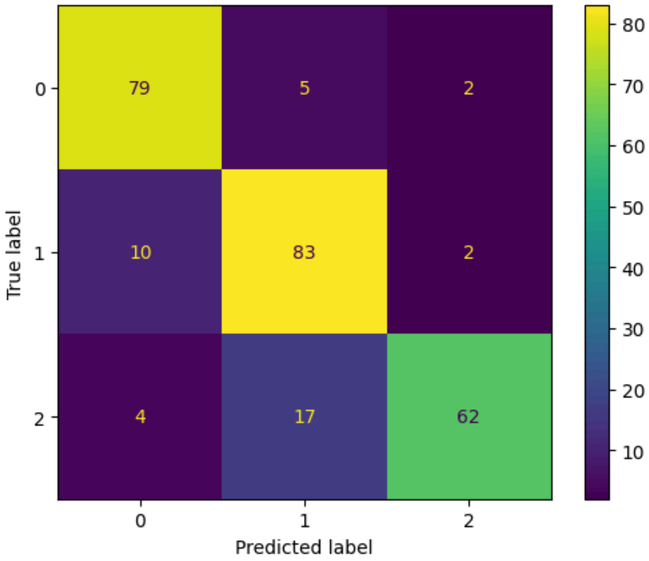
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.33, random_state=125)
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
predicted = model.predict([X_test[6]])
print("Actual Value:", y_test[6])
print("Predicted Value:", predicted[0])
```

```
Actual Value: 0
Predicted Value: 0
```

```
from sklearn.metrics import (accuracy_score,confusion_matrix,ConfusionMatrixDisplay,f1_score,)
y_pred = model.predict(X_test)
print("accuray",accuracy_score(y_pred, y_test))
print("f1",f1_score(y_pred, y_test, average="weighted"))
```

```
accuray 0.8484848484848485
f1 0.8491119695890328
```

```
labels = [0,1,2]
cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot();
```


79
5
2
10
83
2
4
17
62
0
1
2
True label
0
1
2
Predicted label
80
70
60
50
40
30
20
10