```
!pip install scikit-learn matplotlib seaborn
```

Show hidden output

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import (
    accuracy_score, confusion_matrix, classification_report, roc_auc_score, roc_curve
    )
```

```
#loading dataset
data = load_breast_cancer()
```

```
#organizing data
label_names = data['target_names']
labels = data['target']
feature_names = data['feature_names']
features = data['data']
```

```
#data preprocessing
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

```
#splitting the data
train, test, train_labels, test_labels = train_test_split(
      features_scaled, labels, test_size=0.33, random_state=42
)
```

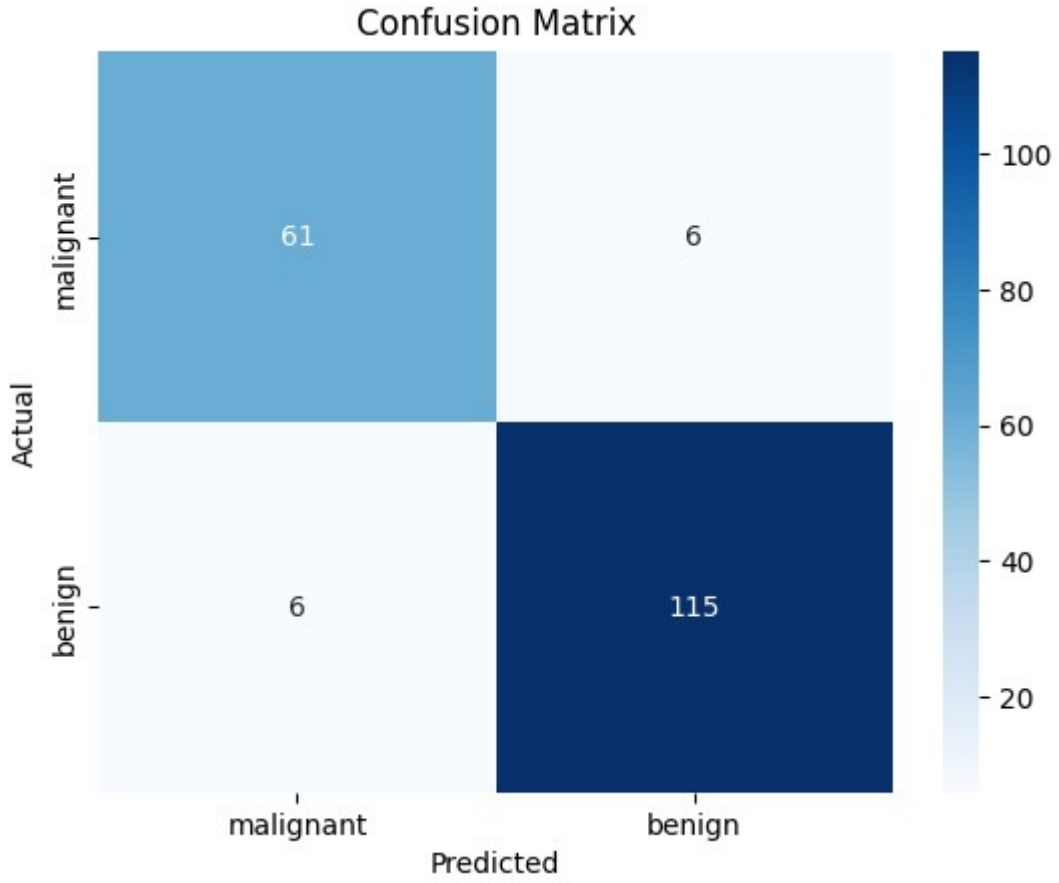
```
#training the classifier
gnb = GaussianNB()
model = gnb.fit(train, train_labels)
```

```
#predictions
predictions = gnb.predict(test)
print(predictions)
#accuracy
print(f"Accuracy Score: {accuracy_score(test_labels, predictions):.2f}")
```

```
[1 0 0 1 1 0 0 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0
 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0
 1 1 1 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0
 1 1 0 0 0 1 1 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 1 0 1 0 0
 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0
 0 1 1]
```

```
 ט ו ו]
Accuracy Score: 0.94
```

```
#confusion Matrix
cm = confusion_matrix(test_labels, predictions)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", xticklabels=label_names, yticklabel
plt.title("Confusion Matrix")
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.show()
```




```
#classification Report
print("Classification Report:\n", classification_report(test_labels, predictions))
```

```
Classification Report:
                precision    recall  f1-score   support

           0       0.91      0.91      0.91        67
           1       0.95      0.95      0.95       121

    accuracy                           0.94       188
   macro avg       0.93      0.93      0.93       188
weighted avg       0.94      0.94      0.94       188
```

```
#cross-validation score
cv_scores = cross_val_score(gnb, features_scaled, labels, cv=5)
print(f"Cross-Validation Accuracy: {cv_scores.mean():.2f}")
```

```
print(f"Cross-Validation Accuracy: {cv_scores.mean():.2f}")
```

```
Cross-Validation Accuracy: 0.93
```

```
#ROC Curve
probs = gnb.predict_proba(test)[:, 1]
fpr, tpr, _ = roc_curve(test_labels, probs)
roc_auc = roc_auc_score(test_labels, probs)

plt.figure()
plt.plot(fpr, tpr, color="darkorange", label=f"ROC curve (area = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], color="navy", linestyle="--")
plt.title("Receiver Operating Characteristic (ROC)")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend(loc="lower right")
plt.show()
```