ASSIGNMENT NO 3

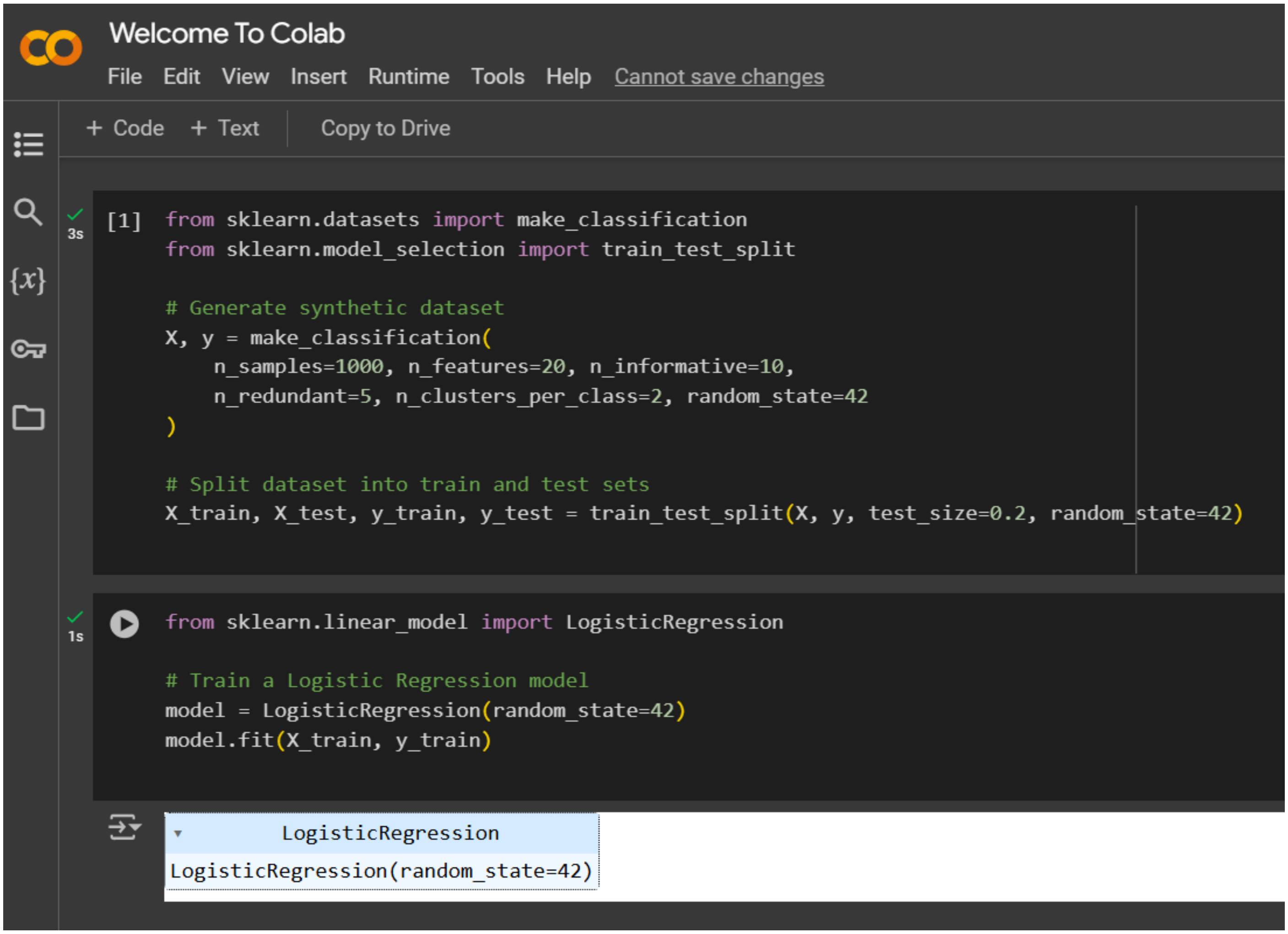
COURSE: ARTIFICIAL INTELLIGENCE

SUBMITTED TO : SIR ZUBAIR

SUBMITTED BY:

NAME	SHEEZA NASEER
ROLL NO	203

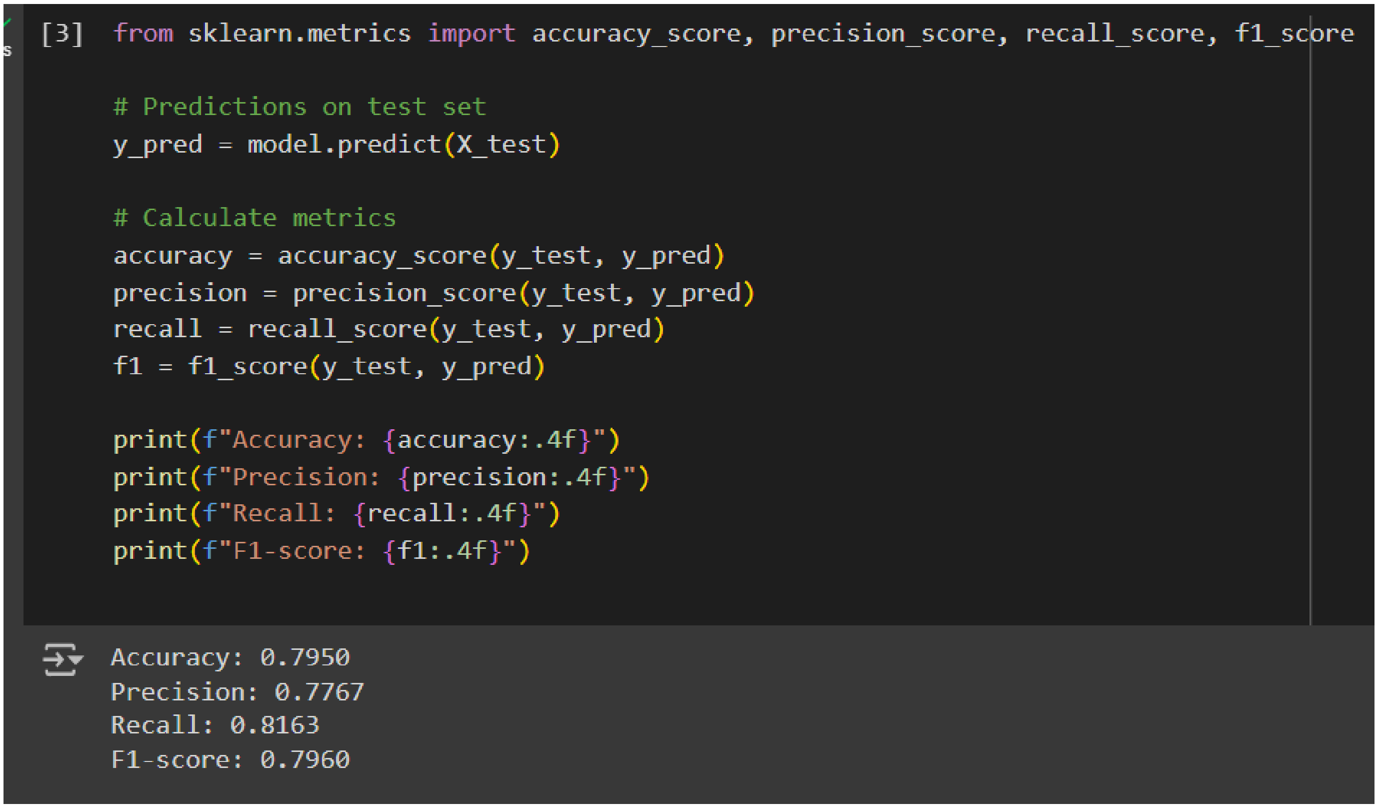
International Institute of Science Art and Technology, Gujranwala


Welcome To Colab
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
3s
1s

```
[1] from sklearn.datasets import make_classification
    from sklearn.model_selection import train_test_split

    # Generate synthetic dataset
    X, y = make_classification(
        n_samples=1000, n_features=20, n_informative=10,
        n_redundant=5, n_clusters_per_class=2, random_state=42
    )

    # Split dataset into train and test sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

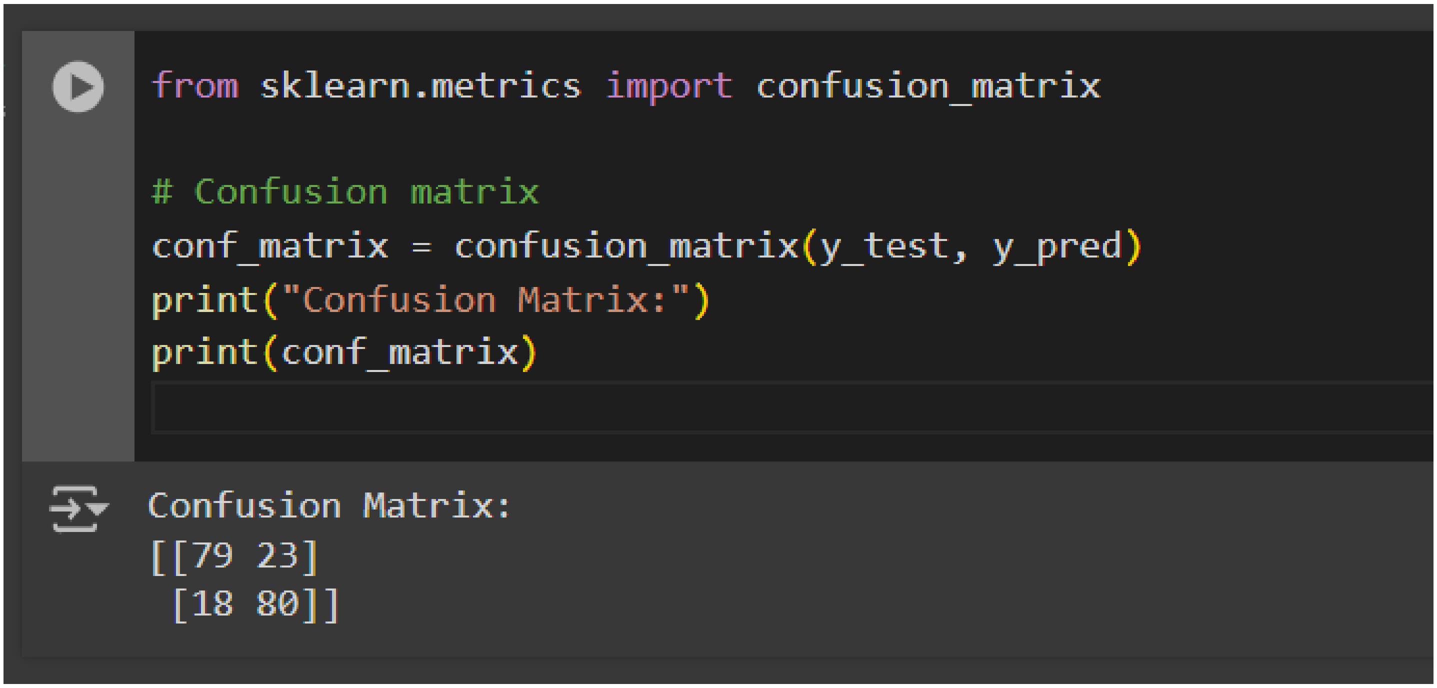
```
from sklearn.linear_model import LogisticRegression

# Train a Logistic Regression model
model = LogisticRegression(random_state=42)
model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression(random_state=42)
```



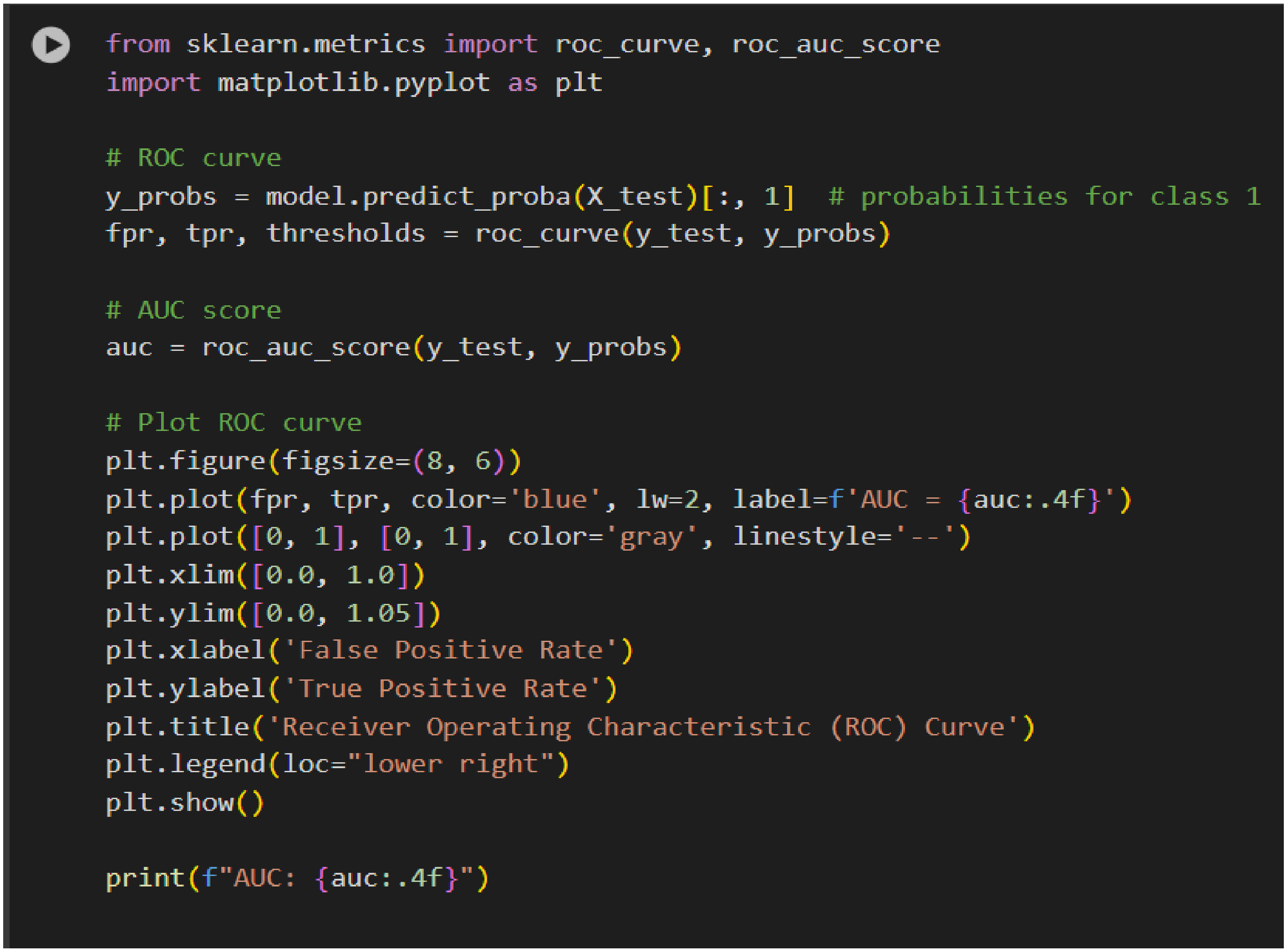
```
[3] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

    # Predictions on test set
    y_pred = model.predict(X_test)

    # Calculate metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1-score: {f1:.4f}")
```

```
Accuracy: 0.7950
Precision: 0.7767
Recall: 0.8163
F1-score: 0.7960
```

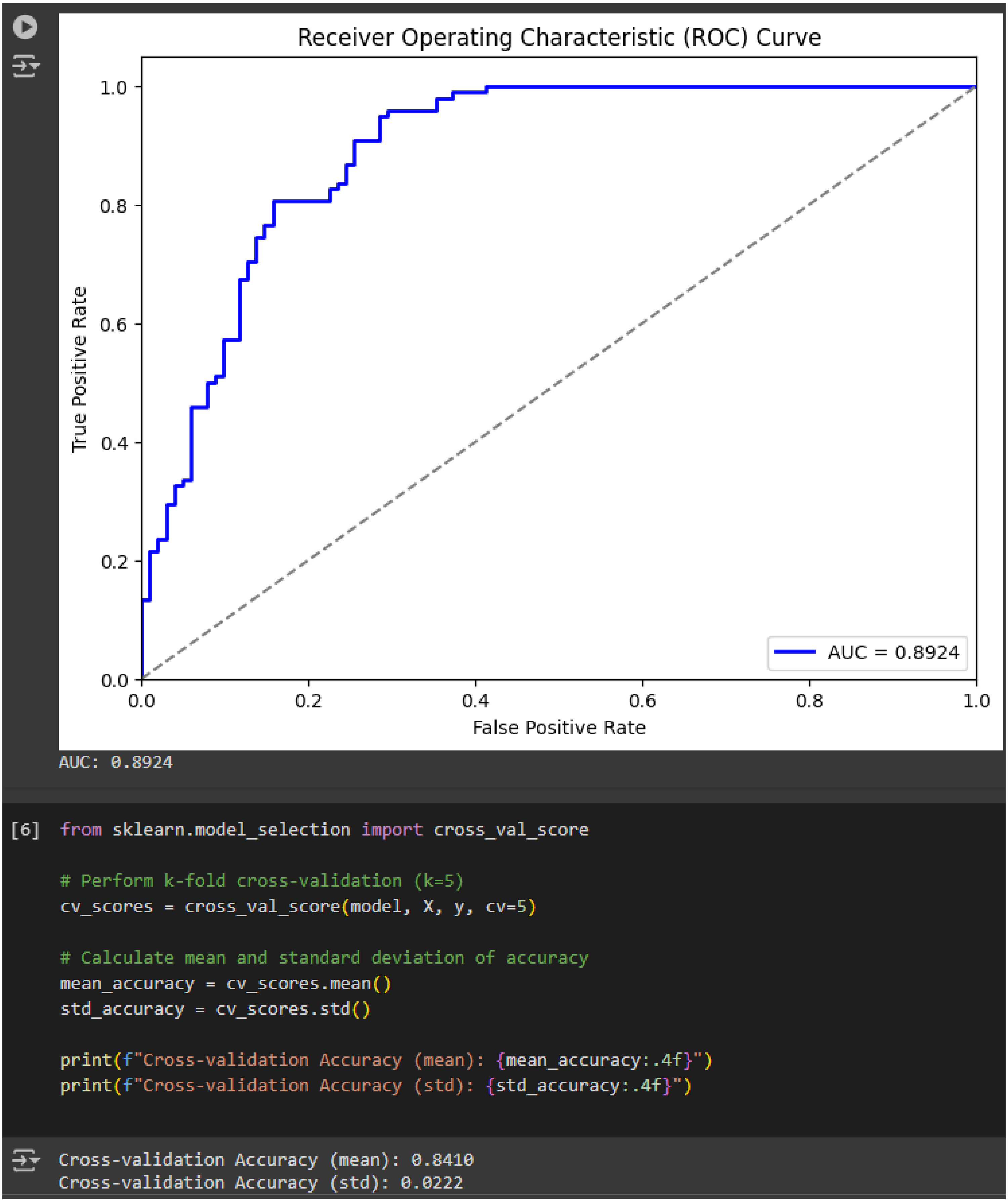


```python
from sklearn.metrics import confusion_matrix

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[79 23]
 [18 80]]
```



```python
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

# ROC curve
y_probs = model.predict_proba(X_test)[:, 1]  # probabilities for class 1
fpr, tpr, thresholds = roc_curve(y_test, y_probs)

# AUC score
auc = roc_auc_score(y_test, y_probs)

# Plot ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', lw=2, label=f'AUC = {auc:.4f}')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()

print(f"AUC: {auc:.4f}")
```



```
AUC: 0.8924
```

```
[6] from sklearn.model_selection import cross_val_score

    # Perform k-fold cross-validation (k=5)
    cv_scores = cross_val_score(model, X, y, cv=5)

    # Calculate mean and standard deviation of accuracy
    mean_accuracy = cv_scores.mean()
    std_accuracy = cv_scores.std()

    print(f"Cross-validation Accuracy (mean): {mean_accuracy:.4f}")
    print(f"Cross-validation Accuracy (std): {std_accuracy:.4f}")
```

```
Cross-validation Accuracy (mean): 0.8410
Cross-validation Accuracy (std): 0.0222
```