1. Accuracy Metrics Calculation

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

# Load the dataset

from sklearn.datasets import load\_iris

data = load\_iris()

X = data.data

y = data.target

# Binarize the output (only consider two classes for simplicity)

y = (y != 0) \* 1

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Standardize the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Train a Logistic Regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = model.predict(X\_test)

# Calculate accuracy, precision, recall, and F1-score

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)

accuracy, precision, recall, f1

**2. Confusion Matrix Interpretation**

from sklearn.metrics import confusion\_matrix

# Generate the confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

cm

1. ROC/AUC Calculation
2. from sklearn.metrics import roc\_curve, roc\_auc\_score
3. import matplotlib.pyplot as plt
4. # Calculate the probabilities for the positive class
5. y\_prob = model.predict\_proba(X\_test)[:, 1]
6. # Calculate the ROC curve
7. fpr, tpr, thresholds = roc\_curve(y\_test, y\_prob)
8. roc\_auc = roc\_auc\_score(y\_test, y\_prob)
9. # Plot the ROC curve
10. plt.figure()
11. plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc\_auc)
12. plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
13. plt.xlim([0.0, 1.0])
14. plt.ylim([0.0, 1.05])
15. plt.xlabel('False Positive Rate')
16. plt.ylabel('True Positive Rate')
17. plt.title('Receiver Operating Characteristic (ROC)')
18. plt.legend(loc="lower right")
19. plt.show()
20. roc\_auc
21. Cross-Validation Reporting
22. from sklearn.model\_selection import cross\_val\_score
23. # Perform 5-fold cross-validation
24. cv\_scores = cross\_val\_score(model, X, y, cv=5, scoring='accuracy')
25. # Calculate mean and standard deviation of accuracy
26. mean\_cv\_score = np.mean(cv\_scores)
27. std\_cv\_score = np.std(cv\_scores)
28. mean\_cv\_score, std\_cv\_score