from sklearn.datasets import load\_breast\_cancer

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

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

# Load the dataset

data = load\_breast\_cancer()

X, y = data.data, data.target

# Split the dataset into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, 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(random\_state=42)

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

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

# Make predictions on the test set

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

# Calculate the accuracy 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)

accuracy, precision, recall, f1

from sklearn.metrics import confusion\_matrix

import seaborn as sns

import matplotlib.pyplot as plt

# Generate the confusion matrix

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

# Plot the confusion matrix

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=data.target\_names, yticklabels=data.target\_names)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

cm

from sklearn.metrics import roc\_curve, roc\_auc\_score

# Predict the probabilities

y\_prob = model.predict\_proba(X\_test)[:, 1]

# Calculate the ROC curve

fpr, tpr, \_ = roc\_curve(y\_test, y\_prob)

roc\_auc = roc\_auc\_score(y\_test, y\_prob)

# Plot the ROC curve

plt.figure()

plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc\_auc)

plt.plot([0, 1], [0, 1], color='navy', lw=2, 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()

roc\_auc

from sklearn.model\_selection import cross\_val\_score

import numpy as np

# Perform 5-fold cross-validation

cv\_scores = cross\_val\_score(model, X, y, cv=5, scoring='accuracy')

mean\_cv\_accuracy = np.mean(cv\_scores)

std\_cv\_accuracy = np.std(cv\_scores)

mean\_cv\_accuracy, std\_cv\_accuracy