```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
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
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix, roc_curve, auc
import seaborn as sns

data = load_breast_cancer()
X = data.data
y = data.target
feature_names = data.feature_names
target_names = data.target_names

df = pd.DataFrame(X, columns=feature_names)
df['target'] = y

print("Dataset Information:")
print(f"Number of samples: {X.shape[0]}")
print(f"Number of features: {X.shape[1]}")
print(f"Target classes: {target_names}")
print(f"Class distribution: {np.bincount(y)}")
print("\n")

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = LogisticRegression(random_state=42, max_iter=200)
model.fit(X_train_scaled, y_train)

y_pred = model.predict(X_test_scaled)
y_pred_prob = model.predict_proba(X_test_scaled)[:, 1]

print("Model Evaluation:")
print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred,
target_names=target_names))

conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=target_names, yticklabels=target_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.tight_layout()

fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area =
{roc_auc:.2f})')
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')
plt.legend(loc="lower right")
plt.tight_layout()

coef = pd.DataFrame(
{'Feature': feature_names,
'Coefficient': model.coef_[0]}
).sort_values('Coefficient', ascending=False)

plt.figure(figsize=(12, 8))
sns.barplot(x='Coefficient', y='Feature', data=coef.head(10))
plt.title('Top 10 Features with Highest Positive Coefficients')
plt.tight_layout()

plt.figure(figsize=(12, 8))
sns.barplot(x='Coefficient', y='Feature', data=coef.tail(10))
plt.title('Top 10 Features with Lowest Negative Coefficients')
plt.tight_layout()

class CustomLogisticRegression:
def __init__(self, learning_rate=0.01, num_iterations=1000):
self.learning_rate = learning_rate
self.num_iterations = num_iterations
self.weights = None
self.bias = None
def sigmoid(self, z):
return 1 / (1 + np.exp(-z))
def fit(self, X, y):
num_samples, num_features = X.shape
self.weights = np.zeros(num_features)
self.bias = 0
```

```
for i in range(self.num_iterations):
linear_model = np.dot(X, self.weights) + self.bias
y_predicted = self.sigmoid(linear_model)
dw = (1 / num_samples) * np.dot(X.T, (y_predicted - y))
db = (1 / num_samples) * np.sum(y_predicted - y)
self.weights -= self.learning_rate * dw
self.bias -= self.learning_rate * db
def predict_prob(self, X):
linear_model = np.dot(X, self.weights) + self.bias
return self.sigmoid(linear_model)
def predict(self, X, threshold=0.5):
probabilities = self.predict_prob(X)
return [1 if i > threshold else 0 for i in probabilities]

custom_model = CustomLogisticRegression(learning_rate=0.01,
num_iterations=1000)
custom_model.fit(X_train_scaled, y_train)
custom_y_pred = custom_model.predict(X_test_scaled)

print("\nCustom Logistic Regression Model:")
print(f"Accuracy: {accuracy_score(y_test, custom_y_pred):.4f}")
print("\nClassification Report:")
print(classification_report(y_test, custom_y_pred,
target_names=target_names))
```

```
ishadpande@Argos:~/Documents/dev/practicals/PS$ /usr/bin/python /home/ishadpande/Documents/dev/practicals/PS/5.py
Dataset Information:
Number of samples: 569
Number of features: 30
Target classes: ['malignant' 'benign']
Class distribution: [212 357]


Model Evaluation:
Accuracy: 0.9790

Classification Report:
              precision    recall  f1-score   support

   malignant       0.96      0.98      0.97        54
      benign       0.99      0.98      0.98        89

    accuracy                           0.98       143
   macro avg       0.98      0.98      0.98       143
weighted avg       0.98      0.98      0.98       143


Custom Logistic Regression Model:
Accuracy: 0.9860

Classification Report:
              precision    recall  f1-score   support

   malignant       0.98      0.98      0.98        54
      benign       0.99      0.99      0.99        89

    accuracy                           0.99       143
   macro avg       0.99      0.99      0.99       143
weighted avg       0.99      0.99      0.99       143
```