**PROGRAM 10 :**

Develop a program to implement k-means clustering using Wisconsin Breast Cancer data set and visualize the clustering result.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_breast\_cancer

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.metrics import confusion\_matrix, classification\_report

# Load dataset

data = load\_breast\_cancer()

X = data.data

y = data.target

# Standardize the features

scaler = StandardScaler()

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

# Apply K-Means clustering

kmeans = KMeans(n\_clusters=2, random\_state=42, n\_init=10)

y\_kmeans = kmeans.fit\_predict(X\_scaled)

# Evaluation

print("Confusion Matrix:")

print(confusion\_matrix(y, y\_kmeans))

print("\nClassification Report:")

print(classification\_report(y, y\_kmeans))

# Reduce dimensions with PCA

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

# Create DataFrame for visualization

df = pd.DataFrame(X\_pca, columns=['PC1', 'PC2'])

df['Cluster'] = y\_kmeans

df['True Label'] = y

# Plot K-Means clustering

plt.figure(figsize=(8, 6))

sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster', palette='Set1', s=100, edgecolor='black', alpha=0.7)

plt.title('K-Means Clustering of Breast Cancer Dataset')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.legend(title="Cluster")

plt.show()

# Plot true labels

plt.figure(figsize=(8, 6))

sns.scatterplot(data=df, x='PC1', y='PC2', hue='True Label', palette='coolwarm', s=100, edgecolor='black', alpha=0.7)

plt.title('True Labels of Breast Cancer Dataset')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.legend(title="True Label")

plt.show()

# Plot K-Means clustering with centroids

plt.figure(figsize=(8, 6))

sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster', palette='Set1', s=100, edgecolor='black', alpha=0.7)

centers = pca.transform(kmeans.cluster\_centers\_)

plt.scatter(centers[:, 0], centers[:, 1], s=200, c='red', marker='X', label='Centroids')

plt.title('K-Means Clustering with Centroids')

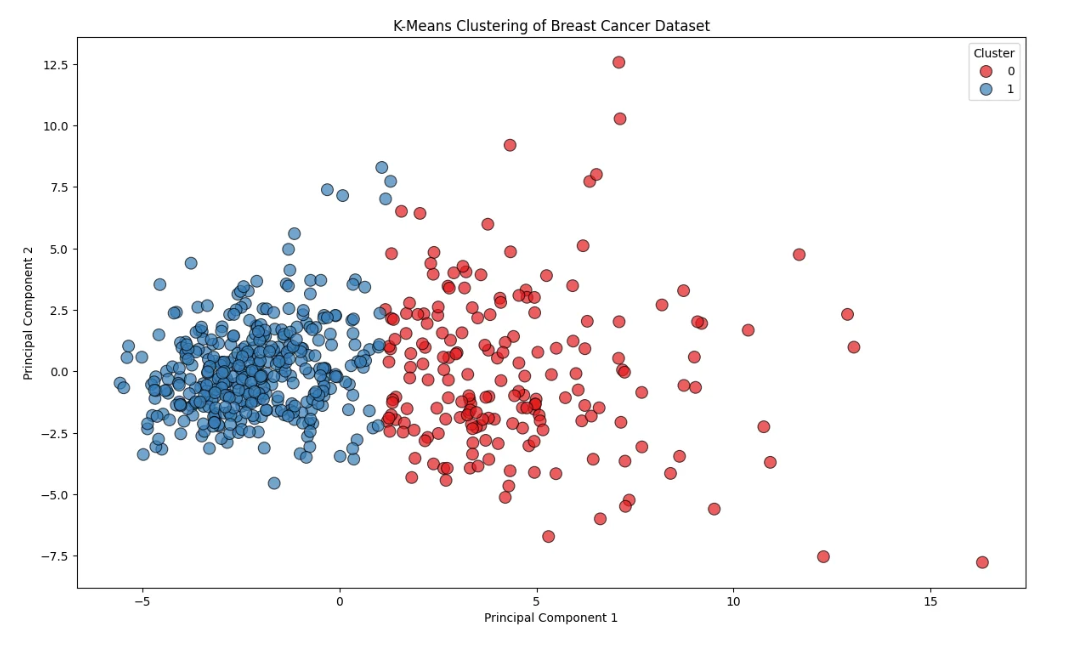
plt.xlabel('Principal Component 1')

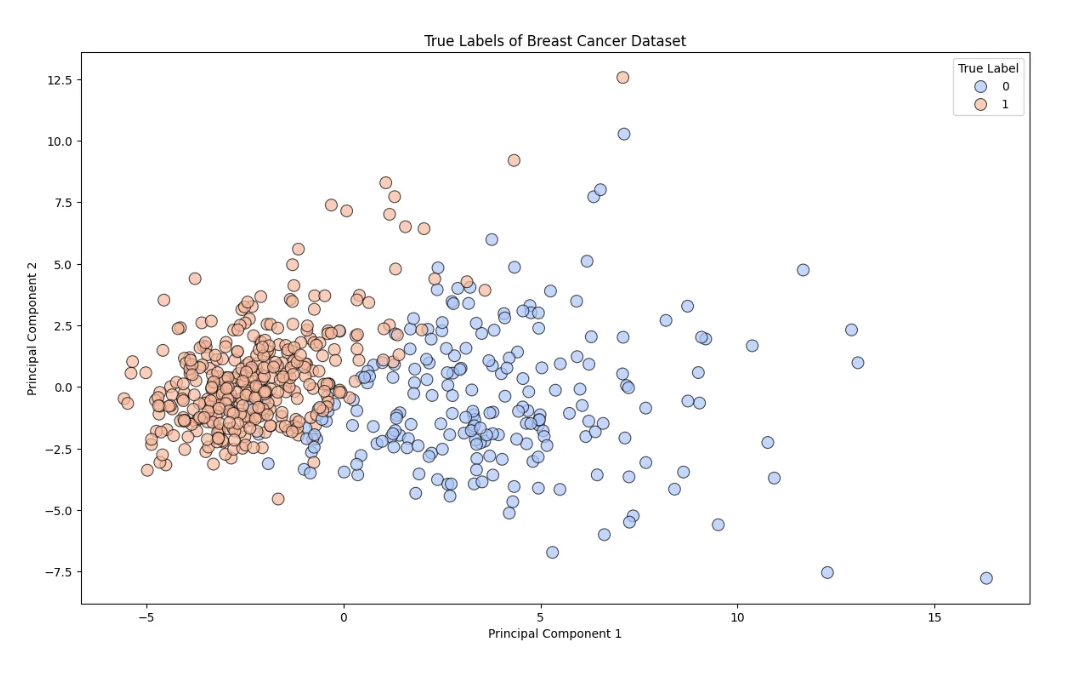
plt.ylabel('Principal Component 2')

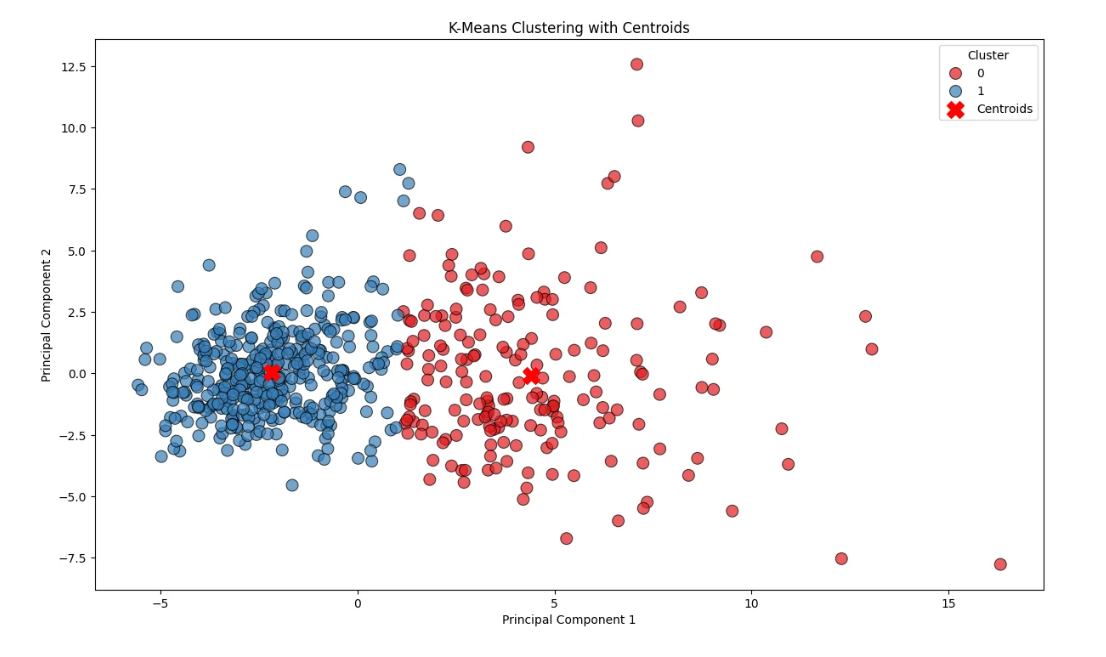
plt.legend(title="Cluster")

plt.show()

OUTPUT :







Confusion Matrix:

[[175 37]

[ 13 344]]Classification Report:

precision recall f1-score support0 0.93 0.83 0.88 212

1 0.90 0.96 0.93 357accuracy 0.91 569

macro avg 0.92 0.89 0.90 569

weighted avg 0.91 0.91 0.91 569