import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import datasets

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

from sklearn.svm import SVC

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

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import StandardScaler

print("All Module Imported Successfully")

data = datasets.load\_breast\_cancer()

#Find the col names

target\_values = data.target

class\_counts = pd.Series(target\_values).value\_counts()

print(class\_counts)

data = datasets.load\_breast\_cancer()

X = data.data  # Features

y = data.target  # Target labels (0 for malignant, 1 for benign)

#Creating the DataFrame

df = pd.DataFrame(X)

#Describing the X Feature

df.describe()

#Standardising the Features of the Dataset using MinMaxScaler

scaler = MinMaxScaler()

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

#Standardising the Features of the Dataset using StandardScaler

scaler = StandardScaler()

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

#Splitting the Data into train and test set

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

#Scaling the Features in the DataSet using MinMaxScaler

scaler = MinMaxScaler()

#Scaling the Features in the DataSet using StandardScaler

scaler = StandardScaler()

#Normalising the Data

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

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

svm\_classifier = SVC(kernel='poly', C=1.0) #Change the Kernal type to see the Desired Result

#Fitting the Normalised data into the Model

svm\_classifier.fit(X\_train\_normalized, y\_train)

y\_pred = svm\_classifier.predict(X\_test\_normalized)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

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

print("Confusion Matrix:\n", conf\_matrix)

class\_report = classification\_report(y\_test, y\_pred)

print("Classification Report:\n", class\_report)