#Importing necessary libraries

import pandas as pd

import numpy as np

import sklearn

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder , MinMaxScaler

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

from sklearn.neighbors import KNeighborsClassifier

#Load the dataset (breast cancer data)

breast\_cancer = pd.read\_csv('/content/breast cancer classification dataset.csv')

breast\_cancer.head(5)

print()

print()

#Observation of data

print('Shape of data (row,column):',breast\_cancer.shape)

print()

print('Summation of null values in each column :')

print(breast\_cancer.isnull().sum())

print()

print('Observing all the data types:')

print(breast\_cancer.info())

print()

print()

#Droping highest null value's and unique values column (here it is-->id,Unnamed: 32) and checking

breast\_cancer = breast\_cancer.drop(['id','Unnamed: 32'],axis=1)

print('Shape of data set after droping necessary column :',breast\_cancer.shape)

print()

print()

#Imputing Missing values and checking

impute = SimpleImputer(missing\_values=np.nan,strategy='mean')

impute.fit(breast\_cancer[['radius\_mean']])

breast\_cancer['radius\_mean'] = impute.transform(breast\_cancer[['radius\_mean']])

impute.fit(breast\_cancer[['fractal\_dimension\_worst']])

breast\_cancer['fractal\_dimension\_worst'] = impute.transform(breast\_cancer[['fractal\_dimension\_worst']])

print('Summation of null values in each column after imputation:')

print(breast\_cancer.isnull().sum())

print()

print()

#Encode categorical feature and checking

print('Checking unique values in categorical feature column:')

print(breast\_cancer['diagnosis'].unique())

print()

labelling = LabelEncoder()

breast\_cancer['diagnosis'] = labelling.fit\_transform(breast\_cancer['diagnosis'])

print('after encoding the values:')

print(breast\_cancer['diagnosis'])

print()

print()

#Scale the data (using MinMax scaller)

scaler = MinMaxScaler()

scaler.fit(breast\_cancer)

breast\_cancer = scaler.transform(breast\_cancer)

print('After Scaling the data set')

print(breast\_cancer)

print()

print()

#Split dataset into feature and label

x = np.hsplit(breast\_cancer, np.array([1,breast\_cancer[1].size]))

label = np.array(x[0])

features = np.array(x[1])

print('After spliting the dataset in label and features:')

print('label shape',label.shape)

print('features shape',features.shape)

print()

print()

# (Optional) train and test

print('Checking Prediction (optional):')

x\_train,x\_test,y\_train,y\_test = train\_test\_split(features,label,test\_size = 0.25,random\_state = 0)

knn = KNeighborsClassifier()

knn.fit(x\_train,y\_train)

print('Accuracy of the model is: ',knn.score(x\_test,y\_test))

print()

print()