import numpy as np

import pandas as pd

from google.colab import drive

drive.mount('/content/drive')

dataSet=pd.read\_csv('/content/drive/My Drive/breast cancer classification dataset.csv')

dataSet.head(3)

dataSet.isnull().sum()

from sklearn.impute import SimpleImputer

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

impute.fit(dataSet[['radius\_mean']])

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

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

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

dataSet=dataSet.drop(['Unnamed: 32'], axis=1)

dataSet.isnull().sum()

dataSet['diagnosis'].unique()

dataSet.head(10)

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

X=dataSet.iloc[:,:-1]

y=dataSet.iloc[:,:-1]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.2,random\_state=1)

print(X\_train.shape)

print(X\_test.shape)

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaler.fit(X\_train)

MinMaxScaler(copy=True, feature\_range=(0, 1))

X\_train\_scaled = scaler.transform(X\_train)

print("per-feature minimum before scaling:\n {}".format(X\_train.min(axis=0)))

print("per-feature maximum before scaling:\n {}".format(X\_train.max(axis=0)))

print("per-feature minimum after scaling:\n {}".format(

X\_train\_scaled.min(axis=0)))

print("per-feature maximum after scaling:\n {}".format(

X\_train\_scaled.max(axis=0)))

feature=[['radius\_mean', 'texture\_mean', 'perimeter\_mean', 'area\_mean', 'smoothness\_mean', 'compactness\_mean', 'concavity\_mean', 'concave', 'points\_mean', 'symmetry\_mean', 'fractal\_dimension\_mean', 'radius\_se', 'texture\_se', 'perimeter\_se', 'area\_se', 'smoothness\_se', 'compactness\_se', 'concavity\_se', 'concave', 'points\_se', 'symmetry\_se', 'fractal\_dimension\_se', 'radius\_worst', 'texture\_worst', 'perimeter\_worst', 'area\_worst', 'smoothness\_worst', 'compactness\_worst', 'concavity\_worst', 'concave', 'points\_worst', 'symmetry\_worst', 'fractal\_dimension\_worst']]

label=[['diagnosis']]