**Experiment 4**

**Aim :** Perform an experiment on data pre-processing.

**Description :** Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

**Following normalization techniques are used:**

1. Rescale Data

2. Binarize Data (Make Binary)

3. Standardize Data

**Rescale Data**

When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.

**Python code to Rescale data (between 0 and 1)**

# importing libraries

import pandas

import scipy

import numpy

from sklearn.preprocessing import MinMaxScaler

# data set link

url =

"https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima

-indians-diabetes.data"

# data parameters

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

# preparating of dataframe using the data at given link and defined columns list

dataframe = pandas.read\_csv(url, names = names)

array = dataframe.values

# separate array into input and output components

X = array[:,0:8]

Y = array[:,8]

# initialising the MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0, 1))

# learning the statistical parameters for each of the data and transforming

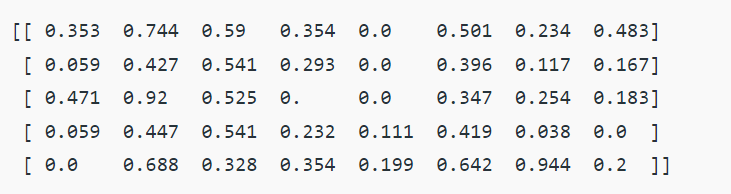
rescaledX = scaler.fit\_transform(X)

# summarize transformed data

numpy.set\_printoptions(precision=3)

print(rescaledX[0:5,:])

OUTPUT



**Binarize Data (Make Binary)**

We can transform our data using a binary threshold. All values above

the threshold are marked 1 and all equal to or below are marked as 0.

**Code: Python code for binarization**

# import libraries

from sklearn.preprocessing import Binarizer

import pandas

import numpy

# data set link

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data"

# data parameters

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

# preparating of dataframe using the data at given link and defined columns list

dataframe = pandas.read\_csv(url, names = names)

array = dataframe.values

# separate array into input and output components

X = array[:, 0:8]

Y = array[:, 8]

binarizer = Binarizer(threshold = 0.0).fit(X)

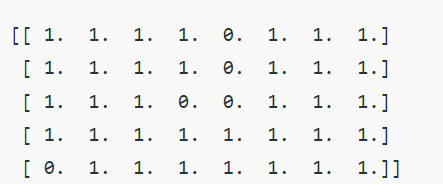
binaryX = binarizer.transform(X)

# summarize transformed data

numpy.set\_printoptions(precision = 3)

print(binaryX[0:5,:])

OUTPUT



**3. Standardize Data**

Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

**Code: Python code to Standardize data (0 mean, 1 stdev)**

# importing libraries

from sklearn.preprocessing import StandardScaler

import pandas

import numpy

# data set link

Url =

"[https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-dia](https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data)

[betes/pima-indians-diabetes.data](https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data)"

# data parameters

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',

'class']

# preparating of dataframe using the data at given link and defined columns list

dataframe = pandas.read\_csv(url, names = names)

array = dataframe.values

# separate array into input and output components

X = array[:, 0:8]

Y = array[:, 8]

scaler = StandardScaler().fit(X)

rescaledX = scaler.transform(X)

# summarize transformed data

numpy.set\_printoptions(precision = 3)

print(rescaledX[0:5,:])

OUTPUT

