“”” **Rain_Prediction**

Name - Shubham Thorat “””

“”” DATASHEET (1st 10 rows)

month	day	FFMC	DMC	DC	ISI	temp	RH	wind	area	rain
mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
mar	fri	91.7	33.3	77.5	9	8.3	97	4	0	0.2
mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0

“””

#Code

importing the libraries

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

importing the data set

dataset=pd.read_csv('university_ranking.csv')

X = dataset.iloc[:,:-1].values

Y = dataset.iloc[:,10].values

/*

X=

```
array([['mar', 'fri', 86.2, 26.2, 94.3, 5.1, 8.2, 51, 6.7, 0.0],
       ['oct', 'tue', 90.6, 35.4, 669.1, 6.7, 18.0, 33, 0.9, 0.0],
       ['oct', 'sat', 90.6, 43.7, 686.9, 6.7, 14.6, 33, 1.3, 0.0],
       ['mar', 'fri', 91.7, 33.3, 77.5, 9.0, 8.3, 97, 4.0, 0.0],
```

['mar', 'sun', 89.3, 51.3, 102.2, 9.6, 11.4, 99, 1.8, 0.0],

['aug', 'sun', 92.3, 85.3, 488.0, 14.7, 22.2, 29, 5.4, 0.0],

['aug', 'mon', 92.3, 88.9, 495.6, 8.5, 24.1, 27, 3.1, 0.0],

['aug', 'mon', 91.5, 145.4, 608.2, 10.7, 8.0, 86, 2.2, 0.0],

['sep', 'tue', 91.0, 129.5, 692.6, 7.0, 13.1, 63, 5.4, 0.0],

['sep', 'sat', 92.5, 88.0, 698.6, 7.1, 22.8, 40, 4.0, 0.0]]

Y=

array([0. , 0. , 0. , 0.2, 0. , 0. , 0. , 0. , 0. , 0.])

*/

handling categorical (encoding) data

from sklearn.preprocessing import LabelEncoder

label_encoder_X = LabelEncoder()

label_encoder_Y = LabelEncoder()

X[:,0] = label_encoder_X.fit_transform(X[:,0])

X[:,1] = label_encoder_X.fit_transform(X[:,1])

/*

X=

array([[7, 0, 86.2, 26.2, 94.3, 5.1, 8.2, 51, 6.7, 0.0],

[10, 5, 90.6, 35.4, 669.1, 6.7, 18.0, 33, 0.9, 0.0],

[10, 2, 90.6, 43.7, 686.9, 6.7, 14.6, 33, 1.3, 0.0],

[7, 0, 91.7, 33.3, 77.5, 9.0, 8.3, 97, 4.0, 0.0],

[7, 3, 89.3, 51.3, 102.2, 9.6, 11.4, 99, 1.8, 0.0],

[1, 3, 92.3, 85.3, 488.0, 14.7, 22.2, 29, 5.4, 0.0],

[1, 1, 92.3, 88.9, 495.6, 8.5, 24.1, 27, 3.1, 0.0],

[1, 1, 91.5, 145.4, 608.2, 10.7, 8.0, 86, 2.2, 0.0],

[11, 5, 91.0, 129.5, 692.6, 7.0, 13.1, 63, 5.4, 0.0],

[11, 2, 92.5, 88.0, 698.6, 7.1, 22.8, 40, 4.0, 0.0]]

***/**

dividing dataset into test and training dataset

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = .2, random_state = 0)

/*

X_train =

array([[11, 3, 90.5, 96.7, 750.5, 11.4, 20.6, 55, 5.4, 24.59],

[11, 5, 90.3, 80.7, 730.2, 6.3, 17.8, 63, 4.9, 0.0],

[1, 6, 91.7, 191.4, 635.9, 7.8, 19.9, 50, 4.0, 82.75],

[7, 3, 89.3, 51.3, 102.2, 9.6, 11.5, 39, 5.8, 0.0],

[1, 2, 92.2, 81.8, 480.8, 11.9, 16.4, 43, 4.0, 71.3]]

Y_train = array([0., 0., 0., 0.])

X-test =

array([[7, 2, 90.6, 50.1, 100.4, 7.8, 15.2, 31, 8.5, 1.94],

[1, 6, 95.1, 141.3, 605.8, 17.7, 20.6, 58, 1.3, 0.0],

[1, 4, 90.7, 194.1, 643.0, 6.8, 16.2, 63, 2.7, 16.33],

[3, 2, 83.9, 8.0, 30.2, 2.6, 12.7, 48, 1.8, 0.0]]

Y_test = array([0., 0., 0., 0.])

*/

feature scaling

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

X_train = sc_X.fit_transform(X_train)

X_test = sc_X.fit_transform(X_test)

/*

X_train =

array([[1.22376001, 0.13759402, -0.02655293, -0.23158779, 0.83368606,
         0.5003457 , 0.29733733, 0.66067453, 0.77617805, 0.31952982],
       [1.22376001, 1.18993347, -0.06158922, -0.47816501, 0.75243841,
        -0.5910774 , -0.1856531 , 1.14554095, 0.49997477, -0.23201558],
       [-1.07985386, 1.7161032 , 0.18366482, 1.22784108, 0.37501705,
        -0.27007061, 0.17658972, 0.35763301, 0.00280885, 1.62403899],
       [0.30231446, 0.13759402, -0.23677067, -0.93125063, -1.76103577,
         0.11513755, -1.27238155, -0.30905832, 0.99714068, -0.23201558],
       [-1.07985386, -0.38857571, 0.27125555, -0.46121282, -0.24574704,
         0.60734796, -0.42714831, -0.06662511, 0.00280885, 1.36721936]])

X_test =

array([[0.21522155, -0.36344089, -0.00370832, -0.95403205, -1.97414518,
        -0.26211086, -0.64314708, -0.90752789, 2.57799633, -0.18962445],
       [-1.12776091, 1.62591976, 0.96045396, 0.5623916 , 0.14719367,
         2.18235783, 0.28536551, 0.83790931, -1.64793111, -0.20727012],
       [-1.12776091, 0.63123944, 0.01771751, 1.44032107, 0.30333496,
        -0.50902689, -0.4712003 , 1.16113842, -0.82622299, -0.05873726],
       [-0.68010009, -0.36344089, -1.43923882, -1.65404778, -2.26879889,
        -1.54607422, -1.07301401, 0.19145109, -1.35446392, -0.20727012]])

*/