# -\*- coding: utf-8 -\*-

"""

Created on Thu Jun 28 21:57:49 2018

@author: shubham b thorat

Rain\_prediction

"""

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

dataset=pd.read\_csv('rain\_prediction.csv')

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

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

"""from sklearn.preprocessing import Imputer

imputer = Imputer(missing\_values="NaN", strategy="mean",axis=0)

imputer = imputer.fit(X[:,2:11], y=None)

X[:,2:11] = imputer.transform(X[:,2:11])"""

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])

"""onehotencoder = OneHotEncoder(categorical\_features= [0])

X = onehotencoder.fit\_transform(X).toarray()

onehotencoder = OneHotEncoder(categorical\_features= [1])

X = onehotencoder.fit\_transform(X).toarray()"""

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)

# 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)

from sklearn.linear\_model import LinearRegression

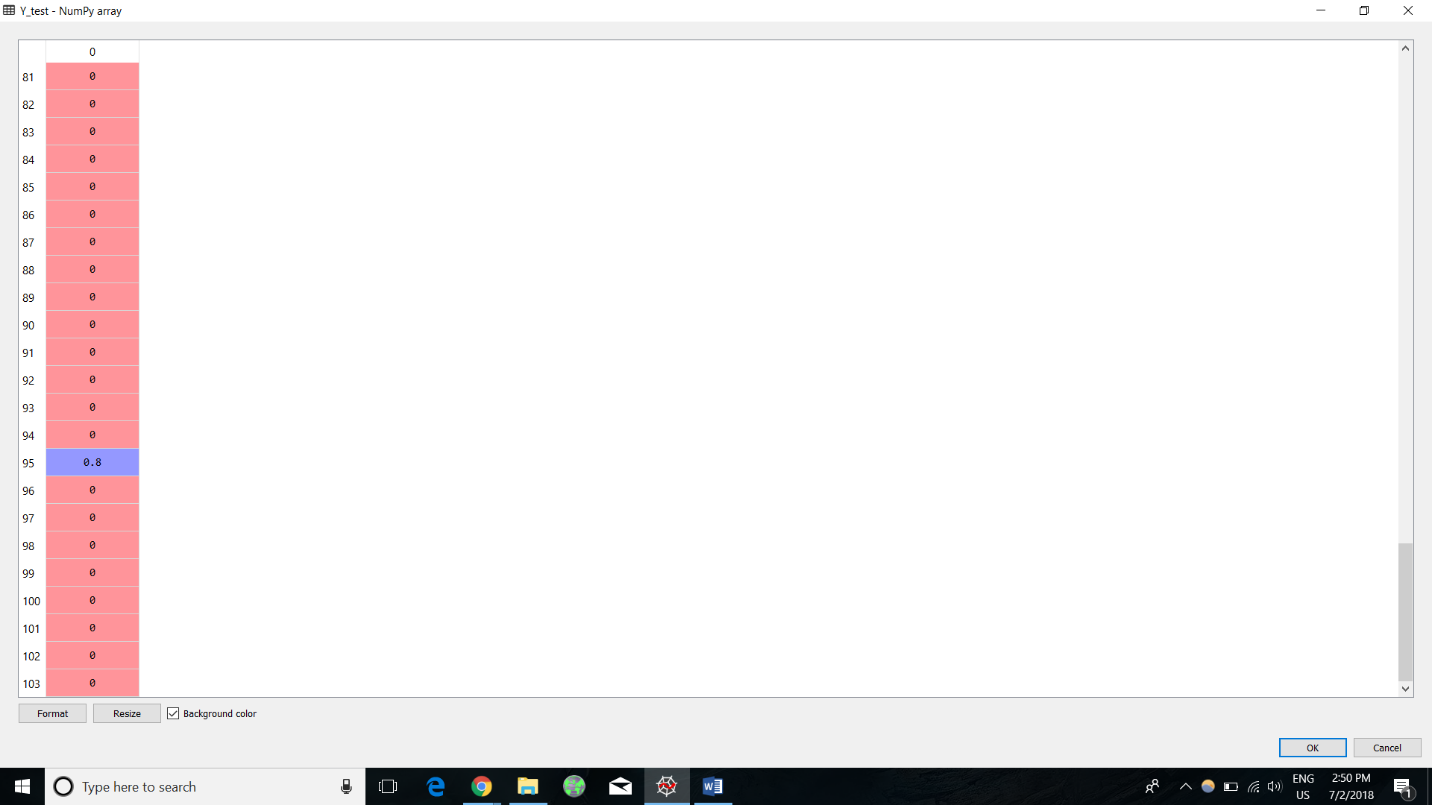
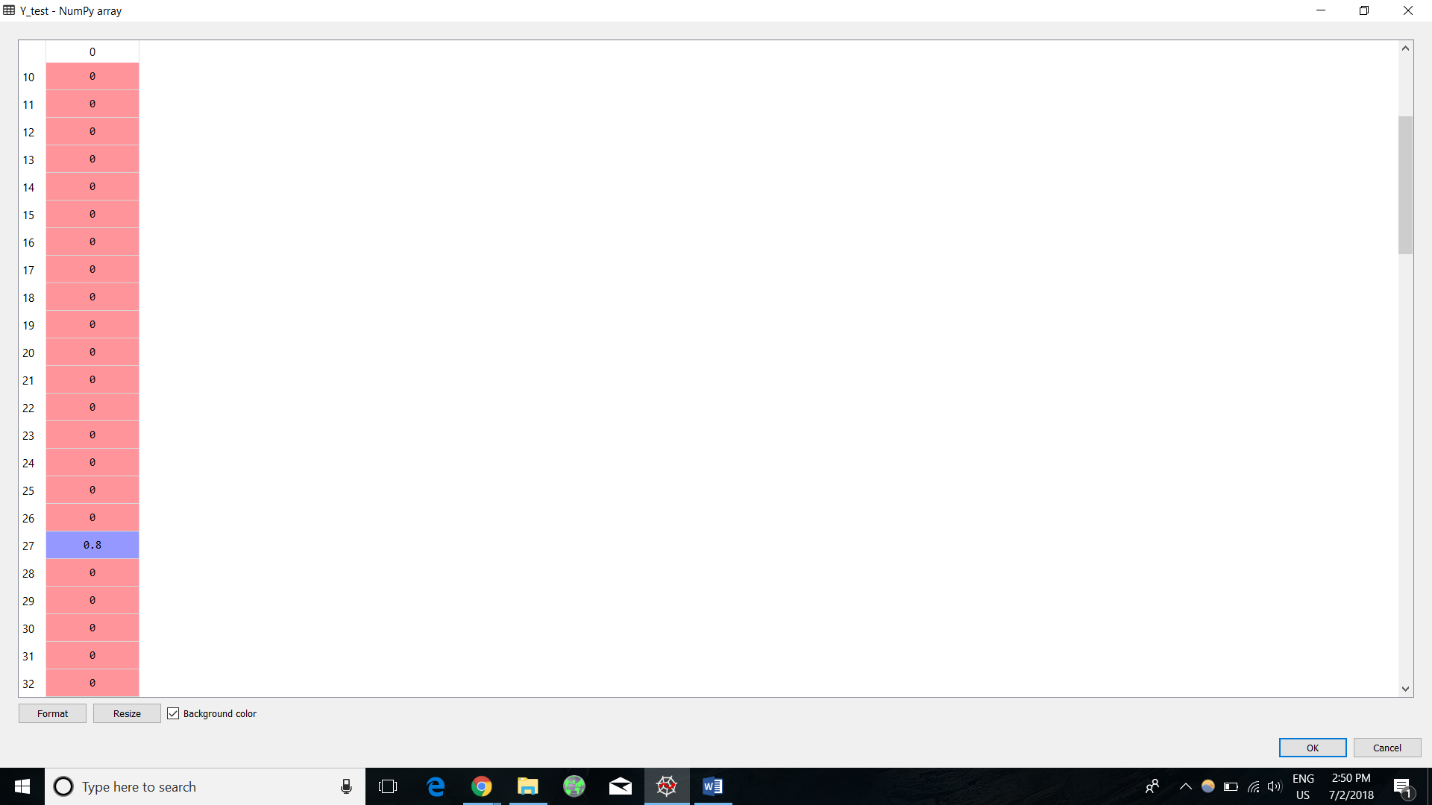
regressor = LinearRegression()

regressor.fit(X\_train,Y\_train)

#predicting the test set result

y\_pred = regressor.predict(X\_test)

y\_test =



Y\_pred =

