```
#ADAGRAD
import numpy as np
import pandas as pd
data= pd.read_csv("C:/Users/mohanarupa/Desktop/dataset.csv")
x=np.array(data.iloc[:,1:2])
y=np.array(data.iloc[:,2:3])
train_x = x[:3]
train_y = y[:3]
test_x = x[3:]
test_y = y[3:]
epochs=100000
m=1
c=1
Gm=Gc=0
n=0.1
e=10**-8
ite=0
while(ite<epochs):
    sample=0
    while sample<len(train_x):
        val=train_y[sample]-m*train_x[sample]-c
        gm= -val*train_x[sample]
        gc= -val
        Gm=Gm+(gm*gm)
        Gc=Gc+(gc*gc)
        delm=(-n/((Gm+e)**0.5))*gm
        delc=(-n/((Gc+e)**0.5))*gc
        m=m+delm
        c=c+delc
        sample+=1
    ite+=1
print("m=",m,"c=",c)
q=0
for i in range(len(x)):
  q += ((y[i]-(m*x[i]+c))**2)
mse_train = q/len(train_x)
print("meansquare error of trained data:",mse_train)
q=0
for i in range(len(test_x)):
  q += ((test_y[i]-(m*test_x[i]+c))**2)
mse_test = q/len(test_x)
print("mean square error of test data:",mse_test)

print("Y predicted for x=0.8 is: ")
y_model =[]
for i in range(0,len(test_x)):
  y_model.append(m*test_x[i]+c)
print(y_model[0])
```

```
m= [2.] c= [3.]
```

```
meansquare error of trained data: [9.09727544e-27]
mean square error of test data: [1.68153591e-26]
Y predicted for x=0.8 is:
[4.6]
```