NAME: Penugonda Abhignanendra **REGNO**:19BCE1677
FACULTY: Dr. Lakshmi Pathi Jakkamputi **SLOT**: L21, L22

Essential of Data Analytics
Tasks for Week-10: Gradient Descent algorithm

Aim: Implementing the Momentum Gradient Descent Algorithm

Algorithm:

1) Clear the space by clearing the list using rm() function.
2) Then declare function MGD using function() with x1,x2,y,m1,m2,c,learning rate(alpha), gamma, number of iterations as the parameters of the function.
3) Declare a variable iterations and assign a value 0.
4) Then declare variables LF, u_m1, u_c and assign value 0.
5) Run a while loop and check the condition if iterations are less than or equal to given number of iterations.
6) Then declare the variable y_pred and assign m1 * x1 + m2 * x2 +c value to y_pred.
7) Then declare variable Lf_new and assign 0.5 * sum of difference of y_pred and y power of 2.
8) Declare and assign variable nu_m1 as gamma * u_m1 + alpha *sum of difference of y_pred and y * x1.
9) Declare and assign variable nu_m2 as gamma * u_m2 + alpha *sum of difference of y_pred and y * x2.
10) Declare and assign variable nu_c as gamma * u_c + alpha *sum of difference of y_pred and y.
11) Assign m1 as difference of m1 and nu_m1.
12) Assign m2 as difference of m2 and nu_m2.
13) Assign c as difference of c and nu_m1.
14) Assign u_m1 as nu_m1.
15) Assign u_m2 as nu_m2.
16) Assign u_c as nu_c.
17) Assign u_c as nu_c.
18) Assign Lf as Lf_new.
19) Then increase iterations to 1.
20) End the loop.
21) Then load the dataset.
22) Plot the two variables that chosen for applying momentum gradient descent using plot() function.

NAME: Penugonda Abhignanendra **REGNO**:19BCE1677
FACULTY: Dr. Lakshmi Pathi Jakkamputi **SLOT**: L21, L22

Essential of Data Analytics
Tasks for Week-10: Gradient Descent algorithm

23) Then call the function Mgd by giving the parameters.
24) Apply linear model to selected variables.
25) Then analyse the summary.

Inference:

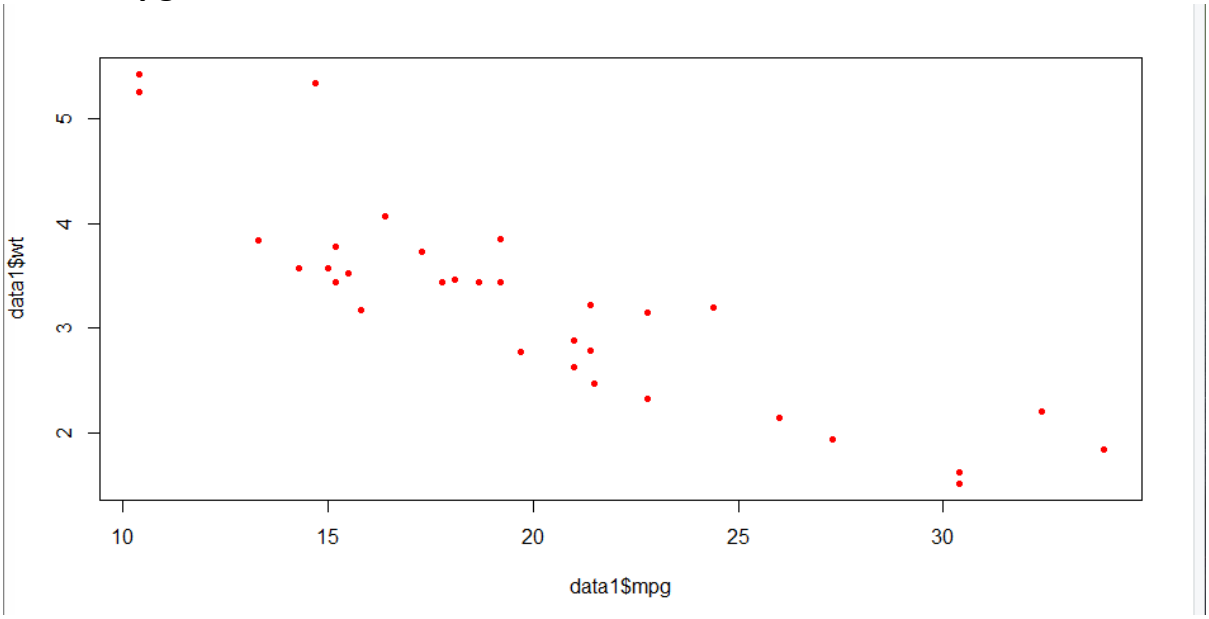
Dataset : mtcars X= hp and wt Y= mpg

Mgd(data1$wt,data1$hp,data1$mpg,-0.2,-0.2,32,0.000002,0,1000000)

According to the observation we saw that intercept and slope of linear regression and momentum gradient descent algorithm doesn't have much difference but momentum gradient descent model performs a bit better than gradient descent model.

Result:

Wt vs mpg:



Linear Model summary:

```
Coefficients:
(Intercept)      data1$hp      data1$wt
   37.22727      -0.03177      -3.87783
```

NAME: Penugonda Abhignanendra **REGNO**:19BCE1677
FACULTY: Dr. Lakshmi Pathi Jakkamputi **SLOT**: L21, L22

Essential of Data Analytics
Tasks for Week-10: Gradient Descent algorithm

Code:

```
rm(list=ls())
Mgd<-function(x1,x2,y,m1,m2,c,alpha,gamma,iter){
 iterations=0
 Lf<-0
 u_m1<-0
 u_m2<-0
 u_c<-0
 while(iterations<=iter){
  y_pred<-m1*x1+m2*x2+c
  Lf_new<-0.5*sum((y_pred-y)^2)
  nu_m1<-gamma*u_m1+alpha*sum((y_pred-y)*x1)
  nu_m2<-gamma*u_m2+alpha*sum((y_pred-y)*x2)
  nu_c<-gamma*u_c+alpha*sum(y_pred-y)
  m1<-m1-nu_m1
  m2<-m2-nu_m2
  c<-c-nu_c
  u_m1<-nu_m1
  u_m2<-nu_m2
  u_c<-nu_c
  Lf<-Lf_new
  iterations=iterations+1
```

Essential of Data Analytics
Tasks for Week-10: Gradient Descent algorithm

```
 }
 return(paste("optimal intercept:",c,"optimatl slope:",m1,m2,"Loss
funciton:",Lf,"iterations:",iterations))
}
data1<-mtcars
plot(data1$mpg,data1$wt,col="red",pch=20)
Mgd(data1$wt,data1$hp,data1$mpg,-0.2,-0.2,32,0.000002,0,1000000)
lr<-lm(data1$mpg~data1$hp+data1$wt)
lr
```