Diabetes dataset using sklearn

```
%matplotlib inline
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model, metrics


diabetes = datasets.load_diabetes()
X = diabetes.data
y = diabetes.target

samp, natt = X.shape

for i in range(natt):
  attr_mean = np.mean(X[:,i])
  attr_std = np.std(X[:,i])
  X[:,i] = (X[:,i] - attr_mean)/attr_std


print(np.mean(X, axis = 0))
print(np.std(X, axis = 0))
```

```
[-9.54490383e-18 -4.21985222e-17 -5.52599696e-17 -4.82268825e-17
  5.52599696e-18 -1.35638107e-17 -2.81323481e-17 -1.48448373e-16
 -1.84932512e-17 -2.67508489e-17]
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
samp, natt = X.shape


samp, natt
```

```
(442, 10)
```

```
regressor = linear_model.LinearRegression(fit_intercept=False, normalize=False)
regressor.fit(X, y)
```

```
LinearRegression(copy_X=True, fit_intercept=False, n_jobs=None, normalize=False)
```

```
regressor.intercept_
```

```
0.0
```

```
regressor.coef_
```

```
array([ -0.47623169, -11.40703082,  24.72625713,  15.42967916,
        -37.68035801,  22.67648701,   4.80620008,   8.422084  ,
         35.73471316,   3.21661161])
```

```
y_pred = regressor.predict(X)
RSS = np.mean((y_pred-y)**2)/(np.std(y)**2)
```

```
Rsq = 1 - RSS
print(Rsq)
mse = np.mean((y_pred-y)**2)
print(mse)
```

```
-3.385293788056333
26004.28740231017
```

Diabetes dataset without using sklearn

```
def MSE(X,y,w):
  y_pred = X.dot(w)
  return (1*np.linalg.norm(y_pred-y)**2)/442

def MSE_gradient(X,y,w):
  return (X.T.dot(X.dot(w)-y))/442


def gradient_descent(init, steps, grad):
  xs = [init]
  for step in steps:
    xs.append(xs[-1] - step * grad(X,y,xs[-1]))
  return xs


n,d = samp,natt
w0 = np.random.normal(0,1,d)
ws = gradient_descent(w0,[0.1]*6000,MSE_gradient)


all_mse = []
for w in ws:
  mse = MSE(X,y,w)
  all_mse.append(mse)

for i in range(len(all_mse)):
  if i%500==0:
    print(f'MSE at interation {i} = {all_mse[i]}')
```

```
MSE at interation 0 = 29139.86111223589
MSE at interation 500 = 26013.848159460198
MSE at interation 1000 = 26008.347652765853
MSE at interation 1500 = 26006.011708990412
MSE at interation 2000 = 26005.01968066765
MSE at interation 2500 = 26004.598386196605
MSE at interation 3000 = 26004.419470910365
MSE at interation 3500 = 26004.34348918772
MSE at interation 4000 = 26004.3112212787
MSE at interation 4500 = 26004.297517746927
MSE at interation 5000 = 26004.291698132663
MSE at interation 5500 = 26004.289226659563
MSE at interation 6000 = 26004.288177074657
```

```
y_prediction = X.dot(ws[-1])
RSS_manual = np.mean((y_prediction-y)**2)/(np.std(y)**2)
Rsq_manual = 1-RSS;Rsq
```

```
-3.385293788056333
```