In [1]:
```
from sklearn.datasets import load_diabetes
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
```

In [2]:
```
X,y = load_diabetes(return_X_y = True)
print(X.shape)
print(y.shape)
```

```
(442, 10)
(442,)
```

In [3]:
```
X_train , X_test , y_train , y_test = train_test_split(X,y,test_size = 0.2 , random_state = 42
```

In [4]:
```
reg = LinearRegression()
reg.fit(X_train , y_train)
```

Out[4]:
```
▾ LinearRegression

LinearRegression()
```

In [5]:
```
y_pred = reg.predict(X_test)
r2_score(y_test , y_pred)
```

Out[5]:
```
0.4526027629719196
```

In [6]:
```
# Now we create our own class

class SGDRegressor :

    def __init__(self , learning_rate = 0.01 , epochs = 100):
        self.coef_ = None
        self.intercept_ = None
        self.lr = learning_rate
        self.epochs = epochs


    def fit(self , X_train , y_train):
        #init our coefs
        self.intercept_ = 0
        self.coef_ = np.ones(X_train.shape[1])

        for i in range(self.epochs):
            for j in range(X_train.shape[0]):
                idx = np.random.randint(0 , X_train.shape[0])

                y_hat = np.dot(X_train[idx] , self.coef_) + self.intercept_

                intercept_der = -2*(y_train[idx] - y_hat)
                self.intercept_ = self.intercept_ - (self.lr*intercept_der)

                coef_der = -2*np.dot((y_train[idx] - y_hat) , X_train[idx])
                self.coef_ = self.coef_ - (self.lr*coef_der)


        print(self.intercept_ , self.coef_)

    def predict(self , X_test):
        return np.dot(X_test , self.coef_) + self.intercept_
```

In [7]:
```
np.random.randint(0,X_train.shape[0])
```

Out[7]:
```
177
```

In [8]:
```
X_train[48]
```

Out[8]:
```
array([ 0.00175052,  0.05068012, -0.00620595, -0.01944183, -0.00982468,
        0.00494909, -0.03971921,  0.03430886,  0.01482098,  0.09833287])
```

In [9]:
```
coef_ = np.ones(X_train.shape[1])
coef_
```

Out[9]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

In [10]:
```
intercept_ = 0
```

In [11]:
```
np.dot(X_train[48] , coef_) + intercept_
```

Out[11]:
```
0.12965077457069332
```

In [12]:
```
sgd = SGDRegressor(learning_rate = 0.01 , epochs = 50)
```

In [13]:
```
sgd.fit(X_train , y_train)
```

```
157.92737879791926 [ 6.24247772e+01 -9.18334027e+01  3.65178712e+02  2.58634443e+02
 -3.29114496e-01 -4.00983089e+01 -1.82676307e+02  1.43383523e+02
  2.93686706e+02  1.37537763e+02]
```

In [14]:
```
y_pred = sgd.predict(X_test)
```

In [15]:
```
y_pred
```

Out[15]:
```
array([163.61316452, 172.2013877 , 162.30547526, 277.64901053,
       147.28321719, 124.42880557, 240.60325445, 205.44172489,
       109.24291968, 135.10625486, 113.03294669, 148.67792859,
        75.88754616, 214.86113026, 124.64862398, 140.85850953,
       224.90785742, 244.41353246, 180.28616804, 214.08642639,
       186.89349572, 110.59879216,  93.02373163, 194.16144794,
       150.69737644, 172.07458759, 187.29033173, 180.52107086,
        68.00618878, 141.19087108, 179.18461855, 108.10239741,
       147.07834243, 188.76331545, 181.45165337, 193.32444356,
       143.80172408, 146.83383879, 174.17993699,  80.16812983,
       102.02788155, 130.87724278, 161.18236972, 173.54019949,
       176.19586264,  85.0451798 ,  94.16823083, 111.07556158,
        78.79337164, 151.19877319, 142.25632999,  80.23030348,
       136.65979689, 121.14986272, 187.33567096, 147.65781055,
       118.22228964, 196.22420682, 121.09957164,  79.96805566,
       190.03853693, 183.35055495, 142.77630194, 132.9180868 ,
       137.61102388, 195.12391956, 183.09076302, 168.45905775,
       109.88566492, 150.36419721, 179.34701129, 204.42315511,
       239.98262184, 148.07699266,  98.02465075, 170.81260253,
       206.23643321, 191.70657153, 172.2180885 , 194.51189548,
       122.88331677, 126.67814222,  73.99360563,  85.47672511,
       124.35778761, 100.072184  ,  82.1000232 ,  78.17060909,
       151.64034806])
```

In [16]:
```
r2_score(y_test , y_pred)
```

Out[16]:
```
0.44428516853002475
```

By the help of sklearn SGDRegressor

In [17]:
```
from sklearn.linear_model import SGDRegressor
```

In [18]:
```
reg = SGDRegressor(max_iter=60 , learning_rate = 'constant' , eta0 = 0.01)
```

In [19]:
```
reg.fit(X_train , y_train)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_stochastic_gradient.py:1561:
ConvergenceWarning: Maximum number of iteration reached before convergence. Consider increasi
ng max_iter to improve the fit.
  warnings.warn(
```

Out[19]:
```
SGDRegressor
SGDRegressor(learning_rate='constant', max_iter=60)
```

In [20]:
```
y_pred = reg.predict(X_test)
```

In [21]:
```
r2_score(y_test , y_pred)
```

Out[21]:
```
0.41974446871951154
```

In [ ]:
```

```