```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

Creating a database

```
X=np.random.randint(1,100,100)
X=X.reshape(-1,1)
X
```

```
array([[17],
       [93],
       [26],
       [49],
       [31],
       [57],
       [ 7],
       [ 6],
       [59],
       [53],
       [12],
       [76],
       [19],
       [ 2],
       [55],
       [22],
       [89],
       [34],
       [33],
       [86],
       [83],
       [15],
       [99],
       [64],
       [83],
       [63],
       [77],
       [55],
       [43],
       [76],
       [74],
       [61],
       [37],
       [85],
       [34],
       [54],
       [37],
       [12],
       [47],
       [24],
       [34],
       [66],
       [63],
       [52],
       [85],
       [39],
       [54],
       [64],
       [26],
       [71],
       [30],
       [52],
       [74],
       [83],
       [24],
       [11],
       [64],
       [80],
```

Create Erroes and to y

```
err=np.random.rand(100)
err=err.reshape(-1,1)
err
```

```
array([[0.84768892],
       [0.37177   ],
       [0.70084722],
       [0.21617727],
       [0.83117651],
       [0.40176657],
       [0.44751653],
       [0.65471835],
```

```
       [0.79516361],
       [0.65730952],
       [0.9126541 ],
       [0.53177142],
       [0.3761964 ],
       [0.23284979],
       [0.42201689],
       [0.07190817],
       [0.31198642],
       [0.92585878],
       [0.28527673],
       [0.40067407],
       [0.78625068],
       [0.60356778],
       [0.04050252],
       [0.07610924],
       [0.85325491],
       [0.48469648],
       [0.53084516],
       [0.94359367],
       [0.25834968],
       [0.97825848],
       [0.88316653],
       [0.65087243],
       [0.32322603],
       [0.01262152],
       [0.31159259],
       [0.67705486],
       [0.12596901],
       [0.10696803],
       [0.65852768],
       [0.68598661],
       [0.62685413],
       [0.37386226],
       [0.50642073],
       [0.41561336],
       [0.82440844],
       [0.70585094],
       [0.03802593],
       [0.91911496],
       [0.91939848],
       [0.3006295 ],
       [0.91161034],
       [0.59736408],
       [0.5002355 ],
       [0.41677203],
       [0.31770282],
       [0.57761615],
       [0.76352345],
       [0.6882036 ],
```

```
y=3*X+1+err*50
```

```
y=y.reshape(-1,1)
y
```

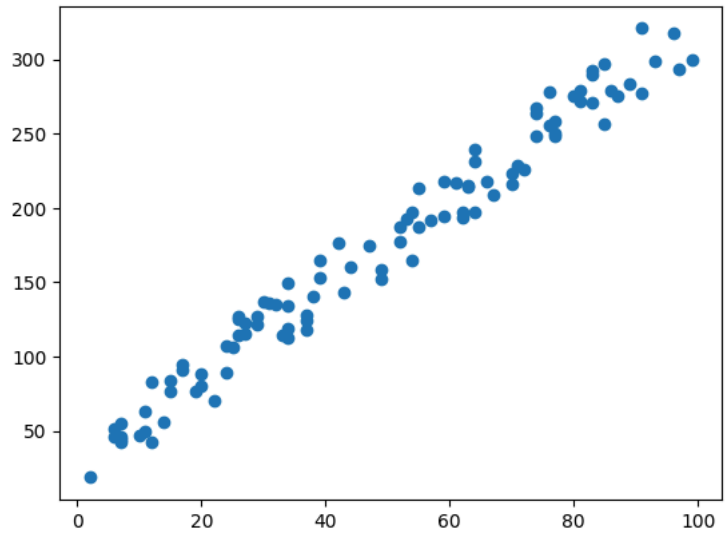
```
array([[ 94.38444592],
       [298.58849991],
       [114.04236117],
       [158.80886374],
       [135.55882553],
       [192.08832849],
       [ 44.37582637],
       [ 51.73591734],
       [217.75818072],
       [192.86547612],
       [ 82.63270523],
       [255.58857094],
       [ 76.80981997],
       [ 18.6424893 ],
       [187.10084427],
       [ 70.59540871],
       [283.59932086],
       [149.29293885],
       [114.26383662],
       [279.03370373],
       [289.31253404],
       [ 76.17838903],
       [300.02512615],
       [196.80546186],
       [292.6627457 ],
       [214.23482399],
       [258.5422578 ],
       [213.1796836 ],
       [142.91748408],
       [277.91292387],
       [267.15832641],
       [216.54362143],
       [128.16130167],
```

```
[256.63107578],
[118.57962951],
[196.85274313],
[118.29845044],
[ 42.34840132],
[174.92638415],
[107.29933043],
[134.34270649],
[217.6931131 ],
[215.32103671],
[177.78066785],
[297.22042206],
[153.29254686],
[164.90129643],
[238.95574821],
[124.96992379],
[229.03147503],
[136.58051719],
[186.86820402],
[248.01177494],
[270.8386013 ],
[ 88.8851409 ],
[ 62.8808077 ],
[231.17617262],
[275.41017995],
```

Visualizing the data

```
plt.scatter(X,y)
```

```
<matplotlib.collections.PathCollection at 0x7a735b7d4820>
```



Split the dataset

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=46)
```

```
X_train.shape,X_test.shape
```

```
((80, 1), (20, 1))
```

```
y_train.shape,y_test.shape
```

```
((80, 1), (20, 1))
```

Scale the features

```
sc=StandardScaler()
sc.fit(X_train)
```

```
▾ StandardScaler
StandardScaler()
```

```
X_train_scaled=sc.transform(X_train)
X_train_scaled[:5]
```

```
array([[ 0.95726103],
       [-1.07886288],
       [ 1.03267303],
       [ 0.01461107],
       [-1.41821686]])
```

```
X_test_scaled=sc.transform(X_test)
```

```
X_test_scaled[:5]
```

```
array([[-0.81492089],
       [ 1.37202701],
       [-1.38051086],
       [-1.38051086],
       [ 1.37202701]])
```

Add column X0

```
X_train_scaled=np.insert(X_train_scaled,0,1,axis=1)
X_test_scaled=np.insert(X_test_scaled,0,1,axis=1)
```

```
X_train_scaled.shape
```

```
(80, 2)
```

```
X_test_scaled.shape
```

```
(20, 2)
```

```
X_train_scaled.min(),X_train_scaled.max()
```

```
(-1.606746851094632, 1.8999109876099918)
```

Iniatilize the parameters and Hyperpara_meters

```
alpha=0.05
n=len(X_train)
theta=np.array([0.5,0.5])
```

```
theta
```

```
array([0.5, 0.5])
```

Algorithm

```
# For a single column
# y_cap=np.dot(X_train_scaled[0],theta)
# theta-alpha*(y_cap-y[0]*X_train_scaled[0])
```

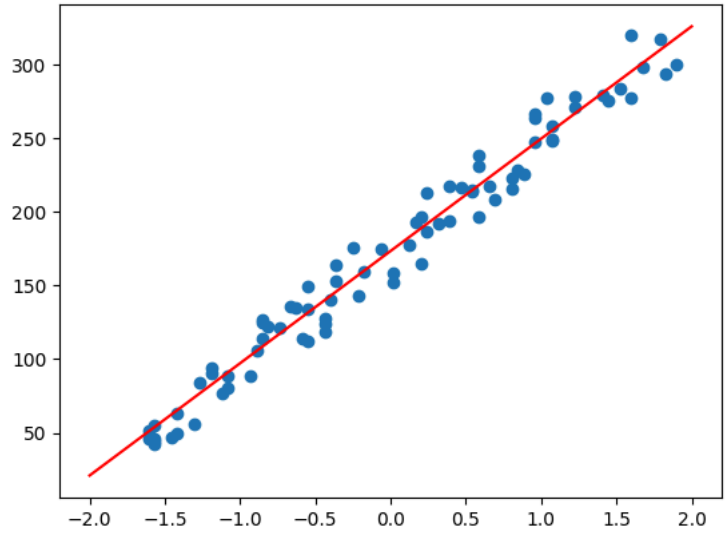
```
for epoch in range(5000):
    for i in range(n):
        y_cap=np.dot(X_train_scaled[i],theta)
        gradient=(y_cap-y_train[i])*X_train_scaled[i]
        theta=theta-alpha*gradient
```

```
theta
```

```
array([173.62097197,  76.33190028])
```

Visualizing the new graph with the line plotted using the update theta

```
plt.scatter(X_train_scaled[:,1],y_train)
X_plot=np.linspace(-2,2,100)
y_plot=(theta[1]*X_plot)+theta[0]
plt.plot(X_plot,y_plot,'r-')
plt.show()
```



R^2 evaluation

```
y_pred=np.dot(X_test_scaled,theta)
y_pred
```

```
array([111.41651205, 278.35040102,  68.24395456,  68.24395456,
       278.35040102, 131.56370555, 272.59406002, 212.15247953,
       212.15247953, 117.17285305, 120.05102355,  39.46224956,
       272.59406002, 263.95954852, 102.78200055, 183.37077454,
       252.44686653,  76.87846605, 272.59406002,  97.02565955])
```

```
y_test=y_test.flatten()
y_test
```

```
array([115.25219125, 256.63107578,  42.34840132,  82.63270523,
       297.22042206, 118.57962951, 292.6627457 , 196.95028866,
       193.3757221 , 127.18422975, 136.58051719,  18.6424893 ,
       270.8386013 , 275.41017995, 107.29933043, 186.86820402,
       255.58857094,  76.17838903, 289.31253404,  70.59540871])
```

```
sstot=(y_test-np.mean(y_test))**2
ssres=(y_test-y_pred)**2
r2=1-(np.sum(ssres)/np.sum(sstot))
r2
```

```
0.9699866700154403
```

```
# a=y_pred-y_test
# (np.sqrt(np.abs(a)))
```

Calculating r^2 using Scikit Model

```
from sklearn.metrics import r2_score
```

```
r2_score(y_pred,y_test)
```

```
0.9668498607119147
```

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

```
model.fit(X_train_scaled,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred=model.predict(X_test_scaled)
# y_pred=y_pred.reshape(-1,1)
# y_pred
```

```
y_test=y_test.reshape(-1,1)
```

```
r_squared = model.score(X_test_scaled, y_test)
r_squared
```

```
0.9706741679565218
```

```
model.coef_
```

```
array([[ 0.        , 76.80608479]])
```

Start coding or generate with AI.