```
import matplotlib.pyplot as plt
import numpy as np
```

```
from sklearn import datasets,linear_model
from sklearn.metrics import mean_squared_error
```

```
diabetes = datasets.load_diabetes()
```

```
diabetes.keys()
```

```
dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename', 'target_filename', 'data_module'])
```

```
print(diabetes.keys())
```

```
dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename', 'target_filename', 'data_module'])
```

```
print(diabetes.data)
```

```
[[ 0.03807591  0.05068012  0.06169621 ... -0.00259226  0.01990749
  -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 ... -0.03949338 -0.06833155
  -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 ... -0.00259226  0.00286131
  -0.02593034]
 ...
 [ 0.04170844  0.05068012 -0.01590626 ... -0.01107952 -0.04688253
   0.01549073]
 [-0.04547248 -0.04464164  0.03906215 ...  0.02655962  0.04452873
  -0.02593034]
 [-0.04547248 -0.04464164 -0.0730303  ... -0.03949338 -0.00422151
   0.00306441]]
```

```
print(diabetes.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one year after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the square root of `n_sample

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (with discussic
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

```
diabetes = datasets.load_diabetes()
```

Double-click (or enter) to edit

```
# diabetes_X =  diabetes.data
diabetes_X =  diabetes.data[:,np.newaxis,2]
```

```
print(diabetes_X)
```

```
 [-0.02991782]
 [-0.0191397 ]
 [-0.04069594]
 [ 0.01535029]
 [-0.02452876]
 [ 0.00133873]
 [ 0.06924089]
 [-0.06979687]
 [-0.02991782]
 [-0.046085  ]
 [ 0.01858372]
 [ 0.00133873]
 [-0.03099563]
 [-0.00405033]
 [ 0.01535029]
 [ 0.02289497]
 [ 0.04552903]
 [-0.04500719]
 [-0.03315126]
 [ 0.097264  ]
 [ 0.05415152]
 [ 0.12313149]
 [-0.08057499]
 [ 0.09295276]
 [-0.05039625]
 [-0.01159501]
 [-0.0277622 ]
 [ 0.05846277]
 [ 0.08540807]
 [-0.00081689]
 [ 0.00672779]
 [ 0.00888341]
 [ 0.08001901]
 [ 0.07139652]
 [-0.02452876]
 [-0.0547075 ]
 [-0.03638469]
 [ 0.0164281 ]
 [ 0.07786339]
 [-0.03961813]
 [ 0.01103904]
 [-0.04069594]
 [-0.03422907]
 [ 0.00564998]
 [ 0.08864151]
 [-0.03315126]
 [-0.05686312]
 [-0.03099563]
 [ 0.05522933]
 [-0.06009656]
 [ 0.00133873]
 [-0.02345095]
 [-0.07410811]
 [ 0.01966154]
 [-0.01590626]
 [-0.01590626]
 [ 0.03906215]
 [-0.0730303 ]]
```

```
diabetes_X_train = diabetes_X[:-30]
diabetes_X_test = diabetes_X[-30:]
```

```
diabetes_y_train = diabetes.target[:-30]
diabetes_y_test = diabetes_X[-30:]
diabetes_X_test = diabetes_X[-30:]

diabetes_y_train =diabetes.target[:-30]
diabetes_y_test = diabetes.target[-30:]
```

```
model = linear_model.LinearRegression()
```

```
model.fit(diabetes_X_train,diabetes_y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
diabetes_y_predicted = model.predict(diabetes_X_test)
print(diabetes_y_predicted)
```

```
[233.80294072 152.62808714 159.73088683 161.76025817 228.72951237
 220.61202701 130.3050024  101.89380365 119.14346004 168.86305786
 226.70014103 116.09940303 163.78962951 115.08471736 121.17283138
 158.71620116 236.84699773 122.18751705  99.86443231 124.21688839
 205.39174197  96.8203753  154.65745848 131.31968807  83.62946159
 171.90711487 138.42248776 138.42248776 190.17145692  84.64414726]
```

```
print("Mean squared error is: ",mean_squared_error(diabetes_y_test,diabetes_y_predicted))
```

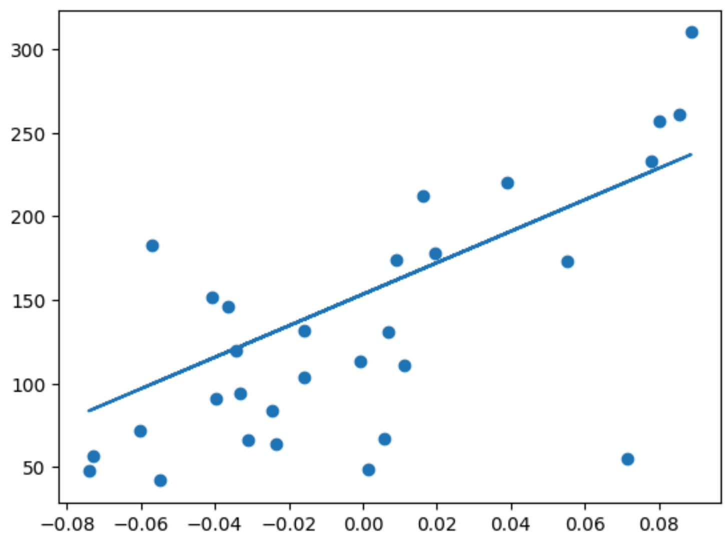
```
Mean squared error is:  3035.060115291269
```

```
print("Weights",model.coef_)
print("Intercept",model.intercept_)
```

```
Weights [941.43097333]
Intercept 153.39713623331644
```

Double-click (or enter) to edit

```
plt.scatter(diabetes_X_test,diabetes_y_test)
plt.plot(diabetes_X_test,diabetes_y_predicted)
plt.show()
```




```
Start coding or generate with AI.
```