Problem Statement (Small)

Linear Regression Sample: The sample below uses only the first feature of the diabetes dataset, in order to illustrate the data points within the two-dimensional plot. Please follow the steps and capture the results on each step.

https://docs.google.com/document/d/1Iwfv0VFyaQTI5fl7OhtSXsrNe4LsvF2J83e8E9cljWU/edit#h

In [ ]:

```
import sys
!{sys.executable} -m pip install numpy
!{sys.executable} -m pip install sklearn
```

```
Requirement already satisfied: numpy in c:\users\synt_\appdata\local\packages\pyth
onsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310
\site-packages (1.22.4)
Requirement already satisfied: sklearn in c:\users\synt_\appdata\local\packages\py
thonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python3
10\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\synt_\appdata\local\packag
es\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\py
thon310\site-packages (from sklearn) (1.1.1)
Requirement already satisfied: joblib>=1.0.0 in c:\users\synt_\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\p
ython310\site-packages (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\synt_\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\p
ython310\site-packages (from scikit-learn->sklearn) (1.22.4)
Requirement already satisfied: scipy>=1.3.2 in c:\users\synt_\appdata\local\packag
es\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\py
thon310\site-packages (from scikit-learn->sklearn) (1.8.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\synt_\appdata\loca
l\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-pac
kages\python310\site-packages (from scikit-learn->sklearn) (3.1.0)
```

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

In [ ]:

```
dataX, dataY = datasets.load_diabetes(return_X_y=True)
```

In [ ]:

```
dataX = dataX[:, np.newaxis, 2]
```

In [ ]:

```
trainX = dataX[:-30]
testX = dataX[-30:]
```

In [ ]:

```
trainY = dataY[:-30]
testY = dataY[-30:]
```

In [ ]:

```
lr = linear_model.LinearRegression()
```

In [ ]:

```
lr.fit(trainX, trainY)
```

Out[ ]:

```
▾ LinearRegression
LinearRegression()
```

In [ ]:

```
predY = lr.predict(testX)
```

In [ ]:

```
print("Coefficients:", lr.coef_)
```

```
Coefficients: [941.43097333]
```

In [ ]:

```
print("Mean squared error: %.2f" % mean_squared_error(testY, predY))
```

```
Mean squared error: 3035.06
```

In [ ]:

```
print("Coefficient of determination: %.2f" % r2_score(testY, predY))
```

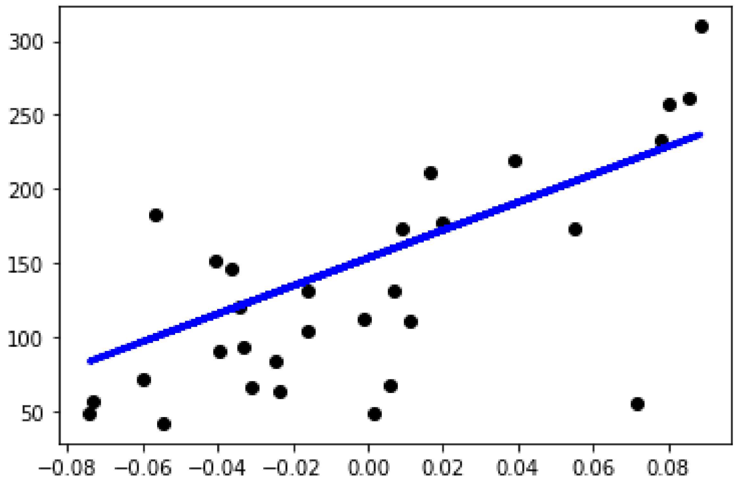
```
Coefficient of determination: 0.41
```

In [ ]:

```
plt.scatter(testX, testY, color="black")
plt.plot(testX, predY, color="blue", linewidth=3)
```

Out[ ]:

```
[<matplotlib.lines.Line2D at 0x22e0fee50f0>]
```


300
250
200
150
100
50
−0.08
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06
0.08