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.

- **Importing Libraries**
  To import necessary libraries for the first task, so execute the following import statements:

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

- **Load the diabetes dataset**

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

- **Use only one feature**

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

- **Split the data into training/testing sets**

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

- **Split the targets into training/testing sets into 70:30 ratio**

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

- **Create linear regression object**

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

- **Train the model using the training sets**

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

- **Make predictions using the testing set**

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

- **Print the coefficients**

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

- **Print the mean squared error**

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

- **Print the coefficient of determination**

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

- **Plot the outputs**

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