Assignment 1 - Linear Regression

Due by 11:59pm, Sep. 25, 2022

Theory Questions (Each question is worth 5 points)

1. Explain the differences between supervised, semi-surpervised, weakly-supervised and unsupervised learning.

Answer:

2. Explain the difference between regression and classification.

Answer:

3. Explain how to perform N-fold cross-validation.

Answer:

4. Explain the difference between training and testing error.

Answer:

5. Explain where the errors come from.

Answer:

6. Explain the standard steps to perform machine learning.

Answer:

7. Let $(x^{(i)}, y^{(i)})_{i=1}^{m}$ be a set of training examples where $y_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^n$, write a linear regression model using model parameter θ_j and features $x_1, \ldots, x_n$.

Answer:

8. Write the objective function that can be used to determine the regression model parameters.

How is this objective function will be used to find model parameters?

Answer:

9. Let θ =[1,-2,3] be the model parameters obtained for linear regression. Let x =[4,6] be a feature vector. Find the value y that will be obtained by the linear regression for the feature vector x.

Answer:

10. Suppose an artifial neuron has weight W = [0.3, 0.5], bias b = 0.1 and a sigmoid activation function. If the input data = [2,3], what is the output value of the neuron?

Answer:

Programming Questions (50 points, each part is worth 10 points)

In this assignment, you are going to implement your own Simple Linear Regression function.

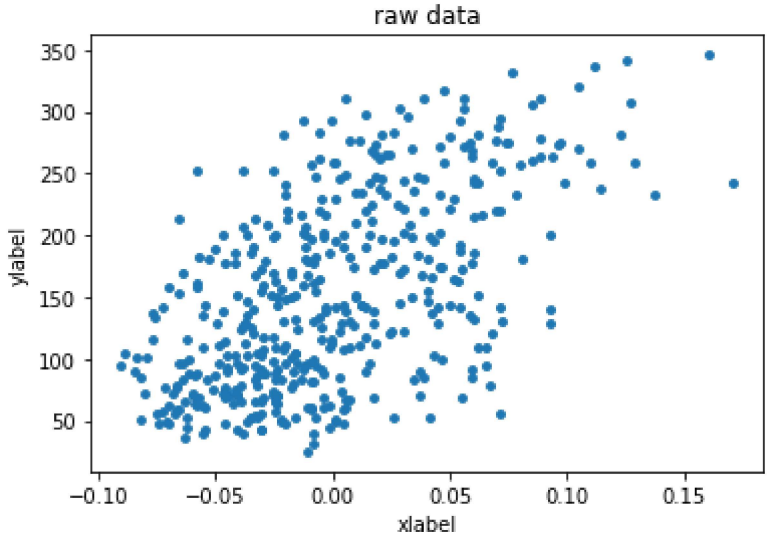
Please notice: **No library versions of linear regression are allowed**.

In [71]:
```
# Do not edit the codes in this cell
# load required library
from sklearn.datasets import load_diabetes
import matplotlib.pyplot as plt
import numpy as np
# load dataset
X, y = load_diabetes(return_X_y=True)
X = X[:, 2]
```

```
**Part I**
_____

Using the given dataset above, draw a scatter plot.
1. Plot title and xlabel, ylabel are required;
2. Show the plot in the Output.
```

Example plot



In [72]:
```
# draw the raw data plot
# TODO
```

Part II

Initialize theta and construct cost function for the upcoming gradient decent.
1. Initialize theta, print it out;
2. Construct Mean Square Error loss function;
3. Calcualte the cost using initial theta, X and y, and print the result out.

In [ ]:
```
# initialize theta
# TODO
```

In [ ]:
```
# define cost function
# TODO
```

In [ ]:
```
# calculate cost by calling the function
# TODO
```

Part III

Gradient descent to find the optimal fit.

1. Initialize learning rate and epoch, try to explain your reasons for the values chosen;
2. Construct gradient descent function, which updates the theta and meanwhile records all the history cost;
3. Call the function for the optimal fit. Print out the final theta and final cost.

Question: How did you choose your lr and epoch number?

Answer:

In [ ]:
```
# gradient descent to find the optimal fit
# TODO
```

Part IV

Plot the cost for each iteration.

1. Plot title and xlabel, ylabel are required;
2. Show the plot in the Output.

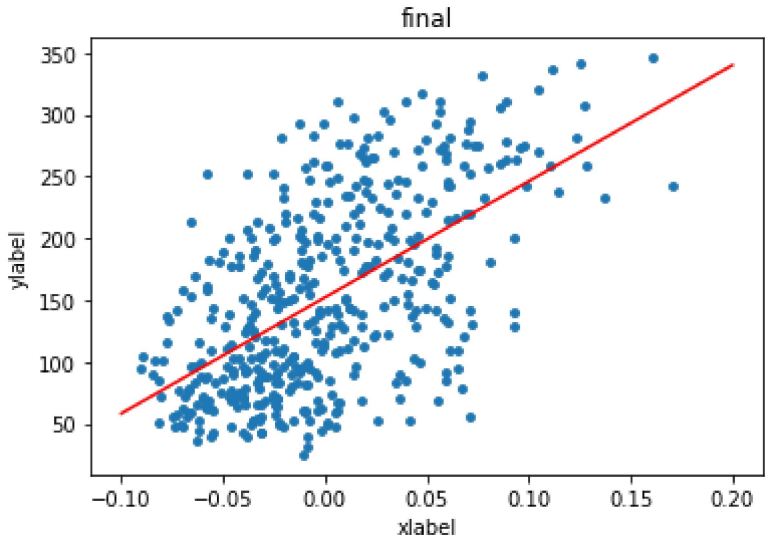
In [ ]:
```
# draw the cost change
# TODO
```

Part V

Based on the raw data plot you already drawn, plot the final linear regression model using the final version of theta.

1. Plot title and xlabel, ylabel are required;
2. Show the plot in the Output.

Example plot



In [ ]:
```
# draw the final linear model
# TODO
```