1．Learning basics of regression in Python

b) Describe and summarize the data in terms of number of data points, dimensions, target, etc

There are 14 attributes in each case of the dataset. They are:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's

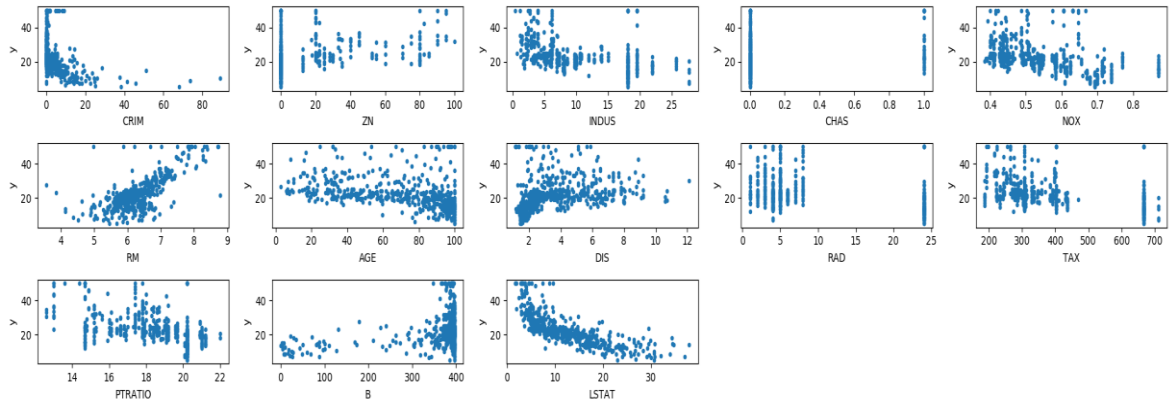
The number of the dataset is: 506

The dimension of the dataset is: (506, 14) 13 of them are features and 1 of them is target.

The number of the target is: 506

The dimension of the target is: (506,)

c) Visualization: present a single grid containing plots for each feature against the target.



f) Tabulate each feature along with its associated weight and present them in a table. Explain what the sign of the weight means in the third column ('INDUS') of this table. Does the sign match what you expected? Why?

Features	Weight
I(BIAS)	13.4331662778
CRIM	-0.0802087715082

ZN	0.0520014015797
INDUS	-0.0229319790032
CHAS	2.43346318301
NOX	-5.28198942762
RM	5.54559496981
AGE	-0.0235668167441
DIS	-1.31560331875
RAD	0.214724839633
TAX	-0.0120959579783
PTRATIO	-0.599070583362
B	0.0102157394502
LSTAT	-0.508573801714

The sign of the weight is negative. It matches what I expect.

As the 'INDUS' denotes proportion of non-retail business acres per town, the higher this index is, the lower the proportion of retail business acres per town, and the price of the owner-occupied homes will be lower. So 'INDUS' ought to have negative influence on MEDV. If the sigh is positive, the house price increase with 'INDUS' criteria increasing. Otherwise, it will decrease. So sign of weight should be negative.

g) Test the fitted model on your test set and calculate the Mean Square Error of the result

Mean square error is around 20. In my testing case, MSE= 20.3562457063

h) Suggest and calculate two more error measurement metrics; justify your choice.

I choose Mean absolute scaled error and Mean absolute percentage error. The answers show this regression algorithm is good but also need to be improved.

Mean absolute scaled error: 2.9533264408

Mean absolute percentage error: 0.166990091302

Mean absolute scaled error (MASE) is a measure of the accuracy of forecasts. It is independent of the scale of the data, and it penalizes positive and negative forecast errors equally. So it's appropriate to measure this issue.

Mean absolute percentage error (MAPE) is a measure of prediction accuracy of a forecasting method in statistics. It fits this issue.

i) According to the result of trained indexes, I think the most significant features are 'NOX' 'RM', because the absolute values of this index are higher among these 13 features.

2. Locally weighted regression

1) Prove that $w^* = \arg\min \frac{1}{2}\sum_{i=1}^{N} a^{(i)}(y^{(i)} - w^T x^{(i)})^2 + \frac{\lambda}{2}\|w\|^2$ is given by the formula:

$$w^* = (X^T AX + \lambda I)^{-1} X^T Ay$$

Solution:

$$\begin{aligned} Loss &= \frac{1}{2}\sum_{i=1}^{N} a^{(i)}(y^{(i)} - w^T x^{(i)})^2 + \frac{\lambda}{2}\|w\|^2 \\ &= \frac{1}{2}(Y - Xw)^T A(Y - Xw) + \frac{\lambda}{2} w^T \bullet w \end{aligned}$$

$$\begin{aligned} 2Loss &= (Y - Xw)^T A(Y - Xw) + \lambda w^T \bullet w \\ &= (Y^T A - w^T X^T A)(Y - Xw) + \lambda w^T \bullet w \\ &= Y^T AY - Y^T AXw - w^T X^T AY + w^T X^T AXw + \lambda w^T \bullet w \end{aligned}$$

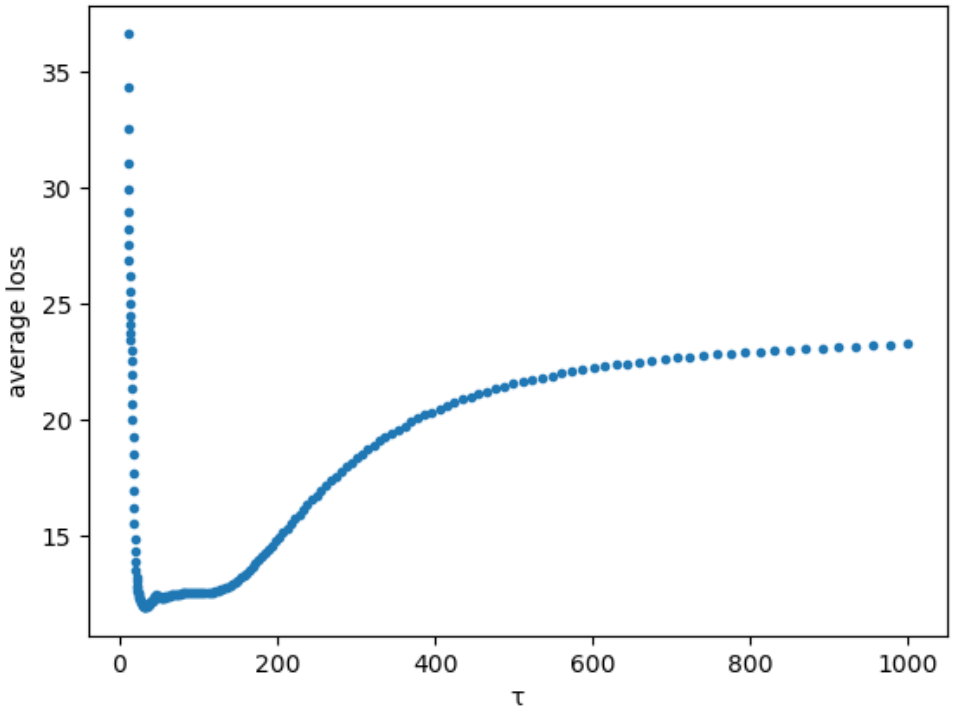
let :

$$\nabla 2Loss(w) = -2X^T AY + 2X^T AXw + 2\lambda w = 0$$

$$(X^T AX + \lambda I)w = X^T AY$$

$$w = (X^T AX + \lambda I)^{-1} X^T AY$$

3) Use k-fold cross-validation to compute the average loss for different values of τ in the range [10,1000] when performing regression on the Boston Houses dataset. Plot these loss values for each choice of τ



4) When $\Upsilon \to \infty$, the algorithm's behavior is getting worse and the error is getting larger and when $\Upsilon \to 0$, the algorithm behavior is also getting worse and the error is increasing. But when Υ goes from 0 to infinity, the algorithm behaves better at the beginning and the error is getting down. The error has a minimum value between 0 and infinity.

3. Mini-batch SGD Gradient Estimator

1) Given a set $\{a_1, a_2, ..., a_n\}$ and random mini-batches L of size m, show that:

$$E_\Gamma\left[\frac{1}{m}\sum_{i\in\Gamma} a_i\right] = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Solution:

$$E_\Gamma\left[\frac{1}{m}\sum_{i\in L} a_i\right] = \frac{1}{m}E_\Gamma\left[\sum_{i\in\Gamma} a_i\right] = \frac{1}{m}E_\Gamma\left[a_1 + a_2 + ... + a_m\right]$$

$$= \frac{1}{m}\left(E[a_1] + E[a_2] + ... + E[a_m]\right)$$

$$= \frac{1}{m}\left(\frac{1}{n}\sum_{i=1}^{n} a_i + \frac{1}{n}\sum_{i=1}^{n} a_i + ... + \frac{1}{n}\sum_{i=1}^{n} a_i\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} a_i$$

2) Show that $E_\Gamma\left[\nabla L(x, y, \theta)\right] = \nabla L(x, y, \theta)$

Solution:

$$E_{\Gamma}[\nabla L_{\Gamma}(x,y,\theta)] = E\left[\nabla\left(\frac{1}{\mathrm{m}}\sum_{i=1}^{n} l\left(x^i,y^i,\theta^i\right)\right)\right]$$
$$= E\left[\frac{1}{\mathrm{m}}\nabla\left(\sum_{i=1}^{n} l\left(x^i,y^i,\theta^i\right)\right)\right]$$
$$= E\left[\frac{1}{\mathrm{m}}\left(\sum_{i=1}^{n} \nabla l\left(x^i,y^i,\theta^i\right)\right)\right]$$

From (1), we know:

$$E_{\Gamma}\left[\frac{1}{m}\sum_{i\in\Gamma} a_i\right] = \frac{1}{n}\sum_{i=1}^{n} a_i$$

So $E_{\Gamma}\left[\nabla L(x,y,\theta)\right] = \nabla\left[\frac{1}{\mathrm{m}}\left(\sum_{i=1}^{n} l\left(x^i,y^i,\theta^i\right)\right)\right] = \nabla L(x,y,\theta)$. The prove is done!

3) Write, in a sentence, the importance of this result.
This result shows that the gradience of the loss function of the whole set is equivalent to the expected gradience of the loss function of batch set. This can help minimize the calculation cost and provide the foundation of mini-batch SGD gradient estimator.

4) a. Write down the gradient, ∇ L above.
From 3.1, 3.2 we know that:

$$\nabla L = E_{\Gamma}\left(\nabla L_{\Gamma} a(x,y,\theta)\right)$$
$$= E\left[\nabla\left(\frac{1}{m}\sum_{i=1}^{m} l(x^i,y^i,\theta)\right)\right]$$
$$= \frac{1}{m}E\left[\sum_{i=1}^{m} \nabla l(x^i,y^i,\theta)\right]$$
$$= \frac{1}{m}E\left[\sum_{i=1}^{m} \nabla (y^i - w^T x^i)^2\right]$$
$$= \frac{1}{m}E\left[\sum_{i=1}^{m} 2(w^T x^i - y^i)x^i\right]$$
$$= \frac{2}{m}\left(X^T(X\bullet w - Y)\right)$$

5) Compare the value you have computed to the true gradient, using both the squared distance metric and cosine similarity. Which is a more meaningful measure in this case and why?
Square Distance Metric: 816.809112922
Cosine similarity: 0.99999965
Cosine similarity is more important. Because Cosine similarity is a measure of similarity between two non-zero vectors that measures the cosine of the angle between

them. It is thus a judgment of orientation and not magnitude. For gradients it's more important to measure their orientation than their magnitude.

6) Plot log σj against log m.