1 Learning basics of regression in Python (3%)

Q: Describe and summarize the data in terms of number of data points, dimensions, target, etc

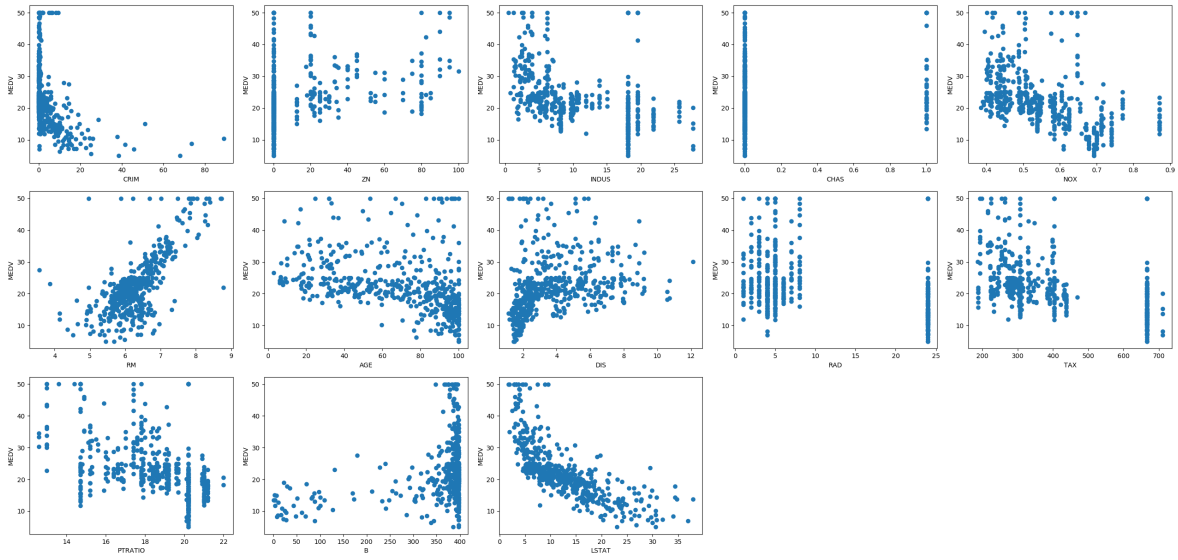
Number of data points: 506

Dimensions: 13

Target: Median Value (attribute 14)

Q: Visualization: present a single grid containing plots for each feature against the target.Choose the appropriate axis for dependent vs. independent variables.

For detail graph, please see Q1_plot.png.



Q: Divide your data into training and test sets, where the training set consists of 80% of the data points (chosen at random).

Please see Q1.py

Q: Write code to perform linear regression to predict the targets using the training data. Remember to add a bias term to your model.

Please see Q1.py

Q: 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?

Bias	4.31761776e+01
CRIM	-1.23984280e-01
ZN	5.15164427e-02
INDUS	3.70500806e-02
CHAS	3.19316149e+00
NOX	-1.98849696e+01
RM	3.21273892e+00
AGE	3.90376026e-03
DIS	-1.63666301e+00
RAD	3.74430453e-01
TAX	-1.46862482e-02
PTRATIO	-9.98734272e-01
B	9.48373885e-03
LSTAT	-5.66101022e-01

The weight of 'INDUS' is 3.70500806e-02, and the sign is positive. And the sign match what I expected, because 'INDUS' represents proportion of non-retail business acres per town. From my point of view, people usually like live in a community with low proportion of retail business. Therefore, the sign of the weight is positive.

Q: Test the fitted model on your test set and calculate the Mean Square Error of the result.

MSE: 15.75343302218935

Q: Suggest and calculate two more error measurement metrics; justify your choice.

I will suggest root mean squared error,RMSE and mean absolute error,MAE.

RMSE: 3.9690594631712623

MAE: 2.8929690793228153

Q: Feature Selection: Based on your results, what are the most significant features that best predict the price? Justify your answer.

I will choose 'RM' as my most significant feature, because 'RM' is has the biggest positive weight and it looks like very linear from the graph that I generate.

2 Locally reweighted regression (6%)

1.)

$\because w* = argmin\frac{1}{2}\sum_{i=1}^{N} a^i(y^i - W^T x^i)^2 + \frac{\lambda}{2}||w||^2$

$\therefore w*^{'} = \sum_{i=1}^{N} a^i(y^i - (x^i)^T w)((-x^i)^T) + \lambda * w$

$\therefore \sum_{i=1}^{N} a^i(y^i - (x^i)^T w)(-(x^i)^T) + \lambda * w = 0$

$\therefore \sum_{i=1}^{N}(-(x^i)^T a^i y^i + (x^i)^T a^i x^i w) + \lambda * w = 0$

$\therefore \sum_{i=1}^{N}((x^i)^T a^i x^i w) + \lambda * w = \sum_{i=1}^{N}(x^i)^T a^i y^i$

In matrix form:

$X^T AXW + \lambda W = X^T y$

$(X^T AX + \lambda I)W = X^T y$

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

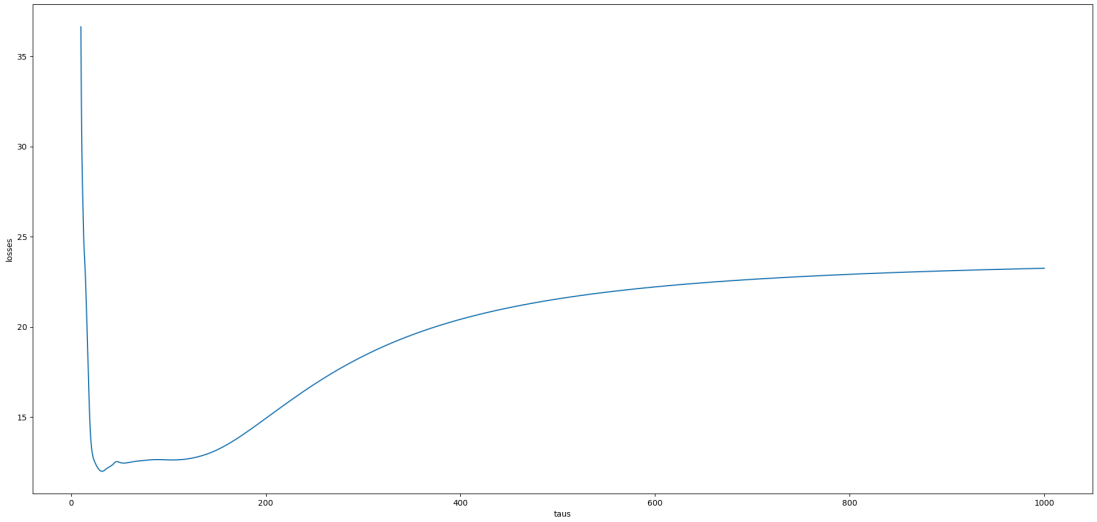
2.)

min loss = 11.988101864973682

Please see Appendix 2 for detail.

3.)

For detail graph, plase see Q2_plot.png



4.)

For $\tau \to \infty$, $-||x - x^i||^2/2\tau^2 \approx 0$, so $exp(-||x - x^i||^2/2\tau^2) \approx 1$. For the denominator part, it basically equal to $n * 1$. Therefore, for $a^i = 1/n$. That makes us get the same W, same W makes us get the same prediction, same prediction makes us get the same loss value. In our case, the prediction will maintain at around 25.

For $\tau \to 0$, the loss value will $\approx \infty$, as it shows from the graph that provided on 3.) Because while $\tau \to 0$, both of numerator and denominator of a^i will become very large. And that makes the squared distance meaningless, so the prediction become very inaccuracy. And that makes the loss value approach .

3 Mini-batch SGD Gradient Estimator (6%)

1.)

$\because \frac{1}{n}\sum_{i=1}^{n} a_i = E(a)$

$\because E_{\mathcal{I}}[\frac{1}{m}\sum_{i\in\mathcal{I}} a_i] = \frac{1}{m}[E(a_1) + E(a_2) + ... + E(a_m)]$

$\therefore E_{\mathcal{I}}[\frac{1}{m}\sum_{i\in\mathcal{I}} a_i] = \frac{1}{m}[m * E(a)] = E(a) = \frac{1}{n}\sum_{i=1}^{n} a_i$

2.)

$\because E_{\mathcal{I}}[\bigtriangledown L_{\mathcal{I}}(x,y,\theta)] = \frac{1}{m}\sum_{i\in\mathcal{I}} \bigtriangledown L_{\mathcal{I}}(x,y,\theta)$

$\therefore E_{\mathcal{I}}[\bigtriangledown L_{\mathcal{I}}(x,y,\theta)] = \frac{1}{m}[E(\bigtriangledown L_{\mathcal{I}}(x_1,y_1,\theta))+E(\bigtriangledown L(x_2,y_2,\theta))+...+E(\bigtriangledown L(x_m,y_m,\theta))]$

$\therefore E_{\mathcal{I}}[\bigtriangledown L_{\mathcal{I}}(x,y,\theta)] = \frac{1}{m}[m * E(\bigtriangledown L(x,y,\theta))] = E(\bigtriangledown L(x,y,\theta))$

$\because \bigtriangledown L(x,y,\theta) = E(\bigtriangledown L(x,y,\theta))$

$\therefore E_{\mathcal{I}}[\bigtriangledown L_{\mathcal{I}}(x,y,\theta)] = \bigtriangledown L(x,y,\theta)$

3.)

Proof that the expected value of a mini-batch in SGD is equal to the true gradient. So we can use mini-batch to predict the true gradient.

4a.)

$\because L(x,y,\theta) = \frac{1}{n}\sum l(x,y,\theta) = \frac{1}{n}\sum (y - w^T x)^2$

$\therefore \bigtriangledown L(x,y,\theta) = \frac{1}{n}\sum_{i=1}^{N} 2(y^i - (x^i)^T w)((-x^i)^T)$

$\therefore \bigtriangledown L(x,y,\theta) = \frac{1}{n}\sum_{i=1}^{N}(-2(x^i)^T y^i + 2(x^i)^T x^i w)$

$\therefore \bigtriangledown L(x,y,\theta) = \frac{1}{n}(2X^T XW - 2X^T y)$

4b.)

Please see Q3.py.

5.)

Cosine similarity: 0.9999995107436891

Squared distance: 308432.0297241211

Cosine similarity is more meaningful measure in this case. Because cosine similarity represent the similarity of gradient's direction. Which is more important in this

case.

For detail, please see Q3.py.

6.)

For detail graph, plase see Q3_plot.png