1.10 ① $(y^T) \underset{1\times n}{} \underset{n\times d}{X} \underset{d\times 1}{w}$ is a scalar.

$$y^T X = [y^T x_1,\ y^T x_2 \cdots y^T x_d]$$

$$y^T X w = y^T x_1 w_1 + y^T x_2 w_2 + \cdots y^T x_d w_d$$

$$\frac{d(y^T X w)}{dw} = \begin{bmatrix} \frac{\partial y^T X w}{\partial w_1} \\ \vdots \\ \frac{\partial y^T X w}{\partial w_d} \end{bmatrix} = \begin{bmatrix} y^T x_1 \\ y^T x_2 \\ \vdots \\ y^T x_d \end{bmatrix} = \begin{bmatrix} x_1^T y \\ x_2^T y \\ \vdots \\ x_d^T y \end{bmatrix} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_d^T \end{bmatrix} y = X^T y$$

② $w^T w = w_1^2 + w_2^2 + \cdots + w_d^2$ is a scalar

$$\frac{d(w^T w)}{dw} = \begin{bmatrix} \frac{\partial (w_1^2 + w_2^2 + \cdots w_d^2)}{\partial w_1} \\ \frac{\partial (w_1^2 + w_2^2 + \cdots w_d^2)}{\partial w_2} \\ \vdots \\ \frac{\partial (w_1^2 + w_2^2 + \cdots w_d^2)}{\partial w_d} \end{bmatrix} = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_d \end{bmatrix} = 2w$$

③ $$w^T X = w^T [x_1, x_2 \cdots x_d] = [w^T x_1, w^T x_2, \cdots, w^T x_d]$$

$$w^T X w = [w^T x_1, w^T x_2, \cdots, w^T x_d] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = w_1 w^T x_1 + w_2 w^T x_2 \cdots w_d w^T x_d$$

Since $$\frac{\partial (w_i w^T x_i)}{\partial w_i} = w^T x_i \cdot \frac{\partial (w_i)}{\partial w_i} + w_i \cdot \frac{\partial (w^T x_i)}{\partial w_i}$$

$$= w^T x_i + w_i (x_{ii})$$

$$\frac{\partial (w_i w^T x_i)}{\partial w_j} = w^T x_i \frac{\partial (w_i)}{\partial w_j} + w_i \frac{\partial (w^T x_i)}{\partial w_j}$$

$$= 0 + w_i x_{ji} \qquad i \neq j$$

Let $X^T = [\vec{x}_1, \vec{x}_2, \cdots \vec{x}_d]$

Then $\frac{d(w^T X w)}{dw} = \begin{bmatrix} w^T x_1 + w_1 x_{11} + \sum_{\substack{i \neq 1 \\ 1 \leq i \leq d}} w_i x_{1i} \\ w^T x_2 + w_2 x_{22} + \sum_{\substack{i \neq 2 \\ 1 \leq i \leq d}} w_i x_{2i} \\ \vdots \\ w^T x_d + w_d x_{dd} + \sum_{\substack{i \neq d \\ 1 \leq i \leq d}} w_i x_{di} \end{bmatrix} = \begin{bmatrix} w^T x_1 + \sum_{1 \leq i \leq d} w_i x_{1i} \\ w^T x_1 + \sum_{1 \leq i \leq d} w_i x_{2i} \\ \vdots \\ w^T x_d + \sum_{1 \leq i \leq d} w_i x_{di} \end{bmatrix}$

$$= \begin{bmatrix} w^T x_1 + w^T \vec{x}_1 \\ w^T x_2 + w^T \vec{x}_2 \\ \vdots \\ w^T x_d + w^T \vec{x}_d \end{bmatrix} = \begin{bmatrix} x_1^T w + \vec{x}_1^T w \\ x_2^T w + \vec{x}_2^T w \\ \vdots \\ x_d^T w + \vec{x}_d w \end{bmatrix} = \begin{bmatrix} x_1^T + \vec{x}_1^T \\ x_2^T + \vec{x}_2^T \\ \vdots \\ x_d^T + \vec{x}_d^T \end{bmatrix} w$$

$$= \left(\begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_d^T \end{bmatrix} + \begin{bmatrix} \vec{x}_1^T \\ \vec{x}_2^T \\ \vdots \\ \vec{x}_d^T \end{bmatrix} \right) w = \left[X^T + (X^T)^T \right] w$$

$$= (X + X^T) w$$

1.2 (1) Let $W := \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$ $\quad X := \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{bmatrix}$ $\quad y := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$

So $\sum_{i=1}^{N} (f_{w,b}(x_i) - y_i)^2 = (XW - y)^T (XW - y)$

Let $\frac{\partial}{\partial W} \left[(XW - y)^T (XW - y) + \lambda \bar{w}^T \bar{w} \right] = 0$

$$\Rightarrow 2X^T (XW - y) + 2\lambda \hat{I}_d W = 0$$

$$(X^T X + \lambda \hat{I}_d) W = X^T y$$

$$W = (X^T X + \lambda \hat{I}_d)^{-1} X^T y$$

where $X^T X + \lambda \hat{I}_d$ is invertible if $\lambda > 0$

(2) $W^* = \arg\min_W J(W)$ and $J(W) = (XW - y)^T (XW - y) + \lambda \bar{w}^T \bar{w}$

set learning rate α, maximum iteration T, initial point W_0

update $W^{(i+1)} \leftarrow W^{(i)} - \alpha \frac{\partial J(W^{(i)})}{\partial W^{(i)}}$

Repeat till convergence.

1.3 (1) $f(x)=x^2$ $\quad f''(x)=2>0$ $\quad x\in\mathbb{R}$ s.t. $\forall x_1,x_2\in\mathbb{R}, \forall\theta\in[0,1] \Rightarrow \theta x_1+(1-\theta)x_2\in\mathbb{R}$

So $f(x)$ is convex

(2) $f(x)=ax+b$ $\quad f''(x)=0\geq 0$ $\quad$ dom f is convex which is similar to (1).

So $f(x)$ is convex

$x_1,x_2\in\mathbb{R}$ $\theta\in[0,1]$ $\theta f(x_1)+(1-\theta)f(x_2)=\theta(ax_1+b)+(1-\theta)(ax_2+b)=a[\theta x_1+(1-\theta)x_2]+b=f(\theta x_1+(1-\theta)x_2)$

$f(x)$ is not strictly convex

(3) $f(x)=|x|$ $\quad x_1,x_2\in\mathbb{R}$ $\quad \theta\in[0,1]$

$f(\theta x_1+(1-\theta)x_2)=|\theta x_1+(1-\theta)x_2|$

$\leq|\theta x_1|+|(1-\theta)x_2|=\theta|x_1|+(1-\theta)|x_2|=\theta f(x_1)+(1-\theta)f(x_2)$ $\quad$ dom f is convex

$f(x)$ is convex.

$x_1,x_2\geq 0$ $\theta\in[0,1]$ $\Rightarrow$ $\theta x_1+(1-\theta)x_2\geq 0$ $\Rightarrow$ $|\theta x_1+(1-\theta)x_2|=\theta x_1+(1-\theta)x_2=\theta|x_1|+(1-\theta)|x_2|$

$f(\theta x_1+(1-\theta)x_2)=|\theta x_1+(1-\theta)x_2|=\theta|x_1|+(1-\theta)|x_2|=\theta f(x_1)+(1-\theta)f(x_2)$

$f(x)$ is not strictly convex.

1.4 $f(x|\mu,b)=\frac{1}{2b}e^{-\frac{|x-\mu|}{b}}$

$\ln f(x|\mu,b)=-\ln(2b)-\frac{|x-\mu|}{b}$

$l(\mu,b|x)=-n\ln(2b)-\frac{1}{b}\sum_{i=1}^{n}|x_i-\mu|$

$\frac{\partial l}{\partial\mu}=\frac{1}{b}\sum_{i=1}^{n}\mathrm{sgn}(x_i-\mu)=0 \Rightarrow \hat{\mu}=\mathrm{median}(x)$

$\frac{\partial l}{\partial b}=-\frac{n}{b}+\frac{1}{b^2}\sum_{i=1}^{n}|x_i-\mu|=0$

$\hat{b}=\frac{1}{n}\sum_{i=1}^{n}|x_i-\hat{\mu}|$

Step1

1. Characteristics of this dataset

(1)

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063	22.532806
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062	9.197104
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000	17.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000	21.200000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000

(2) linear correlation(absolute value):

(3)

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
medv	0.388305	0.360445	0.483725	0.175260	0.427321	0.695360	0.376955	0.249929	0.381626	0.468536	0.507787	0.333461	0.737663	1.000000

(4)

(5) “medv”: The average is similar to median, the data distribution is relatively uniform. The max is not too large, the min is not too small

(6) “crim” “zn” “chas” “rad” are not distributed uniformly, max is too large. The absolute value of linear correlation is small.

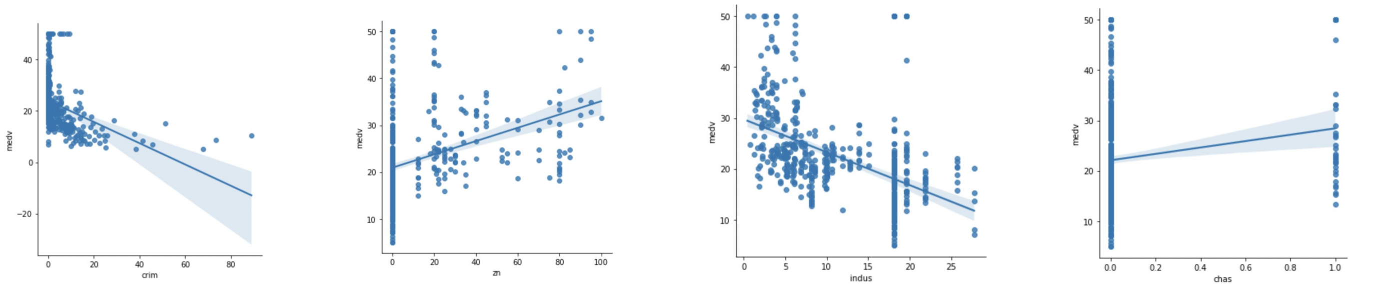
(7) “b” : 75% above of points are greater than 375, but min is 0.3, which too small.

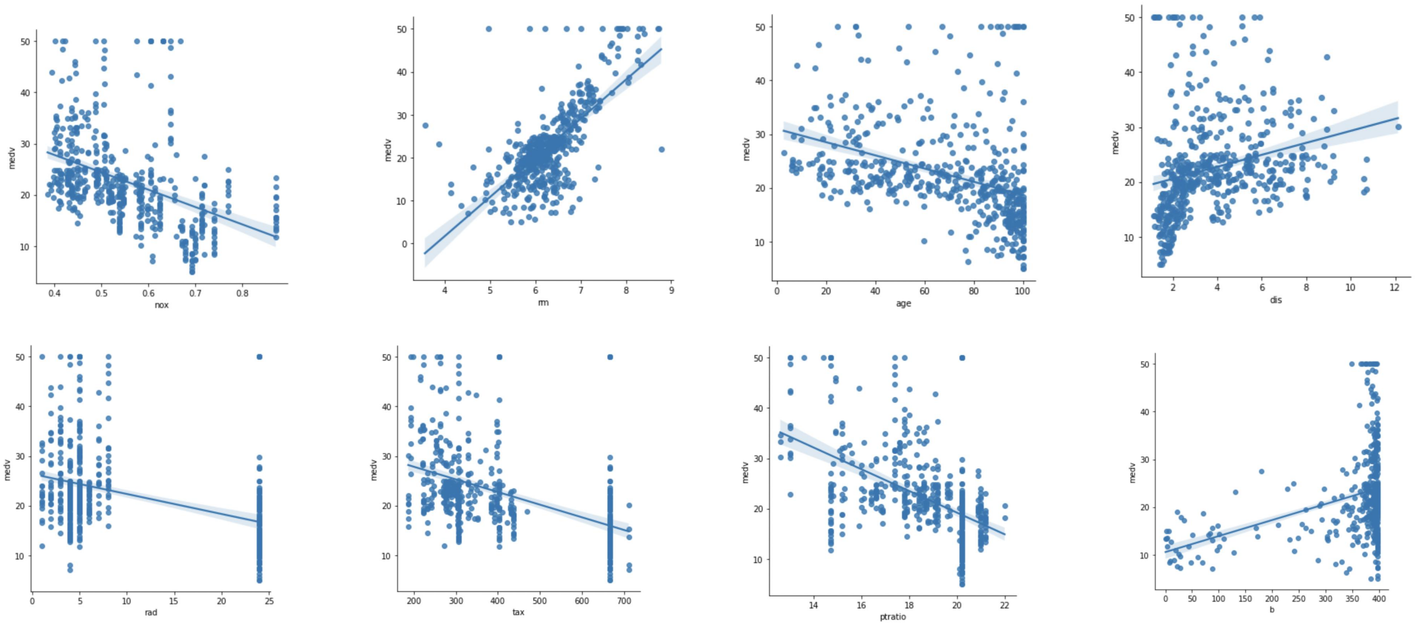
(8) “indus” “nox” “rm” “age” “dis” “tax” “ptratio” “lstat” are distributed relative uniform, the absolute value of linear correlation is relative large. And the term with the largest absolute value of correlation is “lstat”. The more lower status of population there are, the more the cheap own-occupied houses there are. Because the lower status is less possible to afford the expensive house.

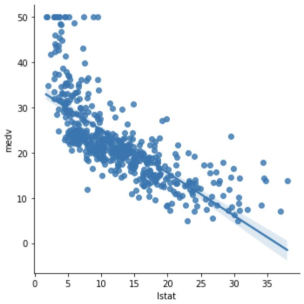
2. I guess the most relevant attribute for MEDV is “lstat”.

Step2

1.



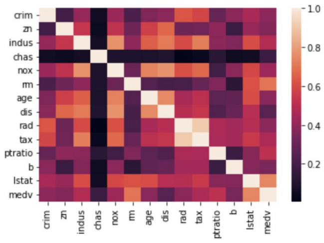




2. figure “indus” “nox” “rm” “age” “dis” “lstat” have relative obvious linear correlationship with “medv”, specially for “rm” “lstat”. The datas of any other figure are distribute too scatteringly.
3. “lstat” is still the most relevant attribute for MEDV.

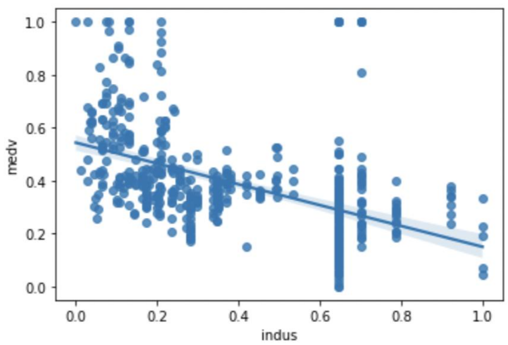
Step3

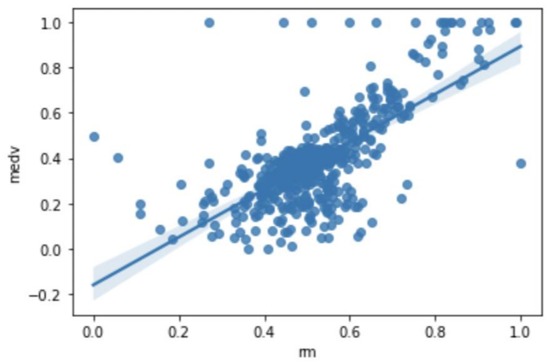
```
sns.heatmap(df.corr().abs())
<AxesSubplot:>
```



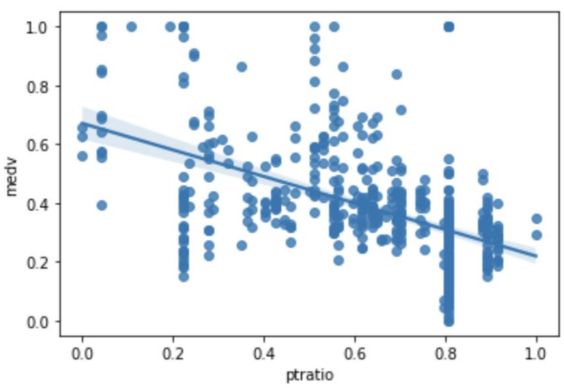
1.
2. we paint the heatmap of the absolute valute of linear correlation. we should pick the shallow boxes.
3. we choose “indus” “rm” “ptratio” “lstat” as predictors.

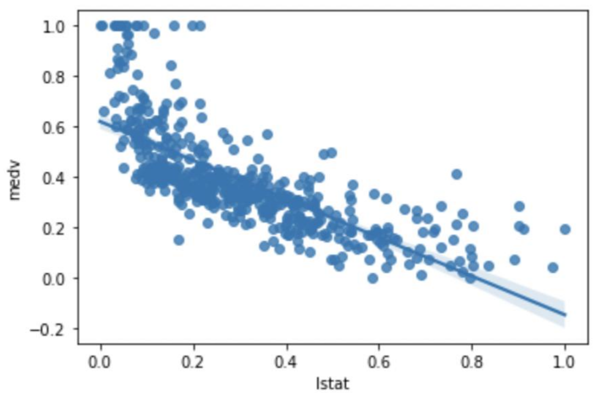
Step4





1.





2. The data points is around the line, which is fit to our expectation generally.

Step5

1. Linear model

   (1) we want to predict “medv” by hypothesis function $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w}$, $\mathbf{X} \in \mathbf{R}^{\mathbf{m}\times(\mathbf{d}+\mathbf{1})}$, $\mathbf{w} \in \mathbf{R}^{\mathbf{d}+\mathbf{1}}$, X is data matrix, $\mathbf{w}$ is weight.

(2) we use MSE and define cost function as $J(w) = \frac{1}{2}(Xw - y)^T(Xw - y)$, $y \in R^m$, y is the predicted vector.

(3) Gradient Descent

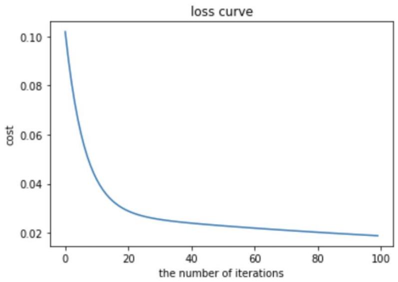
① we use gradient descent method and set learning rate is 0.0001, the number of iterations is 100.

② we pick $\mathbf{w}_0 = \mathbf{0}_{d+1}$

③ $\mathbf{w} \leftarrow \mathbf{w} - learningRate \times \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}, \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{X}^T(\mathbf{Xw} - \mathbf{y})$

(4) RMSE equation is $\sqrt{(1/n)(Xw - y)^T(Xw - y)}$, n is the number of data points.

(5) Training error in terms of RMSE is 0.19405, testing error in terms of RMSE is 0.14647. (only show 5 digits behind point here)

(6) 

the cost is decreasing with the number of iterations, and the speed of decreasing is slower and slower. Because the gradient is smaller and smaller with approaching to the optimal solution. The surface around the optimal solution is smooth.

Step6

1. Training error in terms of RMSE

stepSize\#iterations	10	100	500	1000
0.00001	0.42207949971516906	0.2886967970577309	0.21047468038388592	0.1896069405769589
0.0001	0.28575732233120427	0.18955021655249835	0.1288496680507007	0.12090147896011406
0.001	0.188972375260991	0.12086758667884592	0.1182803390153423	0.11827572674100233

2. Testing error in terms of RMSE

stepSize\#iterations	10	100	500	1000
0.00001	0.3890831259682921	0.2528961281252281	0.2528961281252281	0.1641702600437153
0.0001	0.2499214516597396	0.164121741109645	0.11282164702140571	0.10783282006595289
0.001	0.16360124515339328	0.10783530879673986	0.10537740384720519	0.10534322566854573

3. we can see that

(1) when #iterations is fixed, the RMSE is decreasing with stepSize increasing to some extent.

(2) when stepSize is fixed, the RMSE is decreasing with #iteraions increasing to some extent.

(3) with #iterations increasing, the speed of decreasing of RMSE is slower and slower.

(4) if stepSize is too big, the result may be not converged when I try other stepSizes.