1.1

① $Y=\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ $X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$ $W=\begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$

$Y^TX=\begin{bmatrix} \sum_{i=1}^{n} x_{i1}y_i & \sum_{i=1}^{n} x_{i2}y_i & \cdots & \sum_{i=1}^{n} x_{id}y_i \end{bmatrix}$ $1\times d$

$Y^TXW=\sum_{i=1}^{d} w_i \sum_{j=1}^{n} x_{ji}y_j$ 1×1

$\frac{d(Y^TXW)}{dW}=\begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$, $\forall k$ $\frac{\partial f}{\partial w_k}=\frac{\partial (w_k \sum_{j=1}^{n} x_{kj}y_j)}{\partial w_k}$

$\frac{\partial f}{\partial w_i}=\frac{2(w_i \sum_{j=1}^{n} x_{ji}y_j)}{2w_i}+\frac{2(\sum_{k\neq i} w_k \sum_{j=1}^{n} x_{jk}y_j)}{2w_i}$

$=\sum_{j=1}^{n} x_{ji}y_j+0=\sum_{j=1}^{n} x_{ji}y_j$ for $\forall i\in\{1,\dots,d\}$

$\frac{d(Y^TXW)}{dW}=\begin{bmatrix} \sum_{j=1}^{n} x_{j1}y_j \\ \sum_{j=1}^{n} x_{j2}y_j \\ \vdots \\ \sum_{j=1}^{n} x_{jd}y_j \end{bmatrix}=X^TY$

Q.E.D.

② $W^TW=\sum_{i=1}^{d} w_i^2$

$\frac{d(W^TW)}{dW}=\begin{bmatrix} \frac{2(W^TW)}{2w_1} \\ \frac{2(W^TW)}{2w_2} \\ \vdots \\ \frac{2(W^TW)}{2w_d} \end{bmatrix}=\begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_d \end{bmatrix}=2\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}=2W$

Q.E.D.

③ $W^TX=\begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix}\begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix}=\begin{bmatrix} \sum_{i=1}^{d} w_i x_{i1} & \cdots & \sum_{i=1}^{d} w_i x_{id} \end{bmatrix}$

$W^TX=\begin{bmatrix} \sum_{i=1}^{d} w_i x_{i1}, & \sum_{i=1}^{d} w_i x_{i2} & \cdots & \sum_{i=1}^{d} w_i x_{id} \end{bmatrix}$

$W^TXW=\sum_{j=1}^{d} w_j \sum_{i=1}^{n} x_{ij}w_i$ $\frac{d(W^TXW)}{2W}=\begin{bmatrix} \frac{2f(W^TXW)}{2w_1} \\ \vdots \\ \frac{2f(W^TXW)}{2w_d} \end{bmatrix}$

$\frac{2(W^TXW)}{2w_k}=\frac{2(w_k \sum_{i=1}^{n} x_{ik}w_k)}{2w_k}+\frac{2(\sum_{j\neq k} w_j \sum_{i=1}^{n} x_{ij}w_j)}{2w_k}$

$=\sum_{i=1}^{n} x_{ik}w_k+\sum_{j\neq k} w_j \sum_{i=1}^{n} x_{ij}$

$=\sum_{i=1}^{n} x_{ik}w_i+w_kx_{kk}+\sum_{j\neq k} w_j \sum_{i=1}^{n} x_{kj}$

$=\sum_{i=1}^{n} x_{ik}w_i+\sum_{j=1}^{d} x_{kj}w_j=X_{k\cdot}W+(X^T)_{k\cdot}W$

$=\frac{d}{}$ for $\forall k\in\{1,2,\dots,d\}$

so $\frac{2(W^TXW)}{2W_k}=\begin{bmatrix} X_{1\cdot}W+(X^T)_{1\cdot}W \\ X_{2\cdot}W+(X^T)_{2\cdot}W \\ \vdots \\ X_{d\cdot}W+(X^T)_{d\cdot}W \end{bmatrix}=\begin{bmatrix} X_{1\cdot} \\ X_{2\cdot} \\ \vdots \\ X_{d\cdot} \end{bmatrix}W+\begin{bmatrix} (X^T)_{1\cdot} \\ (X^T)_{2\cdot} \\ \vdots \\ (X^T)_{d\cdot} \end{bmatrix}W$

$=XW+X^TW=(X+X^T)W$

1.2

(1) Let $X=\begin{bmatrix} 1 & x_1^T \\ 1 & x_2^T \\ \vdots & \vdots \\ 1 & x_N^T \end{bmatrix}$ $N\times(d+1)$ $Y=\begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix}$ $N\times 1$

$\bar{W}=\begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$ $(d+1)\times 1$ $f_{w,b}(x_i)$ $f_{\bar{w},b}(X)=X\bar{W}$

$J(\bar{W})=\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x_i)-y_i)^2+\lambda\bar{W}^T\bar{W}=\frac{1}{N}\|X\bar{W}-Y\|_2^2+\lambda\bar{W}^T\bar{W}$

$\nabla J(\bar{W})=\frac{1}{N}\cdot(X\bar{W}-Y)^T(X\bar{W}-Y)+\lambda\bar{W}^T\bar{W}$

so $\nabla J(\bar{W})=\frac{1}{N}2X^T(X\bar{W}-Y)+2\lambda\bar{W}=\frac{1}{N}(2X^TX+2N\lambda I)\bar{W}-\frac{1}{N}2X^TY$

$\nabla J(\bar{W})=0\Rightarrow(2X^TX+2N\lambda I)\bar{W}=2X^TY$

$\bar{W}=(X^TX+N\lambda I)^TX^TY$

(2) ① for each step $\bar{W}_t$, calculate $\nabla J(\bar{W})$

② choose learning rate α by back-tracking

③ update $\bar{W}_t$ by: $\bar{W}_{t+1}=\bar{W}_t-\alpha\nabla J(\bar{W})$

1.3

(1) since $x \in R$, the domain is a convex set.

$\forall x, y \in R$

$f(\lambda x + (1-\lambda)y) = \lambda^2 x^2 + (1-\lambda)^2 y^2$

since $\lambda \in [0,1]$

$0 \le \lambda^2 \le \lambda$, $0 \le (1-\lambda) \le 1 \Rightarrow \lambda^2 x^2 \le \lambda x^2$, $(1-\lambda)^2 y^2 \le (1-\lambda) y^2$

since $x^2 \ge 0$, $y^2 \ge 0$

$\lambda^2 x^2 \le \lambda x^2$, $(1-\lambda)^2 y^2 \le (1-\lambda) y^2$

$f(\lambda x + (1-\lambda)y) \le \lambda x^2 + (1-\lambda) y^2$

$f'(x) = 2x \quad f''(x) = 2$

$\forall d \in R \quad d^T f''(x) d = 2d^2 \ge 0$

so $f(x)$ is PSD $\Rightarrow f(x)$ is convex

(2) $\forall x, y \in$ domain is R, hence convex

$\forall x, y \in R$

$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y) = \lambda(ax+b) + (1-\lambda)(ay+b)$

$= a(\lambda x + (1-\lambda)y) + \lambda b + (1-\lambda) b$

$= a(\lambda x + (1-\lambda)y) + b = f(\lambda x + (1-\lambda)y)$

$\Downarrow$

$f(\lambda x + (1-\lambda)y) = \lambda f(x) + (1-\lambda) f(y)$

$\Downarrow$

$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y)$

so $f(x)$ is convex

since $\exists x, y$, s.t. $f(\lambda x + (1-\lambda)y) = \lambda f(x) + (1-\lambda) f(y)$

$f(x)$ is not strictly convex

(3) $f(x)$ $x \in R \Rightarrow$ domain convex

$\forall x, y \in R \quad f(\lambda x + (1-\lambda)y) = \lambda |x| + |\lambda x + (1-\lambda) y|$

$\le \lambda |x| + |(1-\lambda) y| = \lambda |x| + (1-\lambda) |y|$

$= \lambda f(x) + (1-\lambda) f(y)$

hence $f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y)$

hence $f(x)$ is convex

when $x \ge 0$, $y \ge 0$

$f(\lambda x + (1-\lambda)y) = |\lambda x + (1-\lambda) y| = \lambda x + (1-\lambda) y = \lambda |x| + (1-\lambda) |y|$

hence $\exists x, y \in R$ $f(\lambda x + (1-\lambda)y) = \lambda f(x) + (1-\lambda) f(y) \Rightarrow$ not strictly

1.4

$x_i \sim \text{Laplace}(u, b) \Rightarrow f(x_i | u, b) = \frac{1}{2b} \exp(-\frac{|x-u|}{b})$

$(u,b)_{MLE} = \arg\max_{(u,b)} \log(\prod_{i=1}^{N} f(x_i | u, b))$

$\log \prod_{i=1}^{N} f(x_i | u, b) = \log \prod_{i=1}^{N} \frac{1}{2b} \exp(-\frac{|x_i - u|}{b})$

$= \log(\frac{1}{2b})^N \cdot \exp(-\frac{|x-u|}{b}) = \log(\frac{1}{2b})^N + \log \exp(-\frac{|x-u|}{b})$

$= -N \log 2b - \frac{|x-u|}{b}$

$(u,b)_{MLE} = \arg\max_{(u,b)} (-N \log 2b - \frac{|x-u|}{b})$

$= \arg\max_{u}$

$= \log(\frac{1}{2b})^N \prod_{i=1}^{N} \exp(-\frac{|x-u|}{b}) = -N \log 2b + \log e$

$= \log \prod_{i=1}^{N}$

$= \log(\frac{1}{2b})^N \prod_{i=1}^{N} \exp(-\frac{|x_i - u|}{b}) = \log(\frac{1}{2b})^N + \log \prod_{i=1}^{N} \exp(-\frac{|x_i - u|}{b})$

$= -N \log 2b + \sum_{i=1}^{N} -\frac{|x_i - u|}{b}$

hence

$(u,b)_{MLE} = \arg\max_{(u,b)} -N \log 2b - \sum_{i=1}^{N} \frac{|x_i - u|}{b}$

$= \arg\min_{(\mu,b)} N \log 2b + \sum_{i=1}^{N} \frac{|x_i - u|}{b}$

Report-Assignment1

Linear Model

In this assignment, I use ridge linear regression to fit the data. Specifically, given an input selected features $x \in R^d$, where d is the number of features, the predicted housing price $\hat{y}$ can be predicted by

$$\hat{y} = \bar{x}^T \bar{w}$$

where

$$\bar{x} = [1, x_1, x_2, ..., x_d]^T \in R^{d+1}$$

$$\bar{w} = [b, w_1, w_2, ..., w_d]^T \in R^{d+1}$$

Loss Function

Since I use ridge linear regression, the parameters $\bar{w}$ is trained by minimizing the loss function:

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (f_{\bar{w}}(x_i) - y_i)^2 + \frac{\lambda}{2} \bar{w}^T \bar{w}$$

where m is the number of training samples.
The loss function can also be written as

$$J(w) = \frac{1}{2m} ||X\bar{w} - Y||^2 + \frac{\lambda}{2} \hat{w}^T \hat{w}$$

where

$$X = \begin{matrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \dots \\ \bar{x}_m^T \end{matrix} \in R^{m \times (d+1)}$$

$$Y = \begin{matrix} y_1 \\ y_2 \\ \dots \\ y_m \end{matrix} \in R^{m \times (d+1)}$$

$$\hat{w} = [0, w_1, w_2, ..., w_d]^T \in R^{d+1}$$

Gradient Descend

The procedure of gradient descend can be summarized as:

- Calculate the gradient of loss function w.r.t. weight

$$\nabla J(\bar{w}) = \frac{\partial J}{\partial \bar{w}}$$

- Calculate learning rate α by Armijo Line Search
- Update w by $\bar{w} = \bar{w} - \alpha \nabla J(\bar{w})$
- Check whether the procedure should be terminated
- Go back to step 1 if not terminated

The gradient can be calculated as:

$$\nabla J(\bar{w}) = \frac{1}{m} X^T (X\bar{w} - Y) + \lambda \hat{w}$$

The learning rate α for each step is selected from a set of weights $\{1, \sigma, \sigma^2, \sigma^3, ...\}$ by tring through the set starting from 1, until

$$J(w^k - \alpha_k \nabla J(w^k)) - J(w^k) \leq -\gamma \alpha_k \nabla J(w^k)^T \nabla J(w^k)$$

Whether to terminate the procedure is determined by the loss difference before and after the weight is updated. The procedure is terminated when

$$||J(w^k) - J(w^k - \alpha_k \nabla J(w^k))||^2 \leq threshold$$

Therefore, in total, there are 4 hyperparamters: λ, σ, γ and $threshold$.

Features Selection

In order to avoid overfitting, instead of simly using all features, I only select several features for price prediction. In this report, only part of analysis plots will be shown. To see all plots, please refer to code.ipynb. Features that has an obvious linear relaionship with the price will be chosen. Specifically, features rm, age, dis and lstat are chosen. As is shown in the following plot, these features have strong relationship with price.

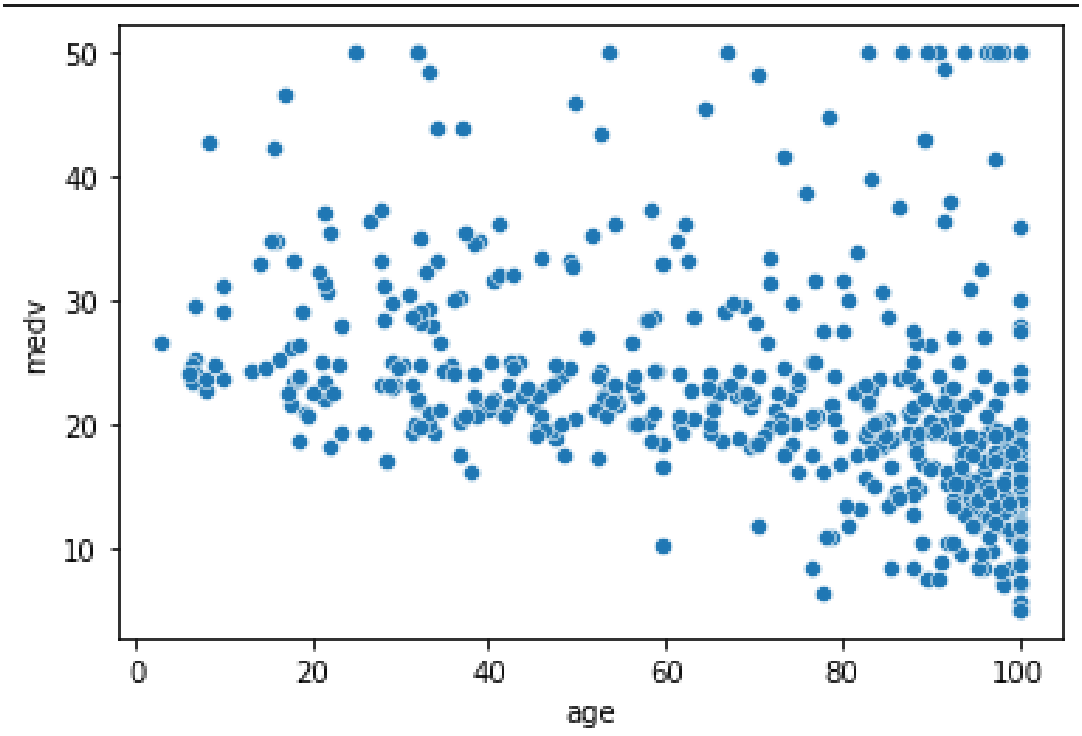

Figure 1: Relationship between age and medv

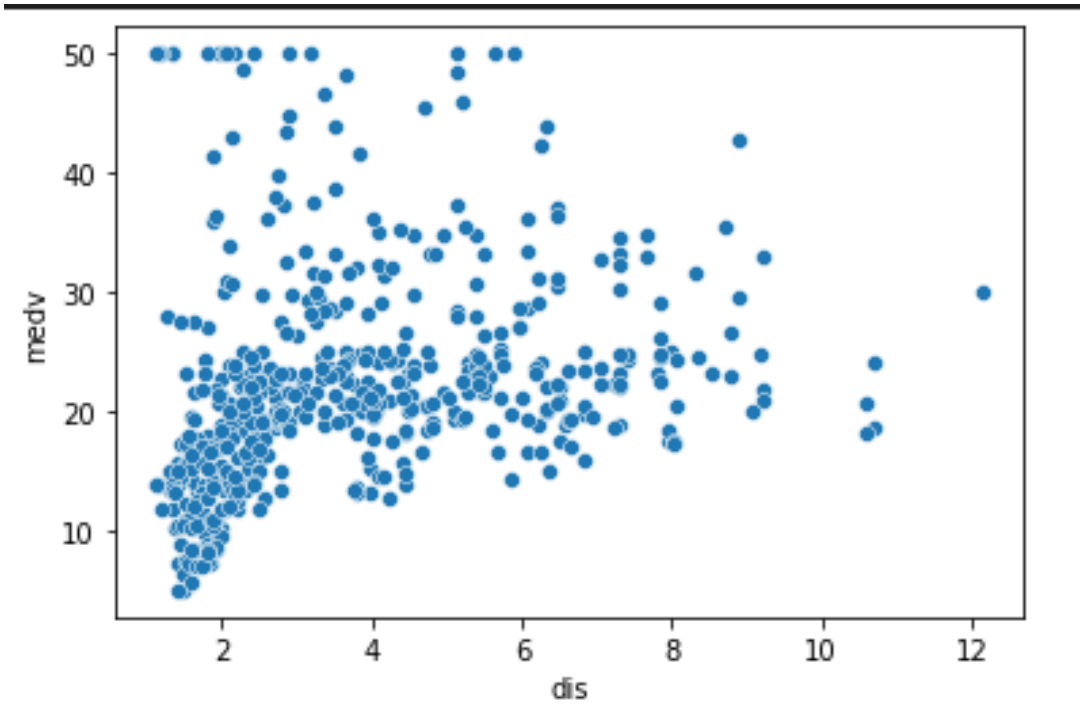

Figure 2: Relationship between dis and medv

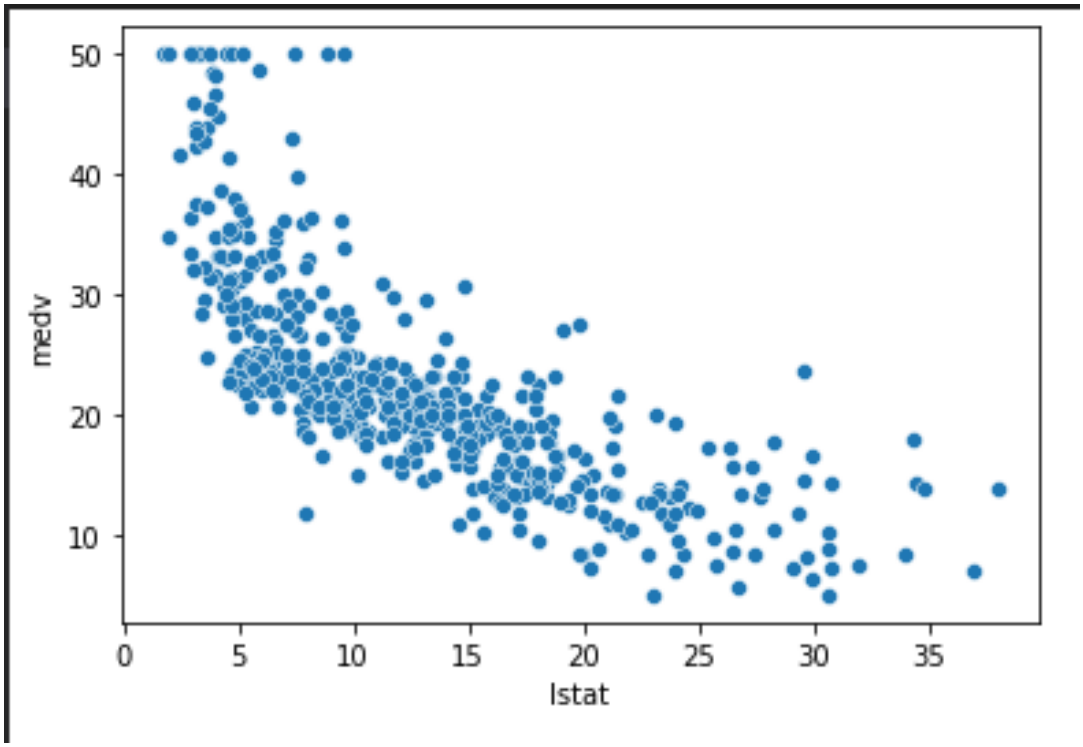

Figure 3: Relationship between lstat and medv

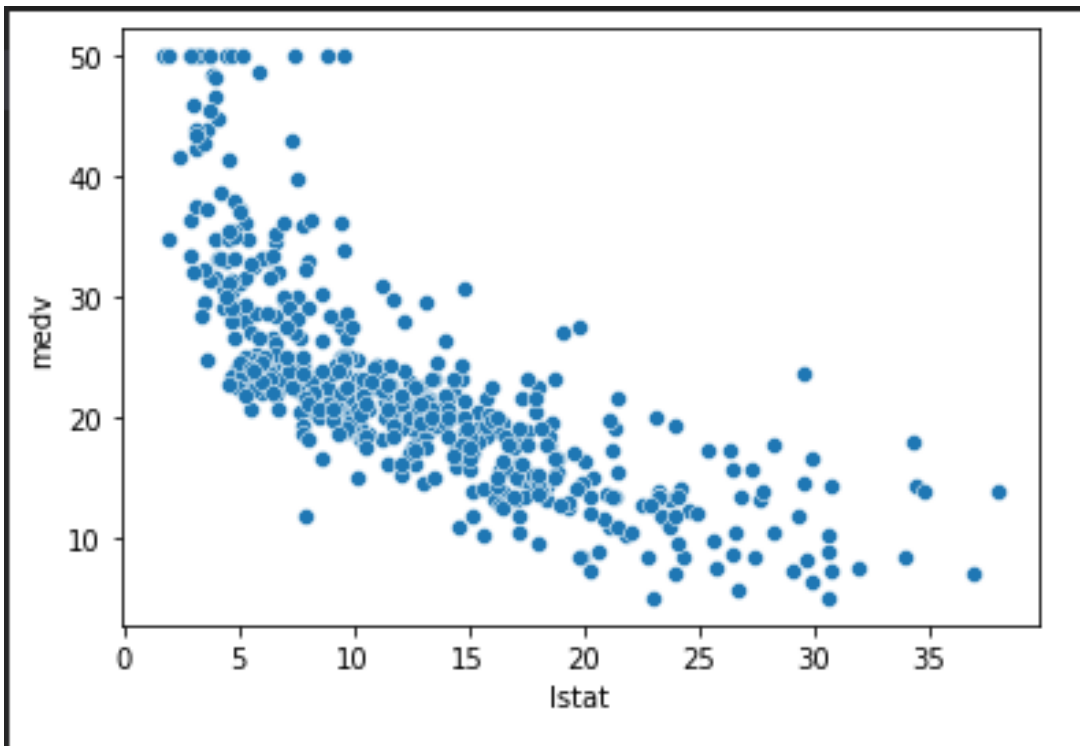

Figure 4: Relationship between rm and medv

As is shown in the heatmap, these features are not depenednt on each other. None of the two features are of the same distribution.

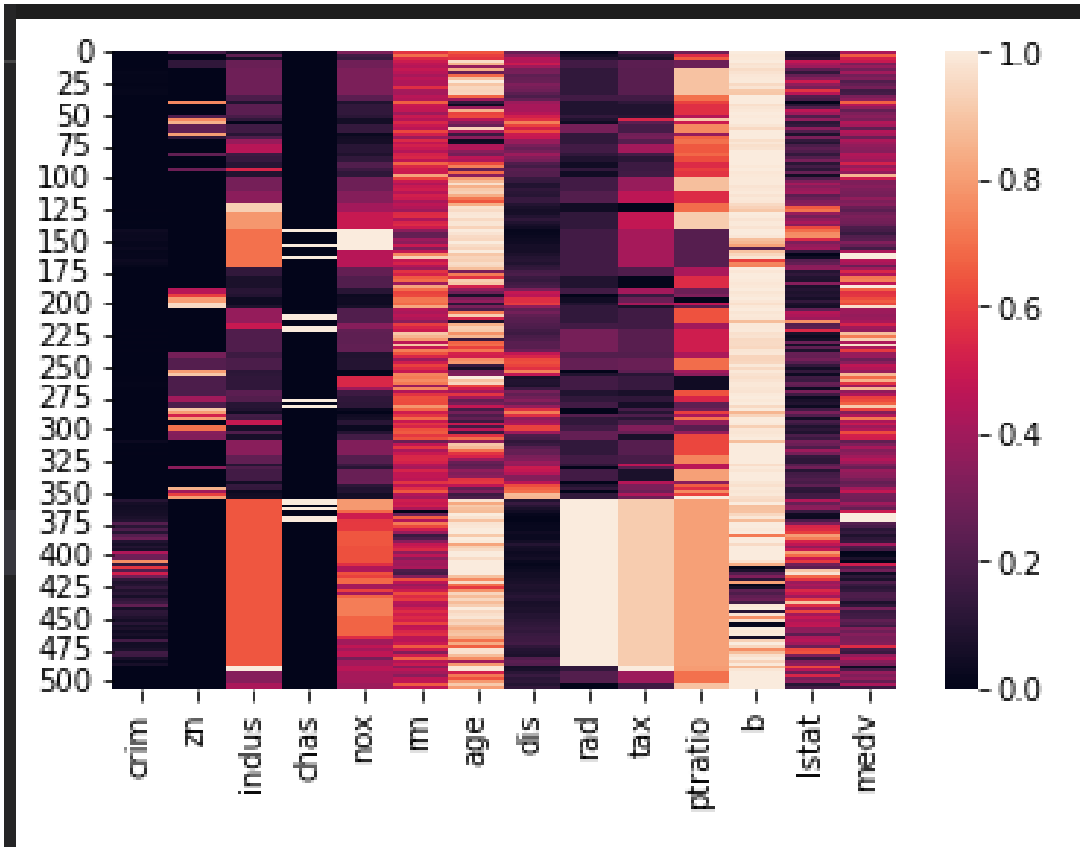

Figure 5:Heatmap

However, there exists a weak correlation between some pairs, as is shown in relevance plot. Here only several of these plots will be shown. To see all plots, please refer to code.ipynb.

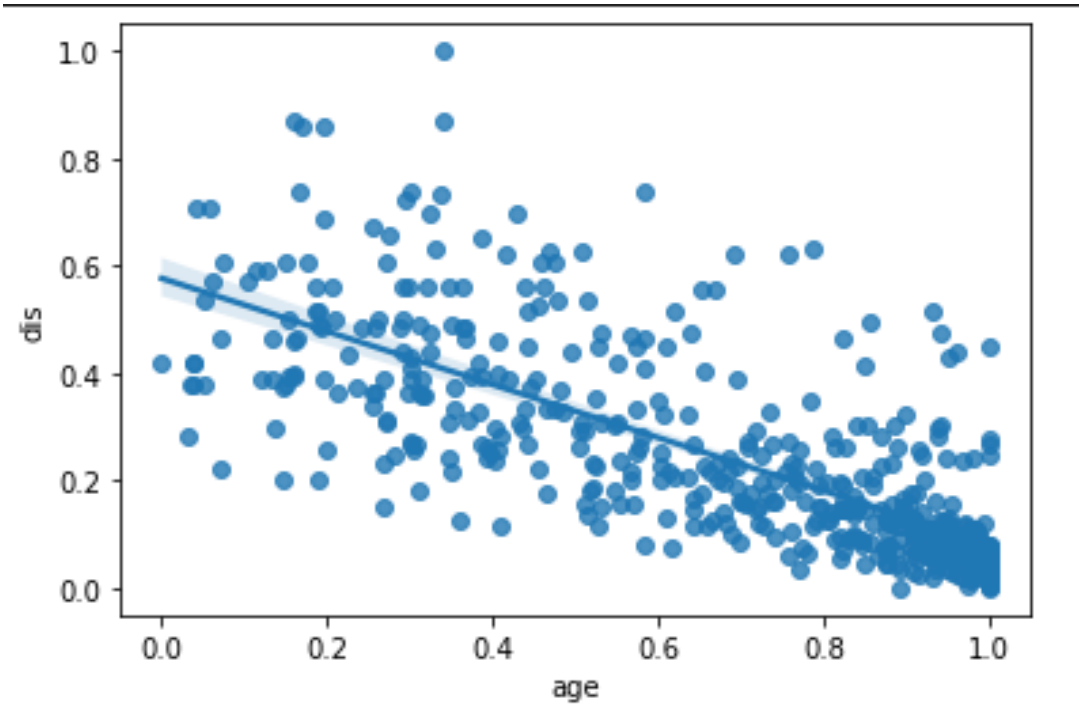

Figure 6:Relation between age and dis

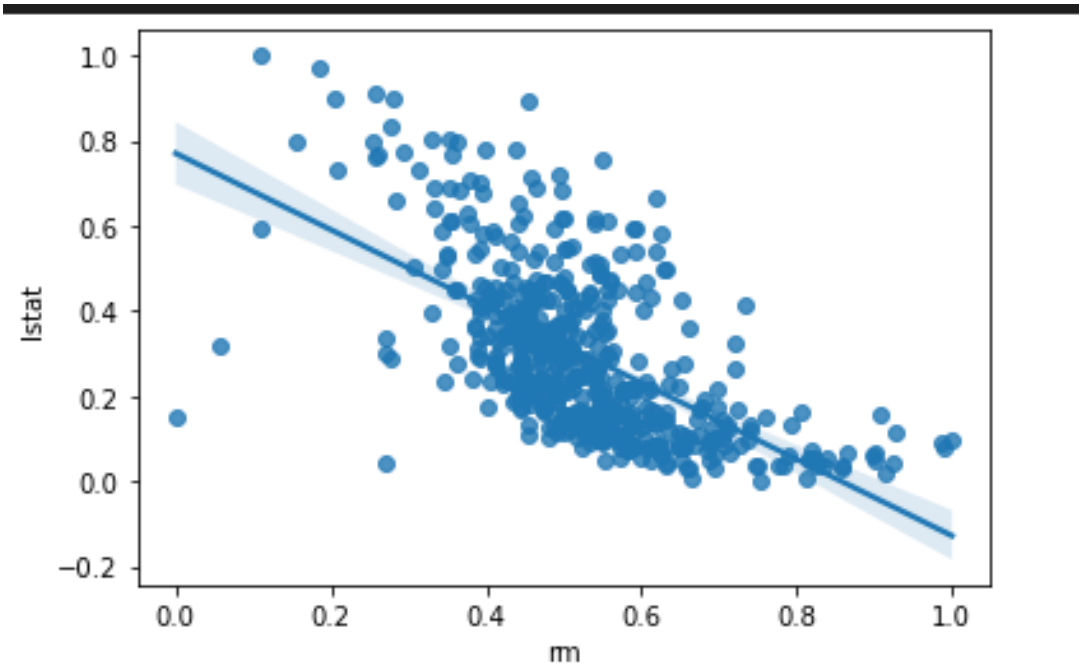

Figure 7:Relation between rm and lstat

Training and Testing

10 experiments are conducted. In each experiment, the data is randomly split into training and testing set in a ratio of 8:2. In each experiment, the linear regression model is trained from random initialization, and the tested by the testing set.

Training

In training, the hyperparameters are set as

$$\lambda = 0.01$$

$$\sigma = 0.5$$

$$\gamma = 0.1$$

$$threshold = 10^{-6}$$

In 10 experiments, the training is terminated at around 170 to 200 iterations. Two plots of loss function value over iterations are shown here. To see all plots, please refer to code.ipynb.

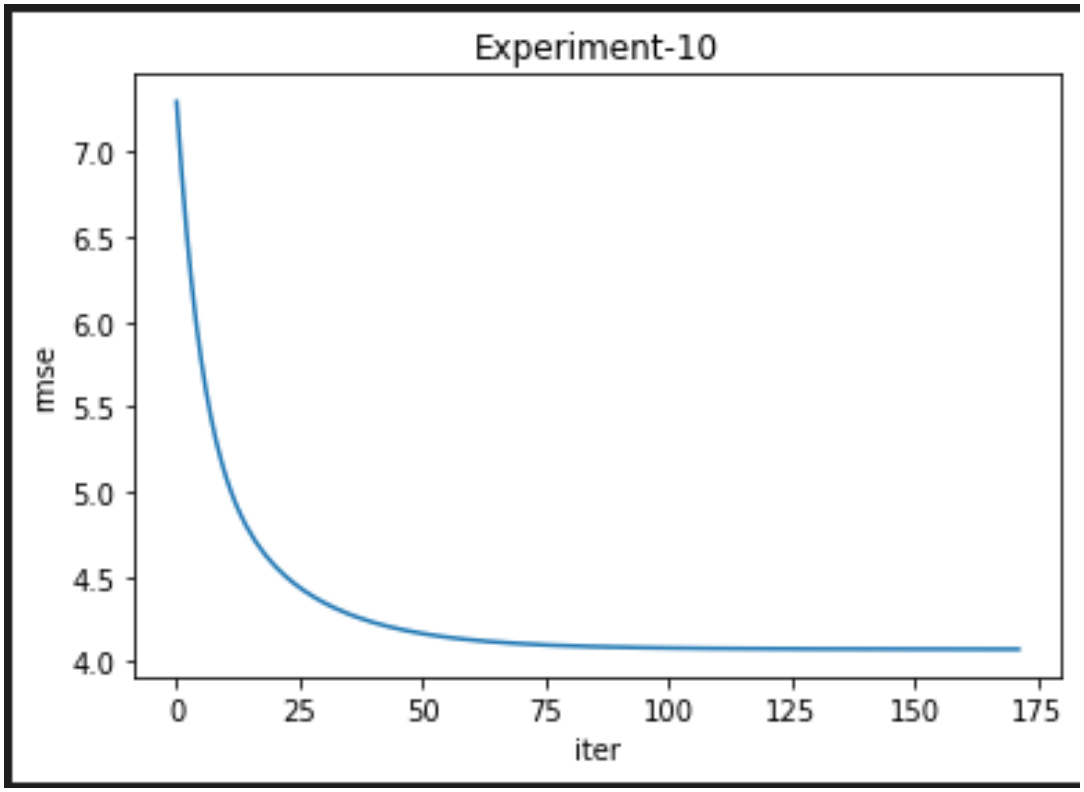

Figure 8:Loss over iteration at experiment1

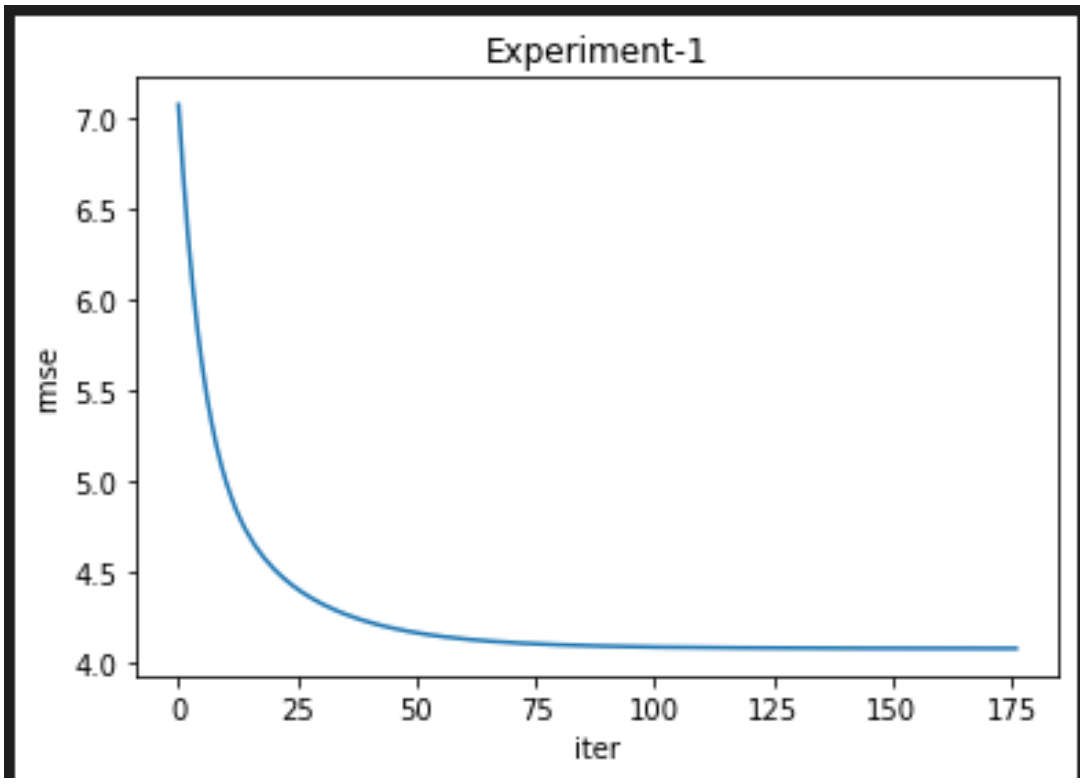

Figure 9:Loss over iteration at experiment10

Testing

RMSE is used in testing the model.

$$rmse(w) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(f_{\bar{w}}(x_i) - y_i)^2}$$

where k is the number of testing sample. The mean testing rmse over 10 experiments is 3.807362106194172. To see the testing results of all 10 experiments, please refer to code.ipynb.