1. Logistic Regression

(a) Negative Log Likelihood (as loss function)

Probability of a single training sample (x_n,y_n)

$$p(y_n|x_n; b; w) = \begin{cases} \sigma(b + w^T x_n) \; if \; y_n = 1 \\ 1 - \sigma(b + w^T x_n) \; otherwise \end{cases}$$

Compact expression, exploring that y_n is either 1 or 0

$$p(y_n|x_n; b; w) = \sigma(b + w^T x_n)^{y_n}[1 - \sigma(b + w^T x_n)]^{1-y_n}$$

The negative log likelihood can be written as,

$$L(w) = -log\left(\prod_{i=1}^{n} P(Y = y_i|X = x_i)\right)$$

So, the cross entropy (loss function) for negative log likelihood is,

$$\varepsilon(b, w) = -\sum_{i=1}^{n}\{y_i log\sigma(b + w^T x_i) + (1 - y_i)\log[1 - \sigma(b + w^T x_i)]\}$$

(b) Gradient Descent to find the update rule for w

Appending $1 \; to \; x, \; x \leftarrow [1 \; x_1 \; x_2 \; \dots \; x_n]$ and $b \; to \; w, \; w \leftarrow [b \; w_1 \; w_2 \; \dots \; w_n]$,

$$\varepsilon(b, w) = -\sum_{i=1}^{n}\{y_i log\sigma(w^T x_i) + (1 - y_i)\log[1 - \sigma(w^T x_i)]\}$$

Computing the gradients we get,

$$\frac{\delta\epsilon(w)}{\delta w} = -\sum_{i=1}^{n}\{y_i[1 - \sigma(w^T x_i)]x_i - (1 - y_i)\sigma(w^T x_i)x_i\}$$

$$= \sum_{i=1}^{n}\{\sigma(w^T x_i) - y_i\}x_i$$

To see if it converges to global minimum, we'll check if it's 2nd derivative is ≥ 0

$$\frac{\delta^2 L(w)}{\delta w^2} = -\frac{\delta \sum_{i=0}^{n}\left(y_i - \sigma(w^T x_i)\right)x_i}{\delta(w)}$$

$$= -\sum_{i=0}^{n} -\sigma\,(w^T x_i)(1-\sigma(w^T x_i))x_i^2$$

$$= \sum_{i=0}^{n} \sigma\,(w^T x_i)(1-\sigma(w^T x_i))x_i^2$$

We know that $0 <= \sigma(a) <= 1$ and $(x_i)^2 >= 0$

$$\frac{\delta^2 L(w)}{\delta w^2} \geq 0$$

Thus we can conclude that the gradient descent converges to a global minimum.

(c) Negative log likelihood $L(w_1, \dots w_k)$ for multi-class classification

Given K different classes, we have the posterior probability for class K as,

$$P(Y = k|X = x) = \frac{\exp(w_k^T x)}{1 + \sum_1^{K-1} \exp(w_t^T x)}, for\ k = 1, \dots., K-1$$

$$P(Y = k|X = x) = \frac{1}{1 + \sum_1^{K-1} \exp(w_t^T x)}, for\ k = K$$

The negative log likelihood $L(w_1, \dots w_k)$ can be written as,

$$L(w_1, \dots w_k) = -\log \prod_{i=1}^{n} P(y_i|x_i)$$

$$L(w_1, \dots w_k) = -\sum_{i=1}^{n} \log P(y_i|x_i)$$

Let y_i be changed to $y_i = [y_{i1}, y_{i2}, \dots y_{iK}]^T$ a K-dimensional vector using 1 of K encoding:

$$y_{ik} = \begin{cases} 1, if\ y_i = k \\ 0, otherwise \end{cases}$$

Therefore, we get,

$$L(w_1, \dots w_k) = -\log \prod_{i=1}^{n} P(y_i|x_i)$$

$$= -\sum_{i=1}^{n} log \prod_{k=1}^{K} P(Y = k|x_i)^{y_{ik}}$$

$$= -\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} log P(Y = k|x_i)$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \frac{\exp(w_{y_i}^T x_i)}{1 + \sum_{l=1}^{K-1} \exp(w_l^T x_i)}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \frac{\exp(w_{y_i}^T x_i)}{\exp(0) + \sum_{l=1}^{K-1} \exp(w_l^T x_i)}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \frac{\exp(w_{y_i}^T x_i)}{\sum_{l=1}^{K} \exp(w_l^T x_i)}$$

Applying the rule $\frac{\log(a)}{\log(b)} = \log(a) - \log(b)$, we get,

$$\boxed{L(w_1, \dots w_k) = \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \left[w_{y_i}^T x_i - \log(1 + \sum_{l=1}^{K} \exp(w_l^T x_i) \right]}$$

(d) Gradient with respect to w_i

We can calculate the gradient descent by,

$$\frac{\delta L(w_1, w_2, \dots., w_K)}{\delta w_i} = -\frac{\delta\left(\sum_{i=1}^{n}\sum_{k=1}^{n} y_{ik}(w_k^T x_i - \log \sum_{i=1}^{k} \exp(w_k^T x_i))\right)}{\delta w_i}$$

$$\boxed{= -\sum_{i=1}^{n}\sum_{k=1}^{n} y_{ik}(1 - \frac{\exp(w_k^T x_i)}{\sum_{l=1}^{K-1} \exp(w_l^T x_i)}}$$

2. Linear / Gaussian Discriminant

(a) The log likelihood function $\mathcal{L}(D)$ and finding $(p_1^*, p_2^*, \mu_1^*, \mu_2^*, \sigma_1^*, \sigma_2^*)$ that maximizes $\mathcal{L}(D)$

We have a Gaussian Discriminant Analysis, given *n* training examples

$D = \{(x_n, y_n)\}_{n=1}^{N}$, with $y_n \in \{1, 2\}$, where

$$P(x_n, y_n) = P(y_n)P(x_n|\, y_n) = \begin{cases} P_1 \dfrac{1}{\sqrt{2\pi}\sigma_1} \exp\left(\dfrac{-(x_n-\mu_1)^2}{2\sigma_1^2}\right), & if\ y_n = 1 \\ P_2 \dfrac{1}{\sqrt{2\pi}\sigma_2} \exp\left(\dfrac{-(x_n-\mu_2)^2}{2\sigma_2^2}\right), & if\ y_n = 2 \end{cases}$$

The log likelihood is given by,

$$\mathcal{L}(D) = \log\left(\prod_{i=1}^{n} P(x_i, y_i)\right)$$

Or ,

$$\mathcal{L}(D) = \sum_{i=1}^{n} \log(P(x_i, y_i))$$

$$\mathcal{L}(D) = \sum_{\substack{i=1 \\ y_i=1}}^{n} \log\left(P_1 \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(\frac{-(x_n-\mu_1)^2}{2\sigma_1^2}\right)\right) + \sum_{\substack{i=1 \\ y_i=2}}^{n} \log\left(P_2 \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(\frac{-(x_n-\mu_2)^2}{2\sigma_2^2}\right)\right)$$

$$\mathcal{L}(D) = \sum_{\substack{i=1 \\ y_i=1}}^{n} \left(\log(P_1) - \log\left(\sqrt{2\pi}\sigma_1\right) - \left(\frac{(x_n-\mu_1)^2}{2\sigma_1^2}\right)\right) + \sum_{\substack{i=1 \\ y_i=2}}^{n} \left(\log(P_2) - \log\left(\sqrt{2\pi}\sigma_2\right) - \left(\frac{(x_n-\mu_2)^2}{2\sigma_2^2}\right)\right)$$

To maximize $\mathcal{L}(D)$ using MLE, we need to take first order derivatives with each parameter and equate them to 0.

By doing this we obtain the following results,

$$\frac{\delta\mathcal{L}(D)}{\delta\mu_1} = \sum_{\substack{i=1 \\ y_i=1}}^{n} \left(-2\left(\frac{(x_n-\mu_1)}{2\sigma_1^2}\right)\right) = 0$$

$$\sum_{\substack{i=1 \\ y_i=1}}^{n} (x_n) = n_1 \cdot \mu_1$$

$$or\ \mu_1 = \frac{\sum_{\substack{i=1 \\ y_i=1}}^{n} (x_n)}{n_1}$$

$$\frac{\delta \mathcal{L}(D)}{\delta \mu_2} = \sum_{\substack{i=1 \\ y_i=2}}^{n} \left(-2 \left(\frac{(x_n - \mu_2)}{2\sigma_2^2} \right) \right) = 0$$

$$\sum_{\substack{i=1 \\ y_i=2}}^{n} (x_n) = n_2 \cdot \mu_2$$

$$or\ \mu_2 = \frac{\sum_{\substack{i=1 \\ y_i=2}}^{n} (x_n)}{n_2}$$

$$\frac{\delta \mathcal{L}(D)}{\delta \sigma_1} = \sum_{\substack{i=1 \\ y_i=1}}^{n} \left(-\frac{1}{\sigma_1} - (-2) \left(\frac{(x_n - \mu_1)^2}{2\sigma_1^3} \right) \right) = 0$$

$$\sigma_1^2 = \frac{1}{n_1} \sum_{\substack{i=1 \\ y_i=1}}^{n} ((x_n - \mu_1)^2)$$

$$\sigma_1 = \sqrt{\frac{1}{n_1} \sum_{\substack{i=1 \\ y_i=1}}^{n} ((x_n - \mu_1)^2)}$$

$$\frac{\delta \mathcal{L}(D)}{\delta \sigma_2} = \sum_{\substack{i=1 \\ y_i=2}}^{n} \left(-\frac{1}{\sigma_2} - (-2) \left(\frac{(x_n - \mu_2)^2}{2\sigma_2^3} \right) \right) = 0$$

$$\sigma_2^2 = \frac{1}{n_2} \sum_{\substack{i=1 \\ y_i=2}}^{n} ((x_n - \mu_2)^2)$$

$$\sigma_2 = \sqrt{\frac{1}{n_2} \sum_{\substack{i=1 \\ y_i=2}}^{n} ((x_n - \mu_2)^2)}$$

By using the property $P_1 + P_2 = 1$, we expand P_1 and P_2 by applying Lagrange's Multiplier on $\hat{P}_1$ and $\hat{P}_2$. Putting this value in the log likelihood equation and simplifying, we get,

$$\mathcal{L}(D) = \sum_{\substack{i=1 \\ y_i=1}}^{n} \left(\log(P_1) - \log\left(\sqrt{2\pi}\sigma_1\right) - \left(\frac{(x_n - \mu_1)^2}{2\sigma_1^2} \right) \right) + \sum_{\substack{i=1 \\ y_i=2}}^{n} \left(\log(P_2) - \log\left(\sqrt{2\pi}\sigma_2\right) - \left(\frac{(x_n - \mu_2)^2}{2\sigma_2^2} \right) \right) + \lambda(\mathrm{P}_1 + \mathrm{P}_2 - 1)$$

$$\frac{\delta\mathcal{L}(D)}{\delta \mathrm{P}_1} = \sum_{\substack{i=1 \\ y_i=1}}^{n} \left(\frac{1}{\mathrm{P}_1} + \lambda \right) = 0$$

$$\frac{n_1}{\mathrm{P}_1} + \lambda = 0 \; or \; \hat{P}_1 = \frac{-n_1}{\lambda}$$

$$\frac{\delta\mathcal{L}(D)}{\delta \mathrm{P}_2} = \sum_{\substack{i=1 \\ y_i=2}}^{n} \left(\frac{1}{\mathrm{P}_2} + \lambda \right) = 0$$

$$\frac{n_2}{\mathrm{P}_2} + \lambda = 0 \; or \; \hat{P}_2 = \frac{-n_2}{\lambda}$$

Now $P_1 + P_2 = 1$. Therefore

$$\frac{-n_1}{\lambda} + \frac{-n_2}{\lambda} = 1$$

$$\lambda = -(n_1 + n_2)$$

Substituting for λ we get

$$\hat{P}_1 = \frac{n_1}{n_1 + n_2}$$

$$\hat{P}_2 = \frac{n_2}{n_1 + n_2}$$

(b) To show that $p(y|x)$ follows a logistic function

Given:

$P(x|y=c_1) = N(\mu_1, \sum)$,$P(x|y=c_2) = N(\mu_2, \sum)$ are multi variate Gaussians.

Also $\mu_1, \mu_2 \in R^D, \sum \in R^{DXD}$.

Consider $c_1 = 1 \; and \; c_2 = 0$, also

$$N(\mu_1, \sum) = 2\pi^{-D/2} \, |\sum|^{-1/2} \exp\left(\frac{-1}{2} * (x-\mu_1)' * \sum^{-1} * (x-\mu_1)\right)$$

$$N(\mu_2, \sum) = 2\pi^{-D/2} \, |\sum|^{-1/2} \exp\left(\frac{-1}{2} * (x-\mu_2)' * \sum^{-1} * (x-\mu_2)\right)$$

$$P(Y=1\,|X) = \frac{P(X\,|Y=1) \cdot P(Y=1)}{P(X)}$$

$P(X)= P(X\,|Y=1) \cdot P(Y=1) + P(X\,|Y=0) \cdot P(Y=0)$

$P(X)= N(\mu_1, \sum) \cdot P(Y=1) + N(\mu_2, \sum) \cdot P(Y=0)$

Also Let $P(Y=1) = p$, *and hence* $P(Y=0) = (1-p)$

Substituting above relations

$$P(Y=1\,|X) = \frac{P(X\,|Y=1) \cdot P(Y=1)}{P(X\,|Y=1) \cdot P(Y=1) + P(X\,|Y=0) \cdot P(Y=0)}$$

$$P(Y=1\,|X) = \frac{1}{1 + \frac{P(X\,|Y=0) \cdot P(Y=0)}{P(X\,|Y=1) \cdot P(Y=1)}} = \frac{1}{1 + \frac{N(\mu_2, \sum) \cdot (1-p)}{N(\mu_1, \sum) \cdot (p)}}$$

$$P(Y=1\,|X) = \frac{1}{1 + \frac{2\pi^{-D/2} \, |\sum|^{-1/2} \exp\left(\frac{-1}{2} * (x-\mu_2)' * \sum^{-1} * (x-\mu_2)\right)(1-p)}{2\pi^{-D/2} \, |\sum|^{-1/2} \exp\left(\frac{-1}{2} * (x-\mu_1)' * \sum^{-1} * (x-\mu_1)\right).p}}$$

$$= \frac{1}{1 + \frac{(1-p)\exp\left(\frac{-1}{2} * (x-\mu_2)' * \sum^{-1} * (x-\mu_2)\right)}{p.\exp\left(\frac{-1}{2} * (x-\mu_1)' * \sum^{-1} * (x-\mu_1)\right)}}$$

$$P(Y = 1\,|X) = \frac{1}{1 + \frac{(1-\ p)\exp(\frac{-1}{2} * (x-\mu_2)' * \sum^{-1} * (x-\mu_2) - \frac{-1}{2} * (x-\mu_1)' * \sum^{-1} * (x-\mu_1))}{p}}$$

Write

$$\frac{1-p}{p}$$

using ln as

$$exp(ln(\frac{1-p}{p})).$$

$$P(Y = 1\,|X) = \frac{1}{1 + \exp\left(\ln\left(\frac{1-p}{p}\right)\right) * \exp(\frac{-1}{2} * (x-\mu_2)' * \sum^{-1} * (x-\mu_2) - \frac{-1}{2} * (x-\mu_1)' * \sum^{-1} * (x-\mu_1))} \ldots\ldots\ldots$$

Considering equation for single dimensional variable we can solve the above given denominator part.

$$\frac{(x-\mu_1)^2}{\sigma^2} - \frac{(x-\mu_2)^2}{\sigma^2} = \frac{x^2 + {\mu_1}^2 - 2x\mu_1 - x^2 - {\mu_2}^2 + 2x\mu_2}{\sigma^2} = \frac{2x(\mu_2 - \mu_1) + {\mu_1}^2 - {\mu_2}^2}{\sigma^2}$$

Writing the equivalent of the above in matrix form for multi-dimensional data we have,

$$(x-\mu_2)' * \sum^{-1} * (x-\mu_2) - (x-\mu_1)' * \sum^{-1} * (x-\mu_1) = 2 * (\mu_2 - \mu_1\ )' * \sum^{-1} * x + \mu_1' * \sum^{-1} * \mu_1 + \mu_2' * \sum^{-1} * \mu_2$$

Using the above equation in denominator of (1)

$$P(Y = 1\,|X) = \frac{1}{1 + \exp -(\ (\mu_2 - \mu_1\ )' * \sum^{-1} * x + \frac{1}{2}\mu_1' * \sum^{-1} * \mu_1 + \frac{1}{2} * \mu_2' * \sum^{-1} * \mu_2 - \ln\left(\frac{1-p}{p}\right))}$$

$$= \frac{1}{1 + \exp(\ -\theta' x + k)}$$

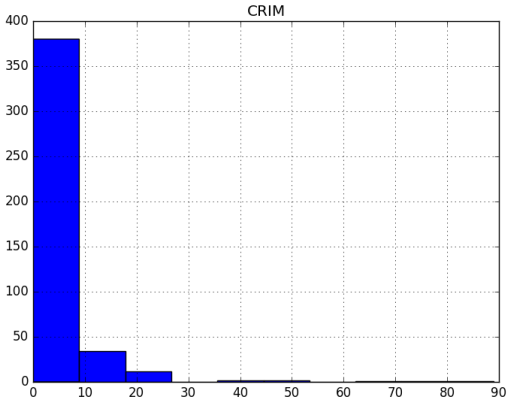
Where $\theta' = (\mu_2 - \mu_1\ )' * \sum^{-1}$ and

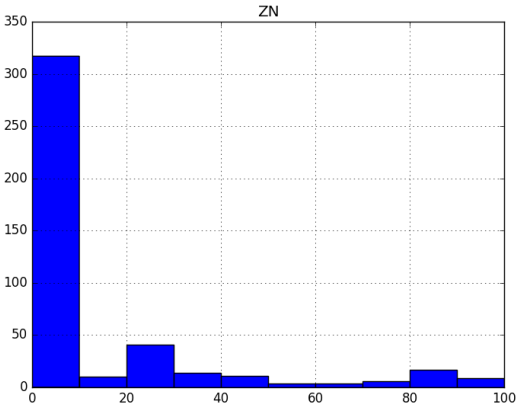
$$k = -\frac{1}{2}\mu_1' * \sum^{-1} * \mu_1 + \frac{1}{2} * \mu_2' * \sum^{-1} * \mu_2 - ln\left(\frac{1-p}{p}\right)$$

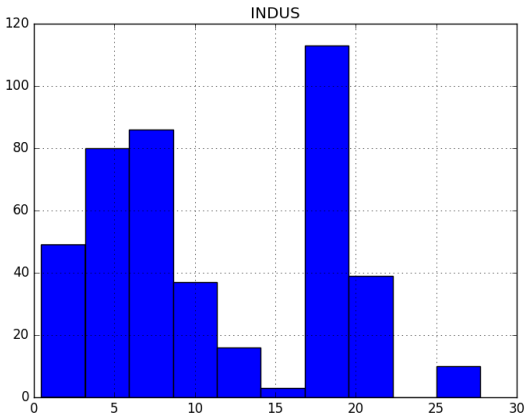
3. Programming

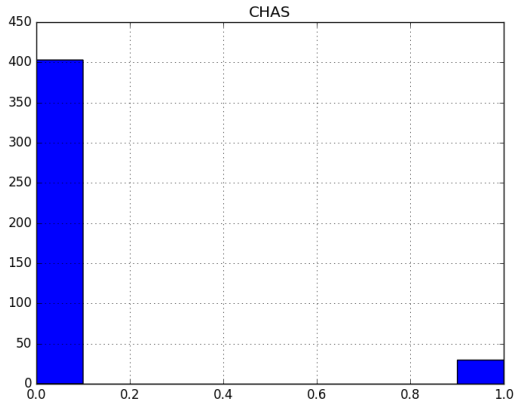
In this assignment we use the Boston Housing Dataset for running our regression algorithms and predict the median value and calculate the mean square errors.

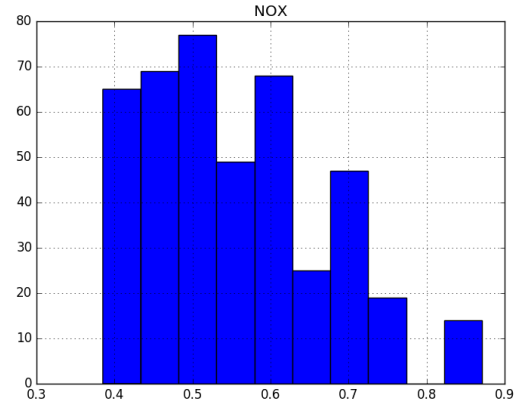
3.1 Histograms

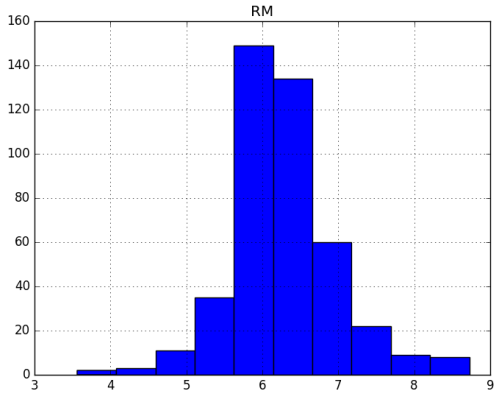


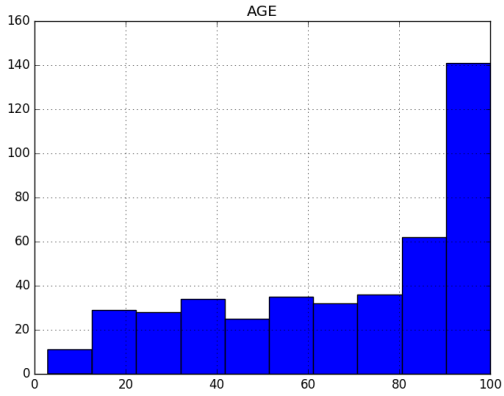


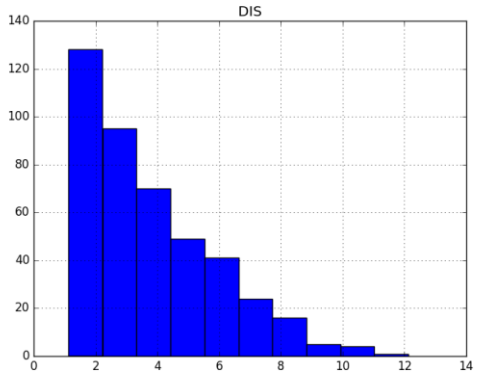


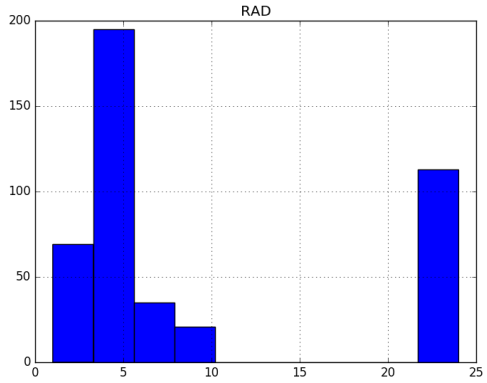

CHAS
450
400
350
300
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0

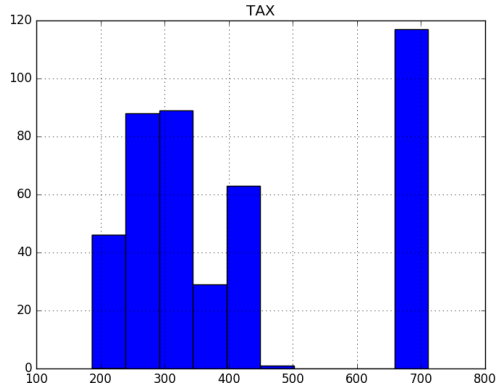

NOX
80
70
60
50
40
30
20
10
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9

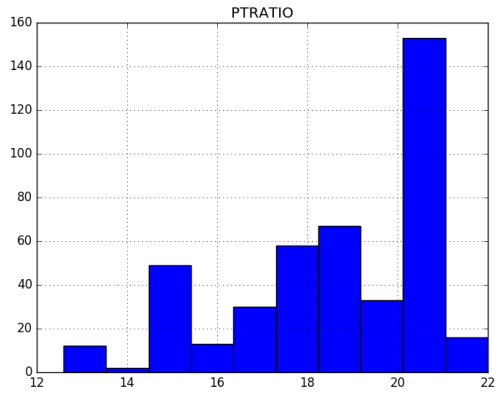

RM
160
140
120
100
80
60
40
20
0
3
4
5
6
7
8
9

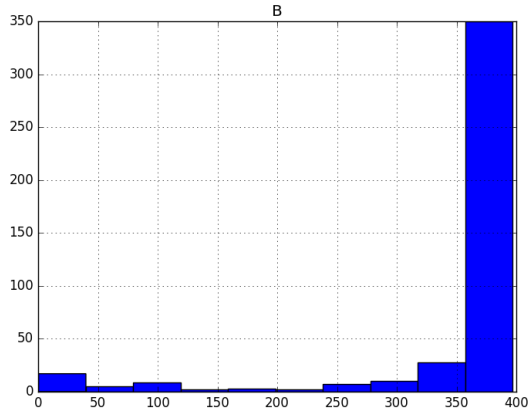

AGE
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100

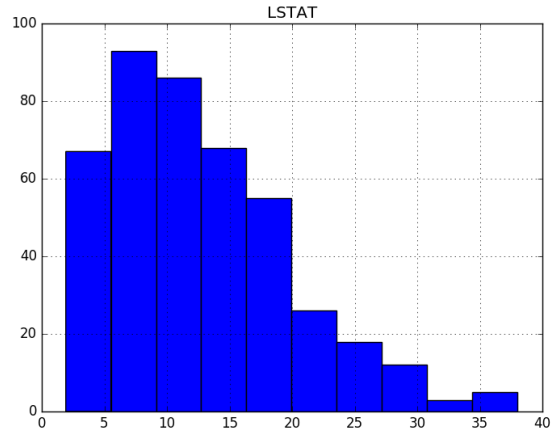

DIS
140
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14


RAD
200
150
100
50
0
0
5
10
15
20
25


TAX
120
100
80
60
40
20
0
100
200
300
400
500
600
700
800


PTRATIO
160
140
120
100
80
60
40
20
0
12
14
16
18
20
22





3.2 Linear Regression and Ridge Regression

As per the given instructions when we run the train data (433 values) against the test data (73 values) we get the following mean square errors' for linear and ridge regression

Algorithm	Mean Square Error for Train vs Train	Mean Square Error for Test vs Train
Linear Regression	20.950144508	28.4179164975
Ridge Regression, $\lambda = 0.01$	20.9501449007	28.4182927619
Ridge Regression, $\lambda = 0.1$	20.9501837112	28.4216969435
Ridge Regression, $\lambda = 1$	20.9539971078	28.4574903672

Here, we can see that there not much of a difference between the MSEs for the test vs train run. This is because the training and test data size are small. Ideally, with the increase in λ the MSE values gradually decrease till a certain point (minima) after which they increase

again. So, by performing 10 fold cross validation for the train dataset we can see how the value of MSE varies for variation in λ.

Ridge Regression with Cross Validation

λ value for Ridge Regression	Mean Square Error for Train vs Train	Mean Square Error for Test vs Train
0.0001	20.8560255283	22.832775315
0.001	20.8560255331	22.8327662748
0.01	20.8560260144	22.8326763226
0.1	20.8560740408	22.8318216626
1	20.8607832677	22.8276600052
10	21.2672908638	23.1560487313

For the above values, the training dataset was divided into 10 bins, out of which 9 were used as training set and the other one was used as the test set. This was repeated by taking different bins as test data and finally the mean of both the train vs train and test vs train MSEs were calculated. Here we can see that the MSE for train vs train reduces till λ= 0.01 and then increases with the increase in lambda values.

Ridge Regression for different values of λ

When we take the entire dataset given and run ridge regression for different values of lambda we get the following values for MSEs.

λ value for Ridge Regression	Mean Square Error for Train vs Train	Mean Square Error for Test vs Train
0.0001	20.950144508	28.4179202582
0.001	20.9501445119	28.4179541062
0.01	20.9501449007	28.4182927619
0.1	20.9501837112	28.4216969435
1	20.9539971078	28.4574903672
10	21.2871171137	28.9854896987

Overall we get to see that Ridge Regression with regularization term λ performs better and avoids over fitting of the given data.

3.3 Feature Selection

The Pearson Coefficients for the columns are:

Column Name	Pearson Coefficient
CRIM	-0.387697
ZN	0.362987
INDUS	-0.483067
CHAS	0.203600
NOX	-0.424830
RM	0.690923
AGE	-0.390179
DIS	0.252421
RAD	-0.385492
TAX	-0.468849
PTRATIO	-0.505271
B	0.343434
LSTAT	-0.739970

(a) The 4 features selected based of Pearson Coefficients are:

Columns Selected: ['LSTAT', 'RM', 'PTRATIO', 'INDUS']

For these values as the test vs train data and train vs train data, we get:

Train Mean Square Error = 26.4066042155

Test Mean Square Error = 31.4962025449

(b) The values for MSE on train vs train data and test vs train data after using residue values,

Columns Selected: ['LSTAT', 'RM', 'PTRATIO', 'CHAS']

Train Mean Square Error = 25.1060222464

Test Mean Square Error = 34.6000723135

(c) The values for MSE on train vs train data and test vs train data using Brute Force Search,

Columns Selected: ['CHAS', 'RM', 'PTRATIO', 'LSTAT']

Train Mean Square Error = 25.1060222464

Test Mean Square Error = 34.6000723135

In feature selection we see that we get the best values for MSE for both train vs train data and test vs train data when we consider the residue values which improves the performance. We also see that the values for the brute force approach is same as the values obtained for feature selection with residue. This confirms that Feature selection with residue gives the best MSE as it has the same values as given by Brute Force method and Brute Force Search feature selection even though is an expensive task it always gives the best MSE values that is the least MSE.

3.4 Polynomial Feature Expansion

When we run the linear regressor on our data after performing polynomial expansion on the train and test data, we get the following MSEs for train vs train data and test vs train data:

Train Mean Square Error = 5.05978429711

Test Mean Square Error = 14.555304972

COLLABORATION

Brain stormed and Collaborated with Adarsha Desai and Ravishankar Sivaraman for this assignment.