1. **Logistic Regression**
   a. **Binary logistic regression model**

Probability of a single training sample (x_i, y_i) is,

$$P(Y = y_i \,|X = x_i; b; w) = \begin{cases} \sigma(b + w^T x_i) & if\ y_i = 1 \\ 1 - \sigma(b + w^T x_i) & otherwise \end{cases}$$

A compact expression to represent this can be written as,

$$P(Y = y_i \,|X = x_i; b; w) = (\sigma(b + w^T x_i))^{y_i}[1 - \sigma(b + w^T x_i)]^{1-y_i}$$

Now we have the negative log likelihood,

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

Which can be written as the cross entropy error by combining the above two equations,

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

   b. **Gradient descent method**

For the sake of convenience, we can represent

$$x \leftarrow [\, 1\; x_1\; x_2\; x_3 \ldots \; x_n]$$

And

$$w \leftarrow [\, b\; w_1\; w_2\; w_3 \ldots \; w_n]$$

Then, the cross entropy error function becomes,

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

The gradient of the cross entropy error function is,

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

Here e_i = $\{\sigma(\, w^T x_i) - y_i\}$ is called the error for the i^{th} training sample

Choosing a proper step size $\eta > 0$ for gradient descent and iteratively updating the parameters following the negative gradient to minimize the error function

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \sum_{i=1}^{n} \{\sigma(\, w^T x_i) - \; y_i\} x_i$$

The solution will converge to a global minimum if $\frac{\delta^2 \varepsilon(w)}{\delta w^2} \geq 0$

$$\frac{\delta^2 \varepsilon(w)}{\delta w^2} = \sum_{i=1}^{n} \{\sigma(\, w^T x_i) - \; y_i\} x_i$$

$$\frac{\delta \varepsilon(w)}{\delta w} = \frac{\delta}{\delta w} (\sum_{i=1}^{n} \{\sigma(\, w^T x_i) - \; y_i\} x_i \,)$$

$$\frac{\delta \varepsilon(w)}{\delta w} = (\sum_{i=1}^{n} \sigma(\, w^T x_i)(1 - \; \sigma(\, w^T x_i)) {x_i}^2 \,)$$

Since $\sigma(\, w^T x_i) \; \in [0,1]$ and ${x_i}^2 \geq 0$, we can conclude that

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

Therefore, the solution will converge to a global maximum if the step size η is chosen properly. A very small η can take a long time to converge and a very large η will result in the values not converging.

c. Negative log likelihood $L(\boldsymbol{w_1}, \dots \boldsymbol{w_k})$ 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,

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

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

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

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

Hence, 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 log 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. Gaussian Discriminant Analysis

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,

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

Or ,

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

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

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

For MLE to maximize $\mathcal{L}(D)$, we have to take the derivatives w.r.t to each parameter and equate each to zero.

So, we get,

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

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

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

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

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

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

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

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

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

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

For $\hat{P}_1$ and $\hat{P}_2$ we use Lagrange Multiplier to expand P_1 and P_2 using the property $P_1 + P_2 = 1$. Substituting this 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

$$\boxed{\hat{P}_1 = \frac{n_1}{n_1 + n_2} \qquad \hat{P}_2 = \frac{n_2}{n_1 + n_2}}$$

b. Multivariate Gaussian distribution

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

And $\mu_1, \mu_2 \in R^D, \Sigma \in R^{DXD}$.

Now consider $c_1 = 1 \; and \; c_2 = 0$, therefore, we have

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

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

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

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

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

Let, P(Y=1) = p, and so P(Y=0)=(1-p)

Substituting these values in the above equations

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

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

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

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

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

We can write $\frac{1-p}{p}$ using ln as exp(ln($\frac{1-p}{p}$)).

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

Consider the equation for single dimensional variable. We can solve the above given denominator part as

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

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

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

Using the above equation in denominator of (1), we get

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

=

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

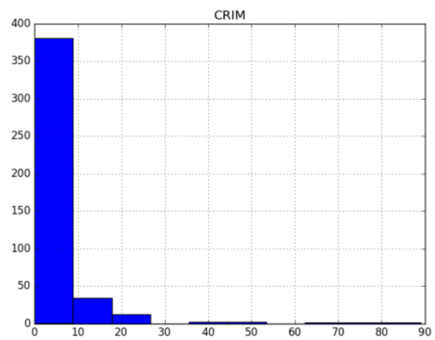
Where $\theta' = (\mu_2 - \mu_1\,)' * \Sigma^{-1}$ and k= $-\frac{1}{2}\mu_1' * \Sigma^{-1} * \mu_1 + \frac{1}{2} * \mu_2' * \Sigma^{-1} * \mu_2 - \ln\left(\frac{1-p}{p}\right)$

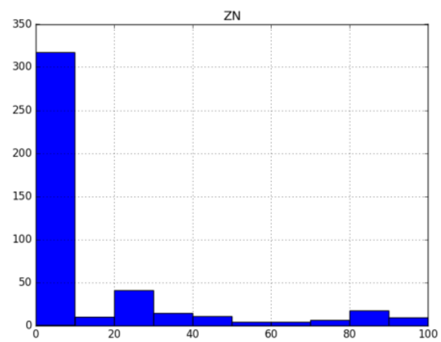
3. Programming

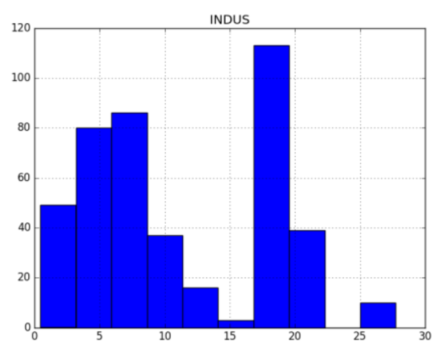
3.1 Data set

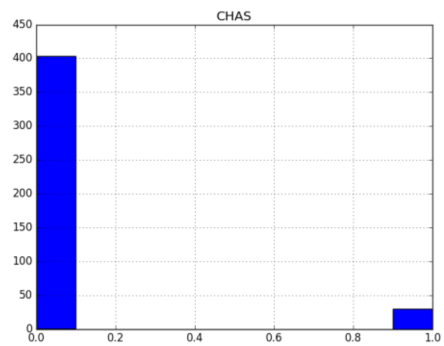
Data Analysis :

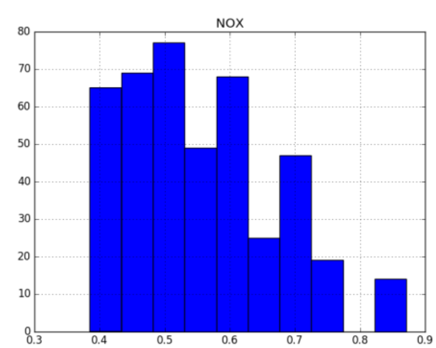
Histograms of all the features are as follows:

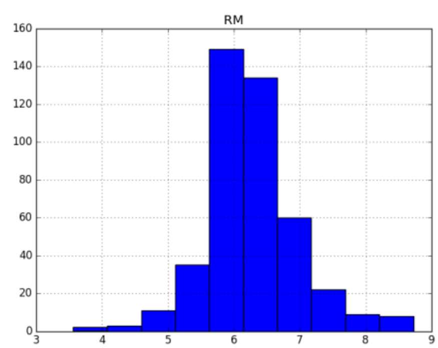


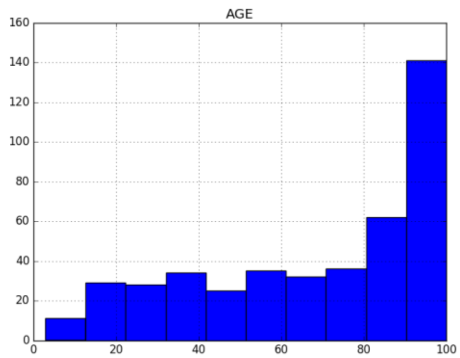


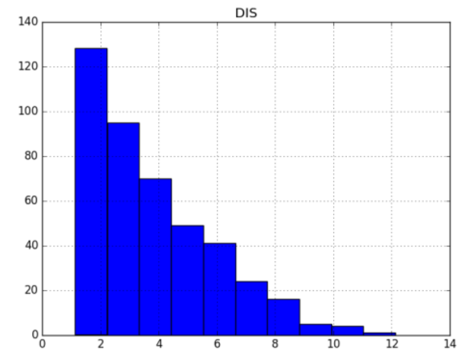


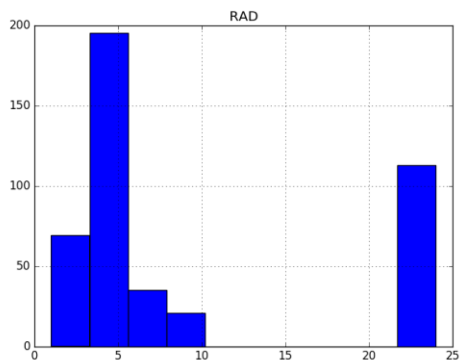


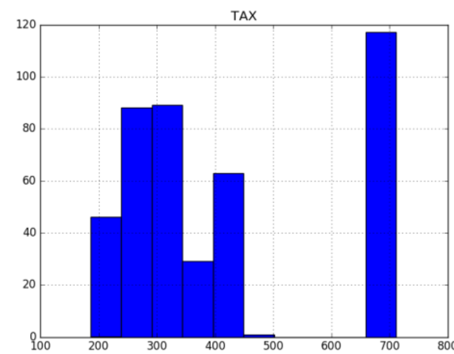


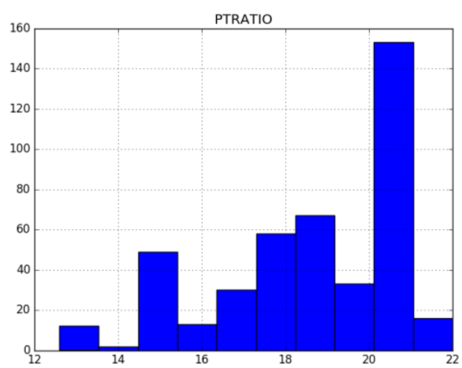


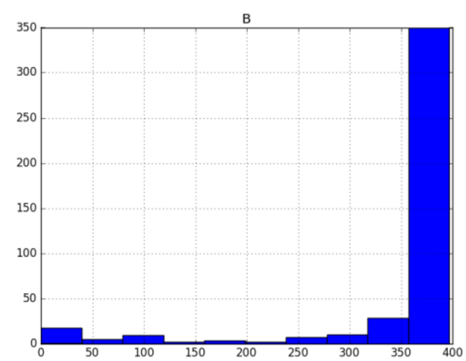

AGE
0 20 40 60 80 100
0 20 40 60 80 100 120 140 160

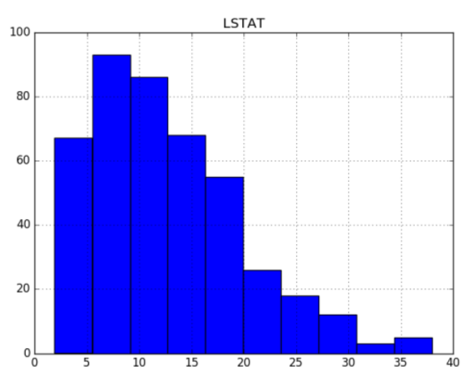

DIS
0 2 4 6 8 10 12 14
0 20 40 60 80 100 120 140


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


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


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


B
0 50 100 150 200 250 300 350 400
0 50 100 150 200 250 300 350


LSTAT
0 5 10 15 20 25 30 35 40
0 20 40 60 80 100

3.2 Linear Regression

a. Linear regression

The Mean Square Error (MSE) for the linear regressor are

Training data V/s	MSE
Training data	20.95014451
Testing data	28.4179165

The linear regressor provides better accuracy when running the regressor for Training data against training data as opposed to Training data against testing data as can be seen from the above results.

b. Ridge Regression

lambda	Training data V/s	MSE
0.01	Training data	20.9501449
	Testing data	28.41829276
0.1	Training data	20.95018371
	Testing data	28.42169694
1	Training data	20.95399711
	Testing data	28.45749037

The ridge regressor provides better accuracy when running the regressor for Training data against training data as opposed to Training data against testing data as can be seen from the above results.

Provided large enough data, ridge regression will outperform linear regression as it will prevent overfitting of data. Though this cannot be noticed in the above results.

Without cross validation, the MSE values are increasing as the values of ‘lambda’ increase

c. Ridge Regression with cross validation

lambda	Training data V/s	MSE
0.0001	Training data	22.806327
	Testing data	28.41792
0.001	Training data	22.806323
	Testing data	28.417954
0.01	Training data	22.806282
	Testing data	28.418293
0.1	Training data	22.805918
	Testing data	28.421697
1	Training data	22.806524
	Testing data	28.45749
10	Training data	23.172236
	Testing data	28.98549

10-fold Cross validation provides a more accurate version of ridge regression. Also, since in most real world scenarios wouldn't present you with a distinguished training and testing. Also for this reason the cross validation give a more accurate representation of now the MSE (Training vs Training) varies with varying lambda values. The MSE initially decreases and then increases.

Comparing this to the MSE for Training vs Testing data shows how testing data (without cross validation) only increases monotonically.

The best lamda values are ususally around [0.1, 1.0] varying as different sets of randomized data is picked. As you can see in this case, the best lambda value is 0.1 with a MSE of 22.805918

3.3 Feature Selection

a. Picking Four features with highest Pearson's correlation coefficients

The features with highest (absolute) pearsons coefficients are ['INDUS', 'RM', 'PTRATIO', 'LSTAT']

Training data V/s	MSE
Training data	26.40660422
Testing data	31.49620254

Above are the linear regression results for Training VS Training and Training Vs testing data.

b. Picking Four features with highest pearsons correlation with residue

The four best features selected are ['CHAS', 'RM', 'PTRATIO', 'LSTAT']

Training data V/s	MSE
Training data	25.10602225
Testing data	34.60007231

c. Brute force Search

The four best features selected in brute force search are ['CHAS', 'RM', 'PTRATIO', 'LSTAT']

Training data V/s	MSE
Training data	25.10602225
Testing data	34.60007231

The correlated residue gives better results than that with the highest pearsons coeffieicents as can be seen with the MSE of training data.

Since we know that brute force search always returns the optimum, we can see that the results of brute force search reinforces the optimality of residual correlation. Since the cost of running selection based on residual correlation is better than brute force search, we can conclude that it is the best approach.

3.4 Feature Expansion

The results of feature expansion through polynomial expansion of the features are as follows.

Training data V/s	MSE
Training data	5.059784297
Testing data	14.55530497

Collaboration

Collaborated on thoughts and ideas with **Adarsha Desai** and **Mahesh Pottippala Subrahmanya**