Machine Learning – ES 654

Spring 2019

IITGN

1Ans.

Machine Learning

Homework 4:

1Ans:-

(a) To learn θ using coordinate descent

Init: $\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

To Show: Calculations for first 3 iterations.

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \end{bmatrix}, \quad y = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$$

Iteration-1:

Optimizing along θ_0.

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$x[:,0] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

we know, $\theta_0 = \frac{\rho_0}{z_0}$, where $\rho_0 = \sum_{i=1}^{N} x_i^j \left(y_i - \hat{y}_i^{(-\theta)}\right)$ & $z_0 = \sum_{i=1}^{N} (x_i^0)^2$

$$\Rightarrow z_0 = 1^2 + 1^2 + 1^2 = 3$$

$$\sum_{i=1}^{N} \hat{y}_i^{-0} = x * \cancel{theta}\, \theta - x[:,0] * \theta[0]$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \sum_i y_i - \hat{y}_i^{-0} = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$$

$$\Rightarrow \rho_j = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix} = 32$$

$$\Rightarrow \theta_0 = \frac{32}{3} = 10.666\bar{6}\ldots$$

Iteration - 2 :-

$$\theta = \begin{bmatrix} 10.6667 \\ 0 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

Optimising along θ_1

$$X[:,1] = \begin{bmatrix} 1 \\ 3 \\ 6 \end{bmatrix}, \quad y = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$$

$$\sum_i \hat{y}_i^{-1} = X^* \theta - X[:,1]^* \theta[1]$$

$$= X^* \theta \qquad (\because \theta[1] = 0)$$

$$= [10.6667 \quad 10.6667 \quad 10.6667]^T$$

$$\sum_i y_i - \sum_i \hat{y}_i^{-1} = \begin{bmatrix} -4.6667 \\ -0.6667 \\ 5.3333 \end{bmatrix}$$

$$\rho_1 = [1 \ 3 \ 6] \begin{bmatrix} -4.6667 \\ -0.6667 \\ 5.333 \end{bmatrix} = 25.333\ldots$$

$$z_1 = \sum_i (x_i^1)^2 = 1^2 + 3^2 + 6^2 = 46$$

$$\Rightarrow \theta_1 = 0.55072$$

Iteration - 3 :-

$$\theta = \begin{bmatrix} 10.6667 \\ 0.55072 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

Optimising along θ_0

$$X[:,0] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad y = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$$

$$\sum_i \hat{y}_i^{-0} = X^* \theta - X[:,0]^* \theta[0]$$

$$= \begin{bmatrix} 11.21739 \\ 12.31884 \\ 13.971014 \end{bmatrix} - \begin{bmatrix} 10.6667 \\ 10.6667 \\ 10.6667 \end{bmatrix} = \begin{bmatrix} 0.55072 \\ 1.65217 \\ 3.30434 \end{bmatrix}$$

$$\sum_i y_i - \hat{y}_i^{-0} = \begin{bmatrix} 5.44927 \\ 8.347826 \\ 12.6956521 \end{bmatrix}$$

$$\delta_0 = [1 \ 1 \ 1] \begin{bmatrix} 5.44927 \\ 8.347826 \\ 12.69565 \end{bmatrix} = 26.49275$$

$$Z_0 = 1^2 + 1^2 + 1^2 = 3$$

$$\Rightarrow \theta_0 = [8.83091787]$$

$$\therefore \quad \theta = \begin{bmatrix} 8.830917 \\ 0.55072464 \end{bmatrix}$$

(b) <u>To Show</u>:- Calculations for SGD for 1 epoch.

INIT: $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\alpha = 0.01$

$X = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \end{bmatrix}$, $y = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$

<u>Iteration-1</u> :- Choose 1^{st} training example $X[0,:] = [1, 1]$

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$y[0] = [6]$, $\Rightarrow$ $y[0] - X[0,:] * \theta = 6$

$$\theta := \theta + 2 * \alpha * (y[0] - X[0,:]) * X[0,:]^T$$

$$:= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2 \times 0.01 \times 6 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.12 \\ 0.12 \end{bmatrix}$$

<u>Iteration-2</u> :- Choose 2^{nd} training example, $X[1,:] = [1, 3]$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 0.12 \\ 0.12 \end{bmatrix}$$

$$y[1] - x[1,:]^{*}\theta = 10 - [1 \;\; 3]\begin{bmatrix}0.12\\0.12\end{bmatrix} = 9.52$$

$$\Rightarrow \quad \theta := \theta - 2 * \alpha * (y[1] - x[1,:]^{*}\theta) * x[1,:]^{T}$$

$$:= \begin{bmatrix}0.12\\0.12\end{bmatrix} - 2 \times 0.01 \times 9.52 \times \begin{bmatrix}1\\3\end{bmatrix}$$

$$= \begin{bmatrix}0.3104\\0.6912\end{bmatrix}$$

Iteration - 3 :- Choose 3^{rd} training example $x[2;:] = [1 \;\; 6]$

$$\theta = \begin{bmatrix}0.3104\\6.912\end{bmatrix}$$

$$y[2] - x[2,:]^{*}\theta] = 16 - \cancel{[16]}[1 \quad 6]\begin{bmatrix}0.3104\\6.912\end{bmatrix}$$

$$= 11.5424$$

$$\theta := \theta - 2 * \alpha * (y[2] - x[2;:]^{*}\theta) * x[2;:]^{T}$$

$$= \begin{bmatrix}0.3104\\6.912\end{bmatrix} - 2 \times 0.01 \times 11.5424 \begin{bmatrix}1\\6\end{bmatrix}$$

$$= \begin{bmatrix}0.541248\\2.0762888\end{bmatrix}$$

$$\Rightarrow \quad \theta = \begin{bmatrix}0.541248\\2.0762888\end{bmatrix}$$

(c) To Learn θ using Normal equation for Ridege regression with $\lambda = 1$.

$$X = \begin{bmatrix}1 & 1\\1 & 3\\1 & 6\end{bmatrix}, \quad y = \begin{bmatrix}6\\10\\16\end{bmatrix}, \quad \lambda = 1$$

$$\theta = (X^{T}X + \lambda I)^{-1} X^{T} y.$$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 6 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \end{pmatrix} = \begin{bmatrix} 3 & 10 \\ 10 & 46 \end{bmatrix}$$

$$X^T X + \lambda I = \begin{bmatrix} 3 & 10 \\ 10 & 46 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 10 \\ 10 & 47 \end{bmatrix}$$

$$(X^T X + \lambda I)^{-1} = \frac{\begin{bmatrix} 47 & -10 \\ -10 & 4 \end{bmatrix}}{188 - 100} = \frac{1}{88} \begin{bmatrix} 47 & -10 \\ -10 & 4 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 6 \end{bmatrix} \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix} = \begin{bmatrix} 32 \\ 132 \end{bmatrix}$$

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

$$= \frac{1}{88} \begin{bmatrix} 47 & -10 \\ -10 & 4 \end{bmatrix} \begin{bmatrix} 32 \\ 132 \end{bmatrix} = \frac{\begin{bmatrix} 184 \\ 208 \end{bmatrix}}{88}$$

$$= \begin{bmatrix} 2.0909 \\ 2.3636 \end{bmatrix}$$

Rough work

47×32: 94, 141, 1504, 1320, 184

~~128~~ 528, 320, 208

2Ans. Custom linear regression implementations

- **a)** Here is the code for regularised normal equation
- **b)** Here is the code for coordinate descent regression
- **c)** Here is the code for coordinate descent lasso regression
- **d)** Here is the code for stochastic gradient descent
- **e)** Here is the code for gradient descent Lasso using autograd

3Ans.

(a) Here is the code for matplotlib animation for stochastic gradient descent

(b) Here is the code for matplotlib animation for coordinate descent

4Ans.

Part-(a): Here is the code for the scikit-learn L2 regularized model

Part-(b): Yes we can now learn the coefficients for the given data, because the matrix (X.T*X+lambda*I) is invertible. Previous time we got the matrix X.T*X as non invertible because of the absence of lambda. Here is the code for the same. Below is the calculation for coefficients.

4 Ans.:- Yes we can calculate the coefficients now

(b)

$$X = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 4 \\ 1 & 3 & 6 \\ 1 & 4 & 8 \end{bmatrix}, \quad y = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}, \quad \lambda = 1.$$

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

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 4 \\ 1 & 3 & 6 \\ 1 & 4 & 8 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 20 \\ 10 & 30 & 60 \\ 20 & 60 & 120 \end{bmatrix}$$

$$(X^T X + \lambda I) = \begin{bmatrix} 5 & 10 & 20 \\ 10 & 31 & 60 \\ 20 & 60 & 121 \end{bmatrix}, \quad X^T y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 14 \\ 40 \\ 80 \end{bmatrix}$$

$$(X^T X + \lambda I)^{-1} = \begin{bmatrix} 0.592 & -0.0392 & -0.0784 \\ -0.039 & 0.803 & -0.3921 \\ -0.0784 & -0.3921 & 0.2156 \end{bmatrix}$$

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

$$= \begin{bmatrix} 0.592 & -0.0392 & -0.0784 \\ -0.039 & 0.803 & -0.3921 \\ -0.0784 & -0.3921 & 0.2156 \end{bmatrix} \begin{bmatrix} 14 \\ 40 \\ 80 \end{bmatrix}$$

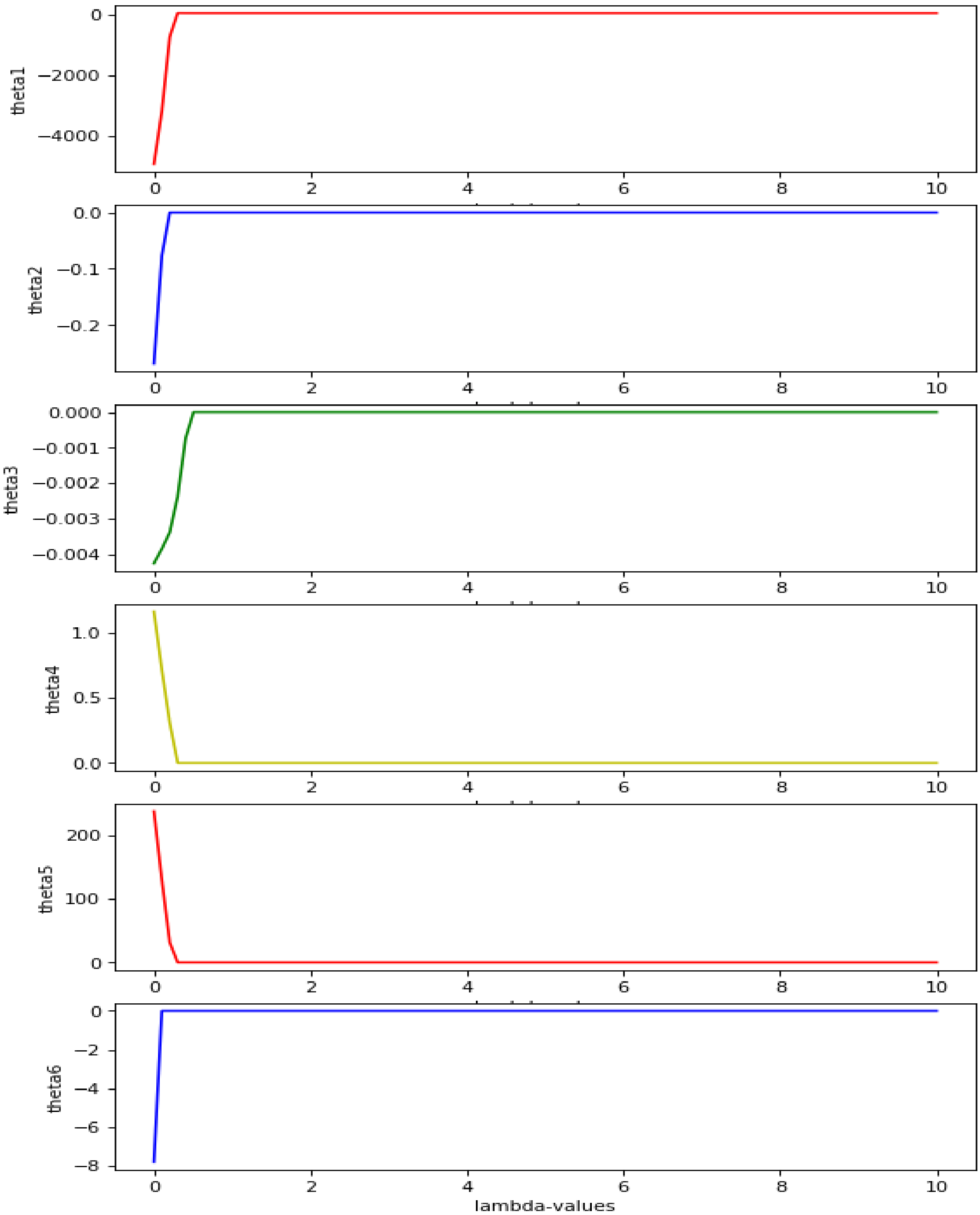
$$\Theta = \begin{bmatrix} 0.44705 \\ 0.23529 \\ 0.47058 \end{bmatrix}$$

5Ans.

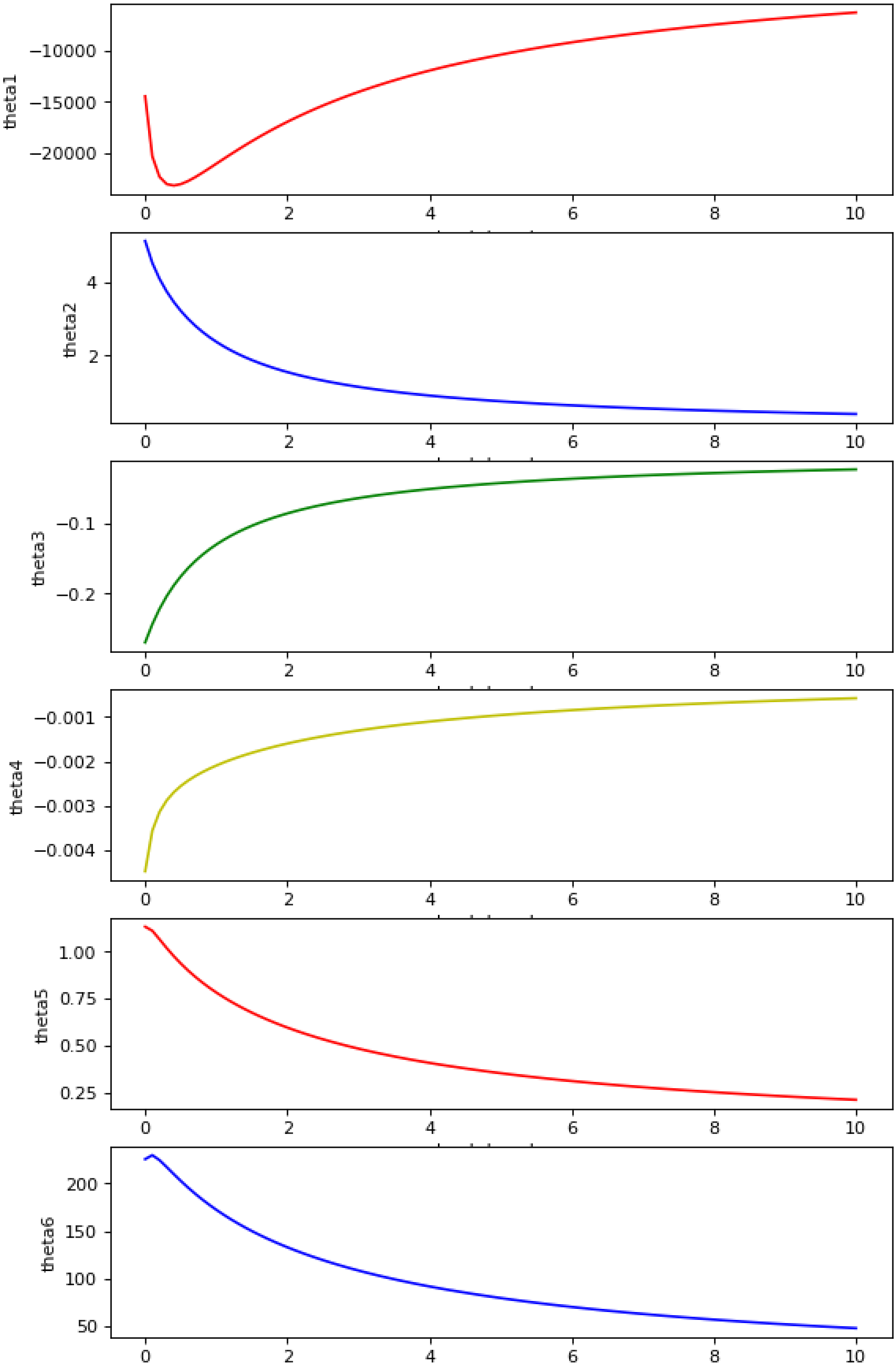
a) Here is the code for the 5-fold cross validation for Ridge regression

b) Here is the code for the 5-fold cross validation for Lasso regression

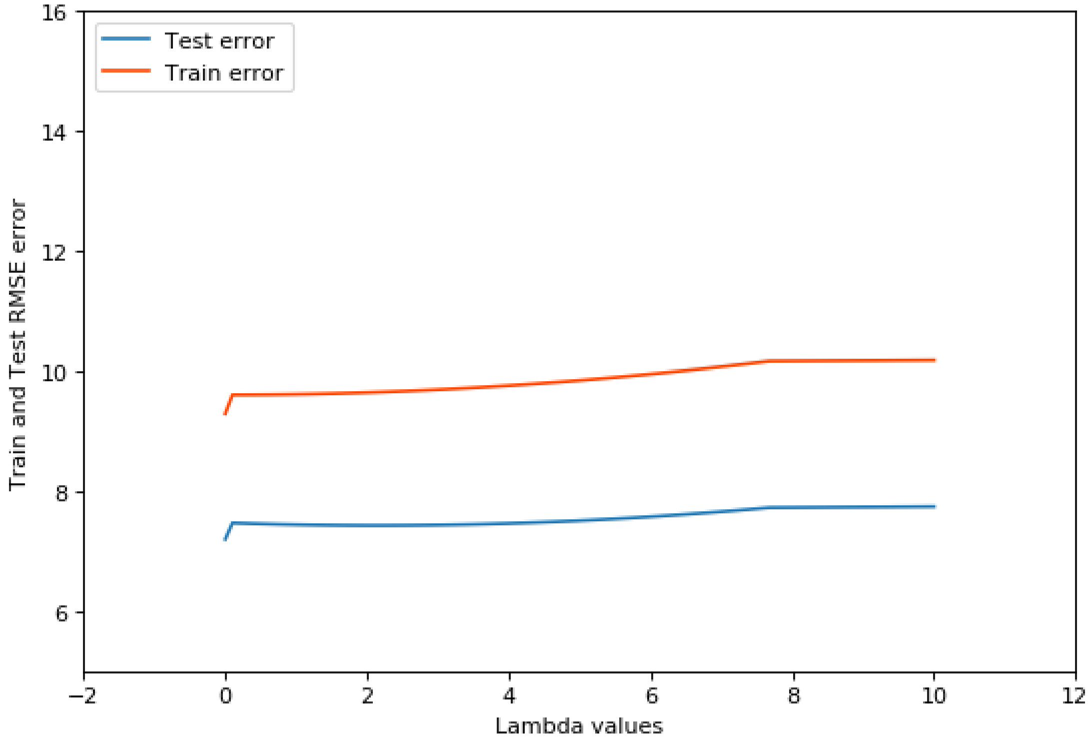
c) Here is the code for the lasso regression's regularisation path. Here is the code for the Ridge regression's regularisation path. Below is the picture with regularisation paths for the Lasso regression. It is observed that incase in case of Lasso regression we have all the thetas converging to the zero making the theta vector too sparse.



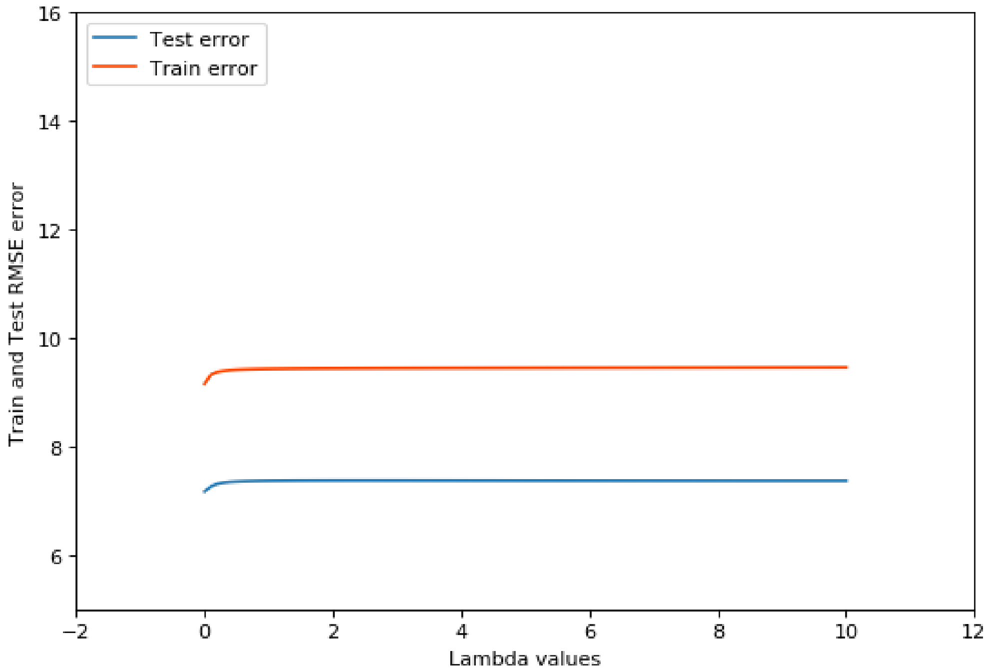
Below is the picture for regularisation paths for Ridge regression.


theta1
theta2
theta3
theta4
theta5
theta6
lambda-values

d) Here is the code for the Lasso train and test RMSE error. Here is the code for the Ridge regression train and test error. Below is the picture for train and test error for Lasso regression.



Below is the picture for train and test error for Ridge regression.



References:

- Linear regression with prior (using gradient descent), Nipun Batra 2019
  Avaliable at: https://nipunbatra.github.io/blog/2017/linear-regression-prior.html [Accessed 18 Feb. 2019].