CSCE 633 Assignment 1

Tang Yunzhi

September 23, 2019

1 Problem 1

I have mistakenly understand the question until the last day. I found out the last xd does not have a coefficient so the entire code for question 1.1-1.2 is not fit in this senario. Can I at least get partial credit?

(1) This part code can be found in hw1q1.py
The residual sum of squares $RSS = e_1^2 + e_2^2 + + e_n^2 = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$

when d=0, $y = a_0$, we have 4 trainning data points:(0,1)(2,4)(3,9)(5,16)
$RSS = (y_1 - \alpha_0)^2 + (y_2 - \alpha_0)^2 + (y_3 - \alpha_0)^2 + (y_4 - \alpha_0)^2$
$= (1 - \alpha_0)^2 + (4 - \alpha_0)^2 + (9 - \alpha_0)^2 + (16 - \alpha_0)^2$
$= (\alpha_0^2 + 2\alpha_0 + 1) + (\alpha_0^2 + 8\alpha_0 + 16) + (\alpha_0^2 + 18\alpha_0 + 81) + (\alpha_0^2 + 32\alpha_0 + 144)$
$= 4\alpha_0^2 + 60\alpha_0 + 242$

now we solve $\frac{\partial RSS}{\partial \hat{\alpha}_0} = 0$
$\frac{\partial RSS}{\partial \hat{\alpha}_0} = 8\alpha_0 + 60 = 0$
so $\alpha_0 = 7.5$

when d=1,$y = a_0 + a_1 x$, we have 4 trainning data points:(0,1)(2,4)(3,9)(5,16)
$RSS = (y_1 - \alpha_0 - \alpha_1 x_1)^2 + (y_2 - \alpha_0 - \alpha_1 x_2)^2 + (y_3 - \alpha_0 - \alpha_1 x_3)^2 + (y_4 - \alpha_0 - \alpha_1 x_4)^2$
$= (1 - \alpha_0 - a_1 * 0)^2 + (4 - \alpha_0 - a_1 * 2)^2 + (9 - \alpha_0 - a_1 * 3)^2 + (16 - \alpha_0 - a_1 * 5)^2$

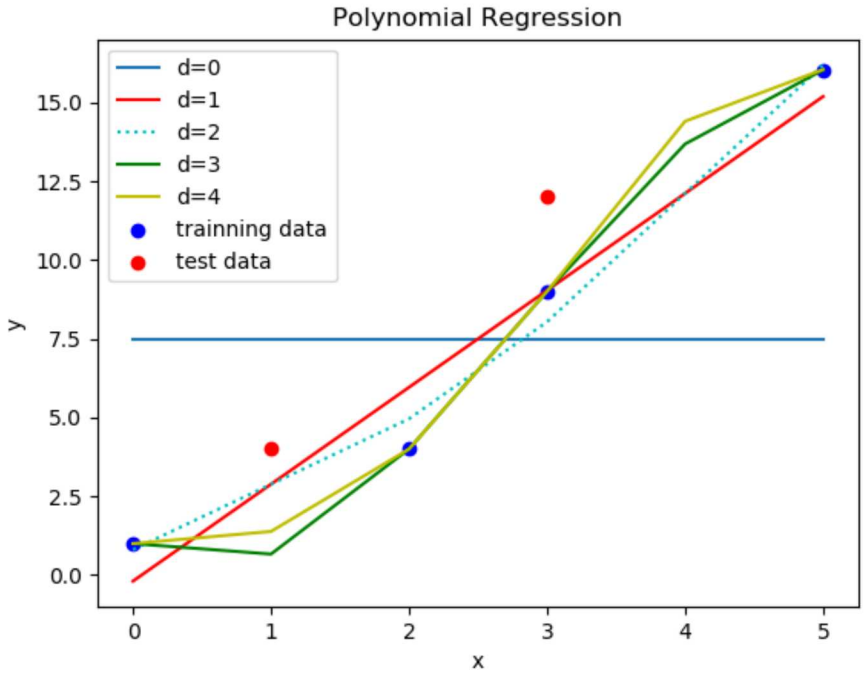
since RSS=$\sum_{i=1}^{n}(y_1 - \alpha_0 - \alpha_1 x_1)^2 = -2n\overline{y} + 2n\hat{\alpha_0} + 2n\hat{\alpha_1}\overline{x}$
$\hat{\alpha_0} = \overline{y} - \hat{\alpha_1}\overline{x}$
From $\frac{\partial RSS}{\partial \hat{\alpha}_1} = 0$
$\hat{\alpha}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$

we can get avg(x)=$\frac{0+2+3+5}{4} = 2.5$
avg(y)=$\frac{1+4+9+16}{4} = 7.5$
$\hat{\alpha}_1 = \frac{(0-2.5)*(1-7.5)+(2-2.5)*(4-7.5)+(3-2.5)*(9-7.5)+(5-2.5)*(16-7.5)}{(0-2.5)^2+(2-2.5)^2+(3-2.5)^2+(5-2.5)^2}$
=3.077
so $\alpha_0 = 7.5 - 2.5 * 3.077 = -0.1925$

Then we use python library skylearn to simulate the polynomial regression:

```
d=1 coefficients: [3.07692308]
d=1 a0: -0.19230769230769074
d=2 coefficients: [1.41025641 0.33333333]
d=2 a0: 0.8076923076923066
d=3 coefficients: [-2.83333333  2.83333333 -0.33333333]
d=3 a0: 0.9999999999999964
d=4 coefficients: [-0.13965341  0.04986408  0.56455997 -0.08978933]
d=4 a0: 0.9999999999999938
```

This verify that I have calculated the right anwser and find out when d=2,$\alpha_0 = 0.808, \alpha_1 = 1.410, \alpha_2 = 0.333$
When d=3:$\alpha_0 = 0.99999, \alpha_1 = -2.833, \alpha_2 = 2.833\alpha_3 = -0.333$
When d=4:$\alpha_0 = 0.99999, \alpha_1 = -0.1396, \alpha_2 = 0.04986\alpha_3 = 0.5645\alpha_4 = -0.0897$
$whend = 0, y = 7.5$
$whend = 1, y = 5.000000 + x$
$whend = 2, y = 2.807692 + -1.923077 * x + x^2$
$whend = 3, y = 0.230769 + 14.141026 * x + -7.166667 * x^2 + x^3$
$whend = 4, y = 1.000000 + -32.833333 * x + 33.833333 * x^2 + -10.333333 * x^3 + x^4$



(2)This part code can be found in hw1(2).py $bias^2 = \sum_1^i (y_i - \hat{y}_i)$ use training points
$variance = \sum_1^i (y_i - \hat{y}_i)$ use test point
Total error = $bias^2 + variance$
Trainning data points:(0,1)(2,4)(3,9)(5,16) and test data points:(1,3),(4,12)

For d=0,y=7.5,
$variance = (3-7.5)^2 + (12-7.5)^2 = 40.5$
$bias^2 = (1-7.5)^2 + (4-7.5)^2 + (9-7.5)^2 + (16-7.5)^2 = 73.75$
Total error= $bias^2 + variance$ =114.25
Trainning error= $\frac{1}{n} * bias^2 = 18.4375$
Test error= $\frac{1}{n} * variance = 20.25$

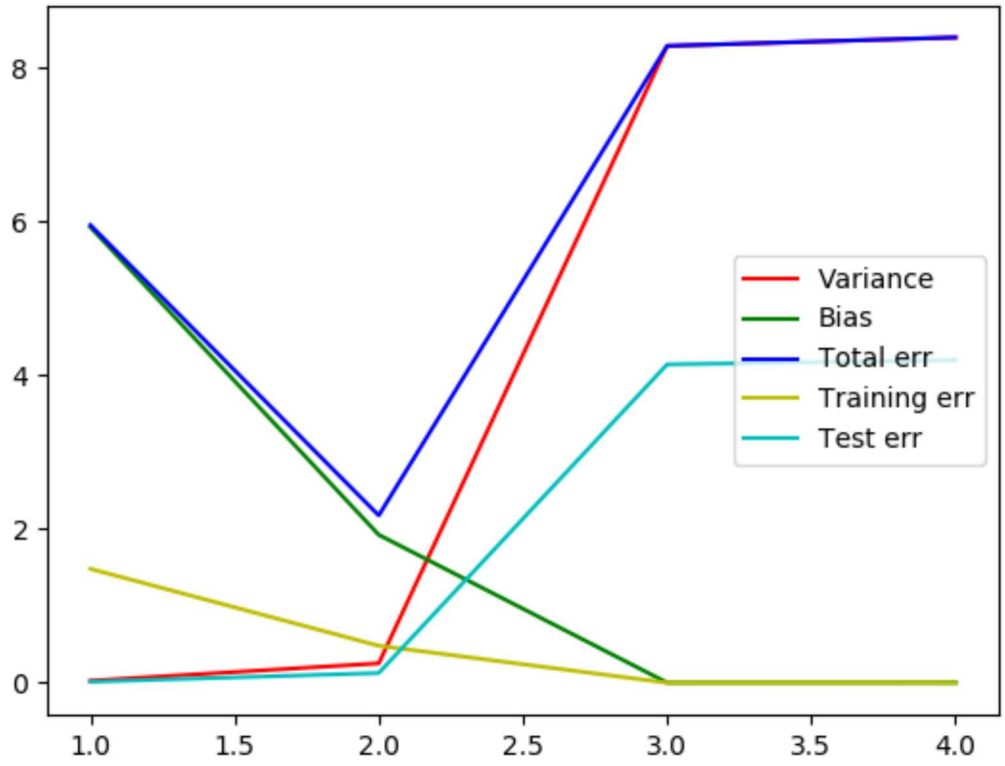
For d=1,y=-0.1925+3.077x,
$variance = (3-2.8846)^2 + (12-12.1153)^2 = 0.0266$
$bias^2 = (1+0.1923)^2 + (4-5.9615)^2 + (9-9.0384)^2 + (16-15.1922)^2 = 5.92307$
Total error= $bias^2 + variance$ =5.94968125
Trainning error= $\frac{1}{n} * bias^2 = 1.4808$
Test error= $\frac{1}{n} * variance = variance/2 = 0.0133$

From here I use programming tool:

For $d = 2, y = 0.80769 + 1.41x + 0.333x^2$,
$variance = 0.25219$
$bias^2 = 1.9232$
Total error= $bias^2 + variance$ =2.1754
Trainning error= $\frac{1}{n} * bias^2 = bias^2/4 = 0.4808$
Test error= $\frac{1}{n} * variance = variance/2 = 0.1261$

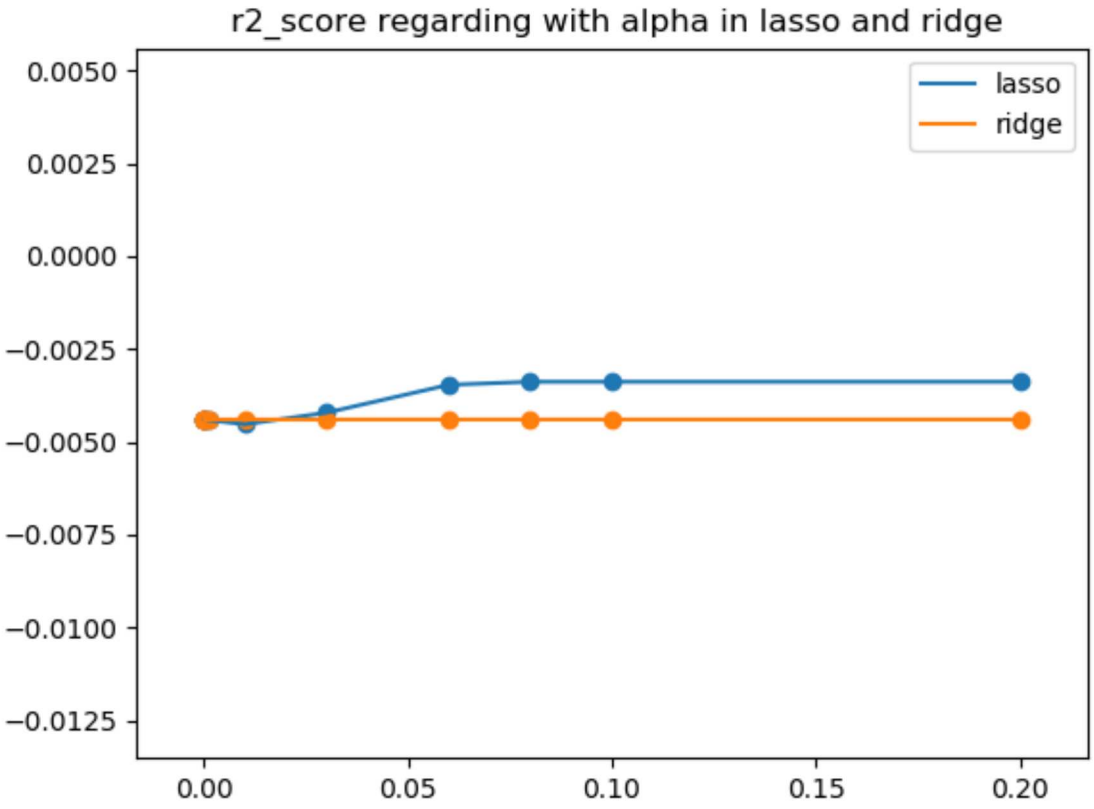
For $d = 3, y = 0.99999 + (-2.833)x + 2.833x^2 + (-0.333)x^3$,
$variance = 8.2787$
$bias^2 = 0.00127$
Total error= $bias^2 + variance$ =8.28
Trainning error= $\frac{1}{n} * bias^2 = bias^2/4 = 0.000319$
Test error= $\frac{1}{n} * variance = variance/2 = 4.139$

For $d = 4, y = 0.99999 + (-0.1396)x + 0.04986x^2 + (0.5645)x^3 + (-0.0897)x^4$,
$variance = 8.388$
$bias^2 = 0.00238$
Total error= $bias^2 + variance$ =8.3904
Trainning error= $\frac{1}{n} * variance = bias^2/4 = 0.000596$
Test error= $\frac{1}{n} * variance = variance/2 = 4.194$
Then we plot with matplotlib:

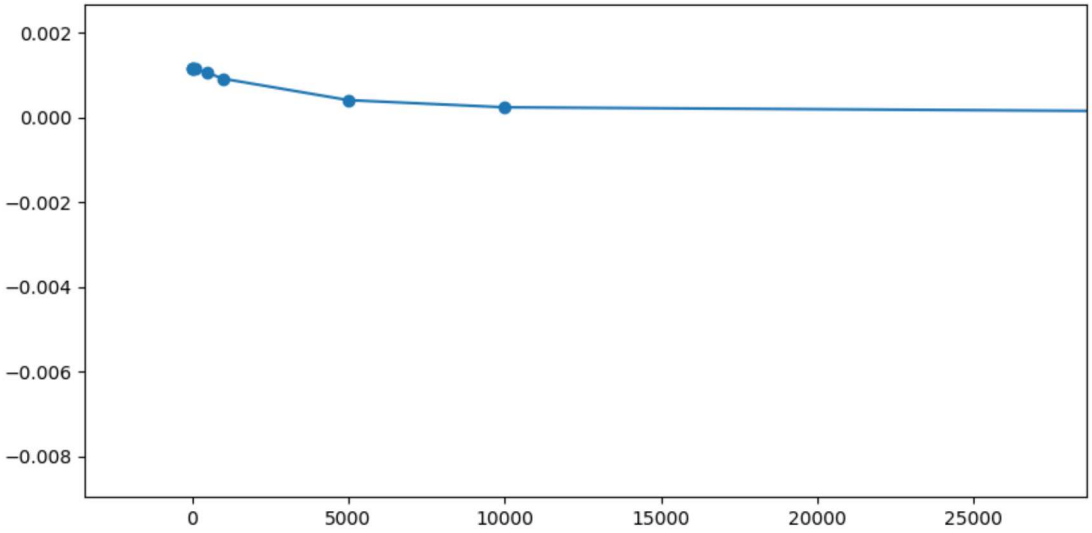


(3) From the graph above we can see the bias-variance tradeoff. From d=1 to d=2,the total error begin to decrease,variance is slowly growing and bias drops.When d=2, total error reaches its lowest point which is the perfect bias-variance tradeoff point and starts to grow up again. The variance start to grow sharply from d=2 to d=3 while the bias start to decrese slower. So when d is smaller than 2, the model is kind of underfiting because of high bias and when d is larger than 2, the model is overfitting because of high variance.

(4)This part of coding can be found in hw1q1(4).py and hw1ridge.py
Firstly, I trained two models one with the L1 norm and one with the L2 norm. In order to get the best alpha, I have tried different sets of alpha to get the optimum alpha using the built-in function $'cross - val - score'$ to the optimum alpha:



From the graph above, I derive that when alpha = 0.06(approximate), the lasso regression r2-score reaches the highest and stopped there. But for ridge regression it stays at the same straight line. So I tried a different way to get r2-score of ridge regression and get the following graph:



So I determined that the alpha should be around 6500 for ridge regression.
Then we use MSE to assess the output:
mean ridge mse= 0.2584088018576527
mean lasso mse= 0.2584022108852252

2 Problem 2

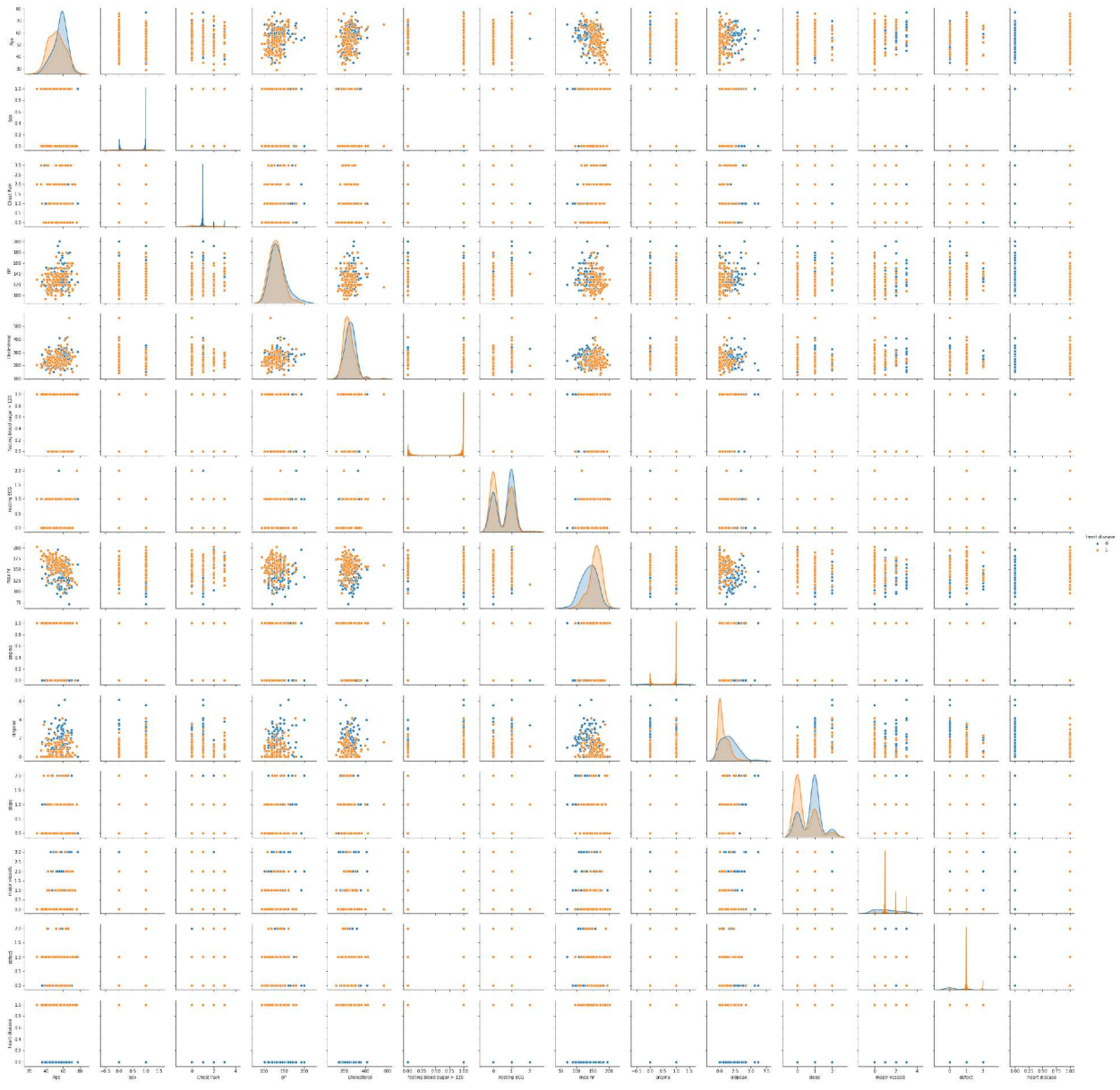
(1) Cross-entropy error $=\epsilon(\beta) = -\sum_{n=1}^{N}(y_n log[\sigma(\beta^T x_n)] + (1-y_n)log[1-\sigma(\beta^T x_n)])$
Because $\beta^T x_n = x+\beta_0$ so $\epsilon(\beta) = -\sum_{n=1}^{N}(y_n log[\sigma(x_n+\beta_0)] + (1-y_n)log[1-\sigma(x_n+\beta_0)])$
$\sigma(\eta) = \frac{1}{1+e^{-\eta}}$ So $\sigma(x+\beta_0) = \frac{1}{1+e^{-(x+\beta_0)}}$
There is only two trainning points: (-3,1)(-1,0)

So we can calculate the cross-entropy error:
$\epsilon(\beta_0) = -((y_1)log[\sigma(x_1+\beta_0)] + (1-y_1)log[1-\sigma(x_1+\beta_0)] + (y_2)log[\sigma(x_2+$
$\beta_0)] + (1-y_2)log[1-\sigma(x_2+\beta_0)])$
$\epsilon(\beta_0) = -(log[\sigma(\beta_0-3)] + log[\sigma(\beta_0-1)])$
Because $\sigma'(x) = \sigma(x) + \sigma(-x)$
We then take a derivitive of $\epsilon(\beta_0)$ which is:
$\epsilon'(\beta_0) = -(\frac{-1}{\sigma(3-\beta_0)} * \sigma(3-\beta_0) * \sigma(\beta_0-3) + \frac{1}{\sigma(\beta_0-1)} * \sigma(\beta_0-1) * \sigma(1-\beta_0))$
$=\sigma(1-\beta_0) - \sigma(\beta_0-3) = 0$
So $1-\beta_0 = \beta_0 - 3$
$\beta_0 = 2$

(2) We have two test data points with features 4, 5
$\sigma(-4+2) = \frac{1}{1+e^{-(-2)}} = 1/8.39 = 0.12 < 0.5 so 0$
$\sigma(5+2) = \frac{1}{1+e^{-7}} = 1/1.0009 = 1$
So the output should be 0 and 1

3 Problem 3

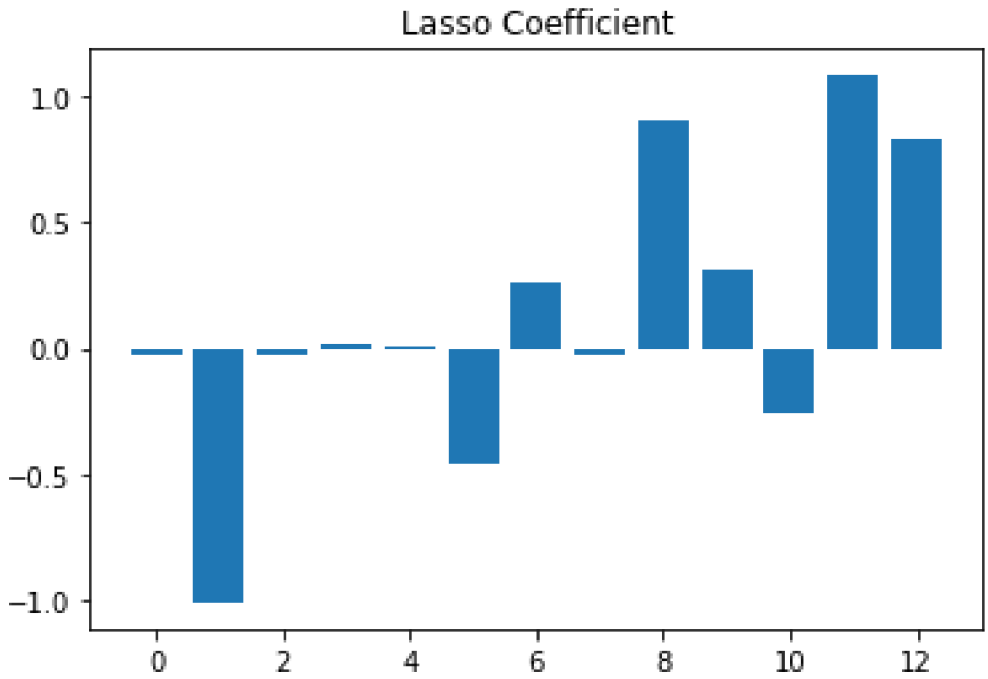
3:(1)
To pair plot the graph I used the panda library to read the csv file and then replaced some of the string variable to integers so we can use it to classify data.
Then I used seaborn library pairplot function to plot the graph.



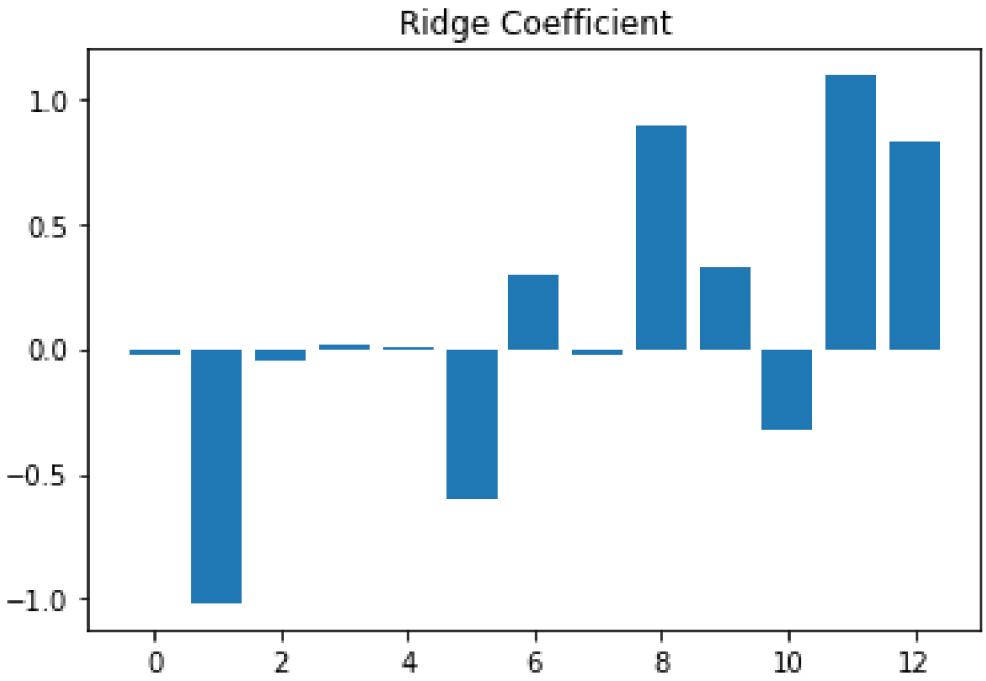
Histograms are shown to see the data distributes along different values. Scat-

ter plots are plotted for continuous variables to do classification on the data. (2)
All the coding in this part can be found in hw1problem3(2).ipynb

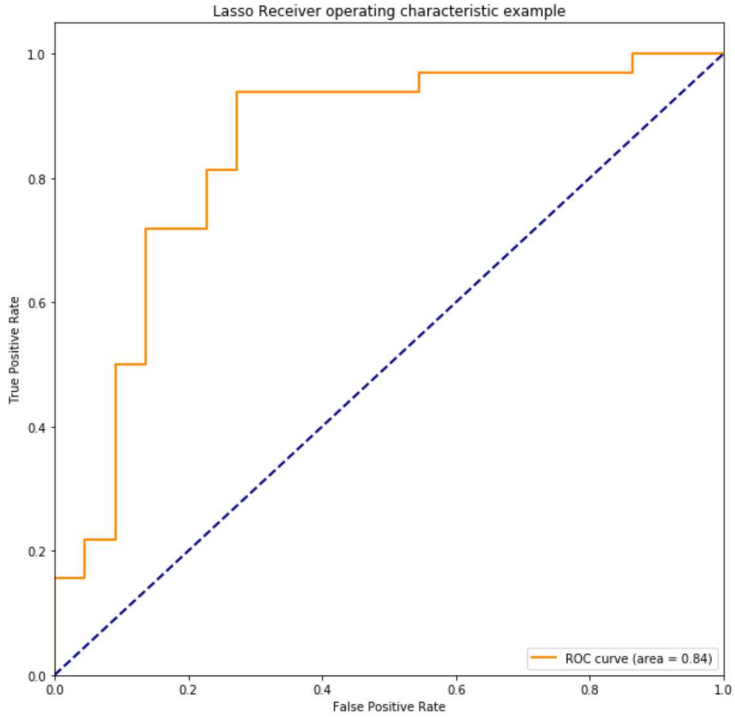
In each bootstraping iteration, I ridgeRegression = LogisticRegression(penalty = 'l2',solver='liblinear') Ridge mean coef 0.10978178293796569
Rasso mean coef 0.1228938652007024
Ridge deviation coef 0.5797631256412025
Rasso deviation coef 0.5626717408005893
After we plot the graph:
Lasso coefficient:

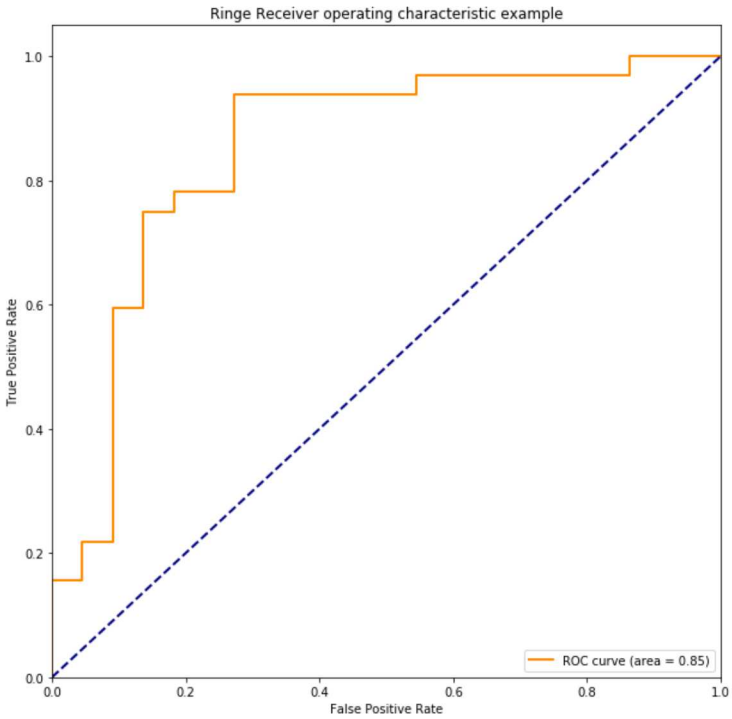


Ridge coefficient:

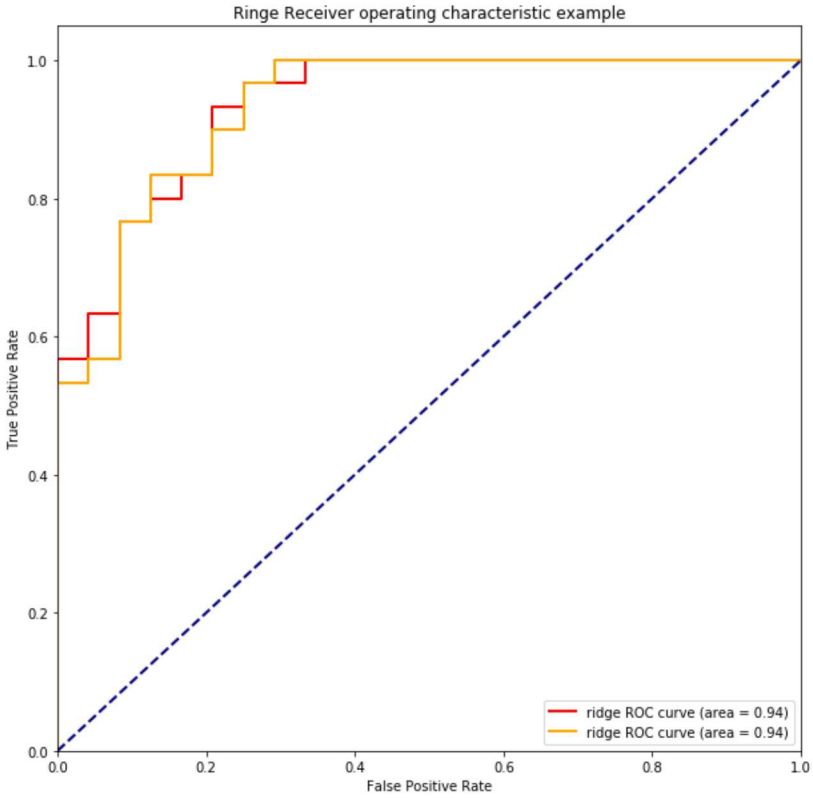


So we find no all features are needed. Only related features obvious in the graph should be needed.
(3) This part of code can also be found in hw1problem3(2).ipynb



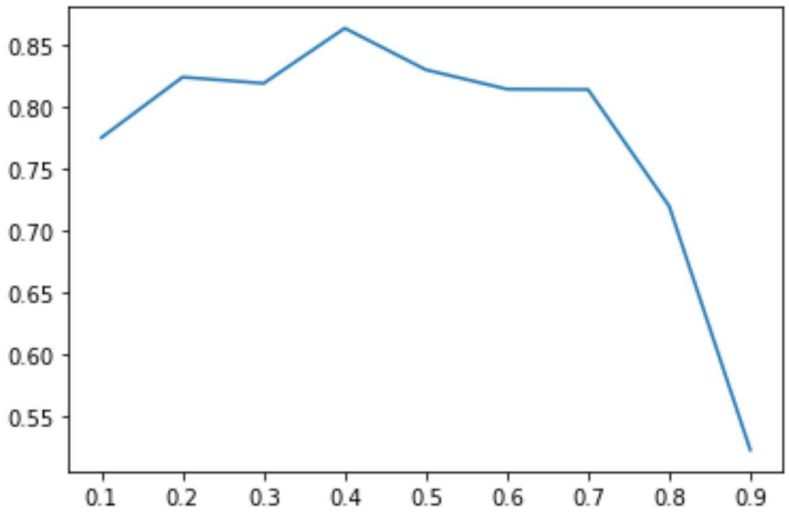


If plot in one graph:

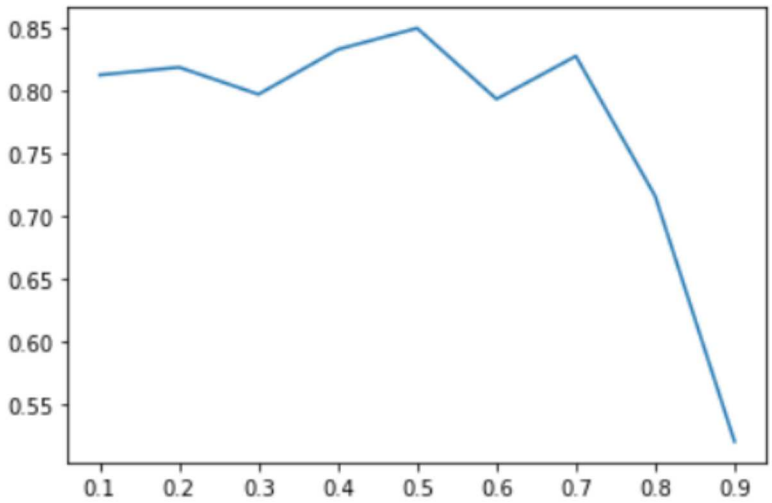


Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of .5 represents a worthless test.

(4)This part code can be found in hw1problem3(2).ipynb Lasso threshold vs F1 score



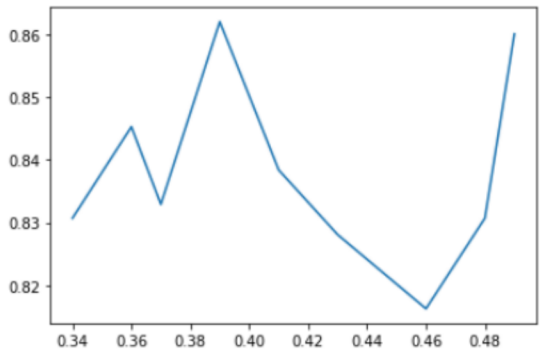
Ridge threshold vs F1 score



From the graph threshold vs F1 score:
For lasso ,the optimal decision threshold to maximize the f1 score is around 0.4
For ridge ,the optimal decision threshold to maximize the f1 score is around 0.5
(5)All the coding in this part can be found in hw1problem3(2).ipynb

A mean and standard deviation for the AUROCs are shown below:
Lasso Regression: Mean of AUROC is 0.8974325443644139
ridge Regression: Mean of AUROC is 0.8989390012686319
Lasso Regression: Standard Deviation of AUROC is 0.03965598790383527
Ridge Regression: Standard Deviation of AUROC is 0.03927244574101741
(6) This part code can be found in hw1problem3(2).ipynb

```
using lasso when threshold is  0.34 mean of F1 score  :  0.8306915060382568
using lasso when threshold is  0.34  standard deviation of F1 score:  0.04327159344578735
using lasso when threshold is  0.36 mean of F1 score  :  0.8452599296426003
using lasso when threshold is  0.36  standard deviation of F1 score:  0.04124206624063743
using lasso when threshold is  0.37 mean of F1 score  :  0.8329080868940428
using lasso when threshold is  0.37  standard deviation of F1 score:  0.03851542239821478
using lasso when threshold is  0.39 mean of F1 score  :  0.8619845345529772
using lasso when threshold is  0.39  standard deviation of F1 score:  0.04915166692547036
using lasso when threshold is  0.41 mean of F1 score  :  0.8383767734878923
using lasso when threshold is  0.41  standard deviation of F1 score:  0.054329796322711885
using lasso when threshold is  0.43 mean of F1 score  :  0.8280763436031398
using lasso when threshold is  0.43  standard deviation of F1 score:  0.04929415081508374
using lasso when threshold is  0.46 mean of F1 score  :  0.8162746361112443
using lasso when threshold is  0.46  standard deviation of F1 score:  0.039450256311413454
using lasso when threshold is  0.48 mean of F1 score  :  0.8307004926809263
using lasso when threshold is  0.48  standard deviation of F1 score:  0.056442317210083026
using lasso when threshold is  0.49 mean of F1 score  :  0.8600349573462864
using lasso when threshold is  0.49  standard deviation of F1 score:  0.04505008334658553
```



```
Using ridge when threshold is  0.44 mean of F1 score  :  0.8119769521682203
 Using ridge when threshold is  0.44  standard deviation of F1 score:  0.053630144804210016
Using ridge when threshold is  0.46 mean of F1 score  :  0.8229832738590834
 Using ridge when threshold is  0.46  standard deviation of F1 score:  0.046693560480651694
Using ridge when threshold is  0.47 mean of F1 score  :  0.8373045957661291
 Using ridge when threshold is  0.47  standard deviation of F1 score:  0.042000909304548806
Using ridge when threshold is  0.49 mean of F1 score  :  0.8386128926451544
 Using ridge when threshold is  0.49  standard deviation of F1 score:  0.02832940652655484
Using ridge when threshold is  0.51 mean of F1 score  :  0.8527734719502966
 Using ridge when threshold is  0.51  standard deviation of F1 score:  0.028774504831558228
Using ridge when threshold is  0.53 mean of F1 score  :  0.8353201479564154
 Using ridge when threshold is  0.53  standard deviation of F1 score:  0.046317831949656996
Using ridge when threshold is  0.56 mean of F1 score  :  0.8101977000434571
 Using ridge when threshold is  0.56  standard deviation of F1 score:  0.04026126067828134
Using ridge when threshold is  0.58 mean of F1 score  :  0.8376120339838573
 Using ridge when threshold is  0.58  standard deviation of F1 score:  0.03103051977243391
Using ridge when threshold is  0.6 mean of F1 score  :  0.8309034953259259
 Using ridge when threshold is  0.6  standard deviation of F1 score:  0.05921509573470792
```