Homework2-INF552-Arpit Sharma

Question 1

(c)

i)

Mean, Standard Deviation, skewness, kurtosis of time series are effective ways to represent and classify such patterns.

Trends like seasonality, periodicity, serial correlation, chaos, nonlinearity, fourier transforms of a range of time series are also good methods.

ii)

I tried using min, max, median, standard deviation and mean as the features.

Bootstrapped 98% confidence intervals of Min

Low: 0.4058229891722333

High: 5.057132508288333

Bootstrapped 98% confidence intervals of Max

Low: 2.0892583810933334

High: 3.4947894504000003

Bootstrapped 98% confidence intervals of Mean

Low: 1.0312801348553333

High: 3.0876193614699994

Bootstrapped 98% confidence intervals of Median

Low: 0.9413418525088333

High: 3.1131769200516666

Bootstrapped 98% confidence intervals of Standard Deviation

Low: 0.48947729260766665

High: 0.9590192196278333

USC ID: 5423-4179-55

iii)

I moved forward with min, max and mean as I observed more standard deviation in these 3 features.

iv)

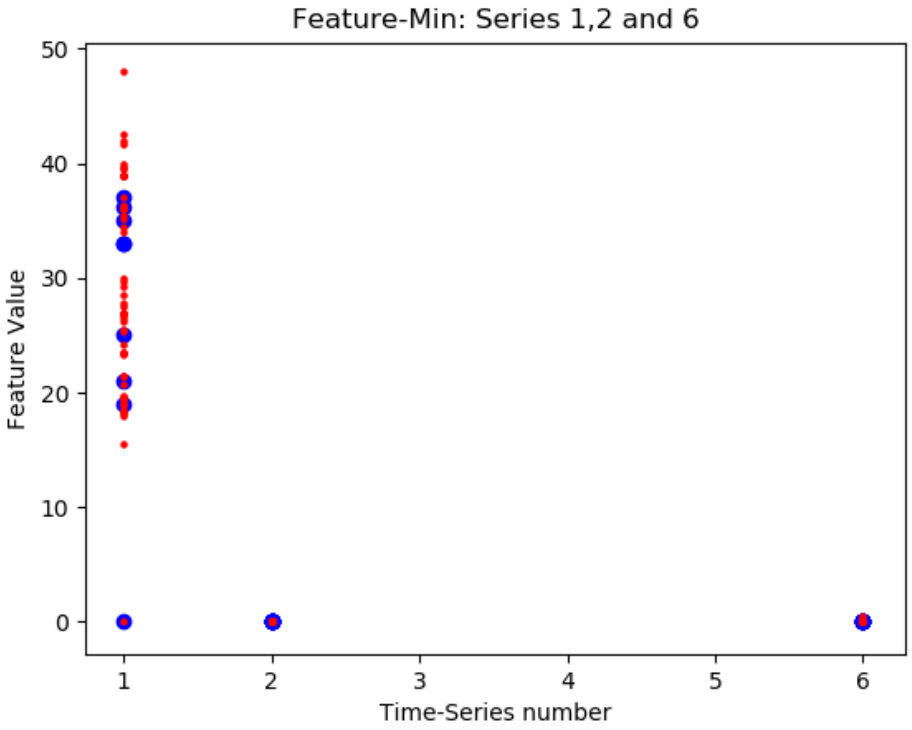
I observed that ‘max’ made a vital contribution to the response variable.

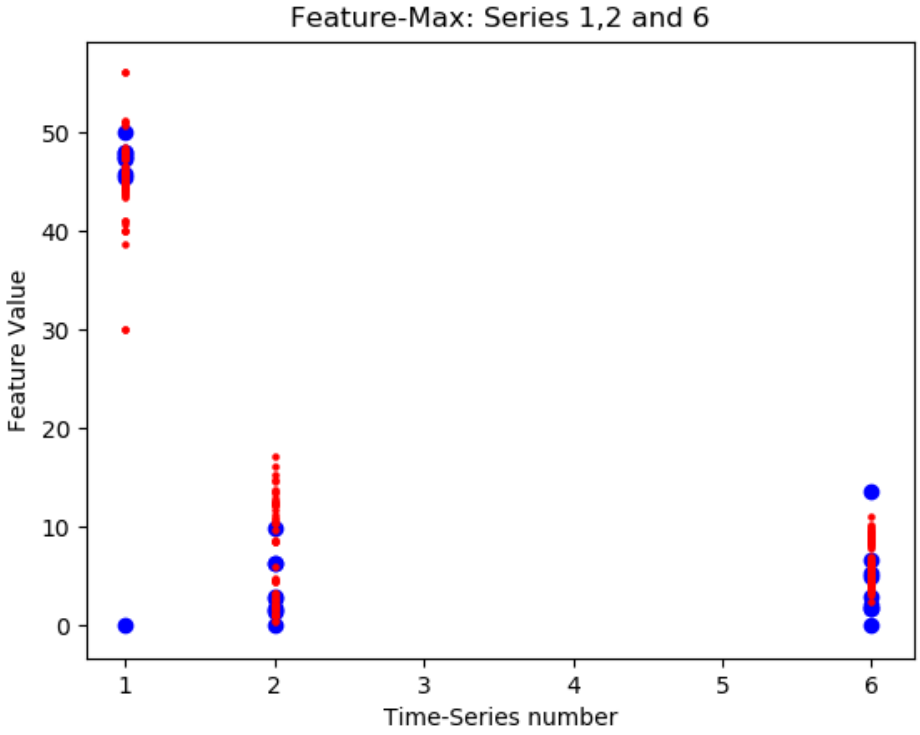
‘min’ of the values did not show heavy significance.

(d.) Binary Classification Using Logistic Regression

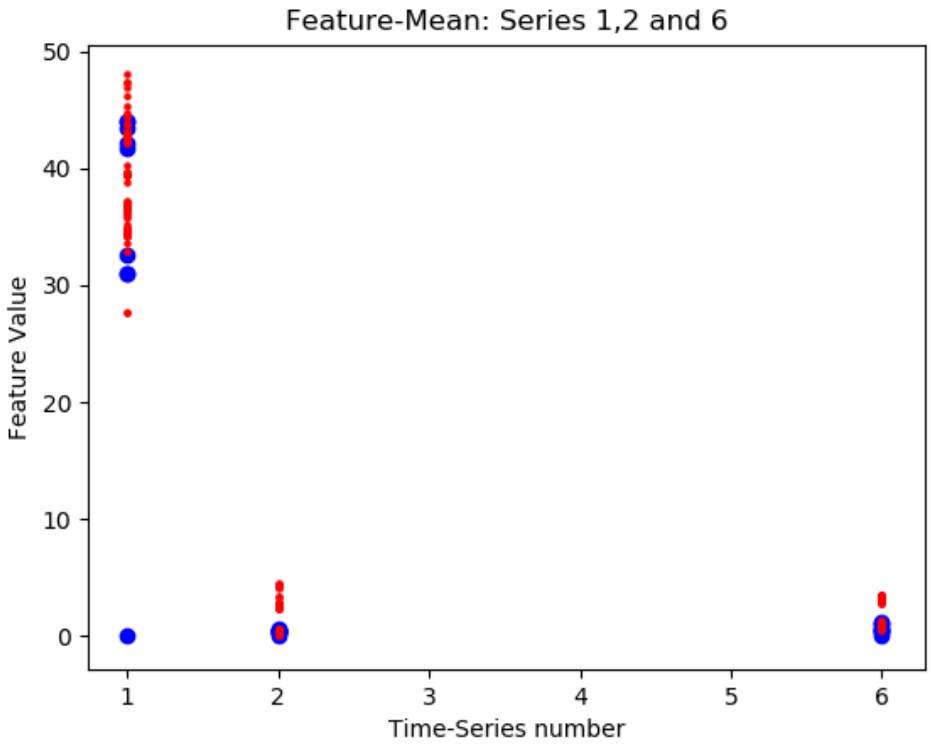
i) Scatter plots for series 1,2 and 6 (Blue represents Bending)

Min, Max and Mean:

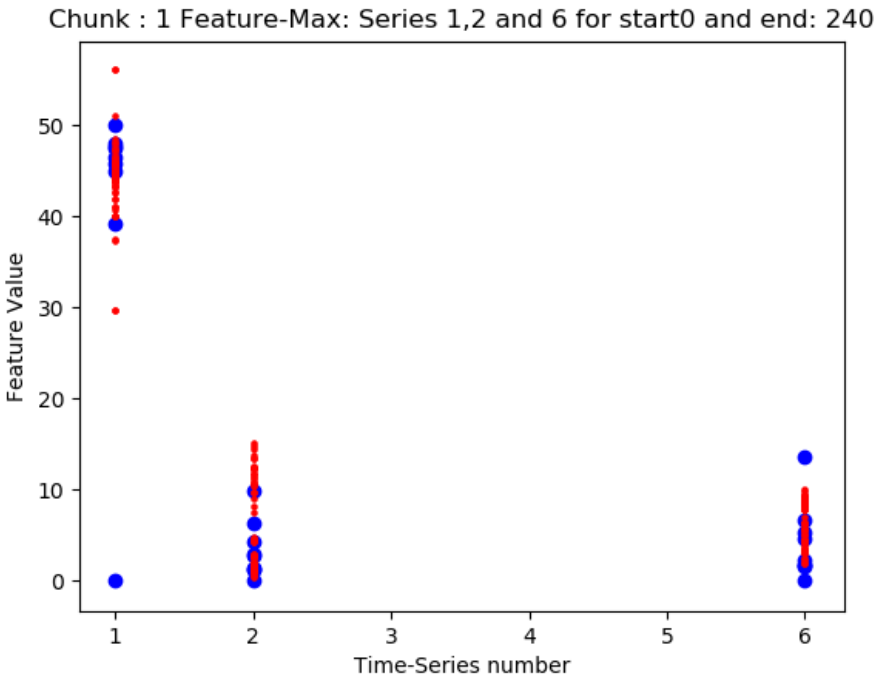


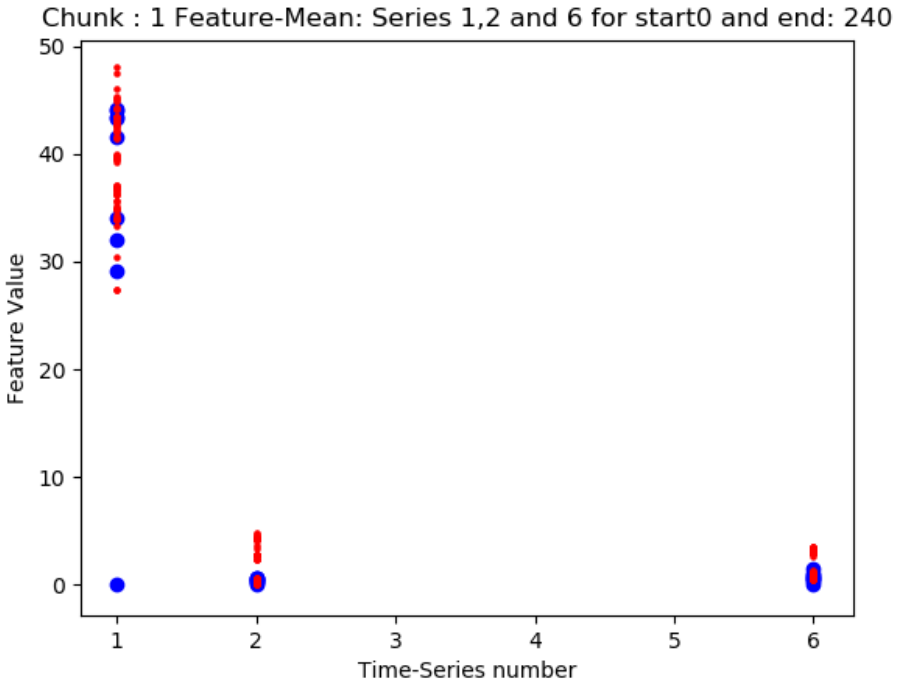


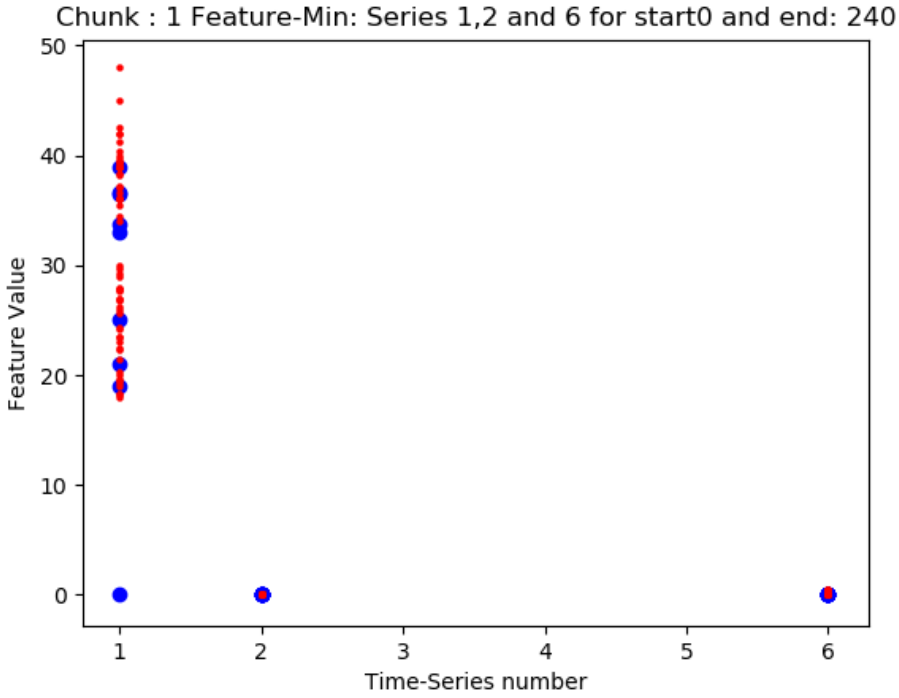
USC ID: 5423-4179-55



ii) Dividing the Dataset into 2 chunks (l=2)
Chunk 1 :



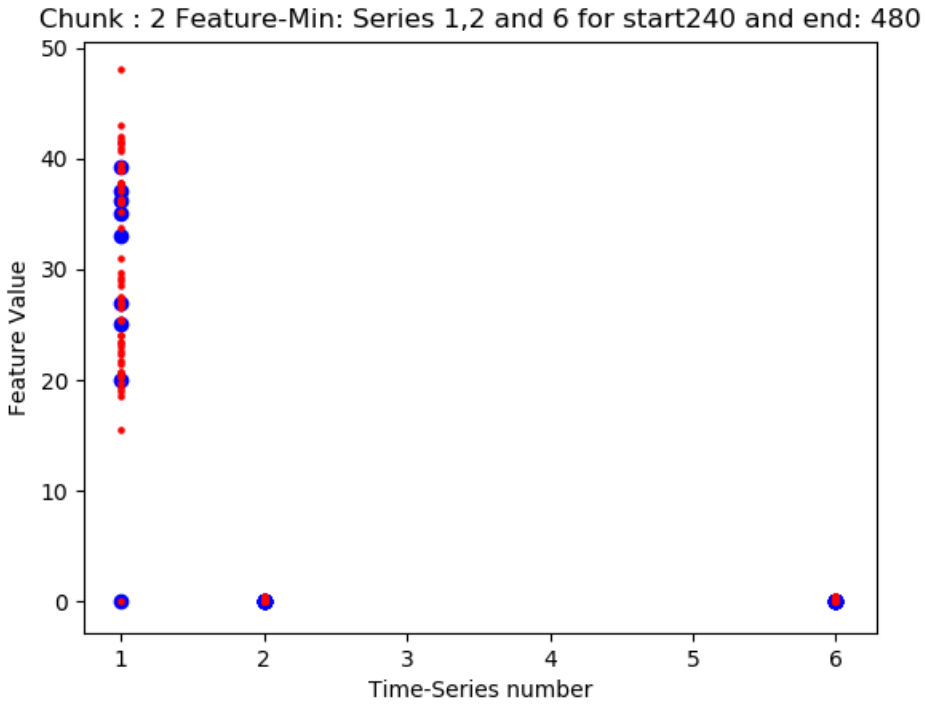


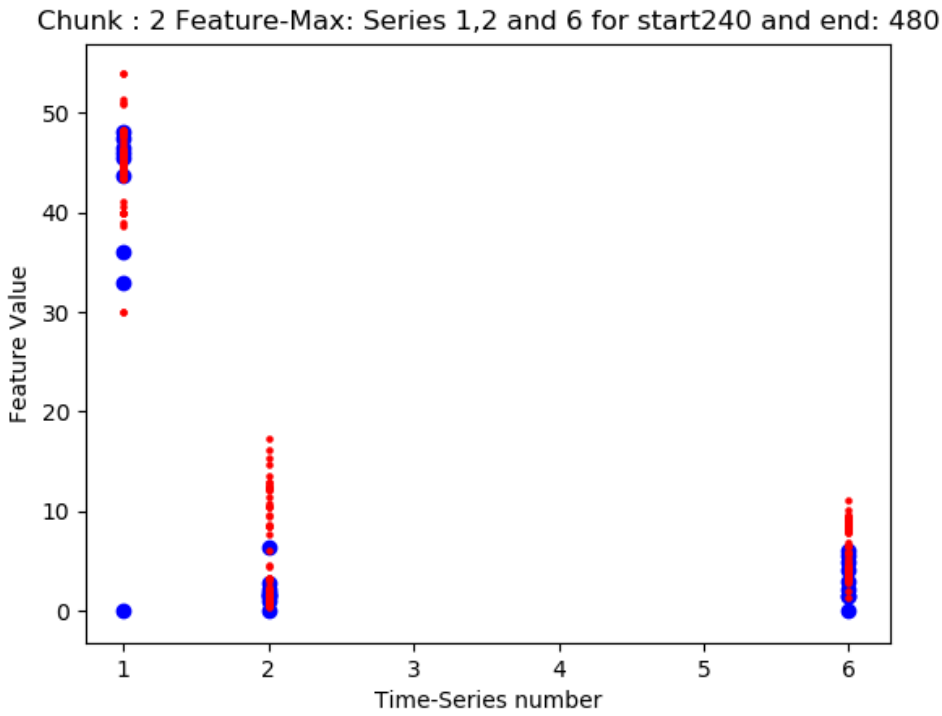


USC ID: 5423-4179-55

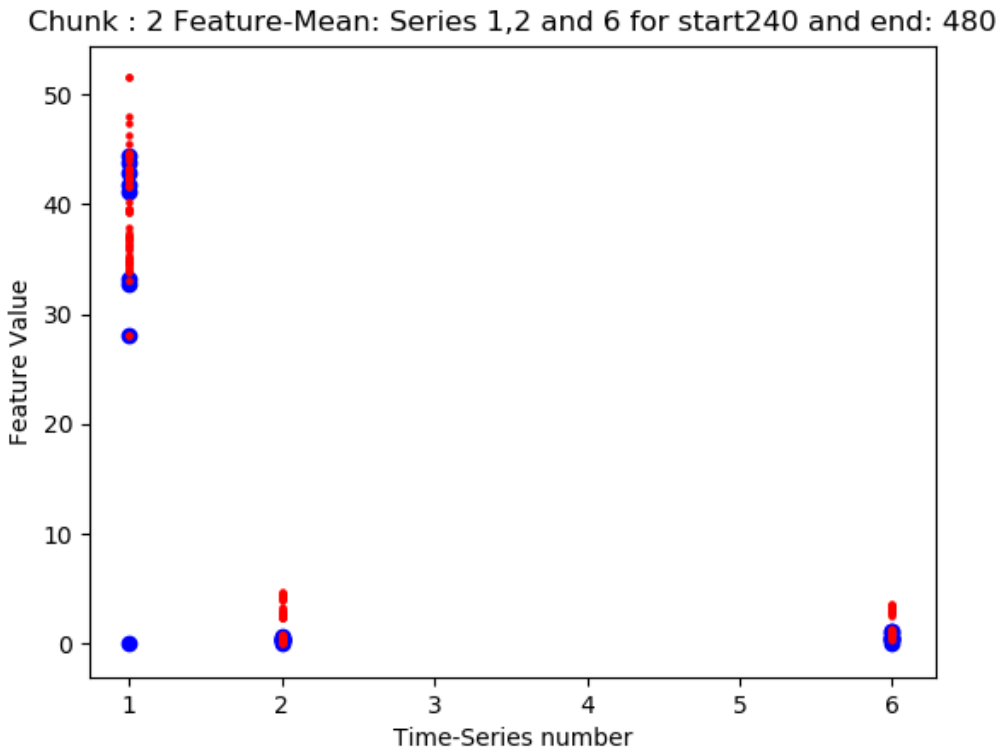
Homework2-INF552-Arpit Sharma

Chunk 2:





USC ID: 5423-4179-55



As compared to using the whole dataset this fairs much better as I could get a whole new range in the values of time-domain features, which is evident while comparing the Max feature plot of them.

I would recommend splitting the data as that modifies the features treating the whole data as 2 new datasets, which can lead to better training of the model.

iii) For l = {1,2, 3,…,10} time series.

Observed P-values

For l=1

```
IN L= 1
In chunk: 1
starting: 0
ending 79
TRAIN: [16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]

 [fold 0] score: 1.00000

p values for X_train_K

[2.18132091e-02 2.01230511e-03 6.69604487e-03 8.60436373e-04
 2.21485427e-03           nan 5.42802041e-03 2.89970770e-01
 9.77730782e-01 3.21178388e-01 7.61067184e-02           nan
 1.49950799e-08 9.20435782e-10 2.25991972e-34 4.46826737e-02
 1.30788340e-02 7.13938329e-01]

[fold 1] score: 1.00000

p values for X_train_K

[1.11546692e-02 1.94778275e-03 4.50985581e-02 2.87697062e-07
 3.16920057e-05           nan 3.61795734e-01 8.42347253e-01
 4.85918807e-03 3.66985570e-02 1.13087116e-03           nan
 1.99335089e-07 1.33531433e-07 2.07232394e-35 3.02588617e-05
 2.38079474e-04 7.13938329e-01]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47]

[fold 2] score: 0.93750
p values for X_train_K

[3.13946900e-02 1.47617327e-03 6.25005545e-02 1.84770359e-06
```

6.11792038e-05 nan 1.89204801e-02 2.86908160e-01
8.30160869e-01 6.17325101e-01 2.95699873e-03 nan
6.92123982e-10 1.64822476e-09 2.18480488e-23 1.20337969e-02
1.29571003e-03 7.25956112e-01]

TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63]

[fold 3] score: 0.87500

p values for X_train_K

[6.94353832e-01 9.40627060e-01 1.81312179e-01 1.85157103e-11
1.94462861e-05 nan 4.64000807e-04 2.09904735e-03
3.45796975e-01 6.02267964e-01 1.52244683e-03 nan
6.99281913e-14 1.47986783e-16 6.21018422e-47 1.52389531e-07
1.70681840e-04 7.02016158e-01]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63]
TEST: [64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]

[fold 4] score: 1.00000

p values for X_train_K

[8.50690127e-01 2.43229179e-01 3.40307999e-01 7.02639956e-03
4.33245587e-02 nan 5.86376259e-04 2.60044190e-03
1.46151108e-03 9.67077033e-01 1.15959176e-01 nan
2.26951307e-15 1.31149812e-19 9.70161864e-74 5.34075501e-03
1.53729255e-02 nan]

For l=2:

IN L= 1
In chunk: 1
starting: 0
ending 79

TRAIN: [16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]

[fold 0] score: 1.00000

p values for X_train_K

[2.18132091e-02 2.01230511e-03 6.69604487e-03 8.60436373e-04
2.21485427e-03 nan 5.42802041e-03 2.89970770e-01
9.77730782e-01 3.21178388e-01 7.61067184e-02 nan
1.49950799e-08 9.20435782e-10 2.25991972e-34 4.46826737e-02
1.30788340e-02 7.13938329e-01]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31]

[fold 1] score: 1.00000

p values for X_train_K

[1.11546692e-02 1.94778275e-03 4.50985581e-02 2.87697062e-07
3.16920057e-05 nan 3.61795734e-01 8.42347253e-01
4.85918807e-03 3.66985570e-02 1.13087116e-03 nan
1.99335089e-07 1.33531433e-07 2.07232394e-35 3.02588617e-05
2.38079474e-04 7.13938329e-01]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47]

[fold 2] score: 0.93750

p values for X_train_K

[3.13946900e-02 1.47617327e-03 6.25005545e-02 1.84770359e-06
6.11792038e-05 nan 1.89204801e-02 2.86908160e-01
8.30160869e-01 6.17325101e-01 2.95699873e-03 nan
6.92123982e-10 1.64822476e-09 2.18480488e-23 1.20337969e-02
1.29571003e-03 7.25956112e-01]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47

```
 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]
TEST: [48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63]

[fold 3] score: 0.87500

 p values for X_train_K

[6.94353832e-01 9.40627060e-01 1.81312179e-01 1.85157103e-11
 1.94462861e-05          nan 4.64000807e-04 2.09904735e-03
 3.45796975e-01 6.02267964e-01 1.52244683e-03          nan
 6.99281913e-14 1.47986783e-16 6.21018422e-47 1.52389531e-07
 1.70681840e-04 7.02016158e-01]

TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63]
TEST: [64 65 66 67 68 69 70 71 72 73 74 75 76 77 78]

[fold 4] score: 1.00000

 p values for X_train_K

[8.50690127e-01 2.43229179e-01 3.40307999e-01 7.02639956e-03
 4.33245587e-02          nan 5.86376259e-04 2.60044190e-03
 1.46151108e-03 9.67077033e-01 1.15959176e-01          nan
 2.26951307e-15 1.31149812e-19 9.70161864e-74 5.34075501e-03
 1.53729255e-02          nan]


IN L= 2
In chunk: 1
starting: 0
ending 39
TRAIN: [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]
TEST: [0 1 2 3 4 5 6 7]

[fold 0] score: 0.87500

 p values for X_train_K

[1.95170009e-01 1.06601156e-01 4.77020510e-02 3.82487588e-04
 1.38485466e-01          nan 5.60797173e-06 4.67875868e-07
 6.13012790e-07 1.69583281e-01 3.24968773e-01          nan
```

```
 1.36191755e-13 1.21991431e-20 4.00596935e-51 1.88803426e-03
 5.31109232e-02           nan]
TRAIN: [ 0  1  2  3  4  5  6  7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]
TEST: [ 8  9 10 11 12 13 14 15]

 [fold 1] score: 1.00000

 p values for X_train_K

[2.06082447e-01 2.82284727e-01 1.86764226e-01 1.20621068e-03
 1.43012797e-01           nan 6.37335235e-06 3.77506429e-06
 3.22131965e-06 1.54551231e-01 3.09381513e-01           nan
 5.92385311e-15 2.84643178e-17 2.14231313e-48 9.02303492e-03
 8.08157365e-02           nan]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]
TEST: [16 17 18 19 20 21 22 23]

[fold 2] score: 1.00000

 p values for X_train_K

[1.95658981e-01 9.73540613e-02 2.00808710e-02 7.78087533e-07
 8.74038884e-03           nan 4.14807936e-03 1.00890327e-03
 6.87560748e-02 5.26809041e-01 2.23424426e-02           nan
 4.14106827e-12 9.95605556e-15 2.26380994e-44 1.70192092e-05
 4.15919160e-03           nan]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 32 33 34 35 36 37 38]
TEST: [24 25 26 27 28 29 30 31]

[fold 3] score: 1.00000

 p values for X_train_K

[3.82447820e-01 2.09080589e-01 8.82591086e-03 1.29206940e-06
 9.90576474e-03           nan 4.67208343e-03 3.64666498e-04
 2.87345718e-02 2.37806503e-01 2.66128799e-02           nan
 1.19516638e-12 6.55586805e-16 1.29210953e-46 2.56586045e-06
 4.22645335e-03           nan]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31]
TEST: [32 33 34 35 36 37 38]
```

```
[fold 4] score: 0.71429

 p values for X_train_K

[1.33869753e-01 1.71256715e-01 2.18625605e-02 1.91723158e-05
 1.70016752e-02          nan 5.65207506e-05 4.73239366e-06
 1.90438249e-05 9.23309991e-01 5.05356749e-02          nan
 2.85296223e-14 2.15178504e-17 2.57963380e-37 1.88389051e-03
 1.93015579e-02          nan]
IN L= 2
In chunk: 2
starting: 39
ending 78
TRAIN: [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]

TEST: [0 1 2 3 4 5 6 7]
  y = column_or_1d(y, warn=True)

[fold 0] score: 1.00000

 p values for X_train_K

[7.31868398e-06 4.48566157e-09 2.63659311e-07 5.82683341e-03
 1.10933422e-03          nan 4.96166236e-01 2.60561710e-03
 1.79830176e-07 3.55014669e-01 2.04468497e-02          nan
 3.60016275e-01 7.50094679e-01 9.11712286e-01 4.83072608e-01
 2.14628892e-02          nan]
TRAIN: [ 0  1  2  3  4  5  6  7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]
TEST: [ 8  9 10 11 12 13 14 15]

[fold 1] score: 1.00000

 p values for X_train_K

[5.01993480e-07 3.46996301e-09 7.86746875e-09 4.20429899e-03
 2.48385692e-03          nan 1.78855186e-01 5.56405280e-03
 3.11370025e-05 3.35106151e-01 2.35886589e-02          nan
 8.09776427e-01 2.98842329e-01 7.66694090e-01 6.46984732e-02
 2.59217139e-02          nan]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 24 25 26 27 28 29 30 31
 32 33 34 35 36 37 38]
```

TEST: [16 17 18 19 20 21 22 23]

[fold 2] score: 1.00000

p values for X_train_K

[1.79666259e-02 1.11158433e-03 5.18047758e-03 4.49963577e-03
3.73483327e-03 nan 4.56539537e-01 1.23635959e-01
3.10165109e-05 6.33674412e-01 4.14850308e-02 nan
4.01017517e-02 2.61704290e-01 1.05139520e-01 1.29292489e-01
3.67797003e-02 nan]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
32 33 34 35 36 37 38]
TEST: [24 25 26 27 28 29 30 31]

[fold 3] score: 1.00000

p values for X_train_K

[3.82317758e-02 2.31765723e-04 9.98690380e-05 6.66648821e-01
4.56566441e-02 nan 5.91459888e-01 1.43157537e-01
6.40193795e-04 7.31706896e-01 1.91624585e-01 nan
2.85279501e-02 1.32273257e-01 1.50391508e-04 8.78093811e-01
1.55222151e-01 nan]
TRAIN: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31]
TEST: [32 33 34 35 36 37 38]

[fold 4] score: 1.00000

p values for X_train_K

[1.41113364e-05 6.59594643e-09 7.72239895e-09 4.93953196e-01
4.59633926e-02 nan 2.65235885e-01 8.05918268e-03
4.71331600e-04 2.97438429e-01 1.59602975e-01 nan
6.49869837e-01 8.64741969e-01 7.56308234e-04 5.73783578e-01
1.15747497e-01 nan]

Similarly, all the values for L=3,4…,10 were observed

I observed for each value of L the chunks of data passed to the model were different, this resulted in different set of features to be pruned each time by the model. I used SelectKBest package in sklearn to select k(6) best features based on their p-values.

I then used the mean of each chunk in L=1,2,3….10 and made a comparison between them. I observed the best score for **L=1**

Final Score for L= 1 : 0.9625
Final Score for L= 2 : 0.9589285714285715
Final Score for L= 3 : 0.9111111111111111
Final Score for L= 4 : 0.9375
Final Score for L= 5 : 0.9466666666666665
Final Score for L= 6 : 0.8944444444444445
Final Score for L= 7 : 0.9380952380952381
Final Score for L= 8 : 0.8625
Final Score for L= 9:: 0.823655425
Final Score for L= 10 : 0.9159626625

Cross-validation might be done in 2 ways: the wrong way and the right way. The only difference is that in the former, we perform the variable selection before cross validation using all the samples. In the latter, we perform the variable selection within a K-fold cross validation loop.

iv)

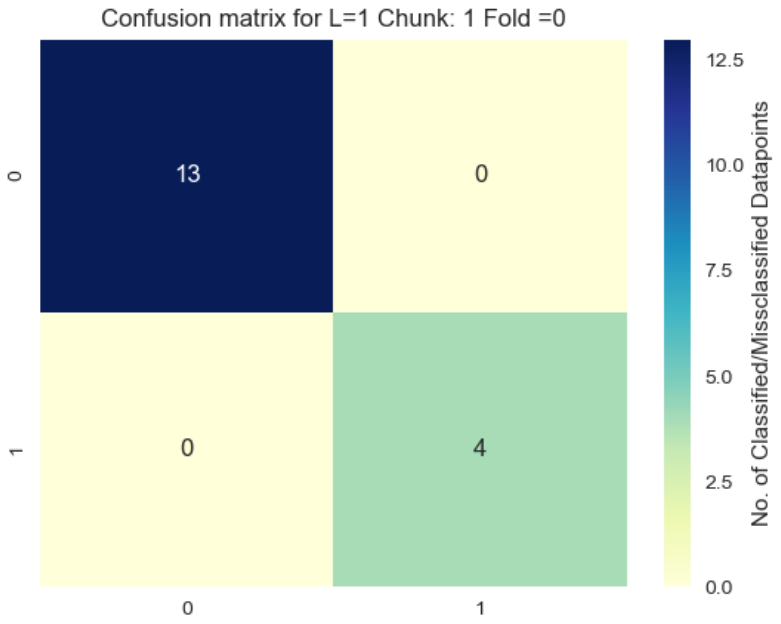
Regression coef and p-values for L=1 for 5 fold CV

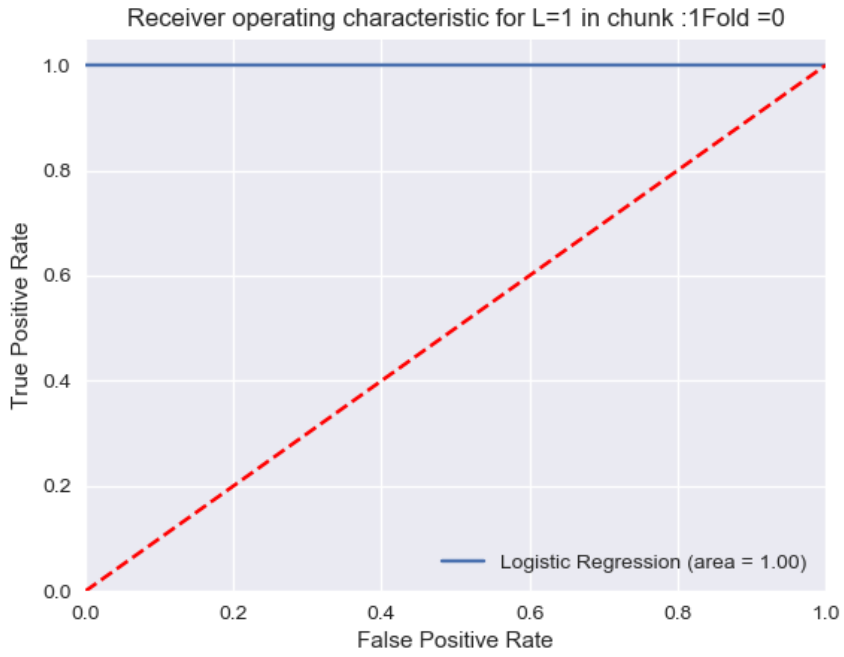
```
IN L= 1
In chunk: 1
starting: 0
ending 81
 [fold 0] score: 1.00000
Regression Coef:
 [[-1.88989006e-01 -3.31441790e-01 -1.87278877e-01 -3.84869958e-01
  -1.62155439e-01  0.00000000e+00  1.47912261e-01 -2.62967806e-01
  -5.25356509e-02  1.46077039e-01 -1.10366775e-01  0.00000000e+00
   5.39161705e-01  4.25511556e-01  7.06919125e-01 -3.79192664e-02
  -1.26477892e-01 -4.01018520e-04]]

 p values for X_train_K

[3.92349208e-02 6.18793704e-04 7.41144085e-04 3.07791459e-03
 1.13343942e-03           nan 2.81880591e-04 2.79224872e-01
 4.16025220e-01 7.78354896e-02 5.82606166e-02           nan
 1.98207096e-10 3.63528323e-09 2.76529009e-28 7.70402802e-02
 1.34686726e-02 6.93303942e-01]

[fold 1] score: 1.00000
Regression Coef:
 [[-8.57949245e-03 -2.71480147e-01 -2.10362480e-01 -4.08587873e-01
  -1.06325146e-01  0.00000000e+00  3.49075579e-02 -3.38705745e-01
  -3.16875295e-01 -7.04519160e-02 -9.73807825e-02  0.00000000e+00
   5.91547740e-01  2.16684833e-01  5.15500706e-01 -1.08906042e-01
  -1.09140024e-01 -1.34679887e-04]]

 p values for X_train_K

[1.17993621e-02 2.14223590e-04 2.83710707e-03 2.97720532e-06
 2.19498330e-05           nan 1.39482986e-01 6.87769284e-01
 5.85553054e-04 2.23247360e-01 1.02569697e-03           nan
 4.06012864e-08 1.12079342e-06 4.58321576e-25 3.54359362e-04
 4.45003302e-04 7.07875580e-01]

 [fold 2] score: 0.93750
```

```
Regression Coef:
 [[-8.60317827e-02 -4.95600069e-01 -8.05139997e-02 -4.07712864e-01
  -2.27954727e-01  0.00000000e+00  1.34393847e-01 -3.70948694e-02
  -7.65638176e-02 -4.42610754e-04 -2.40128948e-01  0.00000000e+00
   4.92214706e-01  5.36217667e-01  2.54024642e-01 -1.67867645e-01
  -2.44817352e-01 -1.06960333e-04]]

 p values for X_train_K

[6.24521028e-02 1.31965524e-03 3.01541196e-02 1.09623081e-06
 1.10994891e-05           nan 8.23118121e-04 1.67578549e-01
 2.25979327e-01 8.48642579e-01 1.27224713e-03           nan
 3.76430645e-13 4.42240752e-11 1.96341189e-24 6.43635564e-03
 5.29433280e-04 6.96354909e-01]

 [fold 3] score: 1.00000
Regression Coef:
 [[-1.44120829e-01 -3.31059872e-01 -1.67794732e-01 -3.96498037e-01
  -1.21723519e-01  0.00000000e+00  8.68105544e-02 -2.12166139e-01
   2.46895622e-02  1.35151057e-01 -1.23383255e-01  0.00000000e+00
   6.15868357e-01  3.47078968e-01  5.85349645e-01 -4.25632700e-01
  -1.46550334e-01 -1.00683338e-04]]

 p values for X_train_K

[7.42926219e-01 6.32242697e-01 6.87074716e-01 6.06519155e-10
 1.33673803e-05           nan 3.83573659e-05 3.01624484e-03
 6.86134011e-01 8.03987221e-01 1.28857636e-03           nan
 2.13093924e-15 1.10854530e-15 4.54785847e-41 1.28874086e-06
 2.23218174e-04 6.84883030e-01]

 [fold 4] score: 1.00000
Regression Coef:
 [[-0.06740508 -0.43109773 -0.18277607 -0.4001777  -0.16158786  0.
   0.0316608  -0.26589794 -0.0574274   0.06066791 -0.17379993  0.
   0.64658364  0.44452797  0.6259898  -0.19496637 -0.19474349  0.       ]]

 p values for X_train_K
[8.42518241e-01 2.65772070e-01 4.33510339e-01 2.51597858e-03
 2.69212313e-02           nan 6.66727786e-04 2.88556899e-03
 2.52562142e-03 9.73825019e-01 9.33670187e-02           nan
 2.19476571e-15 1.43100395e-19 7.07163278e-73 4.32744647e-03
 1.18326240e-02           nan]
#########################################################################
```

Homework2-INF552-Arpit Sharma

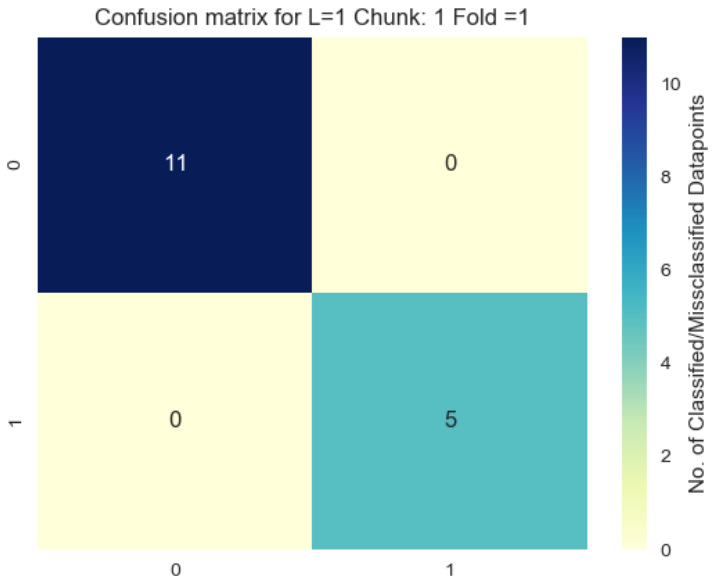
CONFUSION MATRICES AND ROC

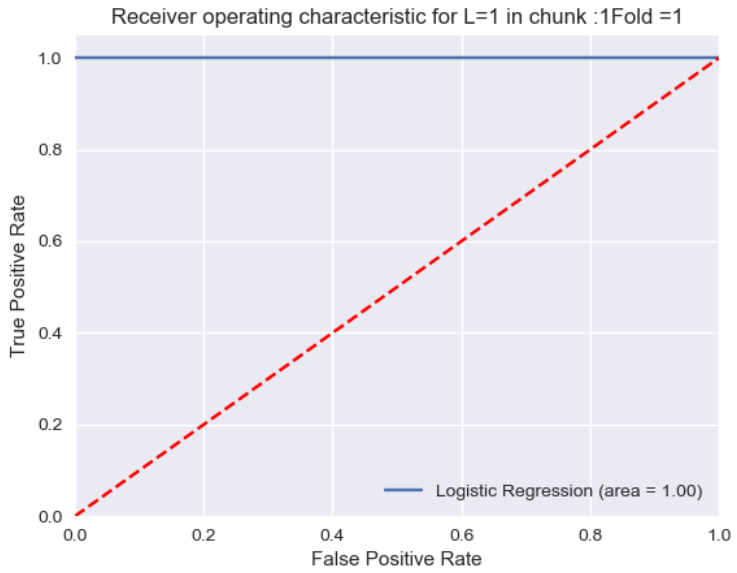




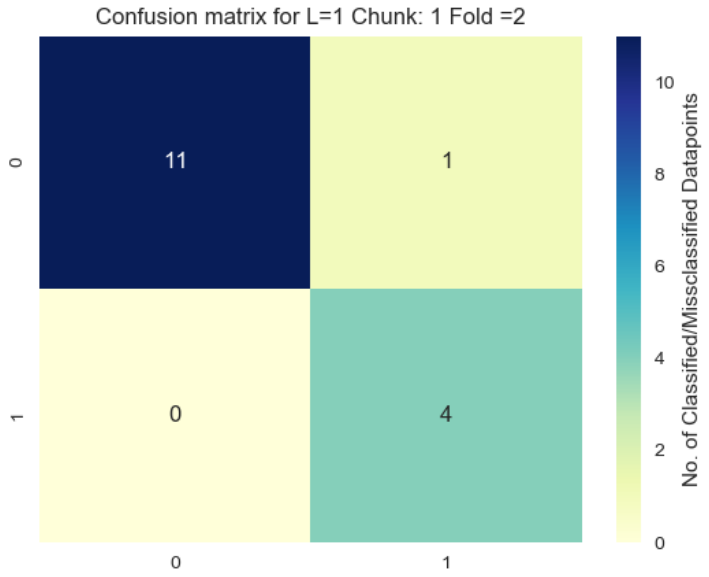
:

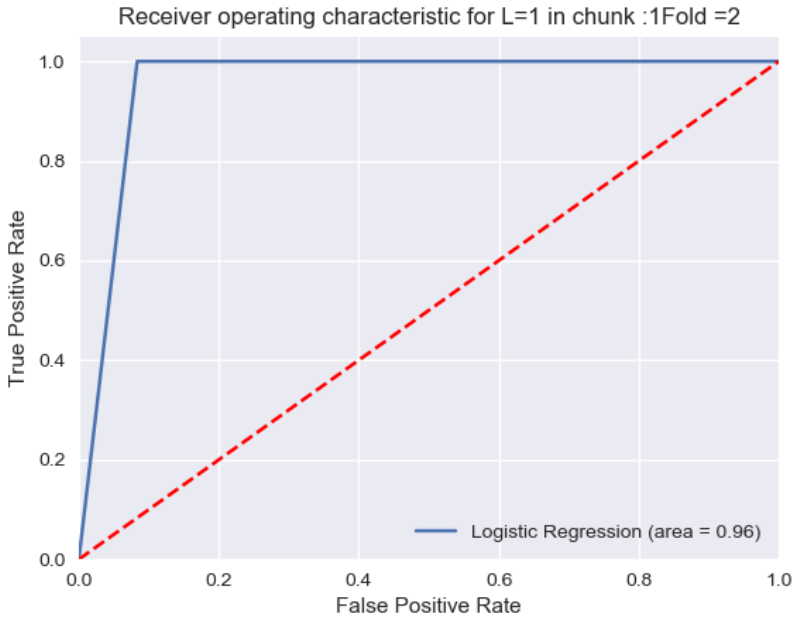
USC ID: 5423-4179-55



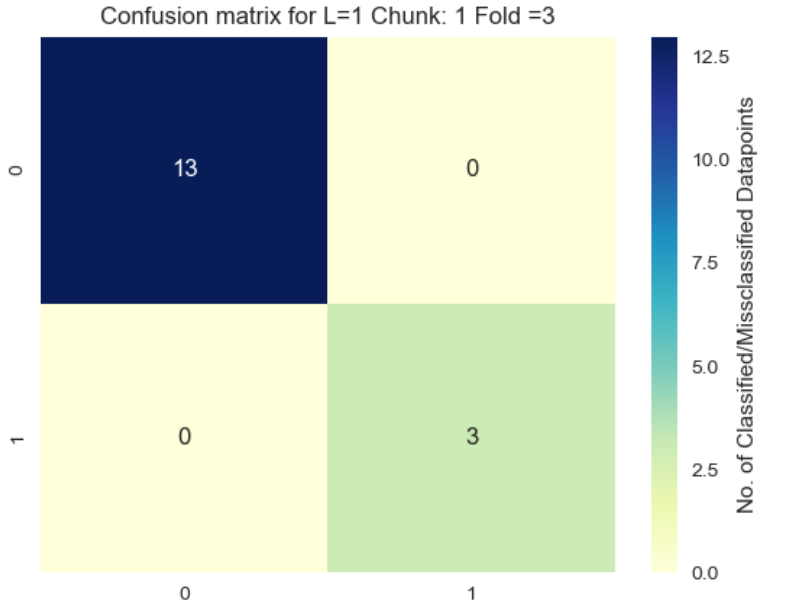


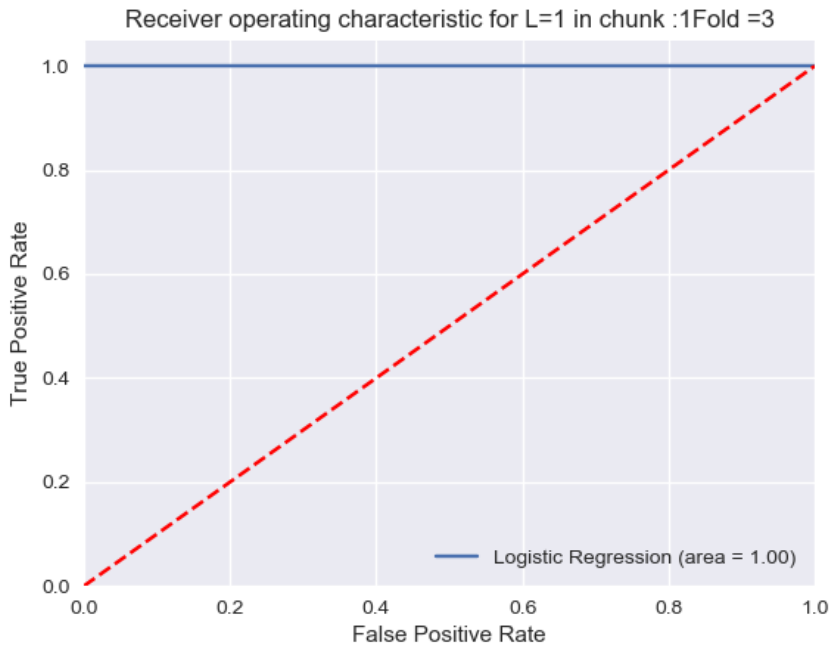
USC ID: 5423-4179-55



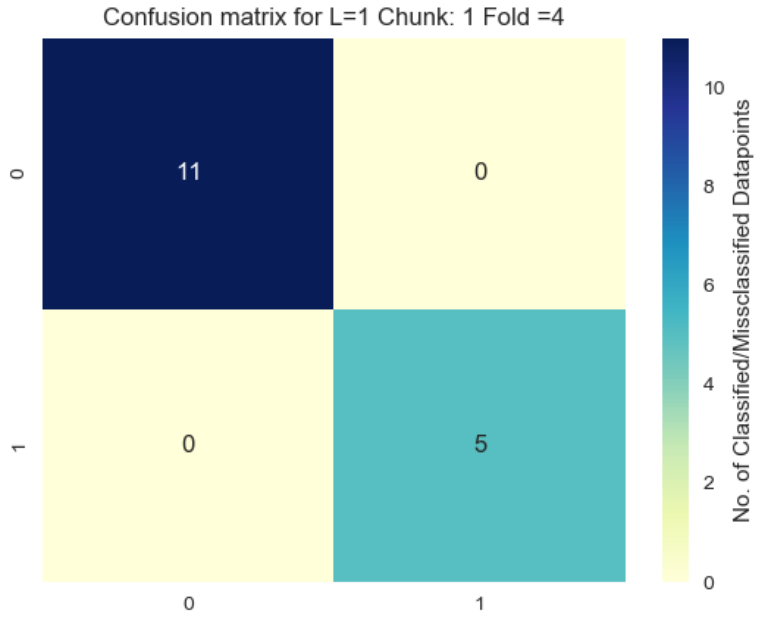


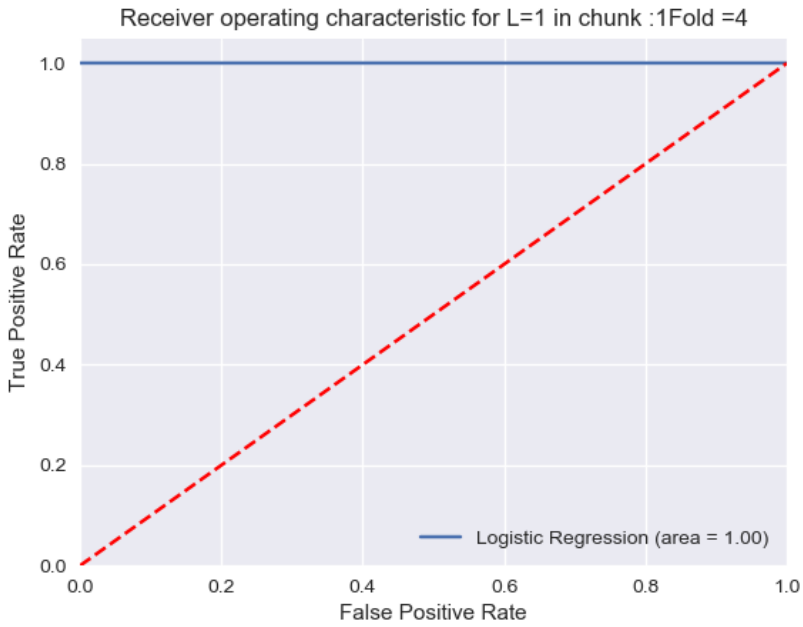
USC ID: 5423-4179-55





USC ID: 5423-4179-55





USC ID: 5423-4179-55

v)

I trained the model on my entire TRAIN set and then used the model to predict on the TEST set.
I also split the test set in the same L as I did the train set and extracted the features that were significant in my observations during **1d(iii)**. I extracted them and encoded them into a new data frame, which I then used for this question.

I observed the following accuracy and error rates.

Net Accuracy for L= 1 is [0.8387096774193549]
Net Error for L= 2 is [0.2]

Out of curiosity I decided to check the accuracy and error rates for L= {1,2,3….,10}

I received the following results:

Net Accuracy for L= 1 is [0.95]
Net Accuracy for L= 2 is [0.95]
Net Accuracy for L= 3 is [0.8888888888888888]
Net Accuracy for L= 4 is [0.8999999999999999]
Net Accuracy for L= 5 is [0.9]
Net Accuracy for L= 6 is [0.888888888888889]
Net Accuracy for L= 7 is [0.7857142857142857]
Net Accuracy for L= 8 is [0.875]
Net Accuracy for L= 9 is [0.9444444444444444]
Net Accuracy for L= 10 is [0.85]

Net Error for L= 1 is [0.05]
Net Error for L= 2 is [0.05]
Net Error for L= 3 is [0.1111111111111111]
Net Error for L= 4 is [0.1]
Net Error for L= 5 is [0.1]
Net Error for L= 6 is [0.1111111111111111]
Net Error for L= 7 is [0.21428571428571427]
Net Error for L= 8 is [0.125]
Net Error for L= 9 is [0.05555555555555555]
Net Error for L= 10 is [0.15]

Clearly, the best accuracy and error rate was obtained when L=1.

Now, to compare them with the accuracy I obtained for the train set.

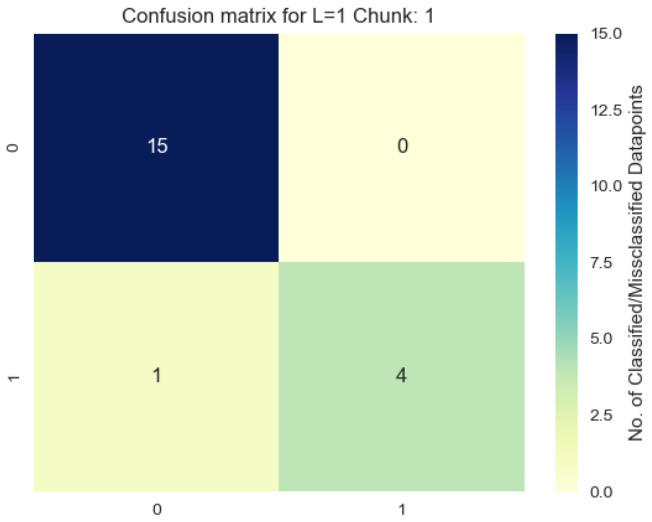
Net Accuracy for L= 1 is [0.95] (TEST SET)

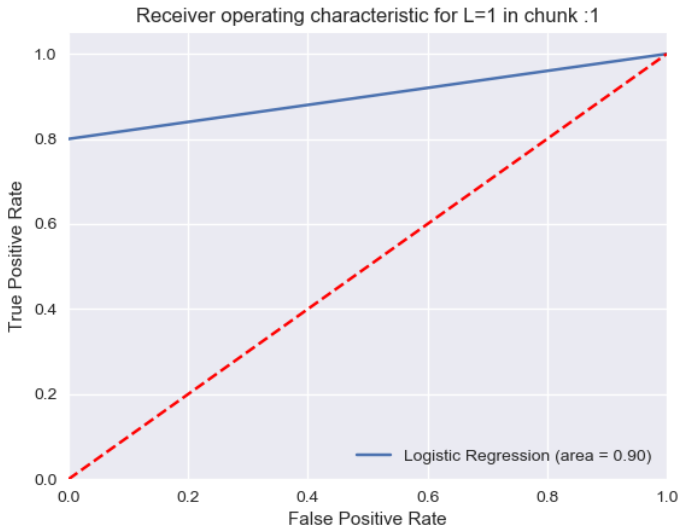
Final Score for L=1: 0.9625 (TRAIN SET)

This comes very close to our train score, this tells us that our model has performed well on the testing data.

vi and vii)

I obtained the following results for confusion matrix and ROC for the test set.





We had just 1 misclassification.
An AUC score of 1 is a result of 0% overlapping degree, and a score of about 0.90 signifies 20% overlapping in data.

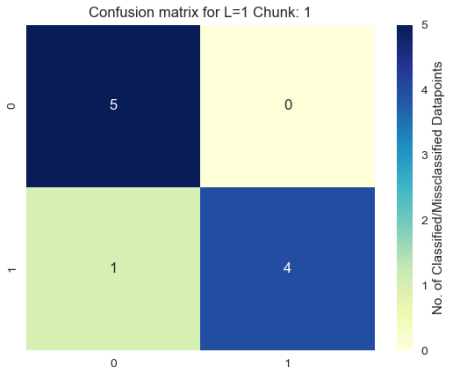
Thus, we can say that we have about 20% overlapping in our data.
Higher overlapping can be caused due to non-separability among classes. Thus, we can conclude that there exists some inseparability among the classes.

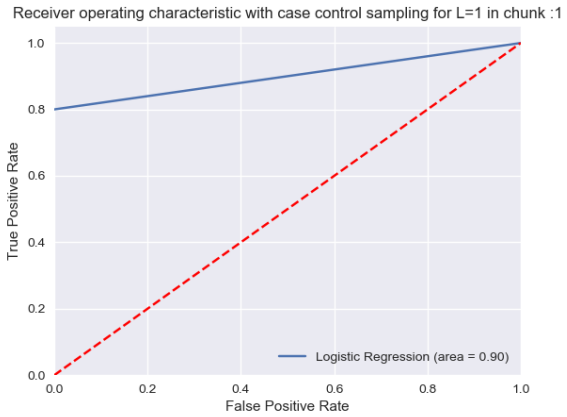
Imbalance:

Clearly, there exists some imbalance in the data as the non-bending class(majority) heavily outnumbers the bending samples(minority). Ratio(1:3)

I have used a randomundersampler in sklearn to under-sample the majority class(0).

After doing this I received the following results.





As observable through the confusion matrix, both my classes are now well-balanced (1:1 ratio)

Part (e) Binary Classification Using L_1-penalized logistic regression

The code tries 10 values of C and cross validates on them as well as on the train set. I obtained the following results for C,Cs(values tried), regression coef, and score for that value of L.

```
In L=1
In chunk: 1
starting: 0
ending 69
[fold 0] score: 0.92857
regression coef-values
[[  0.        -12.1252036   -4.39775649  -2.49544013   0.
    0.32517549   0.          0.          6.5044615    0.
    4.38505777   0.         23.10915204   2.40513284 -10.9224786
    0.        ]]
C [21.5443469]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 1] score: 1.00000
regression coef-values
[[ 0.        -6.62516156  0.          0.          0.          0.
   0.          0.          0.          0.          4.73249187  0.
   5.84474856  0.         -4.04988078  0.        ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 2] score: 1.00000
regression coef-values
[[ 0.        -6.8021539   0.          0.          0.          0.
   0.          0.          0.          0.          6.13451957  0.
   5.1127634   0.         -4.78584327  0.        ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 3] score: 0.85714
regression coef-values
[[  0.        -19.31260283   0.          0.          0.
    0.          0.          3.38400268   7.21266712  -8.68577763
   11.30185231   0.          7.96924647   0.         -3.3104838
    0.        ]]
```

```
C [21.5443469]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 4] score: 0.84615
regression coef-values
[[ 0.         0.         0.         0.         0.         0.
   0.         0.         0.         0.        12.41932742  0.
   3.56433466  0.        -1.53918302  0.       ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]


Score for L= 1  is 0.9263736263736264




################################################################################IN L= 2
################################
In chunk: 1
starting: 0
ending 34

 [fold 0] score: 0.85714
regression coef-values
[[ 0.         0.         0.         0.         0.         0.
   0.         3.55412309  0.         0.         0.         1.69914843
   2.26845942  0.        -3.87446698  0.       ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 1] score: 1.00000
regression coef-values
[[ 0.         0.         0.         0.         0.         0.
   0.         0.         0.         0.         9.50527712  0.25978022
   0.50692354  0.        -1.23946949  0.       ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
```

```
[fold 2] score: 1.00000
regression coef-values
[[ 0.         0.         0.         0.         0.         0.
   0.         0.         0.         0.         8.65694891  0.
   1.14264867  0.        -1.98429351  0.       ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 3] score: 1.00000
regression coef-values
[[ 0.         0.         0.         0.         0.         0.
   0.         0.         0.         0.         7.22082444  0.
   1.7804053   0.        -2.0784629   0.       ]]
C [2.7825594]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 4] score: 0.83333
regression coef-values
[[ 0.         0.         0.        -3.24808571  0.         0.
   0.         1.95655585  0.        -0.30725511 34.26166273  5.36392722
   0.         0.        -8.85624431  0.       ]]
C [1291.54966501]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]


In chunk: 2
starting: 34
ending 68
[fold 0] score: 0.85714
regression coef-values
[[ 0.00000000e+00 -1.12607915e+01 -5.97738231e-03  0.00000000e+00
  -5.79228933e+00  0.00000000e+00  0.00000000e+00 -1.67762719e+00
   0.00000000e+00  0.00000000e+00  9.23310277e-01  0.00000000e+00
   2.17516695e+01  1.75397408e+00  0.00000000e+00  0.00000000e+00]]
C [21.5443469]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 1] score: 0.71429
regression coef-values
```

```
[[  0.        -17.23811824  -3.8199112   -3.8733372   -4.88202328
    0.          0.         -3.46960887   0.          0.
    0.          0.         22.21070089  10.97307841   0.
    0.        ]]
C [166.81005372]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 2] score: 1.00000
regression coef-values
[[  0.         -4.73180721 -23.410815   -15.52928173   0.
    0.        -11.06718438  -3.78898463   0.          0.
   15.45644322   0.         46.16112317   8.27758909  -5.90008187
    0.        ]]
C [1291.54966501]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 3] score: 1.00000
regression coef-values
[[ 0.        -4.71824714 -7.00629675 -3.85423179  0.          0.
   0.        -1.95328685  0.          0.          2.27722907  0.
  20.28642485  2.76442853 -3.93326598  0.        ]]
C [21.5443469]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
[fold 4] score: 0.50000
regression coef-values
[[ 0.        -5.91024503 -7.27102423 -3.55432687  0.          0.
   0.         0.         0.         0.         0.          0.
  24.76786885  4.32852608  0.         0.        ]]
C [21.5443469]
CS_ [1.00000000e-04 7.74263683e-04 5.99484250e-03 4.64158883e-02
 3.59381366e-01 2.78255940e+00 2.15443469e+01 1.66810054e+02
 1.29154967e+03 1.00000000e+04]
```

Score for L= 2 is 0.8761904761904762

Similarly, results were obtained for other values of L.

SCORES:

Score for L= 1 is 0.9663736263736264
Score for L= 2 is 0.8761904761904762
Score for L= 3 is 0.9166666666666666
Score for L= 4 is 0.8666666666666667
Score for L= 5 is 0.7966666666666666
Score for L= 6 is 0.9165646457562424
Score for L= 7 is 0.8875964878999999
Score for L= 8 is 0.8245978458974545
Score for L= 9 is 0.7823235665645688
Score for L= 10 is 0.8125685556987878

Again, we observed the best results when **L is 1.**

ii)

With p-values

Final Score for L=1: 0.9625 (TRAIN SET)

With L-1 penalization

Score for L= 1 is 0.9663736263736264

The L-1 penalization is slightly better in the accuracy. Due to the inbuilt libraries this was much easier to use, it also shuns the features not contributing by setting their regression coefficients to 0 which possibly has resulted in better accuracy.

Personally, I found the L-1 penalization classification easier and simpler to use.

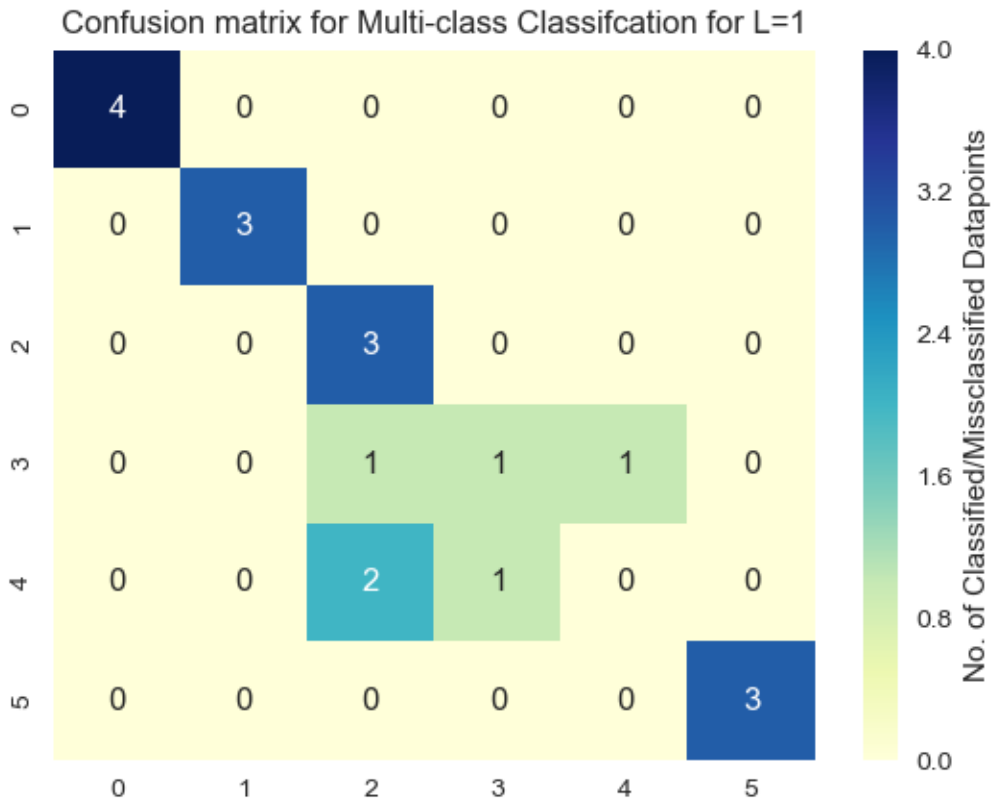
Part (f) Multi-class Classification (The Realistic Case)

(i)

For L=1 (previously obtained)

The test -error is: 0.2631579

With accuracy of: 0.7368421052631579



The confusion matrix for the same can be plotted using seaborn.

ii)

Naïve Bayes with Gaussian prior:

score 0.9473684210526315

The confusion matrix is obtained for the same.

Only 1 data point is misclassified, which is impressive.

Out of curiosity I tried using other values of L.

Results:

Gaussian Naive Bayes for L= 1 :0.9473684210526315

Gaussian Naive Bayes for L= 2 :0.8888888888888888

Gaussian Naive Bayes for L= 3 :0.8333333333333334

USC ID: 5423-4179-55

Homework2-INF552-Arpit Sharma

Gaussian Naive Bayes for L= 4 :0.8125

Gaussian Naive Bayes for L= 5 :0.6666666666666666

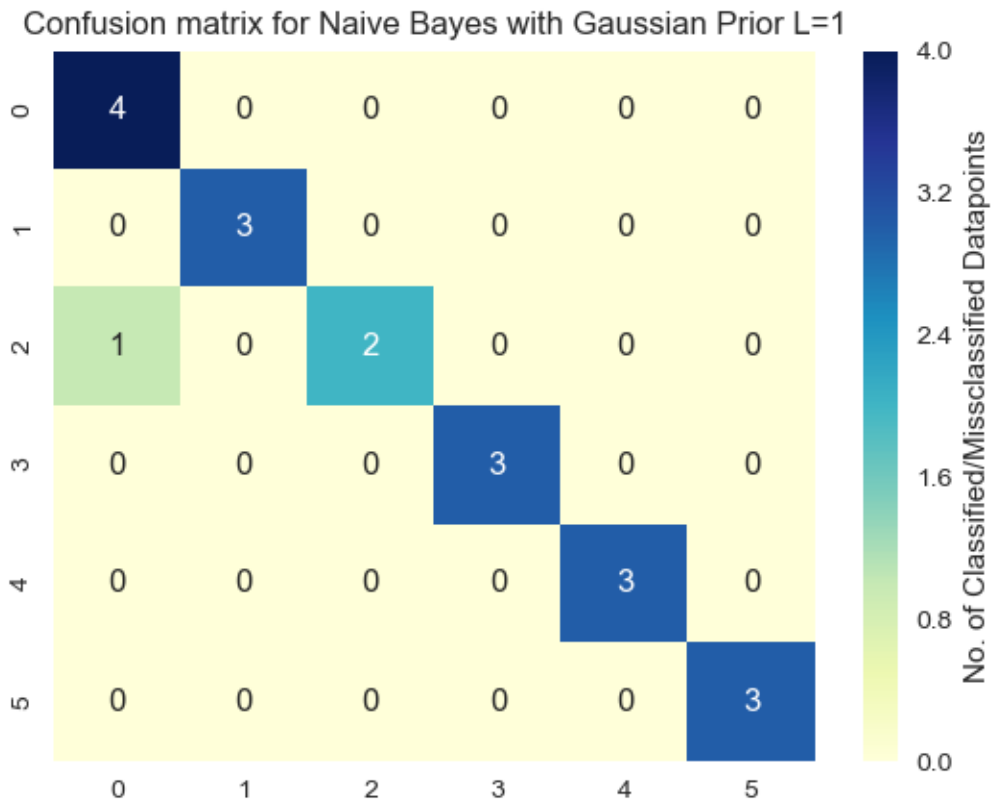
Gaussian Naive Bayes for L= 6 :0.7777777777777777

Gaussian Naive Bayes for L= 7 :0.42857142857142855

Gaussian Naive Bayes for L= 8 :0.5

Gaussian Naive Bayes for L= 9 :0.7222222222222222

Gaussian Naive Bayes for L= 10: 0.2



Naïve Bayes with Multinomial prior

Score for L=1:

0.7368421052631579

The Confusion matrix obtained is shown below:

USC ID: 5423-4179-55

Out of curiosity I tried for other L values and obtained the following results.

Naive bayes with Multinomial classifier for L= 1: 0.7368421052631579

Naive bayes with Multinomial classifier for L= 2: 0.7222222222222222

Naive bayes with Multinomial classifier for L= 3: 0.8333333333333334

Naive bayes with Multinomial classifier for L= 4: 0.6875

Naive bayes with Multinomial classifier for L= 5: 0.6666666666666666

Naive bayes with Multinomial classifier for L= 6: 0.7777777777777777

Naive bayes with Multinomial classifier for L= 7: 0.42857142857142855

Naive bayes with Multinomial classifier for L= 8: 0.625

Naive bayes with Multinomial classifier for L= 9: 0.8888888888888888

Naive bayes with Multinomial classifier for L= 10: 0.1

USC ID: 5423-4179-55

L=9 performed the best.

iii)

Naïve Bayes with Gaussian prior performed best with almost a ***95%*** accuracy of predictions.

The multinomial prior with L=9 also outperformed L-1 penalized classification.

I thus, would use this method for this dataset to perform multi-class classification.

Question 2(ISLR 3.7.4)

a)

It is difficult to say, we need more information for this. Although, as the true relationship is linear, we expect to observe the nature of the linear regressor to be the closest to the True model. Thus, the train RSS for linear model are expected to be lower than the cubic regressor.

b)

Again, we need more information on the test set. But as the cubic regressor will have a higher test RSS, as it would overfit on the training dataset and have more error than the linear model.

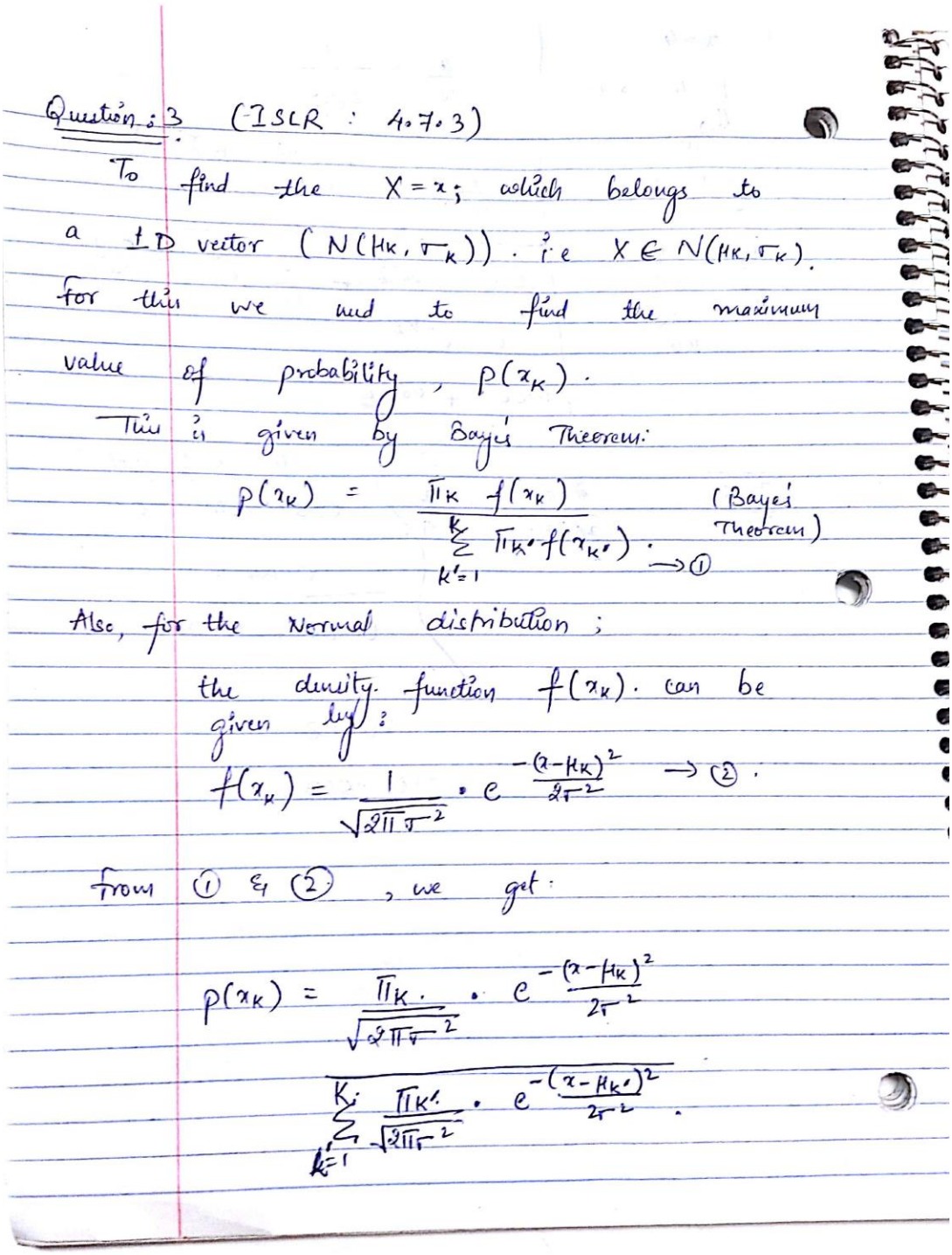
c)

The cubic model will have a lower train RSS as it will overfit by trying to reach all data points due to the wiggly curve,

d)

There is more information required to tell which will be lower. We do not know how far away from linear the true nature of the model is. If it closer to linear then linear model test RSS will be lower than the cubic. If it is closer to the cubic model then the cubic test RSS will be lower. Unless the nature of the model is clearly specified it is difficult to say which test RSS is lower.

Question 3: (ISLR 3.7.4)



Question: 3 (ISLR : 4.7.3)

To find the $X = x_i$ which belongs to a 1D vector $(N(\mu_K, \sigma_K))$. i.e $X \in N(\mu_K, \sigma_K)$.

for this we need to find the maximum value of probability, $p(x_K)$.

This is given by Bayes Theorem:

$$p(x_K) = \frac{\pi_K \, f(x_K)}{\sum_{k'=1}^{K} \pi_{K'} \cdot f(x_{K'})} \quad \text{(Bayes Theorem)} \rightarrow (1)$$

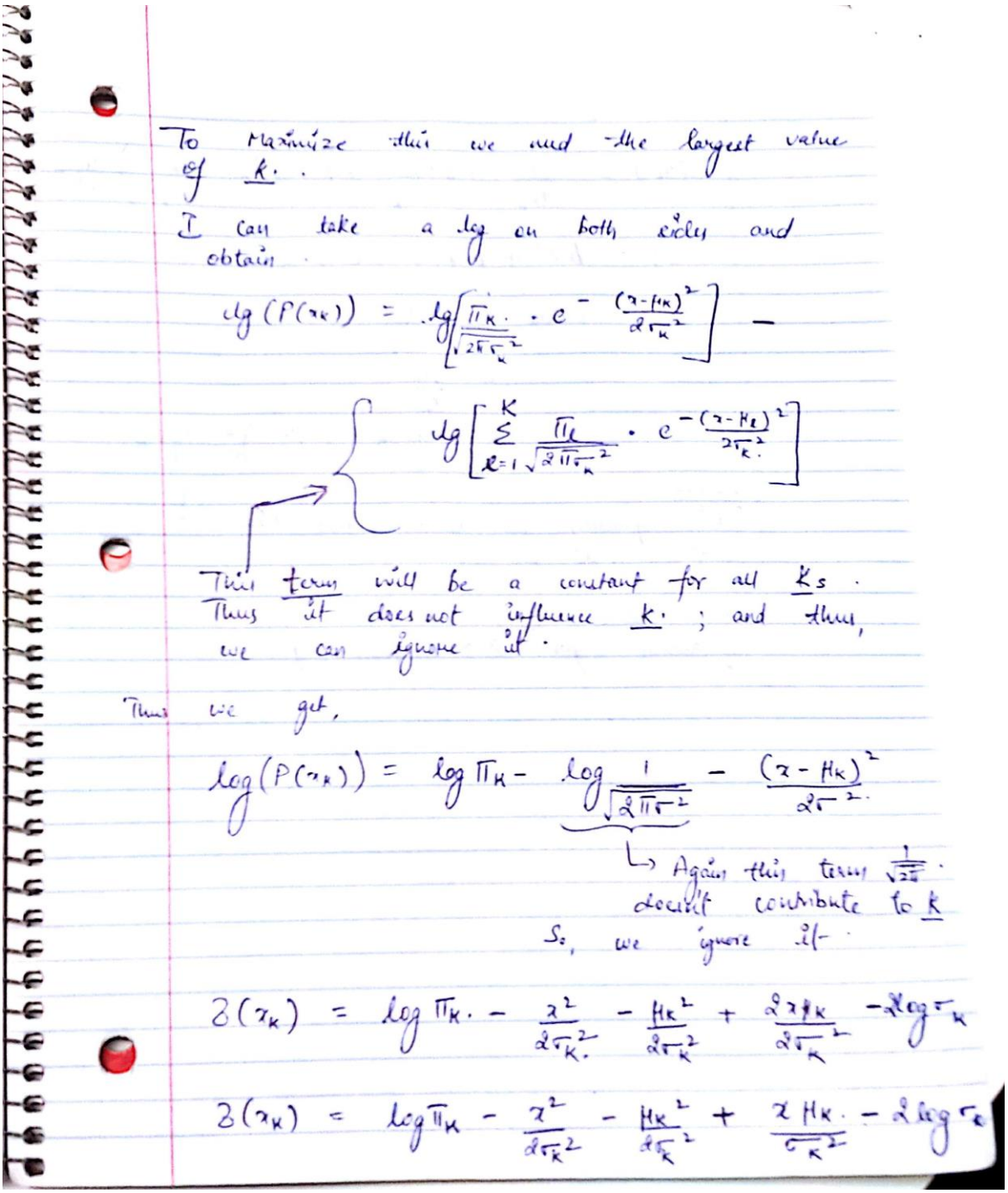
Also, for the Normal distribution;

the density function $f(x_K)$ can be given by:

$$f(x_K) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-(x-\mu_K)^2}{2\sigma^2}} \rightarrow (2).$$

From (1) & (2), we get:

$$p(x_K) = \frac{\frac{\pi_K}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-(x-\mu_K)^2}{2\sigma^2}}}{\sum_{k'=1}^{K} \frac{\pi_{K'}}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-(x-\mu_{K'})^2}{2\sigma^2}}}.$$



To Maximize this we need the largest value of k.

I can take a log on both sides and obtain

$$\log(P(x_k)) = \log\left[\frac{\pi_k}{\sqrt{2\pi\sigma_k^2}} \cdot e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}\right] - \log\left[\sum_{k=1}^{K} \frac{\pi_k}{\sqrt{2\pi\sigma_k^2}} \cdot e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}\right]$$

This term will be a constant for all ks. Thus it does not influence k; and thus, we can ignore it.

Thus we get,

$$\log(P(x_k)) = \log \pi_k - \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(x-\mu_k)^2}{2\sigma^2}$$

↳ Again this term $\frac{1}{\sqrt{2\pi}}$ doesn't contribute to k So, we ignore it.

$$\delta(x_k) = \log \pi_k - \frac{x^2}{2\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \frac{2x\mu_k}{2\sigma_k^2} - 2\log\sigma_k$$

$$\delta(x_k) = \log \pi_k - \frac{x^2}{2\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \frac{x\mu_k}{\sigma_k^2} - 2\log\sigma_k$$

$$\delta(x_k) = \frac{-x^2}{2\sigma_k^2} + x\frac{\mu_k}{\sigma_k^2} + \left(\log \pi_k - \frac{\mu_k^2}{2\sigma_k^2}\right) - \log \sigma_k$$

$$\delta(x_k) = bx^2 + ax + (c);$$

where $b = -\frac{1}{2\sigma_k^2}$,

$a = \frac{\mu_k}{\sigma_k^2}$

$c = \log \pi_k + \left(\frac{-\mu_k^2}{2\sigma_k^2}\right) - \log \sigma_k$.

This equation is clearly not linear.
Infact, it is quadratic.

Thus, Naive Bayes classifier here is quadratic for $X_k \sim N(\mu_k, \sigma_k)$.

Question 4 (ISLR4.7.7)

Question 4 (ISLR 4.7.7)

Here $K = 2$. (Yes or No).
Let $k=1$ signify Yes and
$k=2$ signify No.

$\bar{X}$ for $k=1 = 10$, so $\mu_1 = 10$.

& $\mu_2 = 0$.

$\hat{\sigma}^2 = 36$, And $x = 4$.

$\Pi_1 = \frac{80}{100}$, $\Pi_2 = \frac{20}{100}$.

$$P(x_k) = \frac{\Pi_k \cdot f_k(x_k)}{\sum_{l=1}^{K} \Pi_l f(x_l)}.$$

So, $f_k(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$

Thus plugging in the values; we get.

$$P(k=1 \mid x=4) = \frac{0.8 \cdot \frac{1}{\sqrt{2\pi \cdot 36}} \cdot e^{\frac{-(4-10)^2}{2\times 36}}}{0.8 \times \frac{1}{\sqrt{2\pi \cdot 36}} \cdot e^{\frac{-(4-10)^2}{72}} + \frac{0.2}{\sqrt{2\pi \cdot 36}} \cdot e^{\frac{-(4-0)^2}{72}}}$$

$$= \frac{0.8 \times 0.6065}{0.8 \times 0.6065 + 0.2 \times 0.801}$$

$$\boxed{P(k=1 \mid x=4) = 0.7519.}$$ is the probability.