Name: Subhiksha Rani

USC ID: 9907399097

Homework – 3 Report

1)

- **a)** (Corrected the bending2\dataset4 manually in excel)
- **b)** Divided the data into training and test sets.
- c) Feature Extraction
    - i) Types of time-domain features usually used in time series:
        - (1) Maximum
        - (2) Minimum
        - (3) Mean
        - (4) Median
        - (5) Mode
        - (6) Standard Deviation
        - (7) Variance
        - (8) First-quartile
        - (9) Third-quartile
        - (10) Slope
        - (11) Peak-to-peak
        - (12) Zero cross rating
        - (13) Autocorrelation
        - (14) Cross correlation
        - (15) Linear correlation coefficient

ii) Time-domain features table:

	Instance	Min1	Max1	Mean1	Median1	SD1	1st quart1	3rd quart1	Min2	Max2	...	SD5	1st quart5	3rd quart5	Min6	Max6	Mean6	Median6	SD6	1st quart6	3rd quart6
0	1.0	37.25	45.00	40.624792	40.500	1.476967	39.2500	42.0000	0.0	1.30	...	2.188449	33.0000	36.0000	0.00	1.92	0.570583	0.430	0.582915	0.0000	1.3000
1	2.0	38.00	45.67	42.812812	42.500	1.435550	42.0000	43.6700	0.0	1.22	...	1.995255	32.0000	34.5000	0.00	3.11	0.571083	0.430	0.601010	0.0000	1.3000
2	3.0	35.00	47.40	43.954500	44.330	1.558835	43.0000	45.0000	0.0	1.70	...	1.999604	35.3625	36.5000	0.00	1.79	0.493292	0.430	0.513506	0.0000	0.9400
3	4.0	33.00	47.75	42.179813	43.500	3.670666	39.1500	45.0000	0.0	3.00	...	3.849448	30.4575	36.3300	0.00	2.18	0.613521	0.500	0.524317	0.0000	1.0000
4	5.0	33.00	45.75	41.678063	41.750	2.243490	41.3300	42.7500	0.0	2.83	...	2.411026	28.4575	31.2500	0.00	1.79	0.383292	0.430	0.389164	0.0000	0.5000
5	6.0	37.00	48.00	43.454958	43.250	1.386098	42.5000	45.0000	0.0	1.58	...	2.488862	22.2500	24.0000	0.00	5.26	0.679646	0.500	0.622534	0.4300	0.8700
6	7.0	36.25	48.00	43.969125	44.500	1.618364	43.3100	44.6700	0.0	1.50	...	3.318301	20.5000	23.7500	0.00	2.96	0.555313	0.490	0.487826	0.0000	0.8300
7	8.0	12.75	51.00	24.562958	24.250	3.737514	23.1875	26.5000	0.0	6.87	...	3.693786	20.5000	27.0000	0.00	4.97	0.700188	0.500	0.693720	0.4300	0.8700
8	9.0	0.00	42.75	27.464604	28.000	3.583582	25.5000	30.0000	0.0	7.76	...	5.053642	15.0000	20.7500	0.00	6.76	1.122125	0.830	1.012342	0.4700	1.3000
9	10.0	21.00	50.00	32.586208	33.000	6.238143	26.1875	34.5000	0.0	9.90	...	5.032424	17.6700	23.5000	0.00	13.61	1.162042	0.830	1.332980	0.4700	1.3000
10	11.0	27.50	33.00	29.881938	30.000	1.153837	29.0000	30.2700	0.0	1.00	...	1.745970	17.0000	19.0000	0.00	6.40	0.701625	0.710	0.481103	0.4700	0.9400
11	12.0	19.00	45.50	30.938104	29.000	7.684146	26.7500	38.0000	0.0	6.40	...	5.845911	15.0000	20.8125	0.00	6.73	1.107354	0.830	1.080842	0.4700	1.3000
12	13.0	25.00	47.50	31.058250	29.710	4.829794	27.5000	31.8125	0.0	6.38	...	7.853427	9.0000	18.3125	0.00	4.92	1.098104	0.940	0.831480	0.5000	1.3000
13	14.0	24.25	45.00	37.177042	36.250	3.581301	34.5000	40.2500	0.0	8.58	...	2.890347	17.9500	21.7500	0.00	9.34	2.921729	2.500	1.852600	1.5000	3.9000
14	15.0	28.75	44.75	37.561188	36.875	3.226507	35.2500	40.2500	0.0	9.91	...	2.727377	18.0000	21.5000	0.00	9.62	2.765896	2.450	1.769203	1.4100	3.7700
15	16.0	22.00	44.67	37.058708	36.000	3.710180	34.5000	40.0625	0.0	14.17	...	3.537144	16.0000	21.0000	0.00	8.55	2.983750	2.570	1.815730	1.5000	4.1500
16	17.0	19.00	44.00	36.228396	36.000	3.528617	34.0000	39.0000	0.0	12.28	...	3.166655	14.0000	18.0625	0.00	9.98	3.480687	3.340	1.827769	2.1025	4.5500
17	18.0	26.50	44.33	36.687292	36.000	3.529404	34.2500	39.3725	0.0	12.89	...	2.978238	14.6700	18.5000	0.00	8.19	3.073312	2.690	1.629675	1.9125	4.0875
18	19.0	25.33	45.00	37.114312	36.250	3.710385	34.5000	40.2500	0.0	10.84	...	2.847876	14.7500	18.5000	0.00	9.50	3.076354	2.770	1.824534	1.7000	4.0375
19	20.0	26.75	44.75	36.863375	36.330	3.555787	34.5000	39.7500	0.0	11.68	...	2.655906	15.0000	18.6700	0.00	8.81	2.773312	2.590	1.569919	1.6400	3.6325
20	21.0	26.25	44.25	36.957458	36.290	3.434863	34.5000	40.2500	0.0	8.64	...	2.851673	14.0000	18.2500	0.00	8.34	2.934625	2.525	1.631380	1.6600	4.0300
21	22.0	27.75	44.67	37.142359	36.330	3.762442	34.0000	40.5000	0.0	10.76	...	2.687173	15.0000	18.7500	0.00	8.75	2.825720	2.590	1.637312	1.5900	3.7400
22	23.0	27.00	45.00	36.819521	36.000	3.900459	33.7500	40.2500	0.0	10.47	...	2.781030	15.5000	19.2700	0.00	8.99	2.887562	2.525	1.723094	1.5600	3.7700
23	24.0	27.00	44.33	36.541667	36.000	4.018922	33.2500	39.8125	0.0	10.43	...	3.088141	15.0000	19.5000	0.00	9.18	3.225458	2.870	1.769758	1.8850	4.2625
24	25.0	18.50	44.25	35.752354	36.000	4.614802	33.0000	39.3300	0.0	12.60	...	3.120057	14.0000	18.0625	0.00	9.39	3.069667	2.770	1.748326	1.7975	4.0600
25	26.0	19.00	43.75	35.879875	36.000	4.614878	33.0000	39.5000	0.0	11.20	...	3.537635	14.7500	19.6900	0.00	8.50	3.093021	2.930	1.626034	1.8900	4.0600
26	27.0	23.33	43.50	36.248768	36.750	3.824632	33.4150	39.2500	0.0	9.71	...	3.617405	15.7500	21.0000	0.00	11.15	3.532463	3.110	1.965267	2.1700	4.6250
27	28.0	24.25	45.00	37.177042	36.250	3.581301	34.5000	40.2500	0.0	8.58	...	2.890347	17.9500	21.7500	0.00	9.34	2.921729	2.500	1.852600	1.5000	3.9000
28	29.0	23.50	30.00	27.716375	27.500	1.442253	27.0000	29.0000	0.0	1.79	...	4.074511	5.5000	10.7500	0.00	4.50	0.734271	0.710	0.613688	0.4300	1.0000
29	30.0	24.75	48.33	44.182937	48.000	7.495615	48.0000	48.0000	0.0	3.11	...	3.274539	2.0000	5.5425	0.00	3.91	0.692771	0.500	0.675781	0.3225	0.9400

iii) Standard Deviation of all the features:

	Min SD	Max SD	Mean SD	Median SD	Std Devi SD	1st quart SD	3rd quart SD
0	9.569975	4.394362	5.335686	5.440054	1.772187	6.153874	5.138925
1	0.000000	5.062729	1.574203	1.412293	0.884135	0.946386	2.125399
2	2.956462	4.875137	4.008235	4.036396	0.946654	4.220658	4.171628
3	0.000000	2.183625	1.166135	1.145985	0.458241	0.843405	1.552504
4	6.124001	5.741238	5.675543	5.813782	1.024893	6.096465	5.531720
5	0.045838	2.518921	1.154848	1.086474	0.517601	0.758687	1.523739

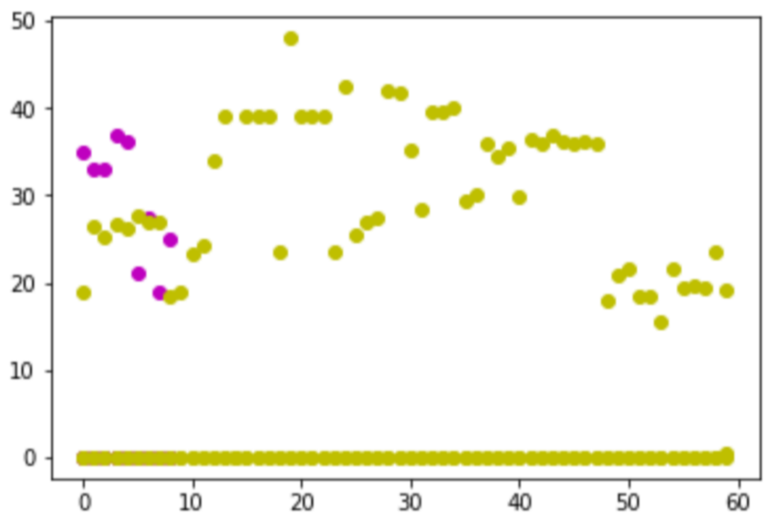
Building a 90% bootstrap confidence interval for the standard deviation of each feature.

```
90% Confidence Interval of  Min SD : (0.993127045404682, 5.724069248223067)
90% Confidence Interval of  Max SD : (3.248443634325581, 4.953024965798376)
90% Confidence Interval of  Mean SD : (1.9158569678215482, 4.53281541241536)
90% Confidence Interval of  Median SD : (1.7410372001391512, 4.504282258240261)
90% Confidence Interval of  Std Devi SD : (0.6471609499515683, 1.2375021080287787)
90% Confidence Interval of  1st quart SD : (1.7002441396095807, 4.927457002221202)
90% Confidence Interval of  3rd quart SD : (2.2313409649702245, 4.510903677961358)
```

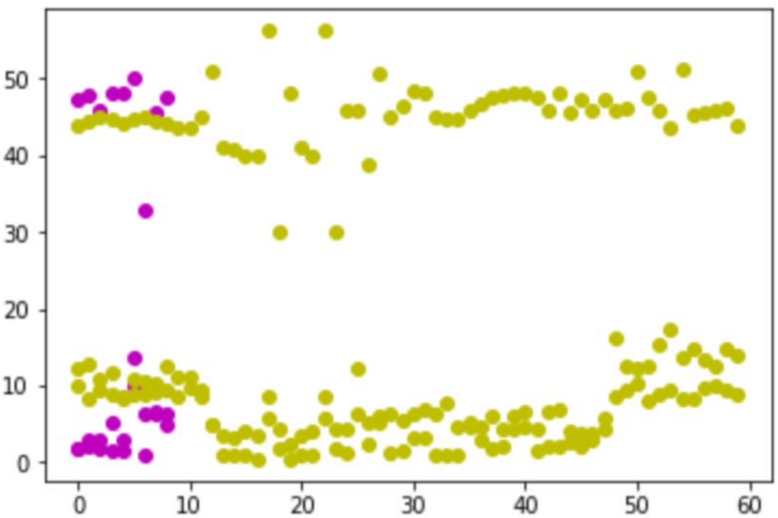
iv) 3 most important time-domain features according to me is Min, Max and Mean because these are the 3 most basic features and it can be used in almost all the scenarios to obtain accurate results. These features can be used to fit our model.

d) Binary Classification using logistic regression

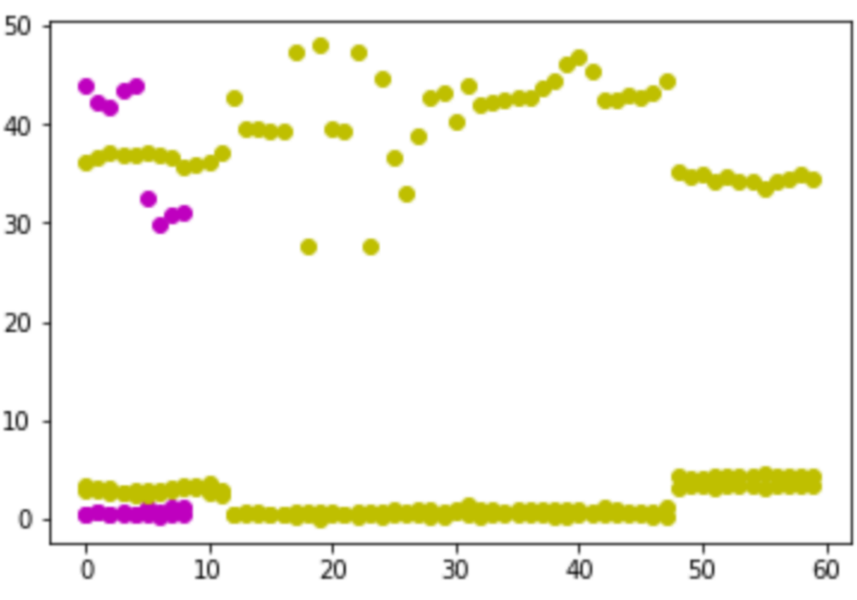
i) Scatter plot of Minimum feature, Pink denotes Bending & yellow denotes Other activities.



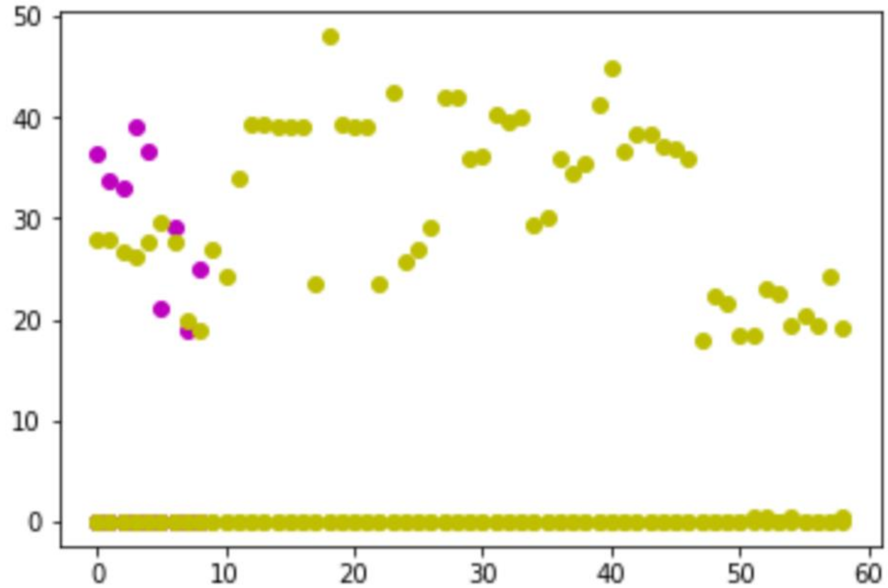
Scatter plot of Maximum feature, Pink denotes Bending & yellow denotes Other activities.



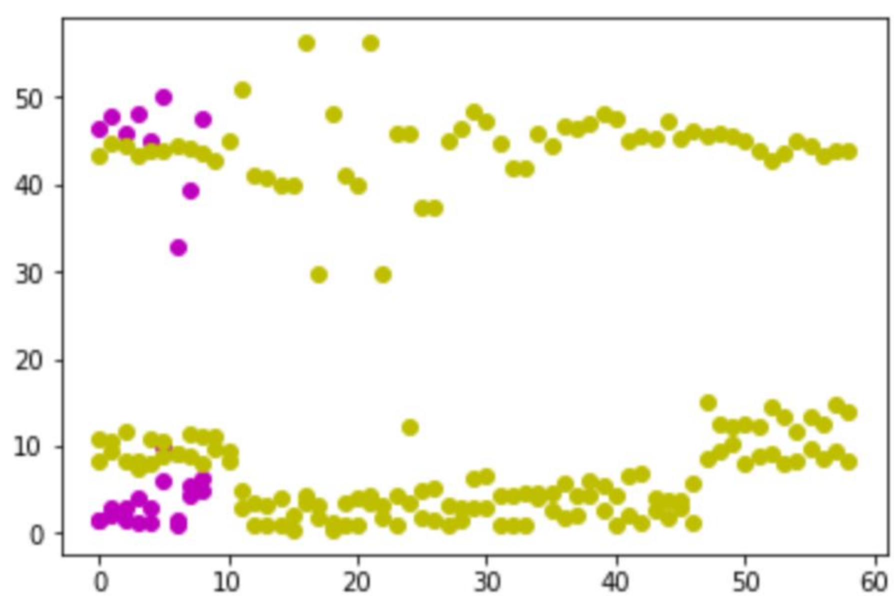
Scatter plot of Mean feature, Pink denotes Bending & yellow denotes Other activities.



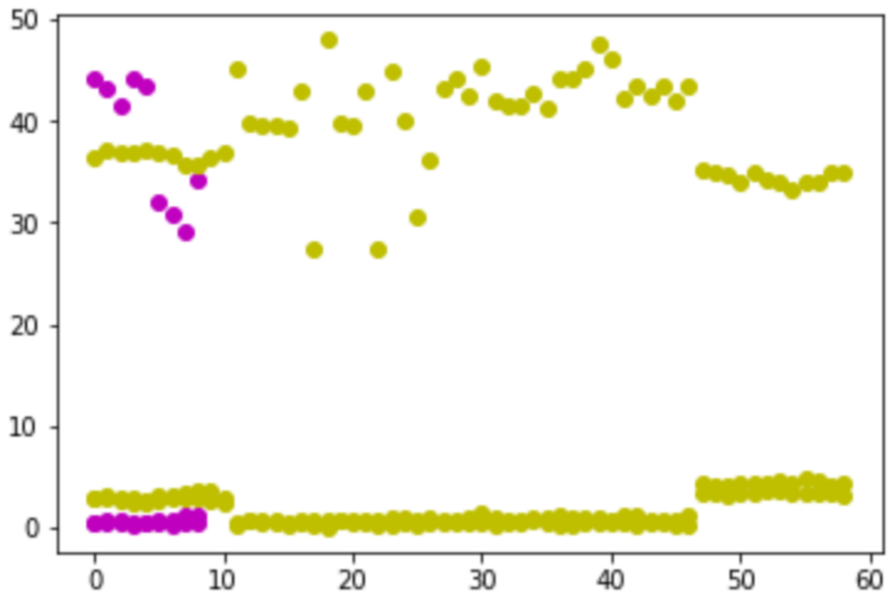
ii) Scatter plot of Minimum feature, Pink denotes Bending & yellow denotes Other activities.



Scatter plot of Maximum feature, Pink denotes Bending & yellow denotes Other activities.



Scatter plot of Mean feature, Pink denotes Bending & yellow denotes Other activities.



Comparing the above graphs to the ones plotted in 1(d)i, we do not see any significant difference between the plots. The graphs look similar for each feature.

iii) Accuracy obtained in 5-fold cross-validation for Training set divided into l = {1,2,…20}:

```
5-fold cross validation average accuracy for l= 1 : 0.986
5-fold cross validation average accuracy for l= 2 : 0.900
5-fold cross validation average accuracy for l= 3 : 0.869
5-fold cross validation average accuracy for l= 4 : 0.891
5-fold cross validation average accuracy for l= 5 : 0.846
5-fold cross validation average accuracy for l= 6 : 0.862
5-fold cross validation average accuracy for l= 7 : 0.870
5-fold cross validation average accuracy for l= 8 : 0.866
5-fold cross validation average accuracy for l= 9 : 0.871
5-fold cross validation average accuracy for l= 10 : 0.912
5-fold cross validation average accuracy for l= 11 : 0.896
5-fold cross validation average accuracy for l= 12 : 0.917
5-fold cross validation average accuracy for l= 13 : 0.869
5-fold cross validation average accuracy for l= 14 : 0.896
5-fold cross validation average accuracy for l= 15 : 0.912
5-fold cross validation average accuracy for l= 16 : 0.890
5-fold cross validation average accuracy for l= 17 : 0.914
5-fold cross validation average accuracy for l= 18 : 0.862
5-fold cross validation average accuracy for l= 19 : 0.853
5-fold cross validation average accuracy for l= 20 : 0.871
```

From the above calculated accuracies, we can concluded that the best (l,p) pair here would be (1,7), where p=7 is the number of features used in recursive feature elimination.

Cross validation for feature selection: Let us take an example of a dataset with 1000 features and 100 samples in it. Common strategies for feature selection with cross validation would be as follows:

I. Find the best 20 features subset that show strong correlation.
II. Using this subset, build a multivariate classifier.
III. Then perform cross validation to estimate prediction error of the final model.

But there is a problem with the above method. The predictors have an unfair advantage, as they were chosen in step 1 on basis of all samples.

This is the wrong way to perform cross-validation, since these predictors "have already seen" the left-out samples.

The right way to perform cross-validation would be as follows:

I. Divide the dataset into 10 subsets of 10 samples each as in 10-fold cross validation.
II. For each group, k = 1, 2….10.
III. Find best 20 features using all of the samples except that in group k.
IV. Using these features, create a multivariate classifier again using all samples except in group k.
V. Use this classifier to predict error in the group k.

Conclusion: The difference in the right & wrong way was that the samples on which the classifier (i.e. the k group) is to be run should be left out during the feature selection step. This ensures that the predictors are not biased, and the prediction would be natural.

Stratified cross-validation: In stratified k-fold cross-validation, the folds are selected so that the mean response value is approximately equal in all the folds. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole. For example in a binary classification problem where each class comprises 50% of the data, it is best to arrange the data such that in every fold, each class comprises around half the instances.

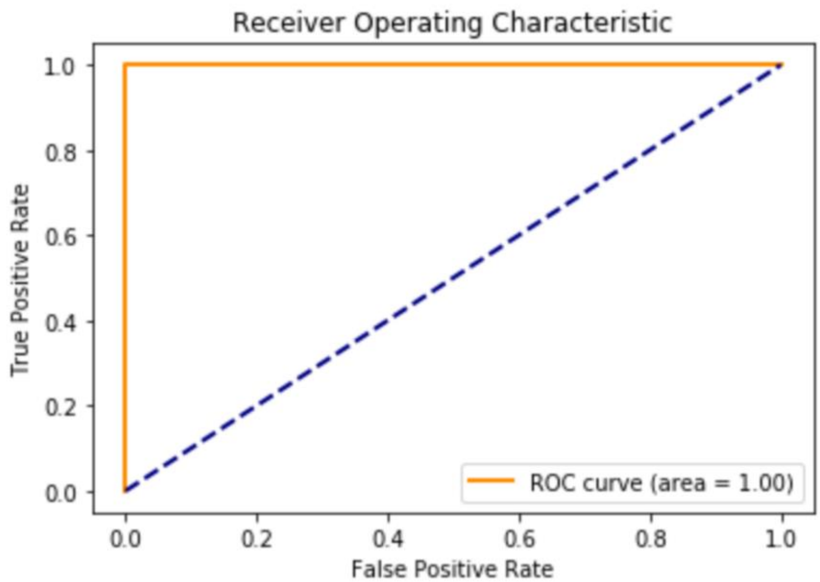
I have not used this method since I have not encountered the problem of class imbalance.

iv) Confusion Matrix:

```
[[ 2  0]
 [ 0 19]]
```

AUC: 1.0

ROC:



Parameters of logistic regression Bi's as well as the p-values associated with them:

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0100	0.006	1.673	0.133	-0.004	0.024
const	-1.643e-12	4.16e-13	-3.951	0.004	-2.6e-12	-6.84e-13
x2	-0.0040	0.016	-0.261	0.801	-0.040	0.032
x3	2.328e-13	2.32e-13	1.004	0.345	-3.02e-13	7.68e-13
x4	0.0277	0.012	2.219	0.057	-0.001	0.057
x5	-0.2924	0.257	-1.138	0.288	-0.885	0.300
x6	-0.0632	0.024	-2.588	0.032	-0.119	-0.007
x7	-0.0099	0.026	-0.381	0.713	-0.070	0.050
x8	0.0048	0.022	0.218	0.833	-0.046	0.055
x9	-0.0761	0.051	-1.484	0.176	-0.194	0.042
x10	0.0406	0.034	1.185	0.270	-0.038	0.119
x11	-0.1523	0.046	-3.329	0.010	-0.258	-0.047
x12	0.4198	0.179	2.347	0.047	0.007	0.832
x13	-0.6090	0.538	-1.133	0.290	-1.849	0.631
x14	-0.2131	0.134	-1.588	0.151	-0.523	0.096
x15	-0.2907	0.841	-0.346	0.738	-2.229	1.648
x16	-0.6784	0.231	-2.932	0.019	-1.212	-0.145

v) Accuracy obtained in 5-fold cross-validation for Test set divided into l = {1,2,…20}:

```
5-fold cross validation average accuracy for l= 1 : 0.950
5-fold cross validation average accuracy for l= 2 : 0.896
5-fold cross validation average accuracy for l= 3 : 0.965
5-fold cross validation average accuracy for l= 4 : 0.973
5-fold cross validation average accuracy for l= 5 : 0.979
5-fold cross validation average accuracy for l= 6 : 0.964
5-fold cross validation average accuracy for l= 7 : 0.970
5-fold cross validation average accuracy for l= 8 : 0.987
5-fold cross validation average accuracy for l= 9 : 0.988
5-fold cross validation average accuracy for l= 10 : 0.974
5-fold cross validation average accuracy for l= 11 : 0.985
5-fold cross validation average accuracy for l= 12 : 0.991
5-fold cross validation average accuracy for l= 13 : 0.992
5-fold cross validation average accuracy for l= 14 : 0.992
5-fold cross validation average accuracy for l= 15 : 0.993
5-fold cross validation average accuracy for l= 16 : 0.990
5-fold cross validation average accuracy for l= 17 : 1.000
5-fold cross validation average accuracy for l= 18 : 0.994
5-fold cross validation average accuracy for l= 19 : 0.994
5-fold cross validation average accuracy for l= 20 : 0.992
Best value of pair (l,p) is ( 17 ,7) with accuracy= 1.0
```

Comparing the accuracy obtained for the Test set above to the accuracy obtained for Training Set, we can say that the 5-fold cross-validation for Test set is more accurate compared to Training Set.

vi) Logistic regression parameters for Test set:

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0031	0.012	0.267	0.790	-0.020	0.026
x2	-0.0004	0.063	-0.006	0.995	-0.124	0.123
x3	0.0154	0.014	1.074	0.284	-0.013	0.044
x4	-0.0328	0.072	-0.455	0.650	-0.175	0.110
x5	0.0005	0.014	0.034	0.973	-0.028	0.029
x6	-0.0801	0.069	-1.169	0.244	-0.215	0.055
x7	-0.0105	0.018	-0.582	0.561	-0.046	0.025
x8	-0.0064	0.026	-0.251	0.802	-0.057	0.044
x9	-0.0038	0.021	-0.176	0.861	-0.046	0.038
x10	-0.0432	0.029	-1.500	0.135	-0.100	0.014
x11	-0.0442	0.021	-2.086	0.038	-0.086	-0.002
x12	0.0430	0.034	1.262	0.208	-0.024	0.110
x13	0.1290	0.086	1.501	0.135	-0.041	0.299
x14	-0.0937	0.225	-0.417	0.677	-0.537	0.349
x15	-0.0829	0.104	-0.796	0.427	-0.288	0.123
x16	0.2734	0.259	1.058	0.292	-0.237	0.784
x17	0.0546	0.105	0.520	0.604	-0.153	0.262
x18	-0.2495	0.280	-0.891	0.374	-0.802	0.303
x19	-0.0576	0.031	-1.879	0.062	-0.118	0.003
x20	0.0412	0.097	0.424	0.672	-0.151	0.233

As we do not see any infinity values in the coefficients listed above, we can say that there is no instability in calculating logistic regression parameters.

vii) Confusion Matrix for test set:

Predicted / True	0	1	All
0	26	0	26
1	0	71	71
All	26	71	97

From the above confusion matrix, we do not see any imbalanced classes.

e) Binary Classification using L1-penalized logistic regression.

i) Accuracy obtained in 5-fold cross-validation for Training set divided into l = {1,2,…20}:

```
5-fold cross validation average accuracy for l= 1 : 0.986
5-fold cross validation average accuracy for l= 2 : 0.892
5-fold cross validation average accuracy for l= 3 : 0.893
5-fold cross validation average accuracy for l= 4 : 0.924
5-fold cross validation average accuracy for l= 5 : 0.890
5-fold cross validation average accuracy for l= 6 : 0.894
5-fold cross validation average accuracy for l= 7 : 0.895
5-fold cross validation average accuracy for l= 8 : 0.904
5-fold cross validation average accuracy for l= 9 : 0.889
5-fold cross validation average accuracy for l= 10 : 0.912
5-fold cross validation average accuracy for l= 11 : 0.893
5-fold cross validation average accuracy for l= 12 : 0.897
5-fold cross validation average accuracy for l= 13 : 0.889
5-fold cross validation average accuracy for l= 14 : 0.899
5-fold cross validation average accuracy for l= 15 : 0.881
5-fold cross validation average accuracy for l= 16 : 0.885
5-fold cross validation average accuracy for l= 17 : 0.892
5-fold cross validation average accuracy for l= 18 : 0.891
5-fold cross validation average accuracy for l= 19 : 0.886
5-fold cross validation average accuracy for l= 20 : 0.896
Best value of pair (l,p) is ( 1 ,7) with accuracy= 0.9857142857142858
```

ii) Comparing the L1-penalized with variable selection using p-values by checking the scores obtained from 5-fold cross-validation of both the models.

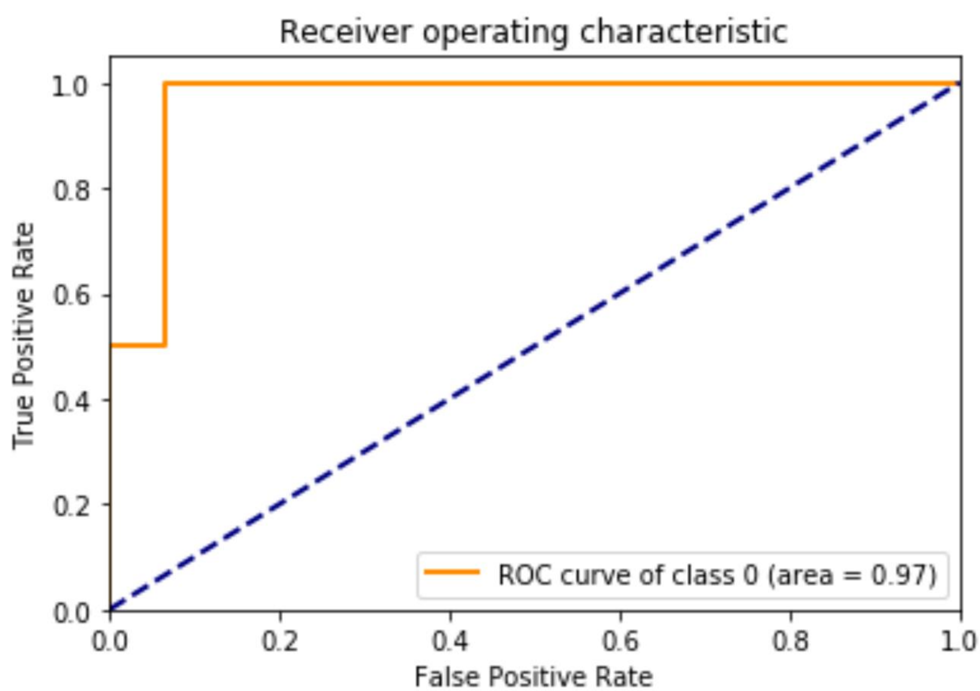
```
Max score obtained with variable selection using p-values:  0.9857142857142858
Mean score obtained with variable selection using p-values:  0.887588444442611
Max score obtained with variable selection using L1-penalized:  0.9857142857142858
Mean score obtained with variable selection using L1-penalized:  0.8993977505072561
```

From the above scores we can see that max score for both models are same. But mean score for L1-penalized is greater than p-value model. So we can say that L1-penalized model is slightly more efficient compared to p-value model & hence L1-penalized model performs better. According to my opinion, both the models are equivalently easy to implement.

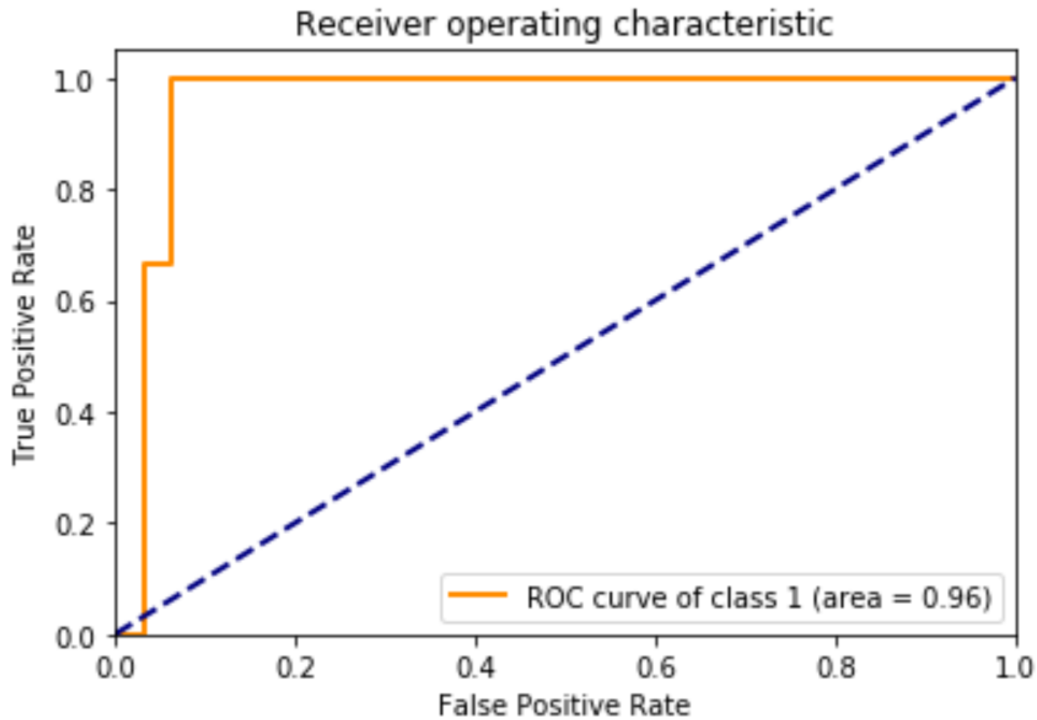
f) Multi-class Classification (The Realistic Case)

i) Best value of pair (l,p) is (1 ,7) with accuracy= 0.8847593582887701
Average score obtained with L1-penalized multinomial regression: 0.7979234659475007
Test Error: 0.7714285714285715

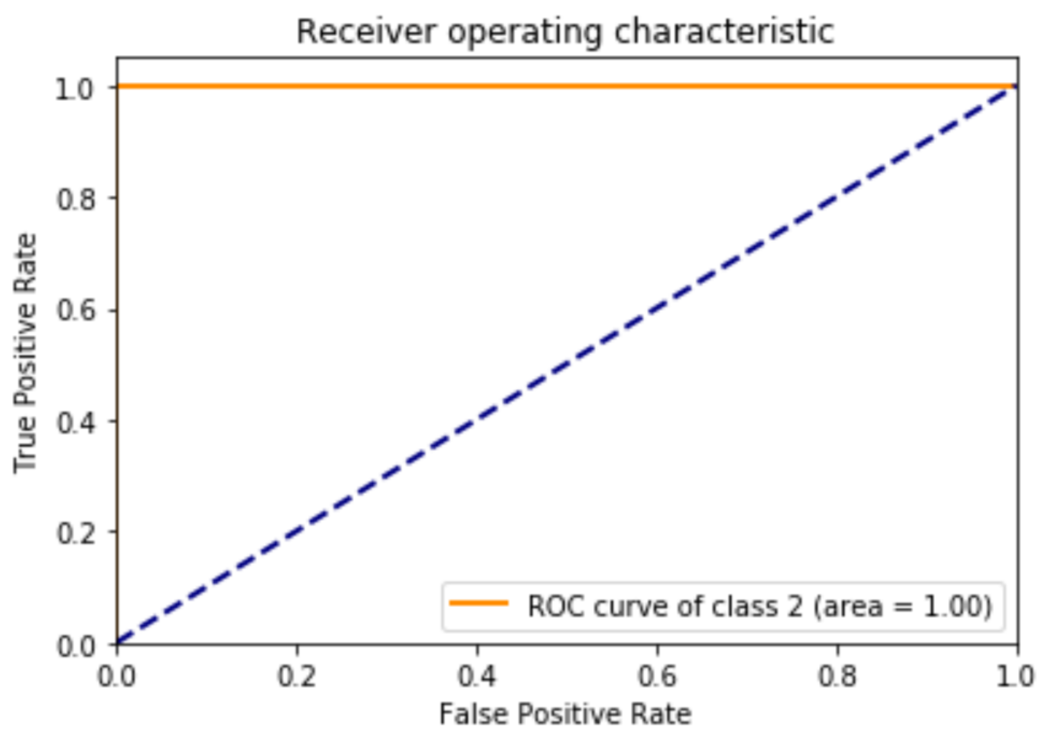
ROC for Class 0 = Bending



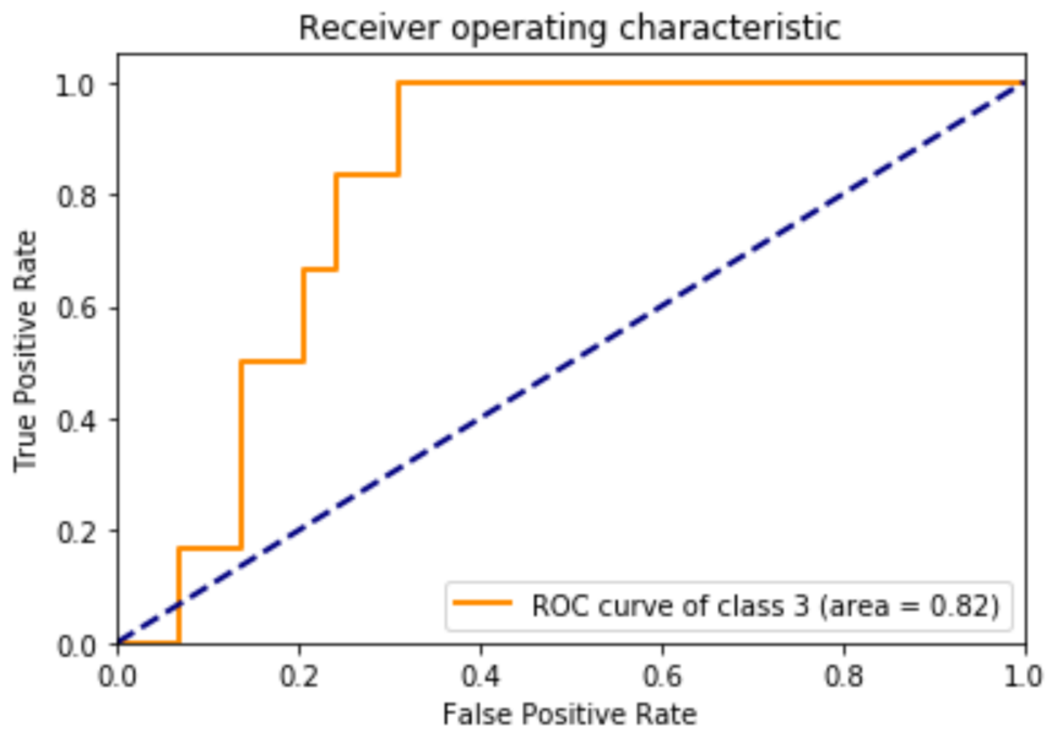
ROC for Class 1 = Cycling



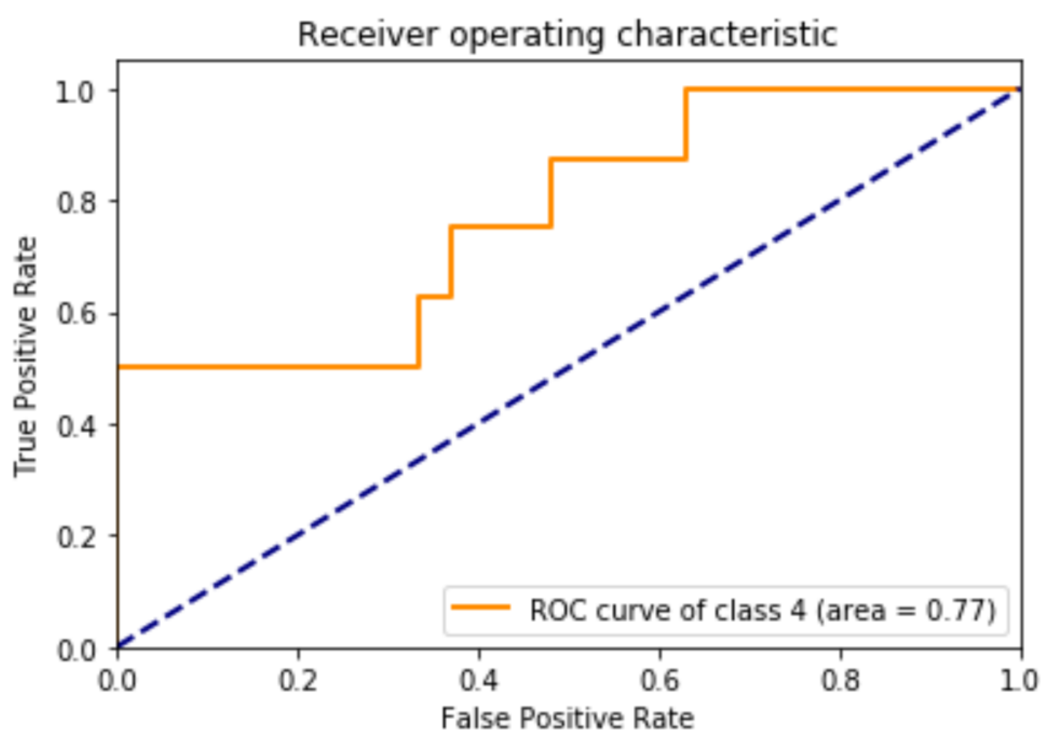
ROC for Class 2 = Lying



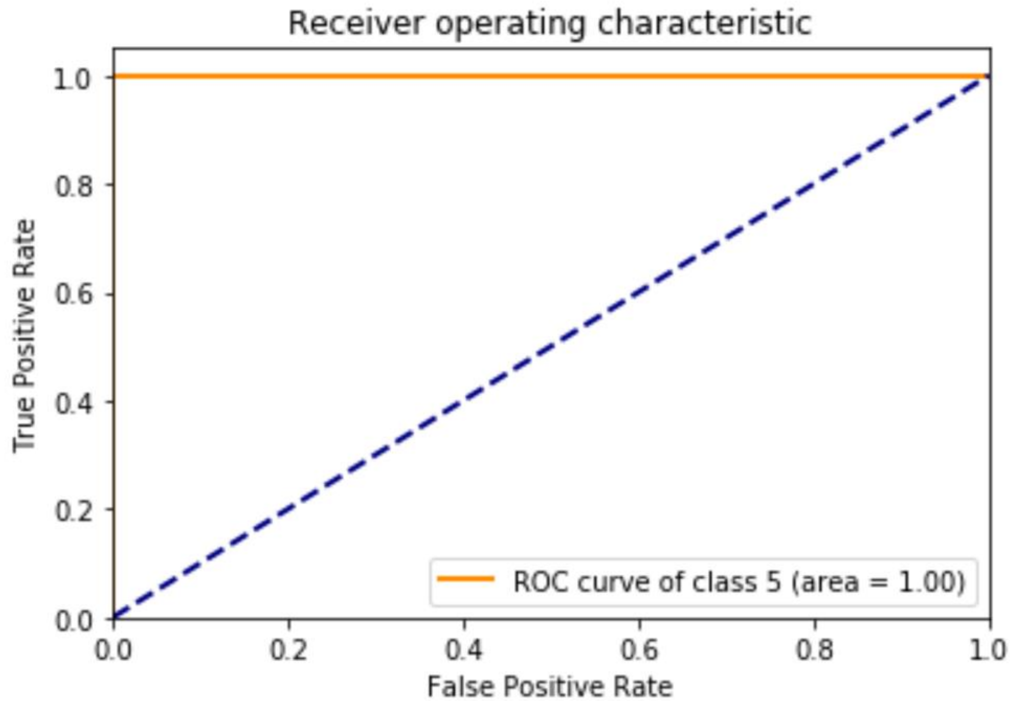
ROC for Class 3 = Sitting



ROC for Class 4 = Standing



ROC for Class 5 = Walking



Confusion Matrix for Class 0 = Bending
[[31 0]
 [2 2]]

Confusion Matrix for Class 1 = Cycling
[[30 2]
 [0 3]]

Confusion Matrix for Class 2 = Lying
[[27 1]
 [0 7]]

Confusion Matrix for Class 3 = Sitting
[[19 10]
 [0 6]]

Confusion Matrix for Class 4 = Standing
[[27 0]
 [6 2]]

Confusion Matrix for Class 5 = Walking
[[28 0]
 [0 7]]

ii) Multi-class Classification using Gaussian Naive Bayers':
Accuracy obtained in 5-fold cross-validation for Training set divided into l = {1,2,…20}:

```
5-fold cross validation average accuracy for l= 1 : 0.816
5-fold cross validation average accuracy for l= 2 : 0.897
5-fold cross validation average accuracy for l= 3 : 0.801
5-fold cross validation average accuracy for l= 4 : 0.791
5-fold cross validation average accuracy for l= 5 : 0.780
5-fold cross validation average accuracy for l= 6 : 0.784
5-fold cross validation average accuracy for l= 7 : 0.799
5-fold cross validation average accuracy for l= 8 : 0.769
5-fold cross validation average accuracy for l= 9 : 0.779
5-fold cross validation average accuracy for l= 10 : 0.757
5-fold cross validation average accuracy for l= 11 : 0.755
5-fold cross validation average accuracy for l= 12 : 0.756
5-fold cross validation average accuracy for l= 13 : 0.760
5-fold cross validation average accuracy for l= 14 : 0.749
5-fold cross validation average accuracy for l= 15 : 0.751
5-fold cross validation average accuracy for l= 16 : 0.752
5-fold cross validation average accuracy for l= 17 : 0.748
5-fold cross validation average accuracy for l= 18 : 0.743
5-fold cross validation average accuracy for l= 19 : 0.735
5-fold cross validation average accuracy for l= 20 : 0.743
Best value of pair (l,p) is ( 2 ,7) with accuracy= 0.8970657528378668
Mean score obtained with Gaussian Naive Bayers' : 0.7731365696660882
```

Multi-class Classification using Multinomial Naive Bayers':
Accuracy obtained in 5-fold cross-validation for Training set divided into l = {1,2,…20}:

```
5-fold cross validation average accuracy for l= 1 : 0.838
5-fold cross validation average accuracy for l= 2 : 0.748
5-fold cross validation average accuracy for l= 3 : 0.757
5-fold cross validation average accuracy for l= 4 : 0.735
5-fold cross validation average accuracy for l= 5 : 0.742
5-fold cross validation average accuracy for l= 6 : 0.735
5-fold cross validation average accuracy for l= 7 : 0.724
5-fold cross validation average accuracy for l= 8 : 0.745
5-fold cross validation average accuracy for l= 9 : 0.741
5-fold cross validation average accuracy for l= 10 : 0.739
5-fold cross validation average accuracy for l= 11 : 0.728
5-fold cross validation average accuracy for l= 12 : 0.730
5-fold cross validation average accuracy for l= 13 : 0.740
5-fold cross validation average accuracy for l= 14 : 0.725
5-fold cross validation average accuracy for l= 15 : 0.728
5-fold cross validation average accuracy for l= 16 : 0.722
5-fold cross validation average accuracy for l= 17 : 0.725
5-fold cross validation average accuracy for l= 18 : 0.728
5-fold cross validation average accuracy for l= 19 : 0.717
5-fold cross validation average accuracy for l= 20 : 0.730
Best value of pair (l,p) is ( 1 ,7) with accuracy= 0.8381461675579323
Mean score obtained with Gaussian Naive Bayers' : 0.7388120357444282
```

Comparing the results of Gaussian Naive Bayers and Multinomial Naive Bayers, we can see that Best accuracy & Average score for Gaussian Naive Bayers' is greater than that of Multinomial Naive Bayers. Hence we can say that Gaussian Naive Bayers performs better than Multinomial Naive Bayers.

2) 3.7.4. I collect a set of data (n=100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. Y=β0+β1X+β2X2+β3X3+ε.
   a) Suppose that the true relationship between X and Y is linear, i.e. Y=β0+β1X+ε. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would

we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.
<u>Ans:</u> As we do not have enough details about the training data, it is difficult to know which training RSS is lower between linear or cubic. Although, as mentioned in the question, the true relationship between X and Y is linear, the RSS for the linear regression may be lower than for the cubic regression since the least squares line is expected to be close to the true regression line.

b) Answer (a) using test rather than training RSS.
<u>Ans:</u> The test RSS depends on the test data, so we do not have enough information to come to a conclusion. Although, we may assume that polynomial regression will have a higher test RSS as the overfit from training would have more error than the linear regression.

c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.
<u>Ans:</u> Polynomial regression has lower train RSS than the linear fit because of higher flexibility: no matter what the underlying true relationship is the more flexible model will follow the points closely and reduce train RSS.

d) Answer (c) using test rather than training RSS.
<u>Ans:</u> We do not have enough information to tell which test RSS would be lower for either regression given the problem statement is defined as not knowing "how far it is from linear". If it is closer to linear than cubic, the linear regression test RSS could be lower than the cubic regression test RSS. Or, if it is closer to cubic than linear, the cubic regression test RSS could be lower than the linear regression test RSS. It is dues to bias-variance tradeoff: it is not clear what level of flexibility will fit data better.

3) 4.7.3: This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class specific mean

vector and a class specific covariance matrix. We consider the simple case where p = 1; i.e. there is only one feature.

Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, X ∼ N(µk, σ2k). Recall that the density function for the one-dimensional normal distribution is given in (4.11). Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.

Ans:

4.7.3

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\,\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k^2)\right)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\,\sigma_l} \exp\left(\frac{-1}{2\sigma_l^2}(x-\mu_l^2)\right)}$$

$$\log(p_k(x)) = \frac{\log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi}\,\sigma_k}\right) + \frac{-1}{2\sigma_k^2}(x-\mu_k)^2}{\log \sum \pi_l \frac{1}{\sqrt{2\pi}\,\sigma_l} \exp\left(\frac{-1}{2\sigma_l^2}(x-\mu_k)^2\right)}$$

$$\log(p_k(x)) \log\left(\sum \pi_l \frac{1}{\sqrt{2\pi}\,\sigma_l} \exp\left(\frac{-1}{2\sigma_l^2}(x-\mu_l)^2\right)\right)$$

$$= \log(\pi_k) + \log\frac{1}{\sqrt{2\pi}\,\sigma_k} + \frac{-1}{2\sigma_k^2}(x-\mu_k)^2$$

$$\delta(x) = \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi}\,\sigma_x}\right) + \frac{-1}{2\sigma_k^2}(x-\mu_k)^2$$

$\therefore$ We can conclude that $\delta(x)$ is a quadratic function of x.

4) 4.7.7: Suppose that we wish to predict whether a given stock will issue a dividend this year (“Yes” or “No”) based on X, last year’s percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was ¯X = 10, while the mean for those that didn’t was ¯X = 0. In addition, the variance of X for these two sets of companies was ˆσ2 = 36. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was X = 4 last year.

Hint: Recall that the density function for a normal random variable is f(x) = √ 1 2πσ2 e−(x−μ)2/2σ2. You will need to use Bayes’ theorem.

Ans:

4.7.7. $$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_l \pi_l \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

$$p_{yes}(x) = \frac{\pi_{yes} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_{yes})^2\right)}{\sum_l \pi_l \exp\left(\frac{-1}{2\sigma^2}(x-\mu_l)^2\right)}$$

$$= \frac{\pi_{yes} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_{yes})^2\right)}{\pi_{yes} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2\right) + \pi_{no} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_{no})^2\right)}$$

$$= \frac{0.80 \exp\left(\frac{-1}{2\times 36}(x-10)^2\right)}{0.80 \exp\left(\frac{-1}{2\times 36}(x-10)^2\right) + 0.20 \exp\left(\frac{-1}{2\times 36} x^2\right)}$$

$$p_{yes}(4) = \frac{0.80\left(\exp \frac{-1}{2\times 36}(4-10)^2\right)}{0.80 \exp\left(\frac{-1}{2\times 36}(4-10)^2\right) + 0.20 \exp\left(\frac{-1}{2\times 36}\times 4^2\right)}$$

$$= 75.2\%$$