Prashanth Manja

USC ID: 3073150764

ML- Assignment -2

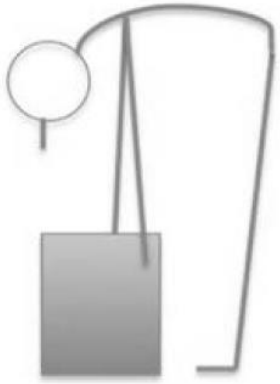
1. Time-domain features are usually used in time series classification:

- Mean
- Maximum
- Minimum
- SD- Standard deviation
- Median
- Variance
- Peak-to-peak
- Slope
- Zero crossing rate
- Autocorrelation
- Cross correlation
- Linear correlation coefficient

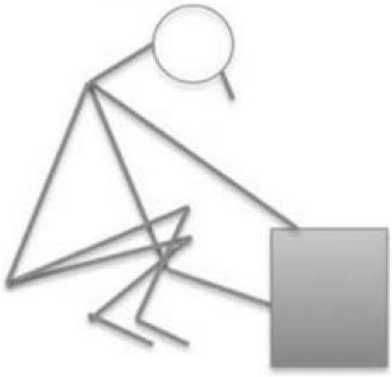
Time domain features considered in this case are:

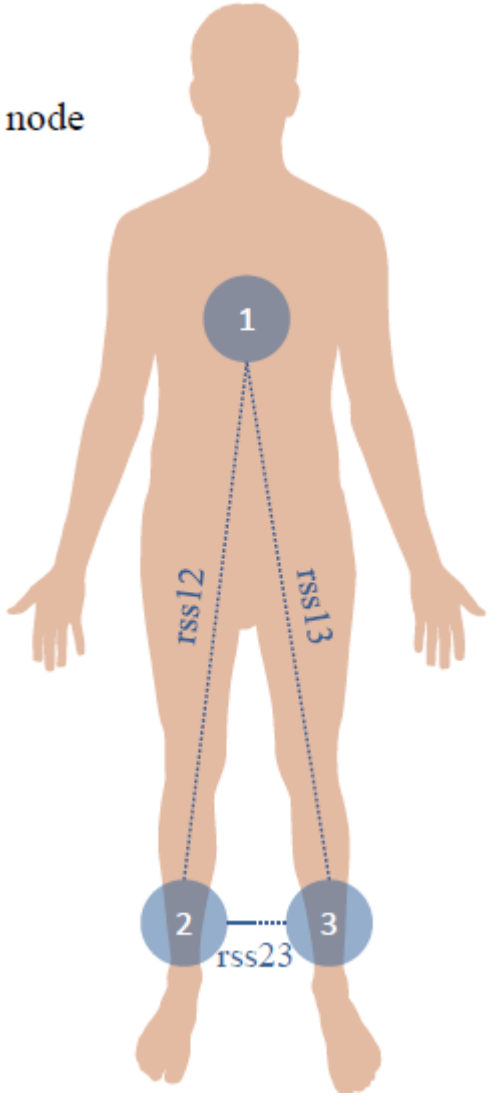
1. Mean
2. Maximum
3. Minimum
4. Median
5. SD- Standard deviation
6. Interquartile 25th percentile
7. Interquartile 75th percentile

bending1



bending2





2. Standard Deviation of all the time domain features considered.

A. For Test set:

min_avg_rss12	12.136206
min_var_rss12	0.000000
min_avg_rss13	2.644618
min_var_rss13	0.000000
min_avg_rss23	8.252947
min_var_rss23	0.000000
max_avg_rss12	4.379342
max_var_rss12	4.870395
max_avg_rss13	5.393220
max_var_rss13	1.733937
max_avg_rss23	6.782153
max_var_rss23	2.505306
mean_avg_rss12	6.790086
mean_var_rss12	1.500529
mean_avg_rss13	4.588252
mean_var_rss13	1.146338
mean_avg_rss23	7.366781
mean_var_rss23	1.119410

B. For Training set:

min_avg_rss12	8.794295
min_var_rss12	0.000000
min_avg_rss13	3.053869
min_var_rss13	0.000000
min_avg_rss23	5.368786
min_var_rss23	0.051766
max_avg_rss12	4.429182
max_var_rss12	5.147841
max_avg_rss13	4.759853
max_var_rss13	2.302408
max_avg_rss23	5.449726
max_var_rss23	2.540166
mean_avg_rss12	4.917692
mean_var_rss12	1.600701
mean_avg_rss13	3.863097
mean_var_rss13	1.179861
mean_avg_rss23	5.120426
mean_var_rss23	1.171401

***for reporting SD other time domain features are skipped here though they are included in building a model.**

3.Confidence_intervel

```
(2.12535392160341, 4.51365382999951)

A. For Test-Set

Bootstrapped 95% confidence intervals
Low: 2.6109164210658893
High: 5.60468610602678

B. For Train-Set

Bootstrapped 95% confidence intervals
Low: 2.3039475317709477
High: 4.4461642733444995
```

Select the three most important time-domain features:

1. Max
2. Mean
3. Stand-Deviation
4. Median

a. Description summary of the Test-Set

	min_avg_rss12	min_var_rss12	min_avg_rss13	min_var_rss13	min_avg_rss23	min_var_rss23	max_avg_rss12	max_var_rss12	max_avg_rss13
count	19.000000	19.0	19.000000	19.0	19.000000	19.0	19.000000	19.000000	19.000000
mean	27.495263	0.0	2.977895	0.0	6.456316	0.0	45.653684	6.094737	23.193158
std	12.136206	0.0	2.644618	0.0	8.252947	0.0	4.379342	4.870395	5.393220
min	0.000000	0.0	0.000000	0.0	0.000000	0.0	30.000000	0.430000	13.000000
25%	20.665000	0.0	0.500000	0.0	0.500000	0.0	44.875000	1.955000	20.835000
50%	28.750000	0.0	2.000000	0.0	4.750000	0.0	46.250000	4.440000	23.750000
75%	36.250000	0.0	5.500000	0.0	7.375000	0.0	48.000000	9.245000	25.290000
max	48.000000	0.0	7.500000	0.0	27.670000	0.0	51.000000	14.500000	35.000000

b. Description summary of the Training-Set:

	min_avg_rss12	min_var_rss12	min_avg_rss13	min_var_rss13	min_avg_rss23	min_var_rss23	max_avg_rss12	max_var_rss12	max_avg_rss13
count	69.000000	69.0	69.000000	69.0	69.000000	69.000000	69.000000	69.000000	69.000000
mean	29.461159	0.0	3.135507	0.0	4.186232	0.006232	45.423333	6.343913	22.705652
std	8.794295	0.0	3.053869	0.0	5.368786	0.051766	4.429182	5.147841	4.759853
min	0.000000	0.0	0.000000	0.0	0.000000	0.000000	30.000000	0.430000	11.330000
25%	23.330000	0.0	0.000000	0.0	0.000000	0.000000	44.330000	1.700000	21.250000
50%	28.500000	0.0	2.500000	0.0	2.500000	0.000000	45.750000	4.500000	23.330000
75%	36.250000	0.0	6.250000	0.0	6.330000	0.000000	47.670000	11.200000	26.330000
max	48.000000	0.0	8.500000	0.0	29.000000	0.430000	56.250000	17.240000	32.750000

4. Training set to classify bending from other activities, a binary classification problem.

A. Minimum Time-Domain Feature:

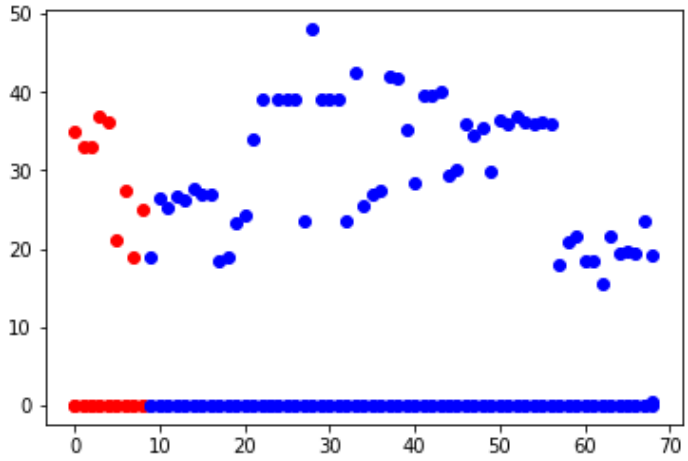
i. Bending: (*displaying part of the set*)

	min_avg_rss12	min_var_rss12	min_var_rss23
0	35.00	0.0	0.0
1	33.00	0.0	0.0
2	33.00	0.0	0.0
3	37.00	0.0	0.0
4	36.25	0.0	0.0

ii. Other Activities:

	min_avg_rss12	min_var_rss12	min_var_rss23
9	19.00	0.0	0.00
10	26.50	0.0	0.00
11	25.33	0.0	0.00
12	26.75	0.0	0.00
13	26.25	0.0	0.00
14	27.75	0.0	0.00
15	27.00	0.0	0.00
16	27.00	0.0	0.00
17	18.50	0.0	0.00
18	19.00	0.0	0.00
19	23.33	0.0	0.00
20	24.25	0.0	0.00

ii. Plot: --Bending - Other activities



B. Max Time-Domain Feature:

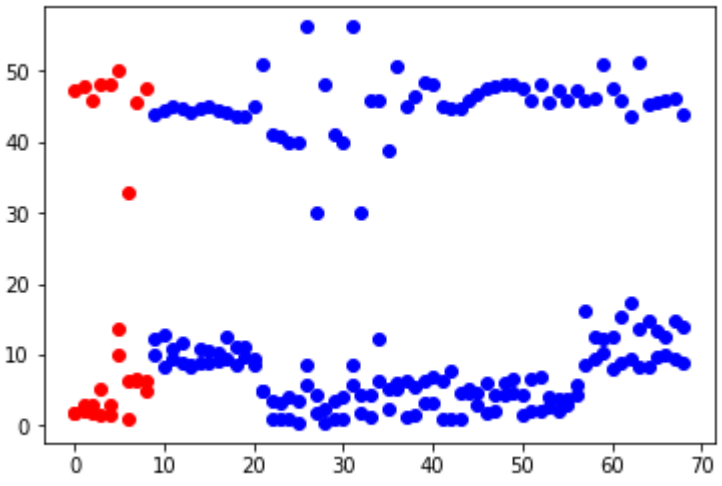
I. Bending:

	max_avg_rss12	max_var_rss12	max_var_rss23
0	47.40	1.70	1.79
1	47.75	3.00	2.18
2	45.75	2.83	1.79
3	48.00	1.58	5.26
4	48.00	1.50	2.96

II.Other Activities:

	max_avg_rss12	max_var_rss12	max_var_rss23
9	44.00	12.28	9.98
10	44.33	12.89	8.19
11	45.00	10.84	9.50
12	44.75	11.68	8.81
13	44.25	8.64	8.34

III.PLOT: **--Bending** **- Other activities**



C. Mean Time-Domain Feature:

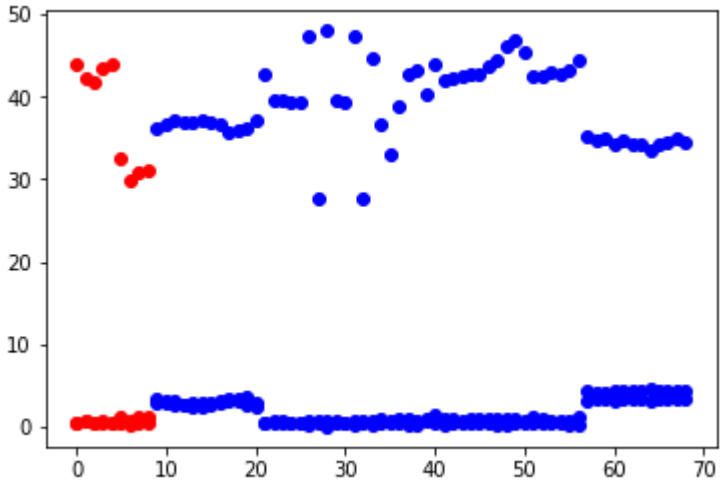
i. Bending

	mean_avg_rss12	mean_var_rss12	mean_var_rss23
0	43.954500	0.426250	0.493292
1	42.179813	0.696042	0.613521
2	41.678063	0.535979	0.383292
3	43.454958	0.378083	0.679646
4	43.969125	0.413125	0.555313

ii. Other Activities:

	mean_avg_rss12	mean_var_rss12	mean_var_rss23
9	36.228396	2.831687	3.480687
10	36.687292	2.973042	3.073312
11	37.114312	2.730000	3.076354
12	36.863375	2.757312	2.773312
13	36.957458	2.420083	2.934625

iii. Plot: **--Bending** **- Other activities**

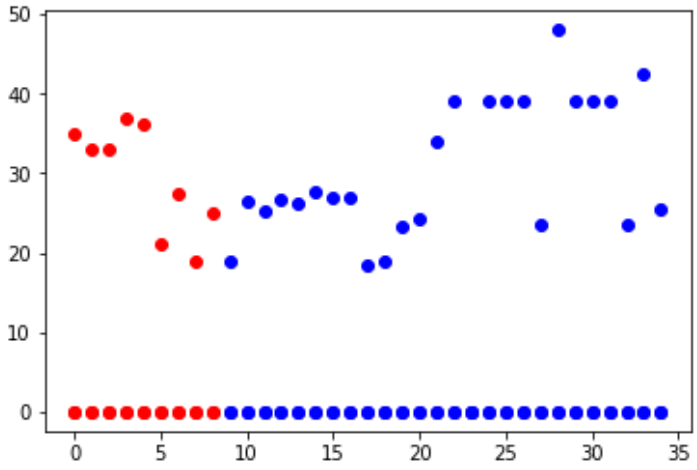


5.Break each time series in your training set into two:

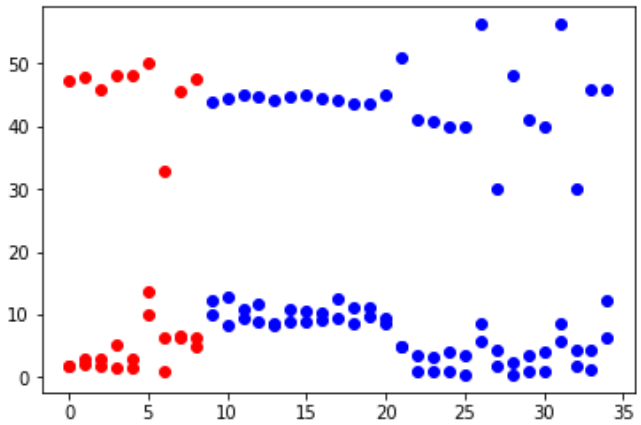
A. FIRST HALF:

Plot for Minimum –1st half of the training set

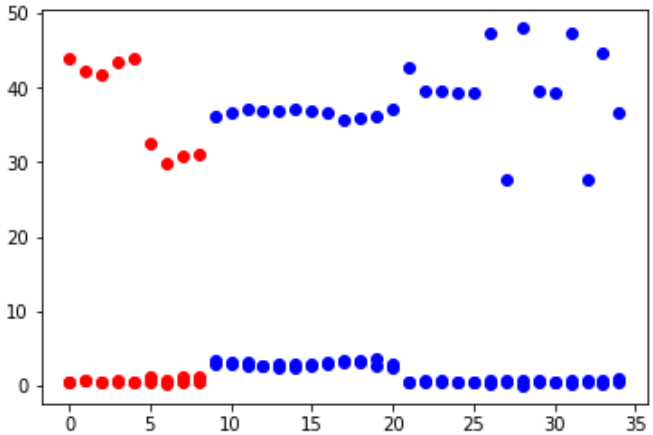
--Bending **- Other activities**



Plot for Maximum –1st half of the training set

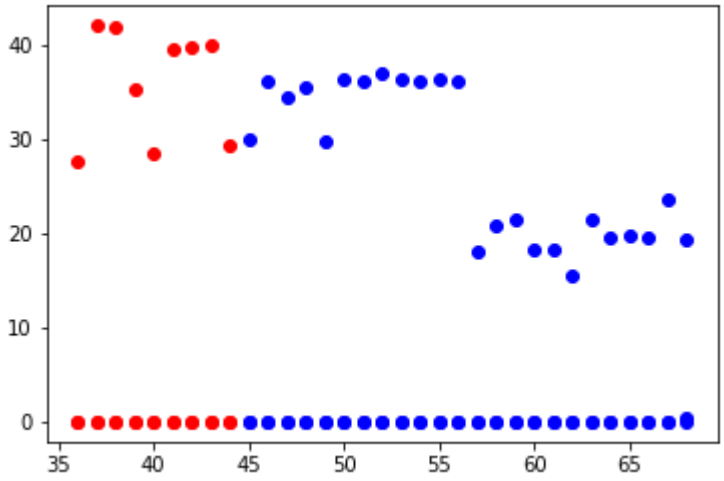


Plot for Mean –1st half of the training set

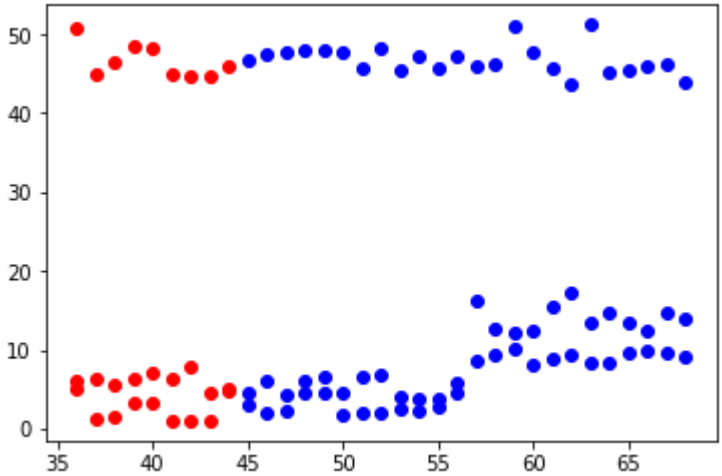


B. Second Half:

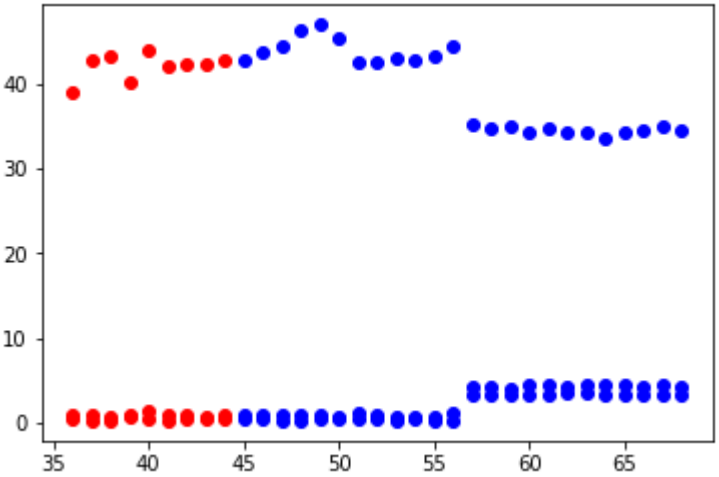
Plot for Minimum -2nd- half of the training set



Plot for Maximum -2nd- half of the training set

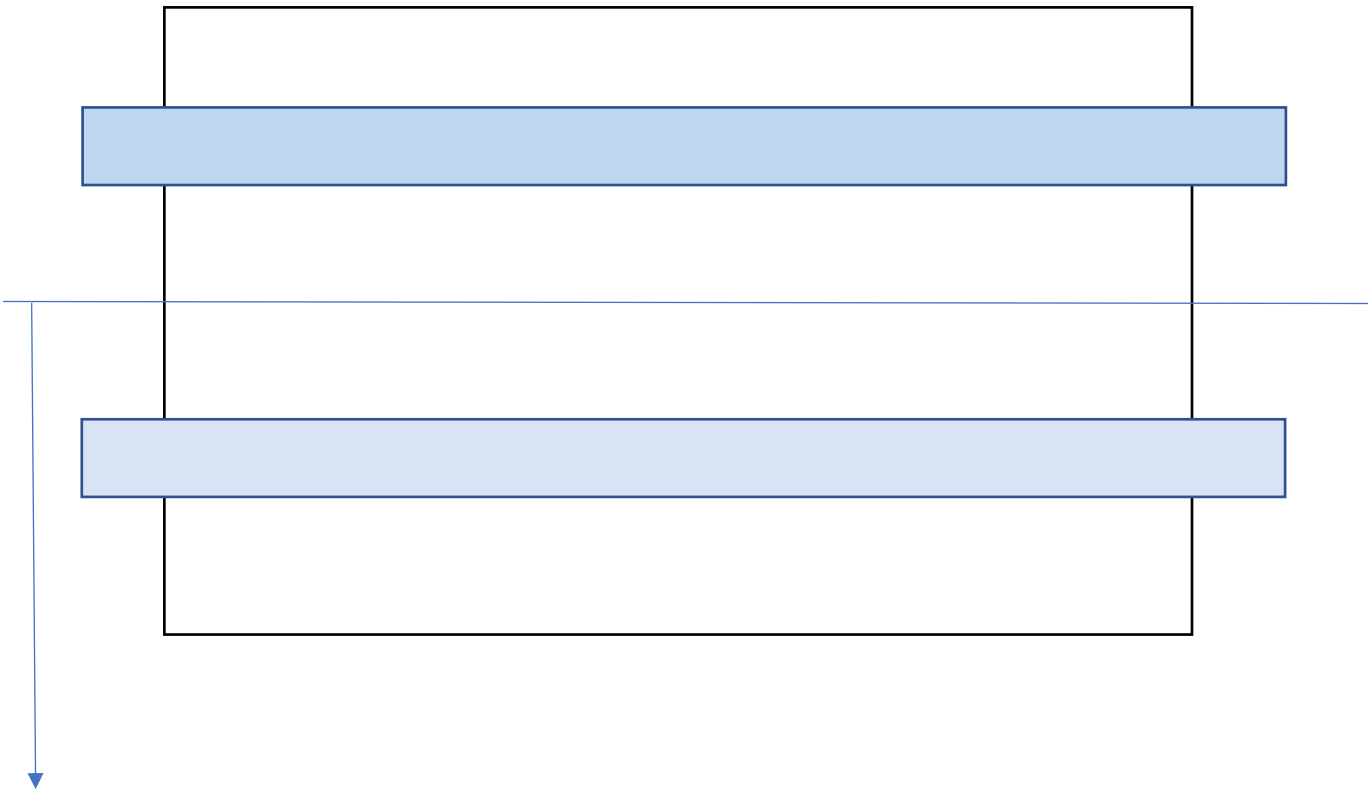


Plot for Mean -2nd- half of the training set

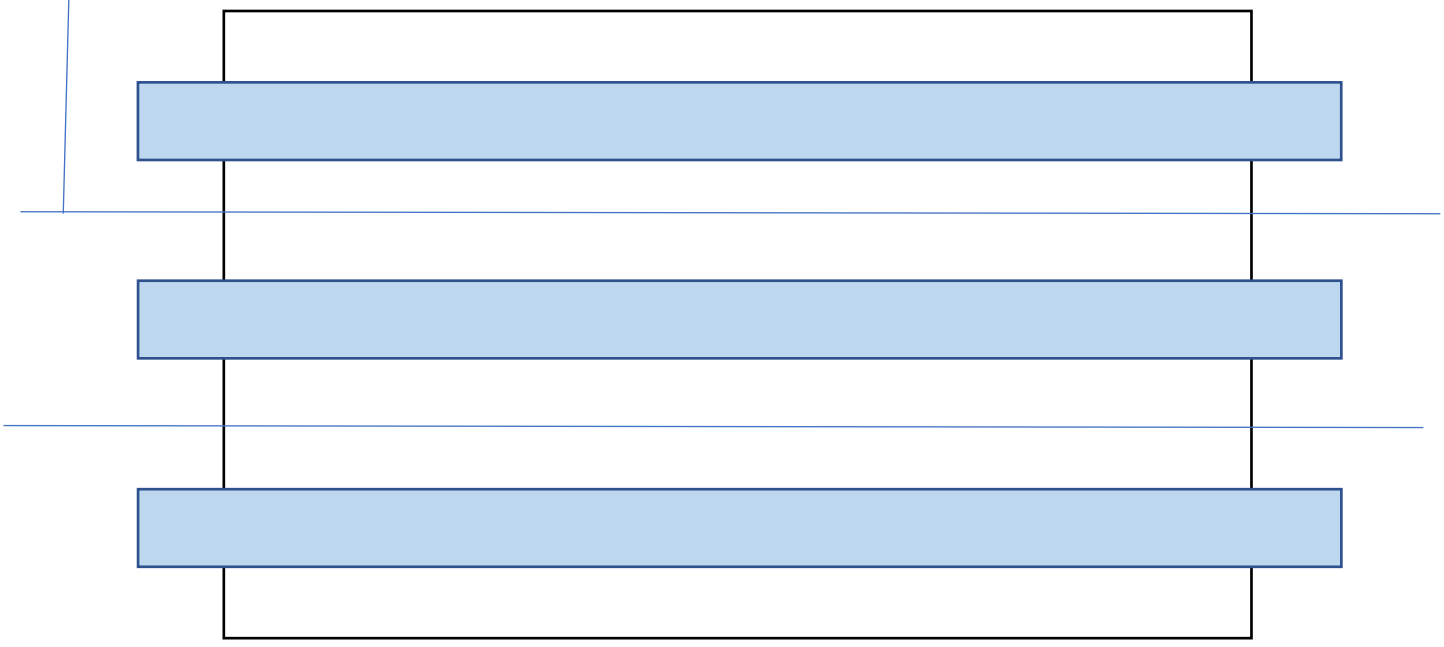


Dividing Training set into l parts (l=1,2,3,4…….10)

a. L=2 & b. L=3



Dividing training dataset into 2,3…10 parts and extracting 1 row from each divided set of the training datafile.



1. **L = 2:**

   a. ***Accuracy of logistic regression classifier on test set: 0.95***
   b. ***Confusion Matrix:***

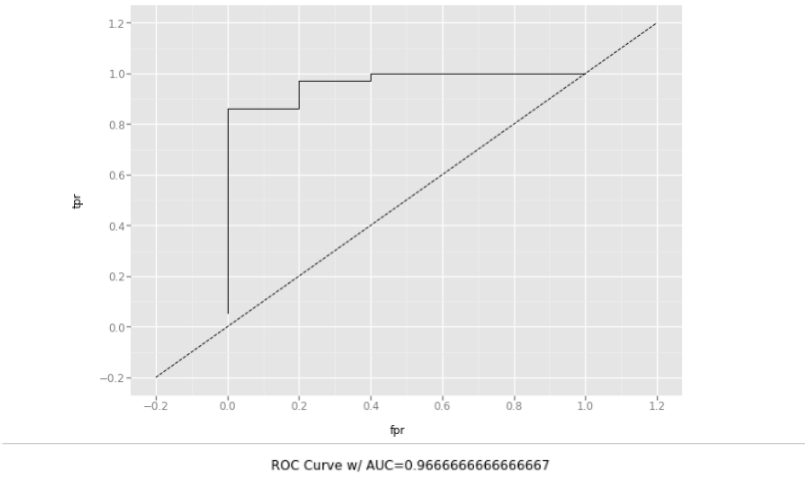
```
[[ 3,  2]
 [ 0, 36]]
```

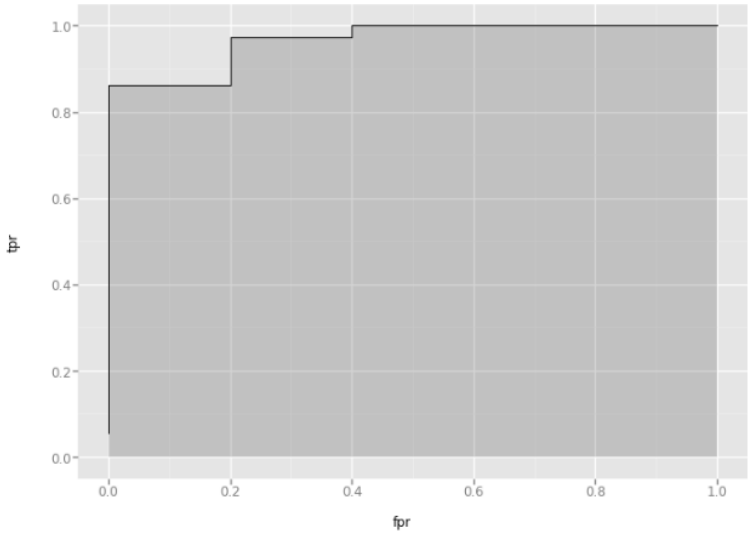
Classification report:

	precision	recall	f1-score	support
0	1.00	0.60	0.75	5
1	0.95	1.00	0.97	36
Avg /total	0.95	0.95	0.95	41

5-fold cross validation average accuracy: 0.926

ROC and AUC:





2. L =3

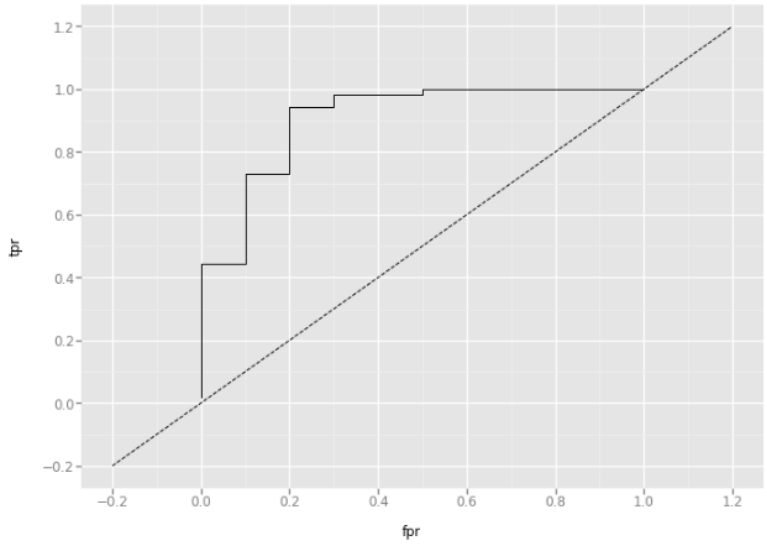
a. Accuracy of logistic regression classifier on test set: 0.92
b .Confusion Matrix:

```
[[ 6,  4],
 [ 1, 51]]
```

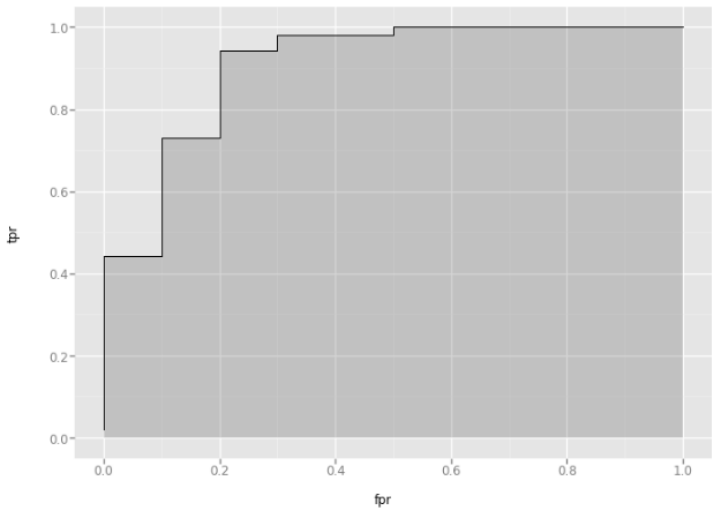
c. Classification Report:

	precision	recall	f1-score	support
0	0.86	0.60	0.71	10
1	0.93	0.98	0.95	52
avg / total	0.92	0.92	0.91	62

5-fold cross validation average accuracy: 0.937



ROC Curve w/ AUC=0.9076923076923077



3. L =4

a.Accuracy of logistic regression classifier on test set: 0.95

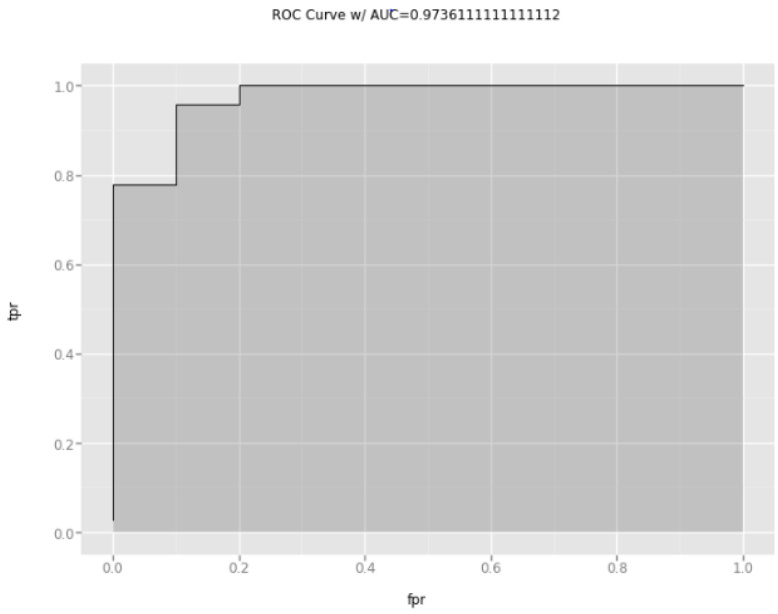
b .Confusion Matrix:

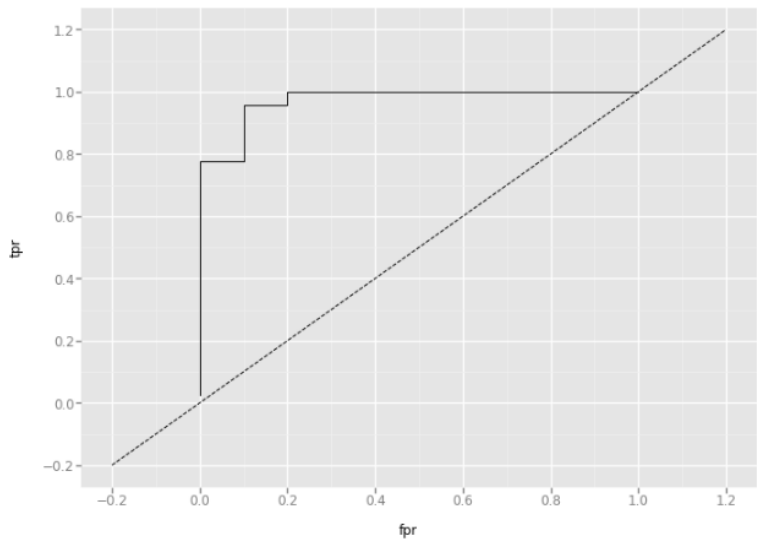
```
[[ 9,  1],
 [ 3, 69]]
```

c. Classification Report:

	precision	recall	f1-score	support
0	0.75	0.90	0.82	10
1	0.99	0.96	0.97	72
avg / total	0.96	0.95	0.95	82

5-fold cross validation average accuracy: 0.937





4. L=5

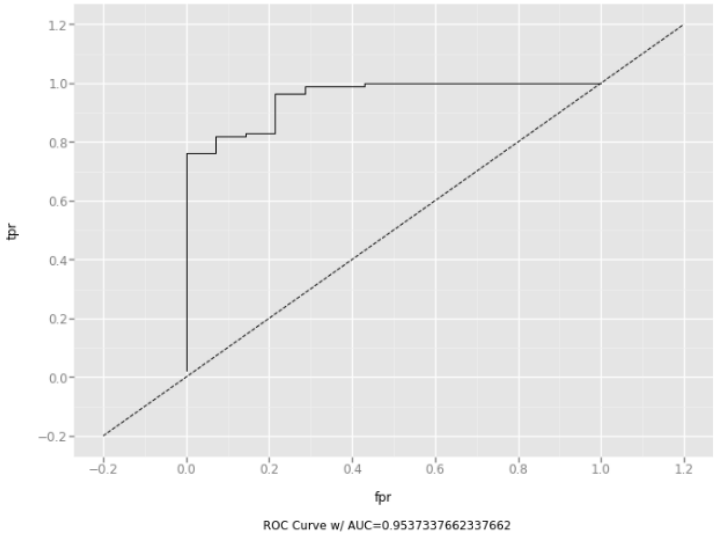
a.Accuracy of logistic regression classifier on test set: 0.89
b. Confusion Matrix:

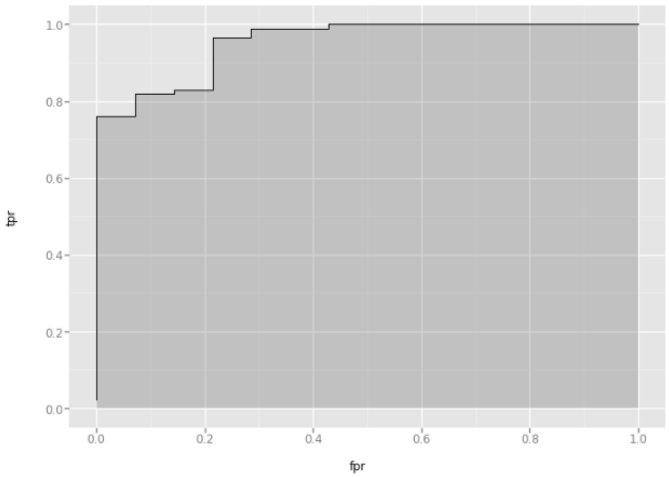
```
([[11,  3],
 [ 8, 80]]
```

c. Classification Report:

	precision	recall	f1-score	support
0	0.58	0.79	0.67	14
1	0.96	0.91	0.94	88
avg / total	0.91	0.89	0.90	102

5-fold cross validation average accuracy: 0.933





5. **L= 6**

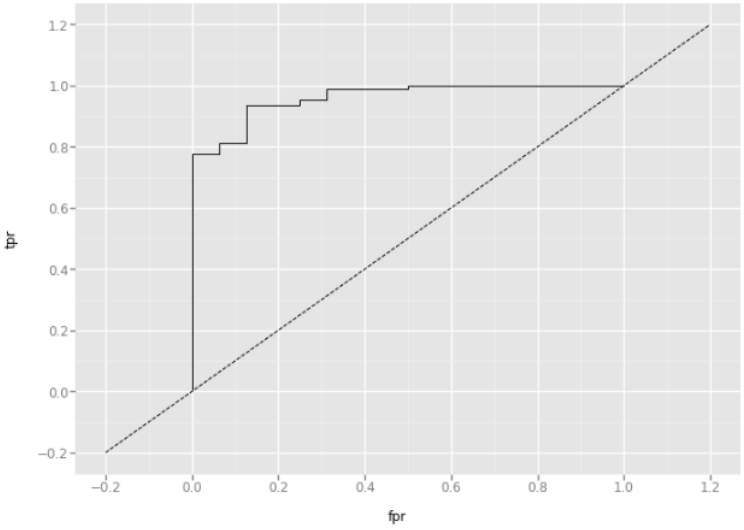
a.Accuracy of logistic regression classifier on test set: 0.93
b. Confusion Matrix:

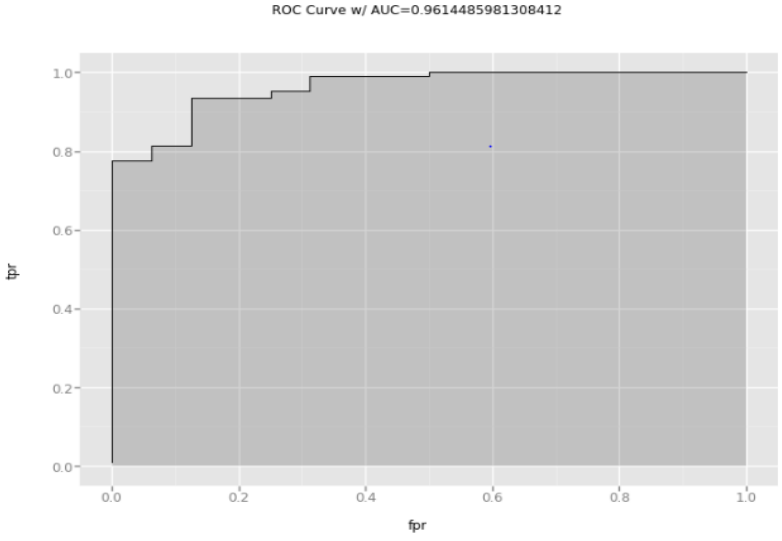
```
([[ 11,   5],
 [  3, 104]]
```

c. Classification Report:

	precision	recall	f1-score	support
0	0.79	0.69	0.73	16
1	0.95	0.97	0.96	107
avg / total	0.93	0.93	0.93	123

5-fold cross validation average accuracy: 0.933





6. L=7

a.Accuracy of logistic regression classifier on test set: 0.94

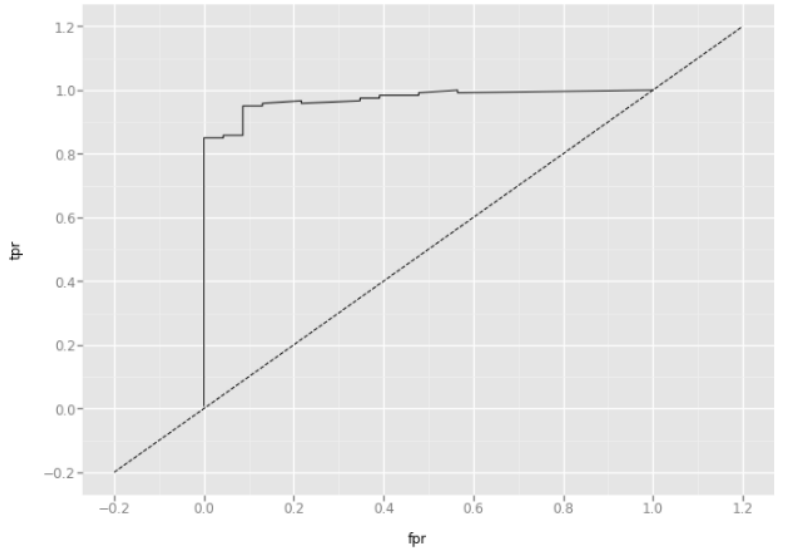
b. Confusion Matrix:

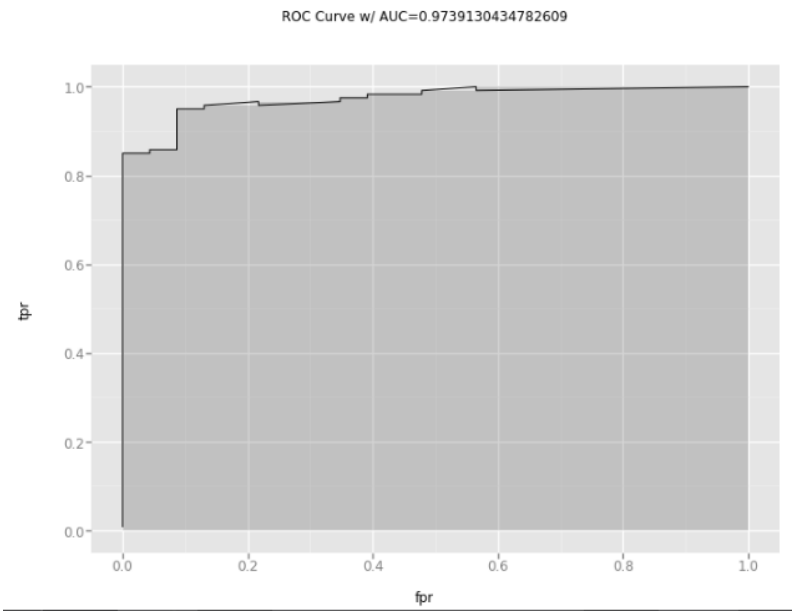
```
([[ 21,   2],
 [  6, 114]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.91	0.84	23
1	0.98	0.95	0.97	120
Avg / total	0.95	0.94	0.95	143

5-fold cross validation average accuracy: 0.928





7. L=8

a. Accuracy of logistic regression classifier on test set: 0.95

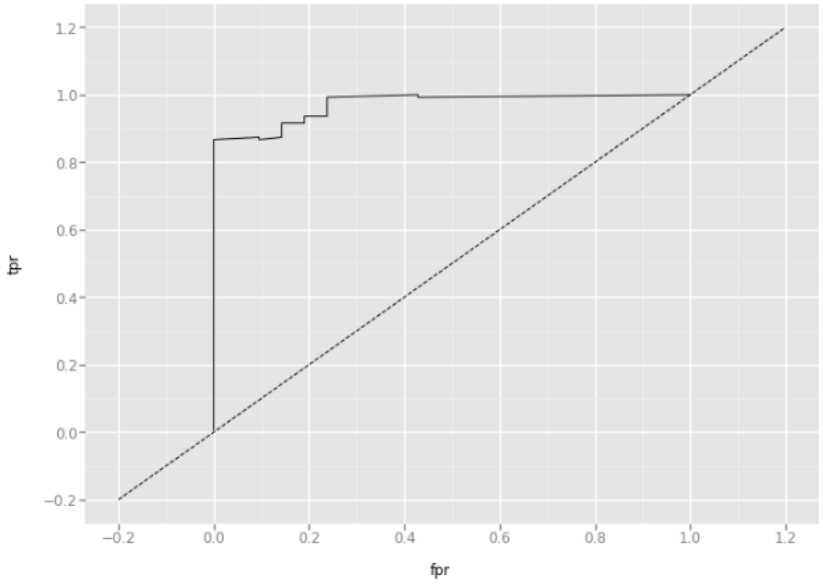
b. Confusion Matrix:

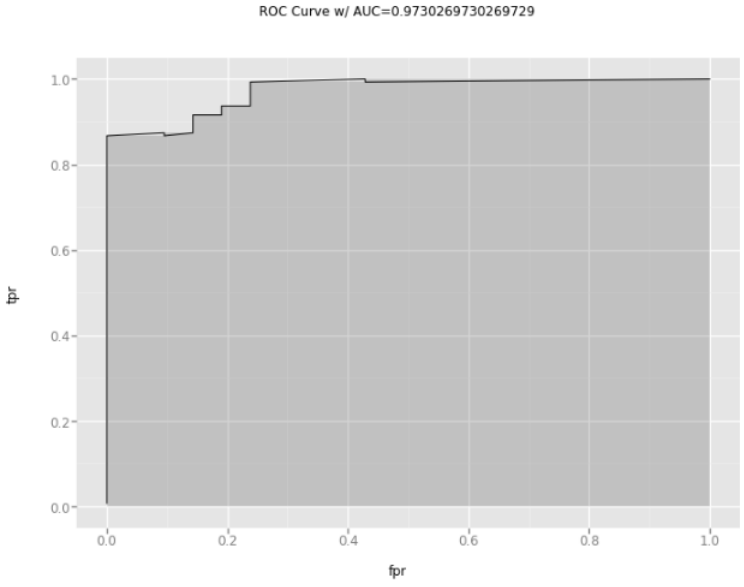
```
([[ 16,   5],
  [  4, 139]],
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.76	0.78	21
1	0.97	0.97	0.97	143
avg /total	0.94	0.95	0.94	164

5-fold cross validation average accuracy: 0.932





8. L=9

a. Accuracy of logistic regression classifier on test set: 0.94

b. Confusion Matrix:

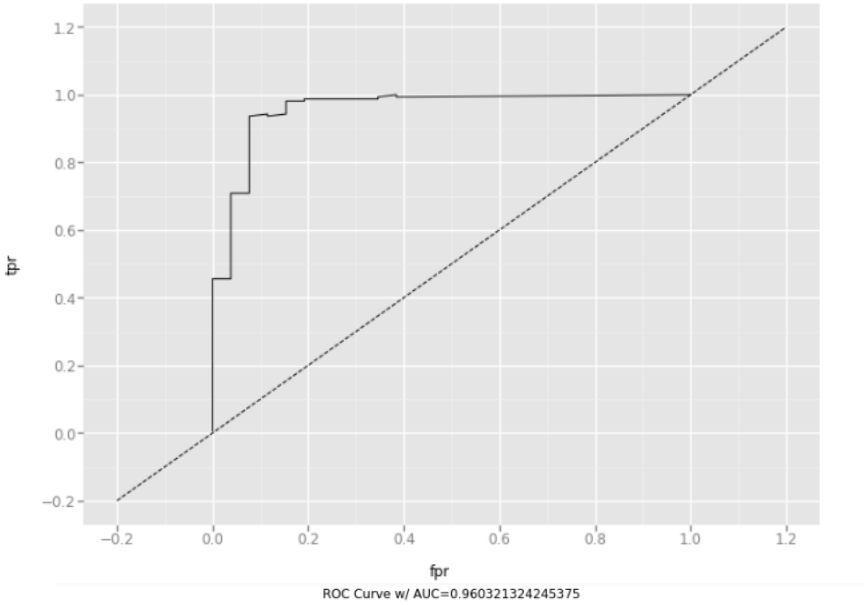
```
([[ 22,   4],
 [  7, 151]]
```

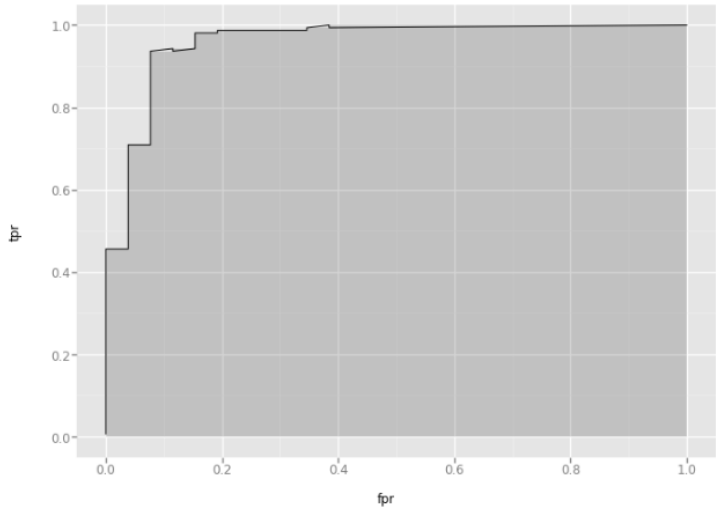
c. ***Classification Report:***

	precision	recall	f1-score	support
0	0.76	0.85	0.80	26
1	0.97	0.96	0.96	158
avg / total	0.94	0.94	0.94	184

5-fold cross validation average accuracy: 0.944

ROC & AUC:





Logit Regression Results

Dep. Variable:	y	No. Observations:	428
Model:	Logit	Df Residuals:	392
Method:	MLE	Df Model:	35
Date:	Sun, 18 Feb 2018	Pseudo R-squ.:	0.8450
Time:	12:38:38	Log-Likelihood:	-25.436
converged:	False	LL-Null:	-164.15
		LLR p-value:	1.671e-39

	coef	std err	z	P>|z|	[0.025	0.975]
x1	0.4324	0.891	0.485	0.628	-1.315	2.179
x2	49.4706	3.55e+04	0.001	0.999	-6.96e+04	6.97e+04
x3	0.3555	0.490	0.726	0.468	-0.604	1.315
x4	61.4359	2.82e+05	0.000	1.000	-5.53e+05	5.53e+05
x5	0.0883	0.693	0.127	0.899	-1.270	1.447
x6	3.2553	8.103	0.402	0.688	-12.626	19.137
x7	0.7737	0.701	1.104	0.269	-0.599	2.147
x8	-2.9566	1.855	-1.594	0.111	-6.592	0.679
x9	-0.9531	0.484	-1.967	0.049	-1.903	-0.004
x10	-2.9067	1.349	-2.155	0.031	-5.550	-0.264
x11	-0.7756	0.601	-1.290	0.197	-1.954	0.403
x12	0.2300	1.440	0.160	0.873	-2.593	3.053
x13	-0.9749	4.484	-0.217	0.828	-9.763	7.813
x14	-20.6053	12.284	-1.677	0.093	-44.682	3.472
x15	1.3895	1.650	0.842	0.400	-1.844	4.623
x16	-12.0171	11.173	-1.076	0.282	-33.915	9.881
x17	-1.4049	2.463	-0.570	0.568	-6.232	3.422
x18	-21.2036	10.210	-2.077	0.038	-41.214	-1.193
x19	0.7291	6.92e+05	1.05e-06	1.000	-1.36e+06	1.36e+06
x20	3.5747	6.64e+07	5.38e-08	1.000	-1.3e+08	1.3e+08
x21	-0.0395	nan	nan	nan	nan	nan
x22	3.1079	nan	nan	nan	nan	nan
x23	0.0386	nan	nan	nan	nan	nan
x24	4.1776	nan	nan	nan	nan	nan
x25	-3.0334	3.835	-0.791	0.429	-10.550	4.483
x26	34.8175	13.547	2.570	0.010	8.267	61.368
x27	0.3742	1.741	0.215	0.830	-3.037	3.786
x28	22.0365	11.323	1.946	0.052	-0.156	44.229
x29	2.4404	2.844	0.858	0.391	-3.134	8.015
x30	8.6692	10.108	0.858	0.391	-11.142	28.481
x31	-1.0921	2.425	-0.450	0.653	-5.846	3.662

9. L=10

a. Accuracy of logistic regression classifier on test set: 0.95

b. Confusion Matrix:

```
([[ 25,   6],
 [  4, 169]],
```

c. Classification Report:

	precision	recall	f1-score	support
0	0.86	0.81	0.83	31
1	0.97	0.98	0.97	173
avg / total	0.95	0.95	0.95	204

5-fold cross validation average accuracy: 0.944

8. **Fitting a Classifier for the selected 'l' and on the pruned set of features:**

After selecting l=9 and refitting a logistic regression model using your pruned set of features.

A. **Accuracy of logistic regression classifier on test set: 0.97**
B. **logistic regression coefficients**

```
[[ 0.20419005 -0.37276074 -0.44371708  0.02798226 -0.48128846  1.835589
32
  -0.37176504 -0.75134339  0.26951062  2.41818192  0.43254436  0.137522
9
   0.71471103 -0.45645945]]
```

C. ***Confusion matrix***

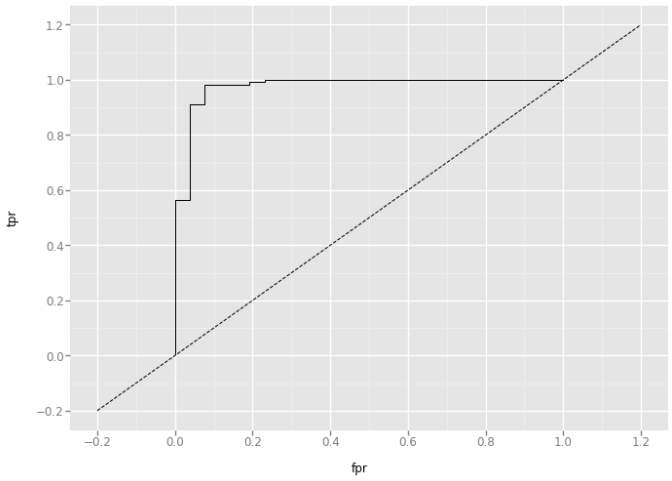
```
[[ 24,   2],
[  4, 154]]
```

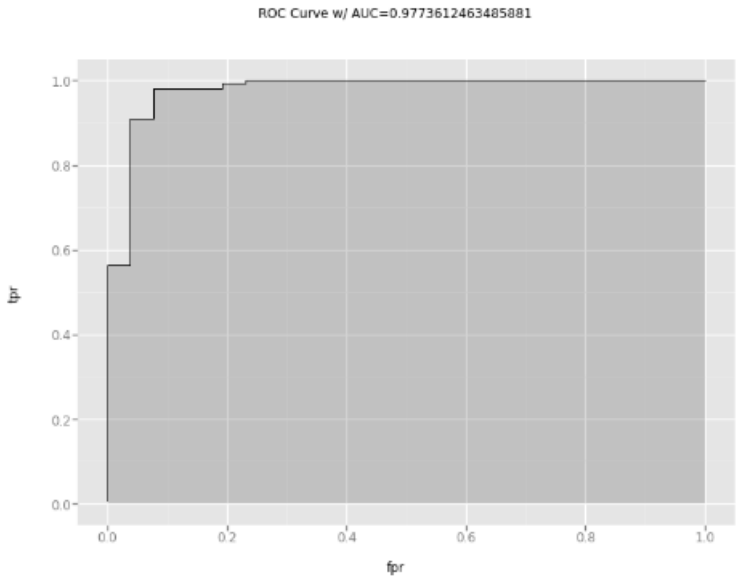
D. ***Classification Report***

	precision	recall	f1-score	support
0	0.86	0.92	0.89	26
1	0.99	0.97	0.98	158
avg / total	0.97	0.97	0.97	184

E . RFE on the selected set :

```
[ True  True  True  True  True  True  True  True  True  True  True  Tru
e
  True  True]
[1 1 1 1 1 1 1 1 1 1 1 1 1]
```





Before and After Feature selection :

a. Confusion Matrix(Before)

```
       0     1
0 ([[ 22,    4],
1 [  7, 151]]
```

b. Confusion Matrix(After)

```
       0    1
0 [[ 24,   2],
1 [  4, 154]]
```

We can clearly see in this binary classification problem that for each class for Example from abov e confusion matrix it is clear that once we have built a classifier based on the pruned set of featur es we can see the prediction of the class activity to the correct class-label improves.

c. ***Classification Report:(Before feature selection)***

	precision	recall	f1-score	support
0	0.76	0.85	0.80	26
1	0.97	0.96	0.96	158
avg / total	0.94	0.94	0.94	184

d. Classification Report:(After feature selection)

	precision	recall	f1-score	support
0	0.86	0.92	0.89	26
1	0.99	0.97	0.98	158
avg / total	0.97	0.97	0.97	184

Cross Validation is model validation technique used to verify the performance of the classifier by a statistical analysis method.
In cross validation, the given dataset is divided into training part and test part .
Different cross validation methods :

- Hold-out method
- Leave-one-out method,
- K-Fold cross validation etc.

Cross validation for feature selection.
Suppose ,we have a dataset with 1000 features and 100 samples in it. Normally, a common strategy for feature selection with cross validation would be as follows:

Wrong way:

1. Look for the best 20 features subset that show strong correlation.
2. With this subset of 20 features, multivariate classifier is built.
3. Further we make use of cross validation to estimate prediction error of the final model.

   Issue: The predictors have an unfair advantage, as they were chosen in step 1 on basis of all samples.
   As given in the course text book –***"leaving samples out after the variables have been selected does not correctly mimic the application of the classifier to a completely independent test set, since these predictors "have already seen" the left out samples.***

Correct Method:

- Divide the dataset into 10 subsets of 10 samples each as in 10-fold cross validation.
- For each group, k = 1, 2….10.
- Find best 20 features using all of the samples except that in group k.
- With the help of these features, create a multivariate classifier again using all samples except in group k.
- Now, use this classifier to predict error in the group k.

Thus, the samples on which the classifier (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.

9. Test the classifier on the test set.

a.Accuracy of logistic regression classifier on test set: 0.96

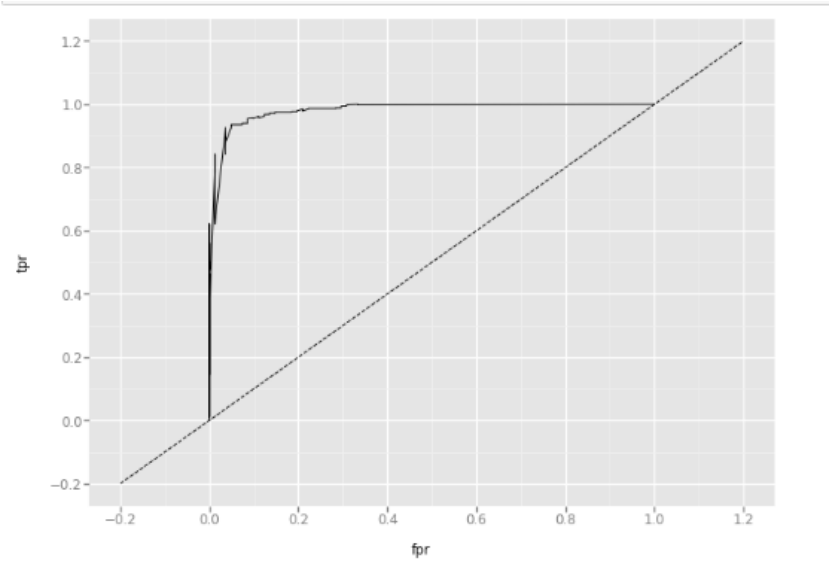
b Confusion Matrix:

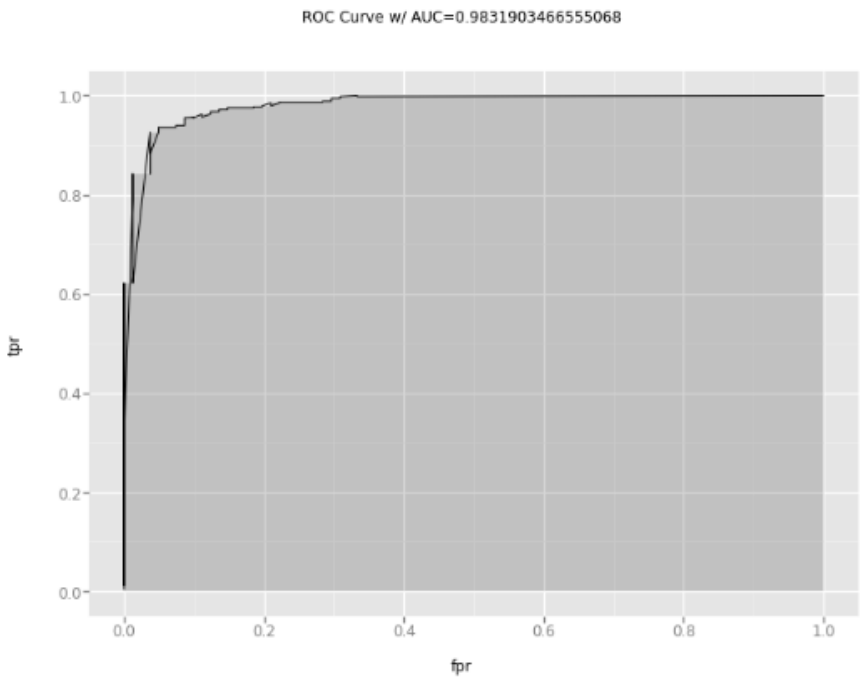
```
        0    1
0 ([[ 66,  15],
1  [ 12, 519]]
```

c. Classification Report:

	precision	recall	f1-score	support
0	0.85	0.81	0.83	81
1	0.97	0.98	0.97	531
avg / total	0.96	0.96	0.96	612

d . ROC and AUC





Optimization terminated successfully.
         Current function value: 0.100896
         Iterations 11

Logit Regression Results

Dep. Variable:	y	No. Observations:	612
Model:	Logit	Df Residuals:	598
Method:	MLE	Df Model:	13
Date:	Sun, 18 Feb 2018	Pseudo R-squ.:	0.7418
Time:	12:49:13	Log-Likelihood:	-61.749
converged:	True	LL-Null:	-239.19
		LLR p-value:	7.276e-68

	coef	std err	z	P>|z|	[0.025	0.975]
x1	0.1725	0.111	1.561	0.119	-0.044	0.389
x2	-0.6500	0.253	-2.572	0.010	-1.145	-0.155
x3	-0.3512	0.219	-1.602	0.109	-0.781	0.079
x4	0.4363	0.495	0.882	0.378	-0.533	1.406
x5	-0.2497	0.426	-0.586	0.558	-1.085	0.586
x6	2.2255	1.238	1.798	0.072	-0.200	4.651
x7	-0.3382	0.618	-0.547	0.584	-1.550	0.873
x8	-0.8670	1.502	-0.577	0.564	-3.811	2.077
x9	0.3031	0.042	7.278	0.000	0.221	0.385
x10	2.8986	0.899	3.225	0.001	1.137	4.660
x11	0.2916	0.361	0.807	0.420	-0.416	1.000
x12	0.0300	0.449	0.067	0.947	-0.849	0.909
x13	0.5463	0.412	1.325	0.185	-0.262	1.355
x14	-2.0566	3.128	-0.657	0.511	-8.188	4.074

Possibly complete quasi-separation: A fraction 0.26 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

Compare the accuracy on the test set with the cross-validation accuracy you obtained previously

- **5-fold cross validation average accuracy of the training: 0.939…i.e 93%**
- **Whereas the Accuracy of logistic regression classifier on test set: 0.96….i.e 96%**
  We can see an increase in the accuracy on the test set

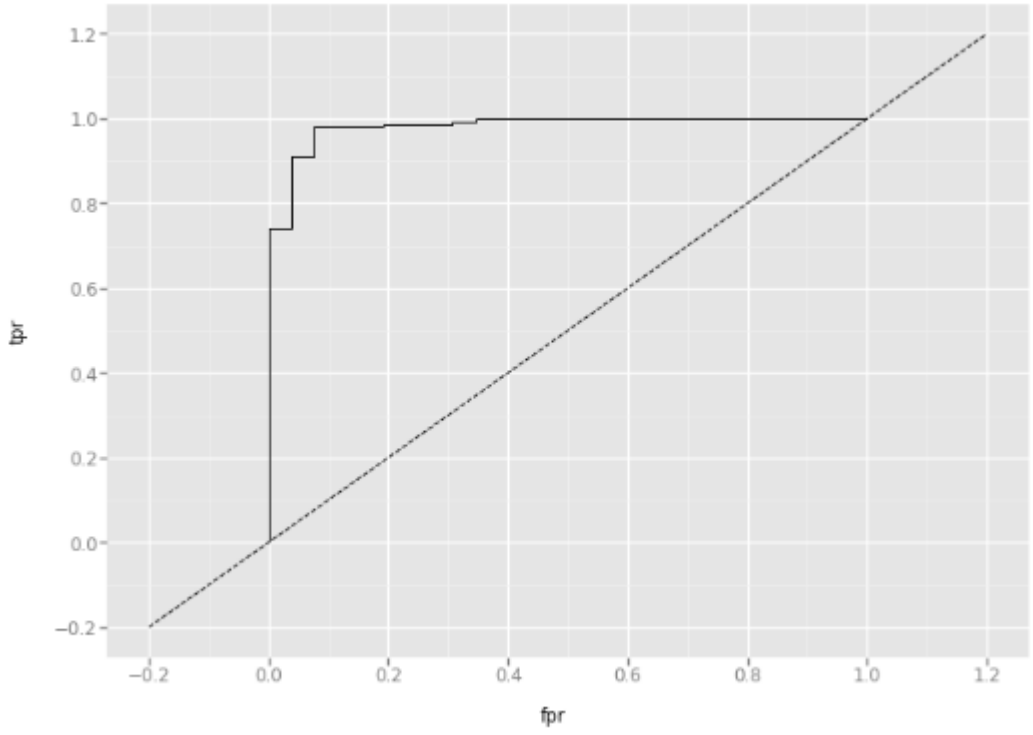
10. Case-control sampling and adjust its parameters:

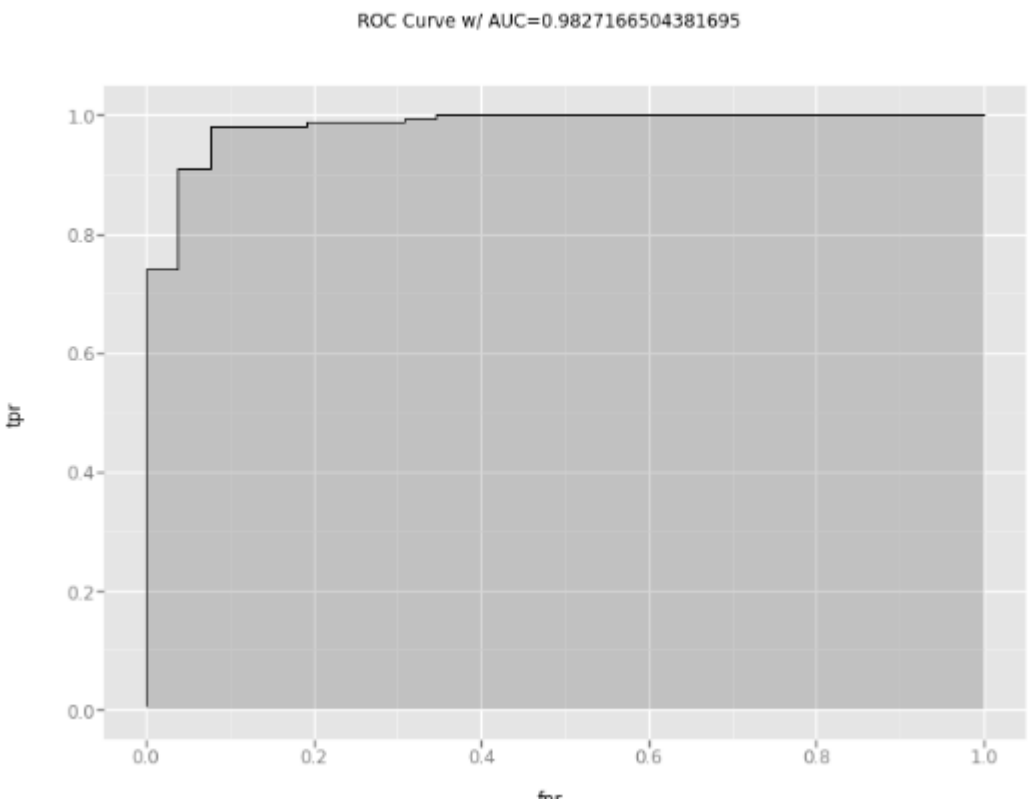
a. Confusion Matrix:

```
([[ 24,   2],
  [ 14, 144]],
```

b. Classification Report:

	precision	recall	f1-score	support
0	0.63	0.92	0.75	26
1	0.99	0.91	0.95	158
avg / total	0.94	0.91	0.92	184




ROC Curve w/ AUC=0.9827166504381695
tpr
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

11. Binary Classification Using L_1-penalized logistic regression:

Confusion Matrix:

```
([[ 24,   2],
  [ 11, 147]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.69	0.92	0.79	26
1	0.99	0.93	0.96	158
avg / total	0.94	0.93	0.93	184

12. Multi-class Classification:

 # '0'- Bending
 # '1' - cycling
 # '2' - lying
 # '3' - sitting
 # '4' - standing
 # '5' - walking

Confusion Matrix:

	0	1	2	3	4	5
0	23,	0,	0,	3,	0,	0],
1	0,	22,	1,	1,	1,	0],
2	0,	1,	27,	5,	3,	0],
3	9,	0,	4,	16,	5,	0],
4	0,	0,	5,	7,	21,	1],
5	0,	0,	0,	0,	0,	29]

Classification Report:

	precision	**recall**	**f1-score**	**support**
0	0.72	0.88	0.79	26
1	0.96	0.88	0.92	25
2	0.73	0.75	0.74	36
3	0.50	0.47	0.48	34
4	0.70	0.62	0.66	34
5	0.97	1.00	0.98	29
avg / total	0.75	0.75	0.75	184

Naive Bayes' classifier.

1. GaussianNB(priors=None)

0.6977124183006536

2. Multinomial

0.7124183006535948

***Note:** Code also contains 1.e and 1.f for pruned set of features (i.e after feature selection) and s ame procedure that was followed for 1.e and 1.f for all features is mimicked just for experimentati on purpose .

****

ISLR Exercises:

****

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. $Y = \beta 0 + \beta 1X + \beta 2X2 + \beta 3X3 + e$.

(a) Suppose that the true relationship between X and Y is linear, i.e. $Y = \beta 0 + \beta 1X + e$. 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.

Ans:

Expectation is that the polynomial regression should have a lower training RSS than the linear regression because it could make a tighter fit against data that matched with a wider irreducible error e.

(b) Answer (a) using test rather than training RSS.

Ans:

Converse to the answer provided to above (a), I would expect the polynomial regression to 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.

Ans:

Because of higher flexibility, Polynomial regression has lower train RSS than the linear fit :no matter what the underlying true relationship is, the more flexible model will closer follow points and reduce train RSS.

(d) Answer (c) using test rather than training RSS.

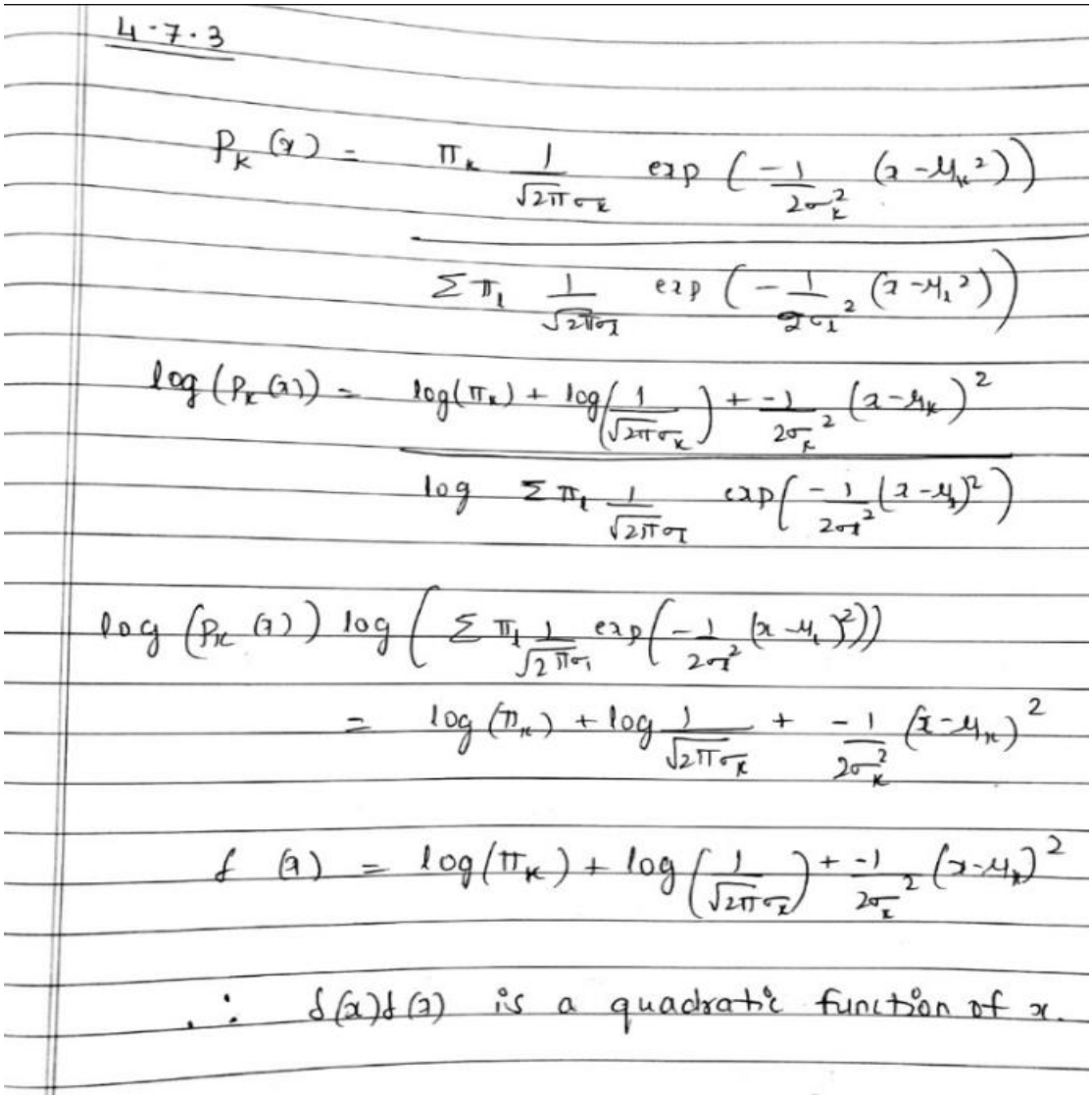
Ans:

- There is not enough evidence from the information provided to tell which test RSS would be lower for either regression given the problem statement is defined as not knowing how far it really 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.
- If it is closer to cubic than linear, the cubic regression test RSS could be lower than the linear regression test RSS.
- It is due to bias-variance tradeoff: it is not clear what level of flexibility will fit data better.

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, $X \sim N(\mu k, \sigma 2k)$. 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.

Solution:



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 $\bar{X} = 10$, while the mean for those that didn't was $\bar{X} = 0$. In addition, the variance of *X* for these two sets of companies was $\hat{\sigma}2 = 36$. 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 $f(x) = \sqrt{1}\ 2\pi\sigma 2\ e-(x-\mu)2/2\sigma 2$. *You will need to use Bayes' theorem.*

4.7.7

$$p_K(x) = \frac{\pi_K \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_K)^2\right)}{\sum \pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_k)^2\right)}$$

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

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

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

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

$$= 75.2\%$$