CSE 574 Programming Assignment 2
Classification and Regression

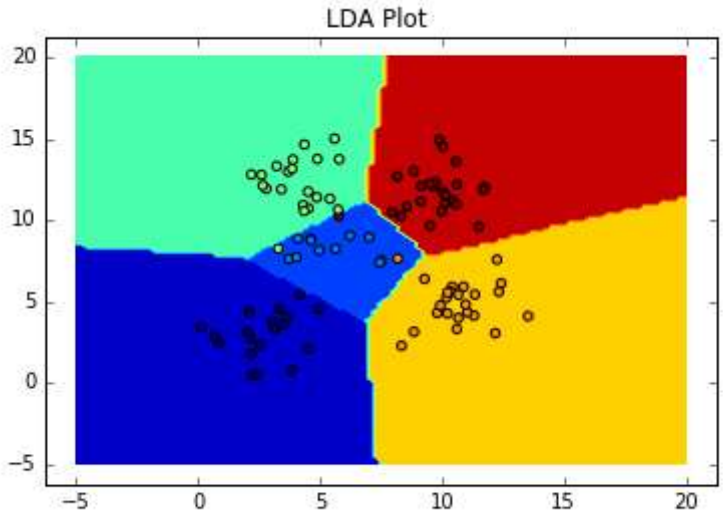
Group number 12
Bhavika Jain - 50170168
Chen Song - 50060333
Pranav Jain - 50169659

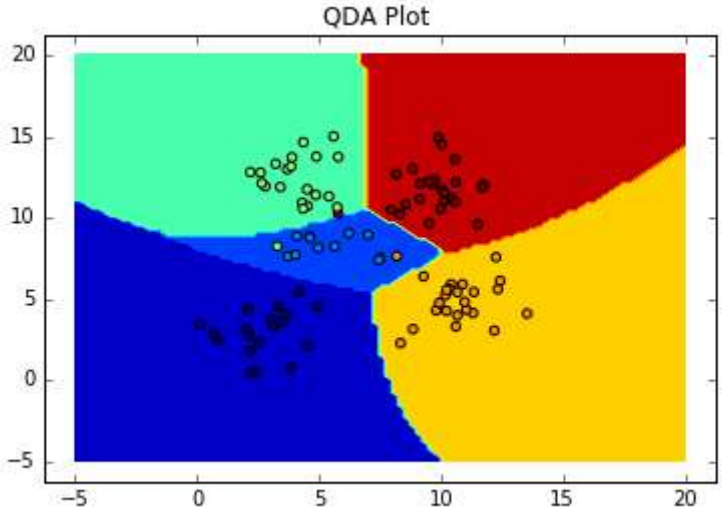
Problem 1 - Experiment with Gaussian discriminators:

We got the following accuracies for LDA and QDA:

Accuracy for LDA: 97%
Accuracy for QDA: 96%

The plots for the discriminating boundaries for LDA and QDA are as shown below:





As seen from plots above, the decision boundaries are lines in LDA and in case of QDA they are curves. The difference between the plots is due to the fact that LDA uses the covariance of the entire data set whereas QDA uses the covariance of each class separately. In LDA, since there is a single covariance matrix, the points were easily classified into different classes. Whereas, in case of QDA, since the boundaries are nonlinear, there is more accurate classification in case when the data is complex. In this case however, since the data was in almost distinct groups, LDA did a slightly better job at classifying.

In cases where the data is more complex though, with possible overlaps and non-linear boundaries, QDA overperforms LDA by a considerable margin. So, if a faster algorithm is required, LDA can be implemented initially. But to get better results, it should be followed by QDA to get more flexible boundaries.

Problem 2 - Experiment with Linear Regression:

We got the following RMSE results for the train and test data respectively:

RMSE for train data:

a. RMSE without intercept: 138.20074835
b. RMSE with intercept: 46.7670855937

RMSE for test data:

a. RMSE without intercept: 326.764994385
b. RMSE with intercept: 60.8920370921

The above results indicate that when an intercept is used, for both train and test data, the RMSE value decreases. This indicates that the error gets reduced with the introduction of intercept which is a desirable property. In case of train data, the error without intercept is nearly 3 times than

that of the error with intercept. Also in case of test data, the error without intercept is nearly 5.4 times than that of the error with intercept. This shows that the error gets reduced by a greater factor in case of test data when using an intercept - which is expected since a trained system should have less error. The RMSE value is higher without intercept as the line learnt passes through origin restricting it to rotation only whereas with intercept, it gives the line the ability to rotate as well as translate. As a result, the hypothesis learnt is much closer to the true concept and error is reduced when an intercept is used. Overall, the accuracy is higher with intercept compared to without intercept.

Problem 3 - Experiment with Ridge Regression:

The table given below shows the RMSE values for different values of lambda for both test and train data.

Lambda	Test RMSE	Training RMSE
0	60.8920371	46.76708559
0.01	54.61177638	48.02949321
0.02	53.86068684	48.51877311
0.03	53.58116823	48.85468415
0.04	53.46026945	49.11332857
0.05	53.41035232	49.32721801
0.06	53.3978484	49.51291236
0.07	53.40739644	49.67974992
0.08	53.43107466	49.83337884
0.09	53.46442201	49.97739748
0.1	53.50474691	50.11419242
0.11	53.55033062	50.24539818
0.12	53.60002129	50.37216394
0.13	53.65301361	50.49531549
0.14	53.70872289	50.6154574
0.15	53.76671005	50.73303941
0.16	53.82663522	50.84840091
0.17	53.88822809	50.9618013
0.18	53.95126849	51.07344124
0.19	54.01557335	51.18347767
0.2	54.08098762	51.2920347
0.21	54.147378	51.39921153
0.22	54.21462838	51.50508835
0.23	54.28263649	51.60973068
0.24	54.35131145	51.71319268
0.25	54.42057188	51.81551966
0.26	54.49034444	51.91675004
0.27	54.56056269	52.01691676
0.28	54.6311662	52.11604846

0.29	54.70209979	52.21417036
0.3	54.77331293	52.31130499
0.31	54.84475924	52.40747266
0.32	54.91639604	52.50269198
0.33	54.98818405	52.5969801
0.34	55.060087	52.69035307
0.35	55.1320714	52.78282598
0.36	55.20410629	52.87441317
0.37	55.27616302	52.96512834
0.38	55.34821508	53.05498467
0.39	55.42023793	53.14399488
0.4	55.4922088	53.23217132
0.41	55.56410664	53.31952602
0.42	55.63591191	53.40607071
0.43	55.70760652	53.49181688
0.44	55.7791737	53.5767758
0.45	55.85059792	53.6609585
0.46	55.92186478	53.74437585
0.47	55.99296098	53.82703853
0.48	56.06387418	53.90895705
0.49	56.13459299	53.99014176
0.5	56.20510685	54.07060285
0.51	56.27540603	54.15035036
0.52	56.34548155	54.22939418
0.53	56.41532512	54.30774406
0.54	56.4849291	54.3854096
0.55	56.55428647	54.46240025
0.56	56.62339077	54.53872535
0.57	56.69223609	54.61439406
0.58	56.760817	54.68941543
0.59	56.82912855	54.76379836
0.6	56.89716621	54.83755162
0.61	56.96492589	54.91068383
0.62	57.03240384	54.9832035
0.63	57.0995967	55.05511899
0.64	57.16650143	55.12643852
0.65	57.23311531	55.1971702
0.66	57.29943591	55.26732198
0.67	57.36546107	55.33690171
0.68	57.4311889	55.4059171
0.69	57.49661775	55.47437573
0.7	57.56174618	55.54228505
0.71	57.62657297	55.6096524
0.72	57.69109712	55.67648499
0.73	57.75531777	55.74278991

0.74	57.81923428	55.80857413
0.75	57.88284615	55.8738445
0.76	57.94615303	55.93860776
0.77	58.00915471	56.00287053
0.78	58.07185115	56.06663932
0.79	58.13424238	56.12992053
0.8	58.19632858	56.19272045
0.81	58.25811005	56.25504526
0.82	58.31958716	56.31690103
0.83	58.38076039	56.37829374
0.84	58.44163032	56.43922927
0.85	58.5021976	56.49971337
0.86	58.56246297	56.55975172
0.87	58.62242722	56.61934989
0.88	58.68209124	56.67851337
0.89	58.74145595	56.73724753
0.9	58.80052236	56.79555768
0.91	58.85929152	56.85344902
0.92	58.91776451	56.91092665
0.93	58.9759425	56.96799562
0.94	59.03382668	57.02466086
0.95	59.09141827	57.08092724
0.96	59.14871855	57.13679953
0.97	59.20572881	57.19228244
0.98	59.26245041	57.24738057
0.99	59.3188847	57.30209848
1	59.37503307	57.35644063

Table 3.1: RMSE for train and test data

Comparison of RMSE using Ridge and OLE Regression:

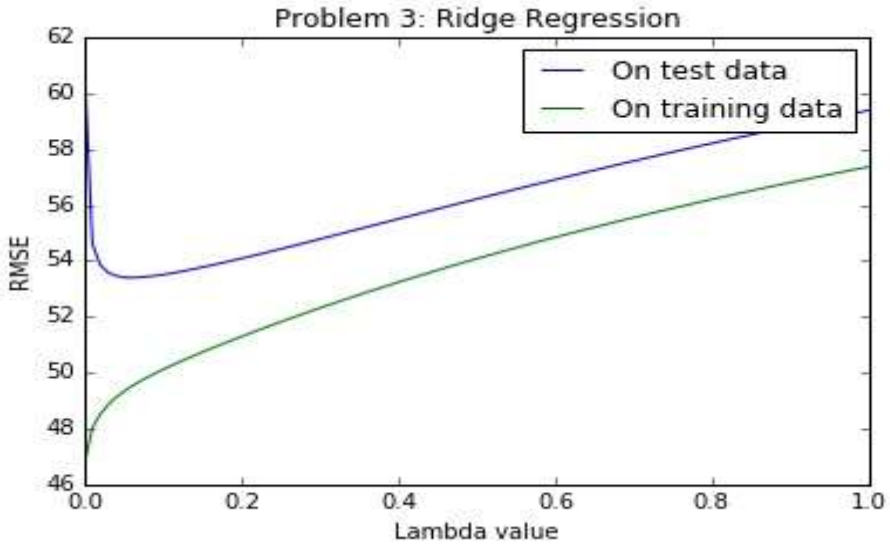


Fig 3.1: Lambda vs RMSE

1. As shown in Table 3.1, the **optimum value of lambda** observed is: **0.06.** At this lambda value, we see that the **RMSE** value obtained is the lowest **: 53.3978484.**
2. For train data, using OLE, RMSE with intercept: 46.7670855937**.** And using Ridge regression, we get the lowest RMSE value for train data as 46.76708559 at lambda = 0**.**
3. From Problem 2, using OLE, for test data, RMSE with intercept: 60.8920370921**.** And, using Ridge regression, lowest RMSE for test data is 53.3978484**.**
4. This shows that RMSE reduces in case of Ridge regression for test data, indicating that Ridge regression performs better than OLE for the given data set**.**

As seen in fig 3.1, initially as **λ** increases, the RMSE value decreases**.** This is expected, as the term $\frac{1}{2} * \lambda * w^T w$ becomes part of the minimization, keeping the weights in control**.** This reduces overfitting on training data and reduces error on test data**.** However, as **λ** increases beyond 0.06, we see that RMSE increases**.** This is an expected behaviour as when **λ** is high it tries to limit the growth of weight vector**.** This causes underfitting of data and RMSE increases on test data**.**

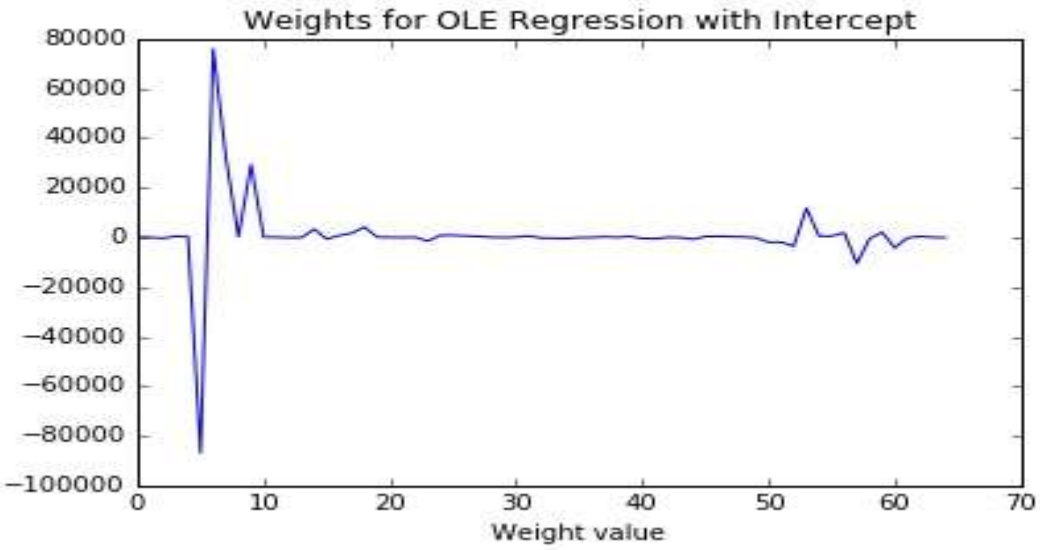


Fig 3.2: Weights for OLE Regression with intercept

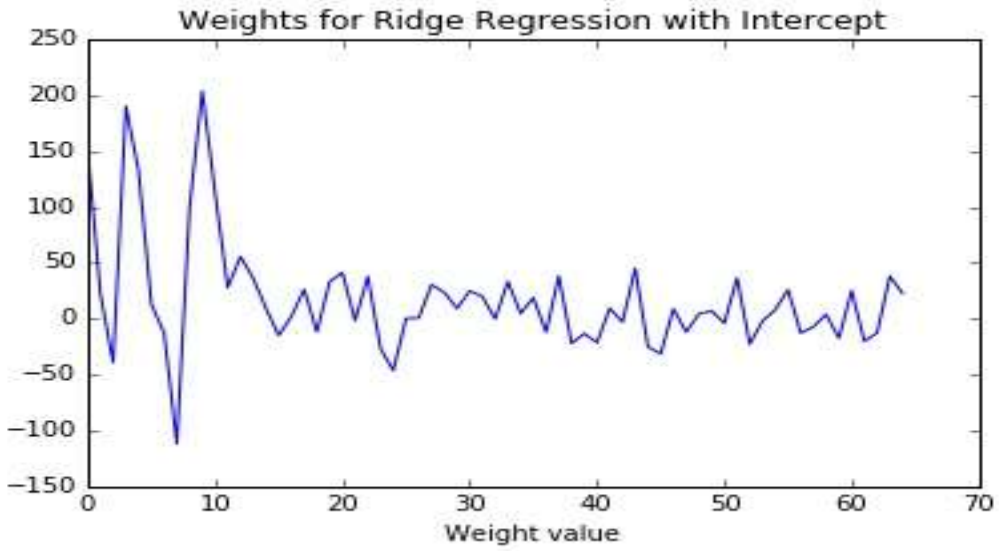


Fig 3.3: Weights for Ridge Regression with intercept

1. Fig 3.2 and Fig 3.3 show that the weights learnt using OLE regression have a much higher magnitude compared to the weights learnt using Ridge regression.
2. Below is the variance in weight vector using OLE and Ridge regression.
   Variance (OLE): 237806821.459.
   Variance (Ridge Regression): 13131.2839474.

The reason for the huge difference between weight vector variance can be attributed to regularization used in Ridge regression. The term $\frac{1}{2} * \lambda * w^T w$ incorporated into J(w) as part of

minimization, keeps the weight in control. Hence the weights observed after learning process in Ridge Regression show such limited magnitudes.

Problem 4 - Using Gradient Descent for Ridge Regression Learning:

The table below shows the RMSE values for test and train data using different values for lambda using Gradient descent approach.

Lambda	RMSE(Test data)	RMSE(Train data)
0	59.66102423	47.05871728
0.01	54.61177354	48.02949327
0.02	53.86068683	48.51877311
0.03	53.58116795	48.85468414
0.04	53.46026934	49.11332857
0.05	53.41035289	49.32721801
0.06	53.39784862	49.51291234
0.07	53.40739631	49.67974992
0.08	53.43107426	49.83337882
0.09	53.46442175	49.97739743
0.1	53.50474683	50.11419241
0.11	53.55033064	50.24539818
0.12	53.6000215	50.37216397
0.13	53.6530138	50.49531551
0.14	53.70872299	50.61545737
0.15	53.76671003	50.73303938
0.16	53.82663522	50.84840089
0.17	53.88822804	50.96180129
0.18	53.95126842	51.07344124
0.19	54.01557326	51.18347768
0.2	54.08098754	51.29203472
0.21	54.14737795	51.39921155
0.22	54.21462831	51.50508836
0.23	54.2826365	51.60973063
0.24	54.35131148	51.71319264
0.25	54.42057191	51.81551964
0.26	54.49034447	51.91675002
0.27	54.56056283	52.01691673
0.28	54.63116634	52.11604844
0.29	54.70209991	52.21417036
0.3	54.77331306	52.31130501
0.31	54.84475933	52.40747268
0.32	54.9163961	52.50269197
0.33	54.98818417	52.5969801
0.34	55.0600871	52.69035306
0.35	55.13207148	52.78282598

0.36	55.20410635	52.87441317
0.37	55.27616304	52.96512835
0.38	55.3482151	53.05498467
0.39	55.42023794	53.14399488
0.4	55.49220882	53.23217132
0.41	55.56410665	53.31952601
0.42	55.63591191	53.40607071
0.43	55.70760652	53.49181688
0.44	55.77917356	53.57677578
0.45	55.8505978	53.66095849
0.46	55.92186469	53.74437584
0.47	55.99296092	53.82703853
0.48	56.06387415	53.90895705
0.49	56.13459297	53.99014176
0.5	56.20510685	54.07060285
0.51	56.27540605	54.15035036
0.52	56.34548158	54.22939418
0.53	56.41532515	54.30774406
0.54	56.48492913	54.3854096
0.55	56.5542865	54.46240025
0.56	56.62339075	54.53872534
0.57	56.69223607	54.61439405
0.58	56.76081698	54.68941542
0.59	56.82912853	54.76379835
0.6	56.89716619	54.83755161
0.61	56.96492586	54.91068383
0.62	57.03240381	54.9832035
0.63	57.09959667	55.05511899
0.64	57.16650139	55.12643852
0.65	57.23311527	55.1971702
0.66	57.29943585	55.26732198
0.67	57.36546099	55.33690171
0.68	57.43118894	55.4059171
0.69	57.4966178	55.47437573
0.7	57.56174624	55.54228505
0.71	57.62657303	55.6096524
0.72	57.69109716	55.67648499
0.73	57.75531781	55.74278991
0.74	57.81923431	55.80857413
0.75	57.88284617	55.8738445
0.76	57.94615304	55.93860776
0.77	58.00915472	56.00287053
0.78	58.07185114	56.06663932
0.79	58.13424237	56.12992053
0.8	58.19632858	56.19272045

0.81	58.25811008	56.25504525
0.82	58.31958718	56.31690104
0.83	58.38076045	56.37829375
0.84	58.44163031	56.43922926
0.85	58.5021976	56.49971337
0.86	58.56246293	56.55975171
0.87	58.62242721	56.61934989
0.88	58.68209124	56.67851336
0.89	58.74145594	56.73724753
0.9	58.80052237	56.79555769
0.91	58.85929159	56.85344902
0.92	58.91776454	56.91092665
0.93	58.97594253	56.96799562
0.94	59.03382677	57.02466087
0.95	59.0914183	57.08092725
0.96	59.14871857	57.13679953
0.97	59.20572882	57.19228243
0.98	59.26245042	57.24738057
0.99	59.3188847	57.30209848
1	59.37503309	57.35644063

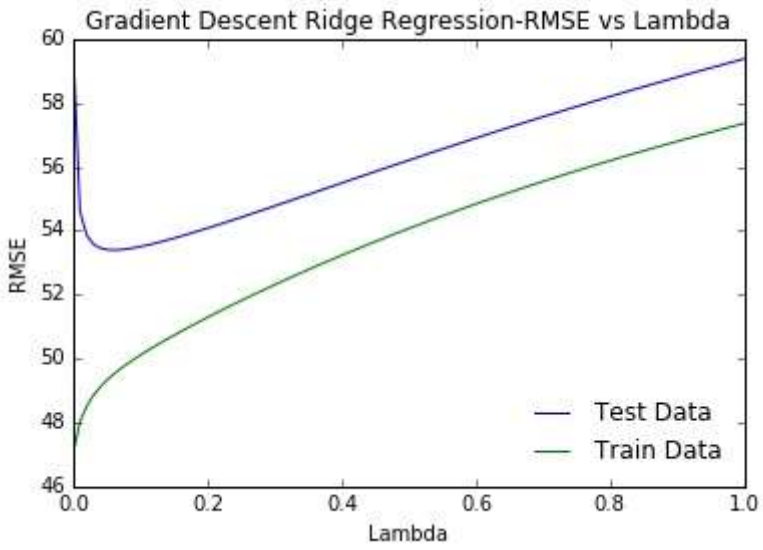


Fig 4.1 : RMSE vs Lambda using Gradient Descent

As seen from the graph above, using Gradient Descent we observe behaviour similar to what we saw in Problem 3 where derivate was used for learning weights. As seen in fig 4.1, RMSE initially decreases with increase in value of $\boldsymbol{\lambda}$, but as $\boldsymbol{\lambda}$ increases beyond 0.06, RMSE increases which is expected.

Overall, we can see that Gradient Descent gives satisfactory results similar to what we saw using Ridge Regression.

Problem 5 - Non-linear Regression:

Comparison of RMSE vs Lambda(Test Data) :

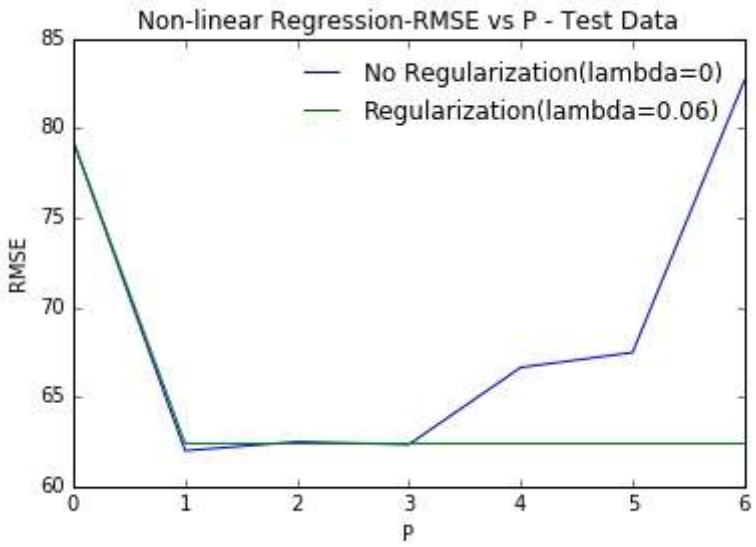


Fig 5.1 : RMSE vs P for Test data using NonLinear Regression

For Test Data :

p	RMSE(No Regularization, λ=0)	RMSE(Regularization, λ=0.06)
0	79.28685132	79.28986043
1	62.00834404	62.41679633
2	62.5070244	62.41461412
3	62.35363292	62.41460339
4	66.658292	62.41460301
5	67.48948346	62.41460301
6	82.66473945	62.41460301

Table 5.1: RMSE values for different values of p

As seen from the results above, when p = 0, the RMSE value is high. This is expected as the regression line is horizontal and no learning is done from training data.

According to table 5.1, without regularization, the error reaches a minimum value at p=1. After that it increases gradually until p=3. After this, it rapidly increases to a value even greater than that at p=0. The increase is due to the fact that there is no regularization, resulting in overfitting of data.

However, with regularization, beyond p = 3, the value of RMSE does not increase, as **λ** provides regularization and avoids overfitting of data.

As seen from table 5.1, using regularization, the error decreases initially, reaching a minimum at p=4. After that it remains constant till p=6. It could be because regularization results into more penalizing of higher weights. This makes the curve almost linear after this minimum is reached.

For Train Data:

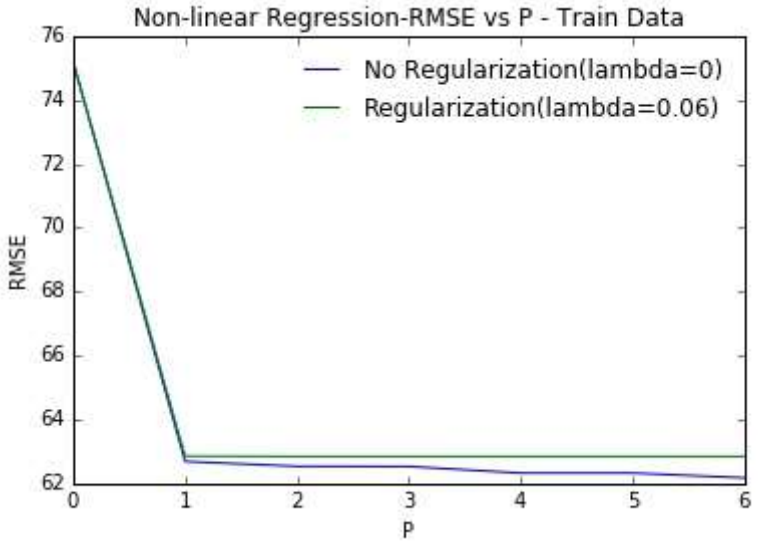


Fig 5.2: Non linear regression RMSE vs P for train data

p	RMSE(No Regularization, λ=0)	RMSE(Regularization, λ=0.06)
0	75.17120818	75.17121728
1	62.69701275	62.86365503
2	62.54470138	62.85449318
3	62.53949684	62.85445514
4	62.33356293	62.85445360
5	62.33303424	62.85445358
6	62.18427011	62.85445358

Table 5.2: RMSE for train data with and without regularization for different values of p

On training data, RMSE value remains low beyond p=3, as training was done on this data itself.

Optimum values of p observed:

P = 1 when **λ** = 0 (No Regularization)
P = 4 when **λ** = 0.06 (With Regularization)

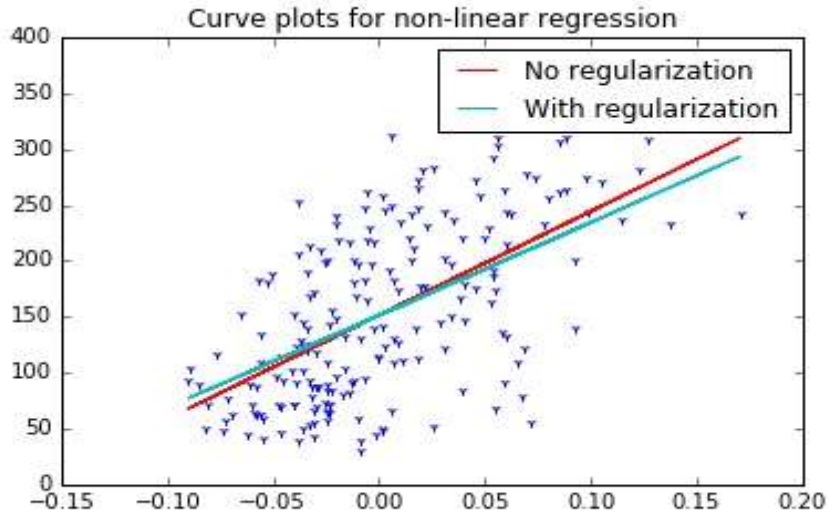


Fig 5.3

Above fig 5.3 shows the curves obtained for optimal p values both for **λ**= 0 and **λ** = 0.06**.** Ideally using regularization, the curve obtained should be non-linear, however the regularization factor(**λ**) , makes it a straight line**.** Hence both the curves appear as straight line**.**

Problem 6 - Interpreting Results:

Comparison between approaches in Problem 2-5 based on training and testing error:

Prob Num	Type	Training RMSE	Testing RMSE
2	OLE With intercept	46.76708559	60.89203709
2	OLE Without intercept	138.20074835	326.7649943
3	Ridge Regression	46.76708559	53.3978484
4	Gradient Descent	47.05871728	53.39784862
5	Non-Linear Regression with regularization	62.85445358	62.41460301
5	Non-Linear Regression without regularization	62.18427011	62.00834404

Table 6.1 : Error comparison

Below table shows the time taken for each type of regression technique.

Method	Time taken
OLE	0. 005003690719604492
Ridge Regression	0.3193521499633789
Gradient Descent	3.6127495765686035
NonLinear Regression	0.015626192092895508

Table 6.2 : Time comparison

We can choose the best setting based on two metrics : **error and time taken.** We will consider testing error since we can tolerate higher errors while training but not while testing.

If accuracy is more important, **Ridge Regression** and **Gradient Descent** seems to be good choices. As seen from table 6.1, the RMSE values obtained are the lowest using these two techniques for test data. Comparing the results of Linear and Non-Linear regression techniques to Ridge Regression and Gradient Descent techniques, we see that error is higher for test data and hence we don't recommend them for the given data set.

If we consider time taken as more important, as seen from table 6.2, we observe that normal Ridge regression runs significantly faster compared to gradient descent ridge regression. Thus, in case faster prediction is required for small data sets, normal ridge regression technique can be used, while the gradient descent could be faster for huge data sets. Also, if time taken has to be minimized, but accuracy can be compromised, Linear Regression can be used to obtain faster result, as the time taken by OLE is the least compared to others (from table 6.2).