Abhishek Kannan, Arjun Sundaresh, Sammok Kabasi
Group Number 44
CSE 574 MACHINE LEARNING

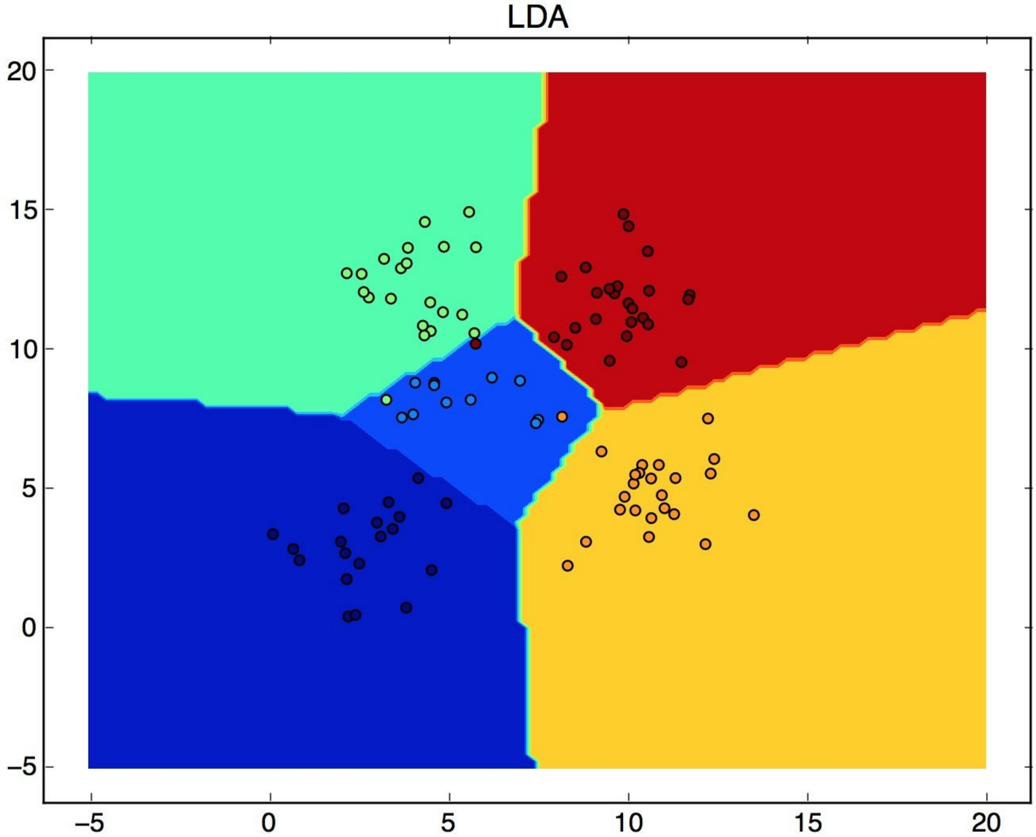
CLASSIFICATION AND REGRESSION REPORT

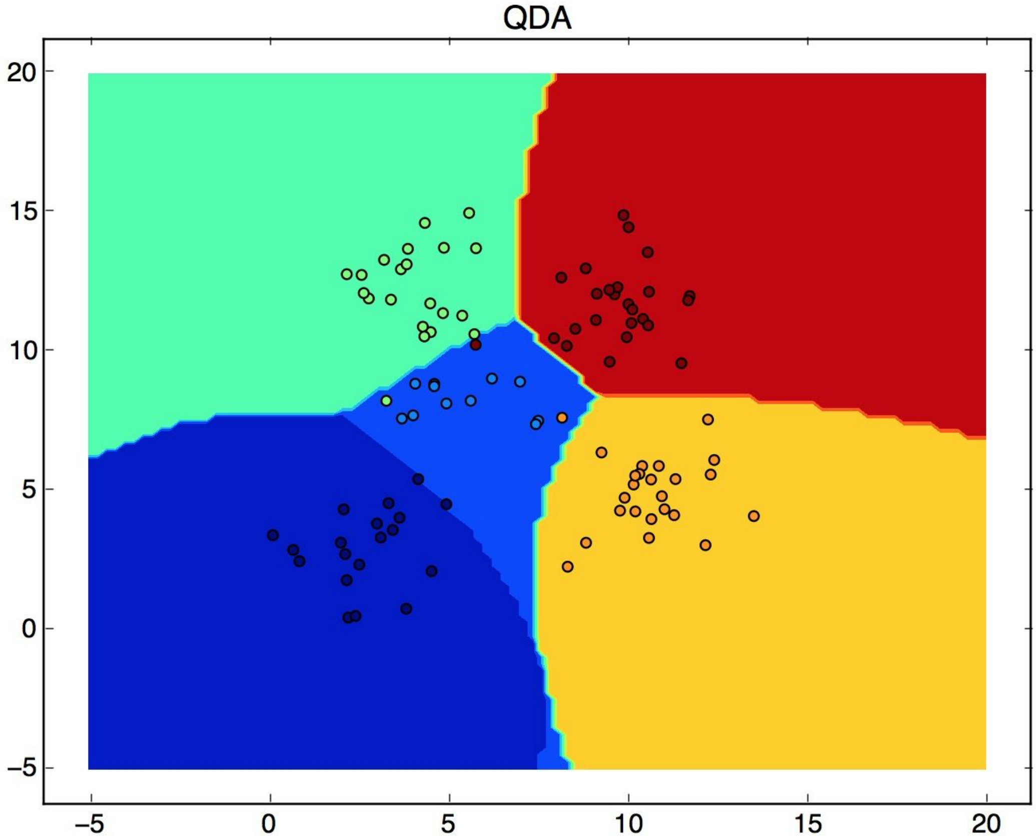
PROBLEM 1 - EXPERIMENT WITH GAUSSIAN DISCRIMINATORS

1.1 Accuracy

Accuracy for LDA = 97%
Accuracy for QDA = 96%

1.2 Boundaries





For LDA, the covariance matrix is the same for all the classes. So, the discriminating boundaries will be straight lines.

For QDA, the covariance matrix is different for each class. So, the discriminating boundaries are curved lines.

PROBLEM 2 - EXPERIMENT WITH LINEAR REGRESSION

2.1 RMSE

RMSE without intercept for training data = 138.20074835
RMSE with intercept for training data = 46.7670855937

RMSE without intercept for testing data = 326.764994388
RMSE with intercept for testing data = 60.892037094

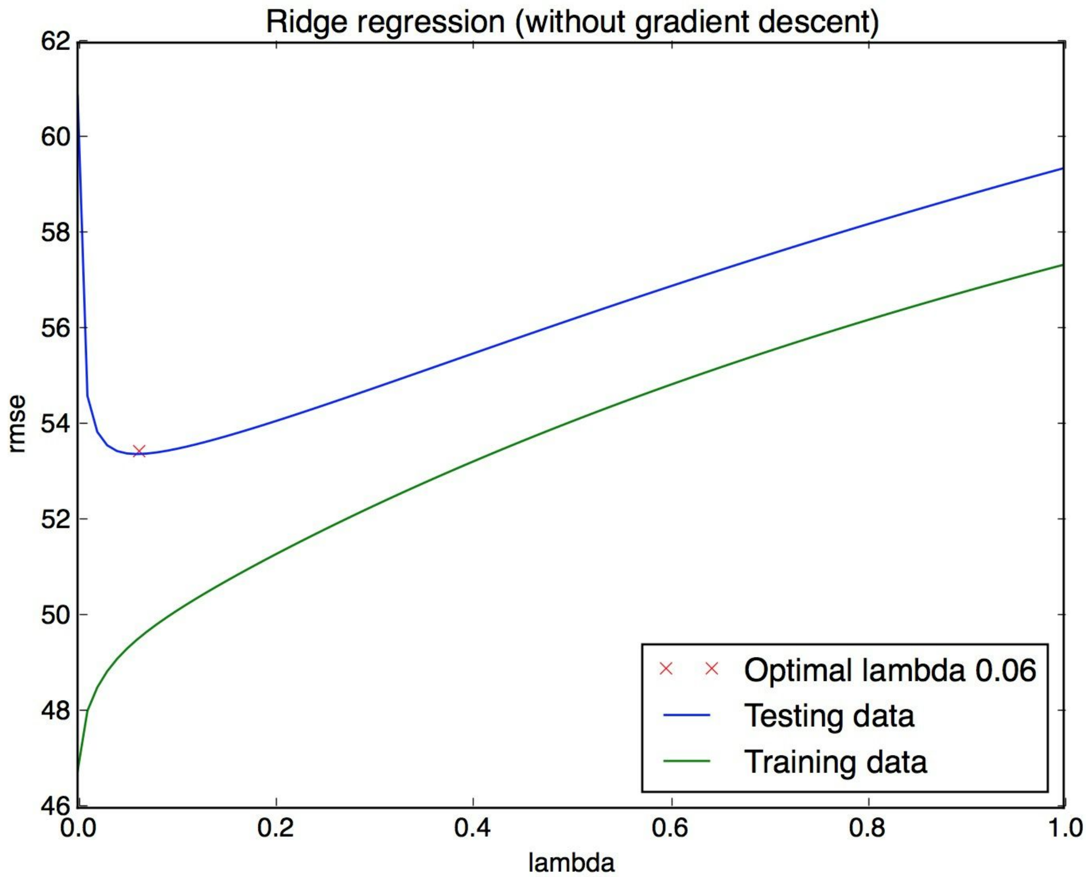
The RMSE is low in the cases where we use intercepts. So, the method using intercepts is better.

PROBLEM 3 - EXPERIMENT WITH RIDGE REGRESSION

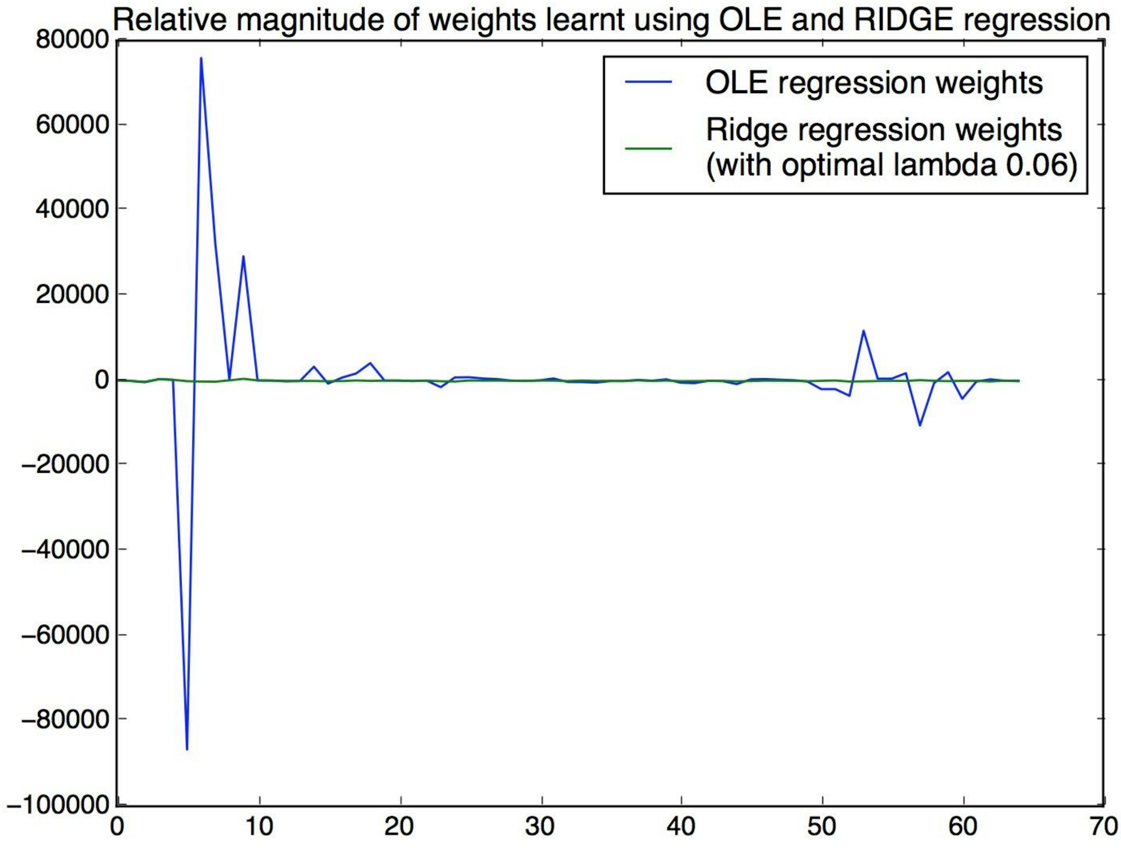
3.1 RMSE for training and test data using the *testoleregression* function

Lambda	Training RMSE	Testing RMSE	Lambda	Training RMSE	Testing RMSE
0	46.76708559	60.89203709			
0.01	48.02949321	54.61177638	0.51	54.15035036	56.27540603
0.02	48.51877311	53.86068684	0.52	54.22939418	56.34548155
0.03	48.85468415	53.58116823	0.53	54.30774406	56.41532512
0.04	49.11332857	53.46026945	0.54	54.3854096	56.4849291
0.05	49.32721801	53.41035232	0.55	54.46240025	56.55428647
0.06	49.51291236	**53.3978484**	0.56	54.53872535	56.62339077
0.07	49.67974992	53.40739644	0.57	54.61439406	56.69223609
0.08	49.83337884	53.43107466	0.58	54.68941543	56.760817
0.09	49.97739748	53.46442201	0.59	54.76379836	56.82912855
0.1	50.11419242	53.50474691	0.6	54.83755162	56.89716621
0.11	50.24539818	53.55033062	0.61	54.91068383	56.96492589
0.12	50.37216394	53.60002129	0.62	54.9832035	57.03240384
0.13	50.49531549	53.65301361	0.63	55.05511899	57.0995967
0.14	50.6154574	53.70872289	0.64	55.12643852	57.16650143
0.15	50.73303941	53.76671005	0.65	55.1971702	57.23311531
0.16	50.84840091	53.82663522	0.66	55.26732198	57.29943591
0.17	50.9618013	53.88822809	0.67	55.33690171	57.36546107
0.18	51.07344124	53.95126849	0.68	55.4059171	57.4311889
0.19	51.18347767	54.01557335	0.69	55.47437573	57.49661775
0.2	51.2920347	54.08098762	0.7	55.54228505	57.56174618
0.21	51.39921153	54.147378	0.71	55.6096524	57.62657297
0.22	51.50508835	54.21462838	0.72	55.67648499	57.69109712
0.23	51.60973068	54.28263649	0.73	55.74278991	57.75531777
0.24	51.71319268	54.35131145	0.74	55.80857413	57.81923428
0.25	51.81551966	54.42057188	0.75	55.8738445	57.88284615
0.26	51.91675004	54.49034444	0.76	55.93860776	57.94615303
0.27	52.01691676	54.56056269	0.77	56.00287053	58.00915471
0.28	52.11604846	54.6311662	0.78	56.06663932	58.07185115
0.29	52.21417036	54.70209979	0.79	56.12992053	58.13424238
0.3	52.31130499	54.77331293	0.8	56.19272045	58.19632858
0.31	52.40747266	54.84475924	0.81	56.25504526	58.25811005
0.32	52.50269198	54.91639604	0.82	56.31690103	58.31958716
0.33	52.5969801	54.98818405	0.83	56.37829374	58.38076039
0.34	52.69035307	55.060087	0.84	56.43922927	58.44163032
0.35	52.78282598	55.1320714	0.85	56.49971337	58.5021976
0.36	52.87441317	55.20410629	0.86	56.55975172	58.56246297
0.37	52.96512834	55.27616302	0.87	56.61934989	58.62242722
0.38	53.05498467	55.34821508	0.88	56.67851337	58.68209124
0.39	53.14399488	55.42023793	0.89	56.73724753	58.74145595
0.4	53.23217132	55.4922088	0.9	56.79555768	58.80052236
0.41	53.31952602	55.56410664	0.91	56.85344902	58.85929152
0.42	53.40607071	55.63591191	0.92	56.91092665	58.91776451
0.43	53.49181688	55.70760652	0.93	56.96799562	58.9759425
0.44	53.5767758	55.7791737	0.94	57.02466086	59.03382668
0.45	53.6609585	55.85059792	0.95	57.08092724	59.09141827
0.46	53.74437585	55.92186478	0.96	57.13679953	59.14871855
0.47	53.82703853	55.99296098	0.97	57.19228244	59.20572881
0.48	53.90895705	56.06387418	0.98	57.24738057	59.26245041
0.49	53.99014176	56.13459299	0.99	57.30209848	59.3188847
0.5	54.07060285	56.20510685	1	57.35644063	59.37503307

3.2 Plot the errors for different values of lambda



3.3 Relative magnitudes of weights learnt using OLE and RIDGE regression



We can see from the plot above that the weights learnt using ridge regression are much smaller than the weights learnt using OLE regression. This can be attributed to the fact that ridge regression normalizes the high weights. Ridge regression does not assign high weights unnecessarily to certain input attributes. So, ridge regression will perform better.

3.4 Comparison of OLE and ridge regression in terms of error

OLE Regression
RMSE with intercept for training data = 46.7670855937
RMSE with intercept for testing data = 60.892037094

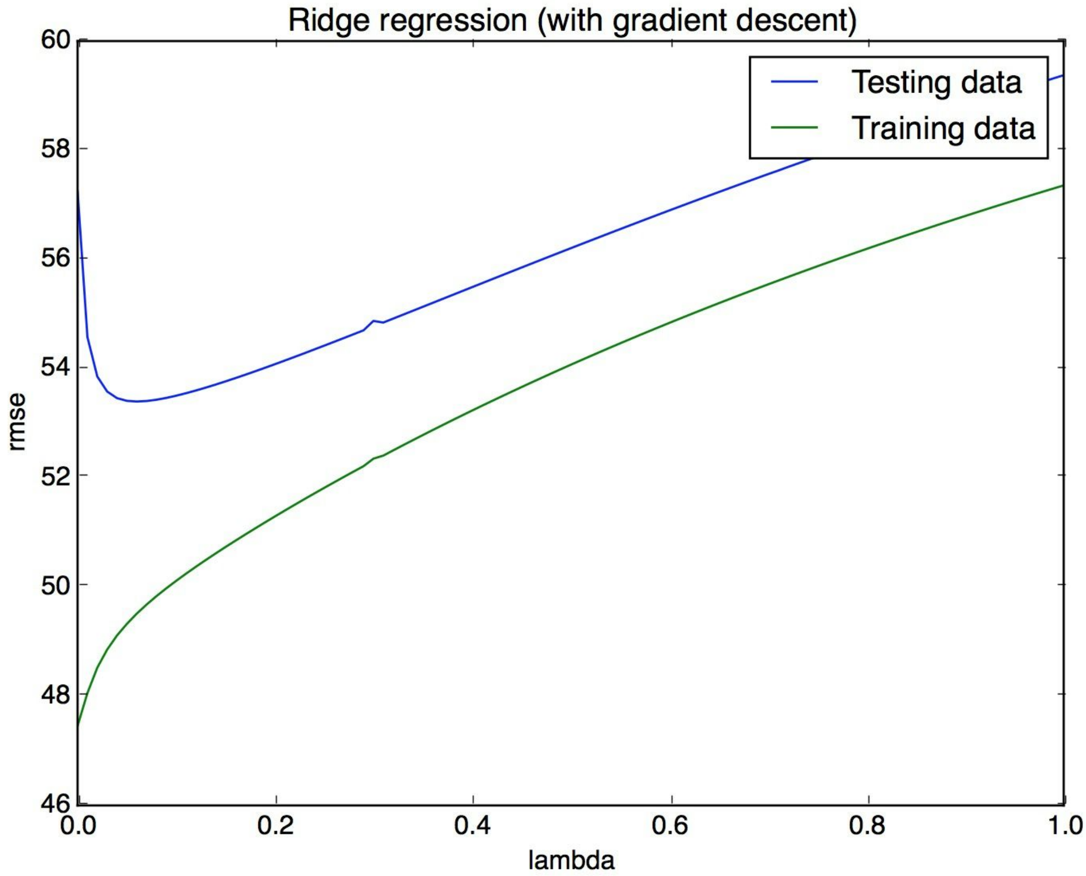
Ridge Regression (with optimal lambda 0.06)
RMSE with intercept for training data = 49.51291236
RMSE with intercept for testing data = **53.3978484**

The testing RMSE obtained using ridge regression is lesser than the testing RMSE obtained using OLE regression. So, ridge regression is a better approach.

3.5 Optimal lambda
The optimal lambda value is **0.06** because it minimizes the testing RMSE for the dataset.

PROBLEM 4 - RIDGE REGRESSION WITH GRADIENT DESCENT

4.1 Plot the errors for different values of lambda



4.2 Comparison with results obtained using normal ridge regression

The results obtained using gradient descent are almost identical to those obtained by normal ridge regression. The lines produced using gradient descent are not as smooth as those produced using normal ridge regression without gradient descent.

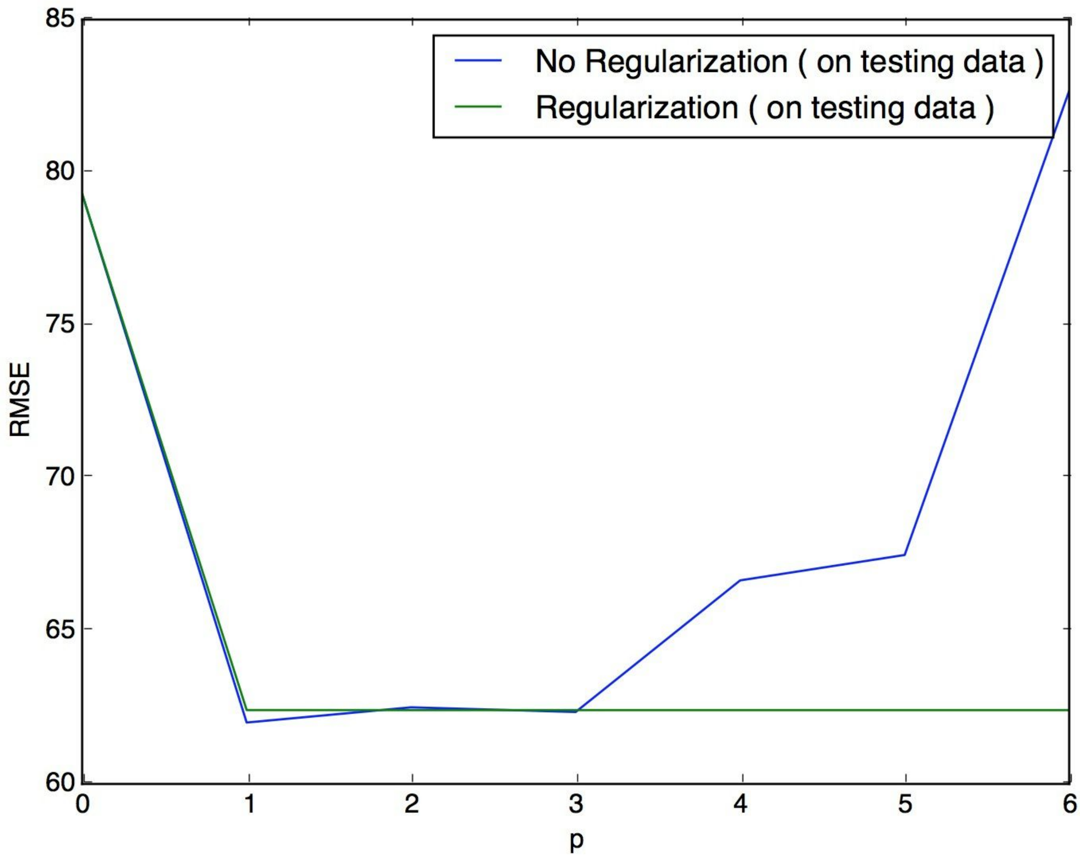
For tasks with big data sets, ridge regression using gradient descent will definitely be a much better option, because calculating the weights using matrix inversion can be expensive and takes a lot of time. Our training matrix cannot be singular if we do not use gradient descent, because the matrix inversion operation requires a non-singular matrix.

PROBLEM 5 - NON-LINEAR REGRESSION

5.1 Prediction on test data set

RMSE for prediction on testing data

p	Lambda = 0	Lambda = 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



Without regularization

RMSE decreases initially, reaching minimum at p = 1, and quickly rises with increasing p values.
This can be attributed to the fact that complex curves overfit the data, and thus does not perform well on unseen (testing) data set.

With regularization (optimal lambda = 0.06 from problem 3 & 4)

RMSE decreases with increasing p values reaching minimum at p = 4. Increasing the value of p further has no effect on RMSE, because regularization normalizes the weights. The final curve we learn will be very smooth and simple. Simple curves always perform well on unseen data.

5.3 Optimal values of p for test data set

Without regularization

The optimal value of p is 1 because the error (RMSE 62.00834404) is minimal at p = 1.

With regularization

The optimal value of p is 4 because the error (RMSE 62.41460301) is minimal at p = 4.

PROBLEM 6 - INTERPRETING RESULTS

6.1 Comparison between various approaches in terms of error

Approach	Training RMSE	Testing RMSE
OLE Regression (without intercept)	138.20074835	326.764994388
OLE Regression (with intercept)	46.7670855937	60.892037094
Ridge Regression (with optimal lambda)	49.51291236	53.3978484
Ridge Regression (with optimal lambda and gradient descent)	47.4618893086	53.397889628
Non-linear Regression (without regularization)	62.18427011	62.00834404
Non-linear Regression (with regularization)	62.85445358	62.41460301

6.2 Metric for choosing the best setting

Testing error

The testing error can be used to measure the performance of the model on unseen data.

Time taken

Time taken for prediction can also be considered as a metric to choose the best setting if our data set is very large.

6.3 Final Recommendation

By observing the testing RMSE values for different approaches, it is clear that ridge regression performs much better than other approaches. So, it is recommended to use *ridge regression for predicting diabetes level*. If we also consider time taken for prediction as a metric and our data set is very large, it is better to use *gradient descent based ridge regression (provided we do not mind the negligible loss in accuracy)* because it will be faster than normal ridge regression which uses matrix inversion. For small data sets, the performance and accuracy of both variations of ridge regression will be almost identical.