UNIVERSITY AT BUFFALO

CSE – 574 INTRODCUTION TO MACHINE LEARNING

PROGRAMMING ASSIGNMENT REPORT

CLASSIFICATION & REGRESSION

GROUP 11

GROUP MEMBERS:

SIDDHESH CHOURASIA: 50415033

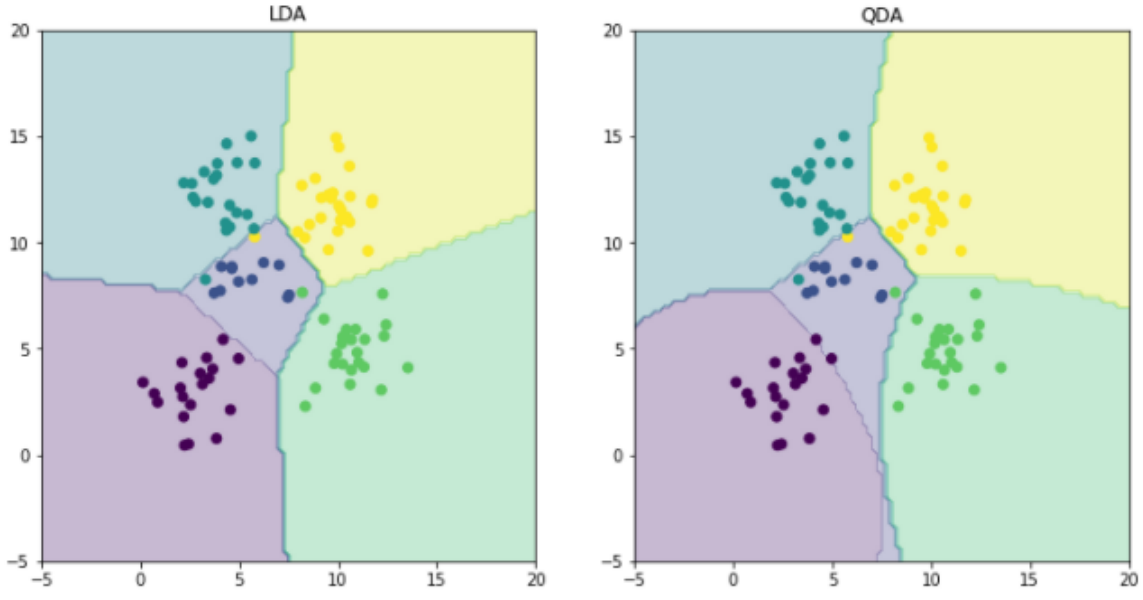
TANUSH TRIPATHI: 50411177

SARVESHWAR SINGHAL: 50418642

Report 1: Implementation of Linear & Quadratic Discriminant Analysis

Linear Discriminant Analysis (LDA) **does not** have class-specific covariance matrices, but **one shared** covariance matrix among the classes. The covariance matrix **can be different** for each class in case of Quadratic Discriminant Analysis (QDA). The classification rule is also for both LDA and QDA. But QDA allows more flexibility for the covariance matrix, due to which data is fit better than LDA. But at the same time, it has more parameters to estimate as compared to that of LDA, since we will have a separate covariance matric for every class.

Results:



	Accuracy
LDA	0.97
QDA	0.96

Observations:

The decision boundary is nearly linear for LDA and non-linear for QDA. From the plot, we can observe that there are 2 points in QDA which is on the boundary line and because of these two data points, there is difference in the accuracy rate which is a percent less in QDA than LDA

Report 2: Using least squares to calculate Mean Squared Errors

A squared loss function is used as an objective function for linear regression and the reason behind choosing this function is because of its convex shape, which makes it easier to find the optimal points where the function is minimum. It is worth to note that a squared loss function has only one global minimum and no local optima. Based on this property, the optimal weight vector can be calculated by setting the gradient/first derivative of the objective function to 0, where the squared loss function has minimum value, and it results in the following equation.

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Results:

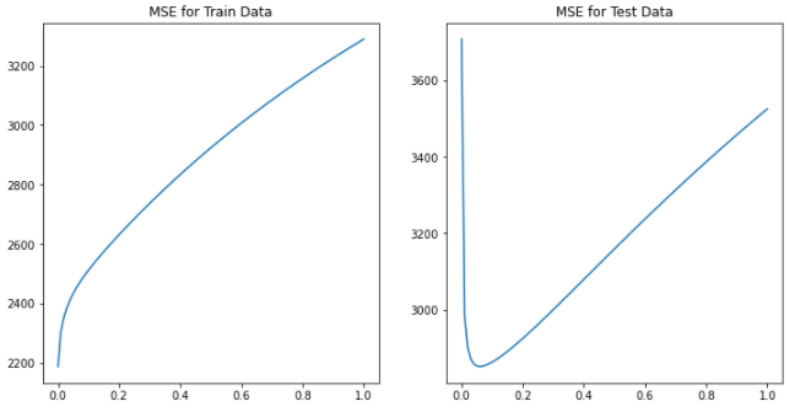
Data	Mean Squared Error without Intercept	Mean Squared Error with Intercept
Train data	19099.44684457	2187.16029493
Test data	106775.3615605	3707.84018174

Observations:

It is observed that the Mean Squared Error is much higher without intercept than with intercept and we can further infer that without intercept, the generalization error is very high. The reason for this is '**without intercept**' term, we are assuming that the prediction will be zero when the input is zero which is not always true in real life scenarios and will result in a poor learning model.

Report 3: Implementation of MSE using Ridge Regression

Results:



Error for different values of lambda

The left plot is for the training data, while the right plot is for test data. In the training data, we can observe that MSE value keeps on increasing as we increasing the value of λ. The MSE will be the lowest when the value of λ is 0.0. Likewise the value of MSE increases in test data as we increase the value of $\lambda = 0.6$, which indicates poor fit for the test dataset.

Lambda	Training MSE	Testing MSE
0.0000000000000000000e+00	2.187160294930390137e+03	3.707840181379404385e+03
1.0000000000000000021e-02	2.306832217933733318e+03	2.982446119711893971e+03
2.0000000000000000042e-02	2.354071343933828302e+03	2.900973587082240101e+03
2.9999999999999999889e-02	2.386780163097976583e+03	2.870941588884394150e+03
4.0000000000000000083e-02	2.412119043000748661e+03	2.858000409573395700e+03
5.0000000000000000278e-02	2.433174436702397998e+03	2.852665735165675869e+03
5.9999999999999999778e-02	2.451528490643497207e+03	2.851330213443847242e+03
7.0000000000000000666e-02	2.468077552526011004e+03	2.852349994057720778e+03
8.0000000000000000167e-02	2.483365646530862705e+03	2.854879739175838040e+03
8.9999999999999999667e-02	2.497740258565791009e+03	2.858444421148574747e+03
1.0000000000000000056e-01	2.511432281988938939e+03	2.862757941425694753e+03
1.1000000000000000006e-01	2.524600038524527463e+03	2.867637909167099224e+03
1.1999999999999999956e-01	2.537354899845916862e+03	2.872962282711428088e+03
1.3000000000000000044e-01	2.549776886783925875e+03	2.878645869386919003e+03
1.4000000000000000133e-01	2.561924527725497228e+03	2.884626914167790801e+03
1.4999999999999999844e-01	2.573841287742292025e+03	2.890859109690363766e+03
1.6000000000000000033e-01	2.585559874972393573e+03	2.897306658951088139e+03
1.7000000000000000122e-01	2.597105192167583027e+03	2.903941126290982538e+03
1.7999999999999999933e-01	2.608496400254901801e+03	2.910739372130537049e+03
1.9000000000000000022e-01	2.619748386225819104e+03	2.917682164132781509e+03
2.0000000000000000111e-01	2.630872823196496029e+03	2.924753221647404189e+03
2.0999999999999999922e-01	2.641878946159088628e+03	2.931938544167440796e+03
2.2000000000000000011e-01	2.652774126329711180e+03	2.939225929865840953e+03
2.3000000000000000100e-01	2.663564300769779493e+03	2.946604623783517127e+03
2.3999999999999999911e-01	2.674254296671457269e+03	2.954065056016314884e+03
2.5000000000000000000e-01	2.684848078094598350e+03	2.961598643409769466e+03
2.6000000000000000089e-01	2.695348935022923342e+03	2.969197636770348709e+03
2.7000000000000000178e-01	2.705759629119314468e+03	2.976855001187918333e+03
2.8000000000000000266e-01	2.716082506704084608e+03	2.984564320794121613e+03
2.8999999999999999800e-01	2.726319586736426118e+03	2.992319721808787563e+03
2.9999999999999999889e-01	2.736472629603949827e+03	3.000115809462218294e+03
3.0999999999999999978e-01	2.746543191088131152e+03	3.007947615588462668e+03
3.2000000000000000067e-01	2.756532664817393652e+03	3.015810554534209132e+03
3.3000000000000000155e-01	2.766442315736602723e+03	3.023700385632482721e+03
3.4000000000000000244e-01	2.776273306536249038e+03	3.031613180925095094e+03
3.5000000000000000333e-01	2.786026718543498191e+03	3.039545297133644908e+03
3.5999999999999999867e-01	2.795703568242508481e+03	3.047493351110586445e+03
3.6999999999999999956e-01	2.805304820335696604e+03	3.055454198173505119e+03
3.8000000000000000044e-01	2.814831398061086475e+03	3.063424912854024569e+03
3.9000000000000000133e-01	2.824284191328954876e+03	3.071402771689597103e+03
4.		

Error for different values of lambda

Observations:

Comparison of Mean Weights for Ridge Regression and Linear Regression: -

Also, on comparing the mean weights, it can be figured out that Ridge Regression is better than OLE as it has lesser mean weight.

Compare the two approaches in terms of error on Training and Test data: -

The MSE for OLE test data with intercept is	3707.84018174
The MSE for OLE train data with intercept is	2187.16029493
The MSE For Ridge Regression Using Intercept for Training Data	2451.5284906
The MSE For Ridge Regression Using Intercept for Testing Data	2851.33021344

Comparison of Mean Weights for Ridge Regression and Linear Regression: -

Mean OLE: - 882.8076239648467

Mean Ridge: - 17.321927262515402

The MSE for Ridge regression is significantly lower for the test data than the one for OLE thus Ridge regression is a better algorithm for the task at hand.

Also, on comparing the mean weights, it can be figured out that Ridge Regression is better than OLE as it has lesser mean weight.

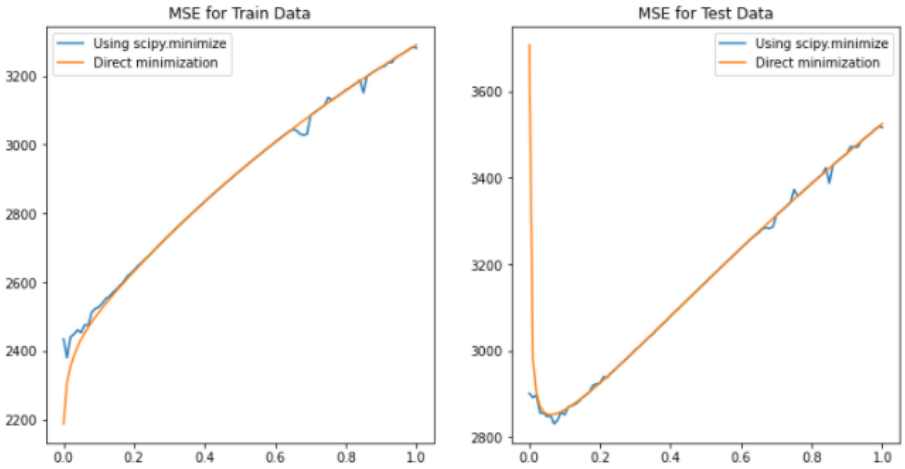
Optimal value for λ:

The value of the optimal lambda is 0.06. This value is optimal because it minimizes the testing MSE for our data set. At this value, the errors converge and start increasing again on increasing the value of lambda.

Lambda	Testing Data	Training Data
0.06	2851.3	2451.52

Report 4: Implementation of Gradient Descent for Ridge Regression

Results:



MSE for Ridge Regression and Gradient Descent

Observations:

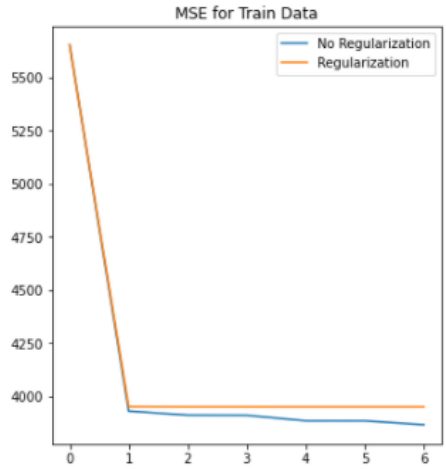
The results obtained in problem 3 are nearly like those obtained in problem 4. It should be noted that the lines produced using gradient descent are not as smooth, compared to regular Ridge regression, having some outliers in a couple of places, though they are few and minor.

One area where problem 3 excels in this data set is that it is much faster than gradient descent, as the minimize function takes time to converge. However, with bigger matrices, calculating the weights through matrix inversion can be computationally expensive, in some cases even problematic, when the matrix is singular. In such scenarios, Ridge regression through gradient descent is perhaps a better option since each step is easy to compute. With this specific data set, direct computation of the weights is faster.

Report 5: Non-Linear Regression

Results:

Below figure shows the MSE on the training set for different higher order polynomials of the input features. We can see the prediction error decreases when increasing the polynomial degree. This is because we train using the same data set, therefore higher order curves will fit better the data points and reduce the error.



MSE for Non-Linear regression on Train data

Below figure shows the MSE on the test set for different higher order polynomials of the input features. As in the previous case, we plotted the error both with and without regularization for different degree polynomials.

MSE for Non-Linear regression on Test data

When lambda is 0 MSE first decreases when P changes from 0 to 1 then starts increasing when P increases. When lambda is optimal initially MSE is very high for P = 0, but as P increases MSE narrows down to 3895 and becomes nearly constant. The optimal value of P is 1 when lambda is zero and 4 when lambda is optimal.

	Training Data	Test Data
No Regularization	3866.88344945	3845.03473017
Regularization	3950.68233514	3895.58266828

When we do not use regularization, the error reaches its minimum at p=1, means regular regression. Then it slowly increases until p=3, after which it rapidly grows and becomes even greater than p=0, the case in which we use a horizontal line. The reason is that with higher order polynomials we have overfitting: the training error decreases, but the learned curve is highly bound to the training data, and with a different data set the error steeply increases. Using regularization, the scenario is different: the error in this case decreases and reaches its minimum at p=5. However, it does not decrease much, especially when compared to the case with no regularization.

Report 6: Interpreting Results

Our goal is to obtain MSE for test data without any overfitting while learning from the train data. From our observation in problem 02 to 05, we have found that if we use Gradient Descent for optimization with regularization parameter λ, then we will achieve the best results.

Comparison between the various approaches in terms of training and testing error: -

Problem	Training MSE	Testing MSE
1 Intercept	2187.16029493	3707.84018174
2 Non-Intercept	19099.44684457	106775.3615605
3 Optimal Ridge Regression	2851.33021344	2451.5284906
4 Gradient Descent	2373.6276057	2833.13531812
5 No Regularization	3866.88344945	3845.03473017
6 Regularization	3950.68233514	3895.58266828

What metric should be used to choose the best setting?

One clear metric for selecting the best algorithm is the testing error, since it shows how accurate the algorithm behaves for classification. According to this metric, we see no reason to recommend regression without using an intercept, since that gives a higher error for both training and test set.

Linear regression with intercept gives the least training error, but when it comes to test error, Ridge regression performs better. Non-linear regression does not perform as well as Ridge regression.

Ridge Regression is the best approach in this scenario (for small datasets).

It is worth to notice that in some cases, like in problem 2 and problem 5, a decreasing training error may lead to conclude that is the best approach. Unfortunately, in many cases, that only means overfitting (for example, question 5), and the dramatic increase of the test error confirms it.

Having a high-test error is unfavorable in any circumstance, and therefore, training error alone is not a good metric. A further metric that is particularly significant in scenarios with limited resources is execution time.

In this case, and in general with relatively small data sets, it does not matter much, but when handling large data sets, execution time(runtime) can be an important issue.

In those scenarios, Ridge regression through matrix inversion may be unfeasible, whereas gradient descent could provide results faster at an acceptable loss of accuracy.