Problem 1:

Observations:

1. For LDA, the boundaries are linear that separate all the classes from each other ie boundaries form straight lines.

2. In case of QDA, the boundaries separating the classes are non-linear.

3. Graphs shows that we can observe that LDA has lines but QDA has curves which is better choice in case of real world data. Because with random data there is very less probability that we can separate data into different classes using straight lines.

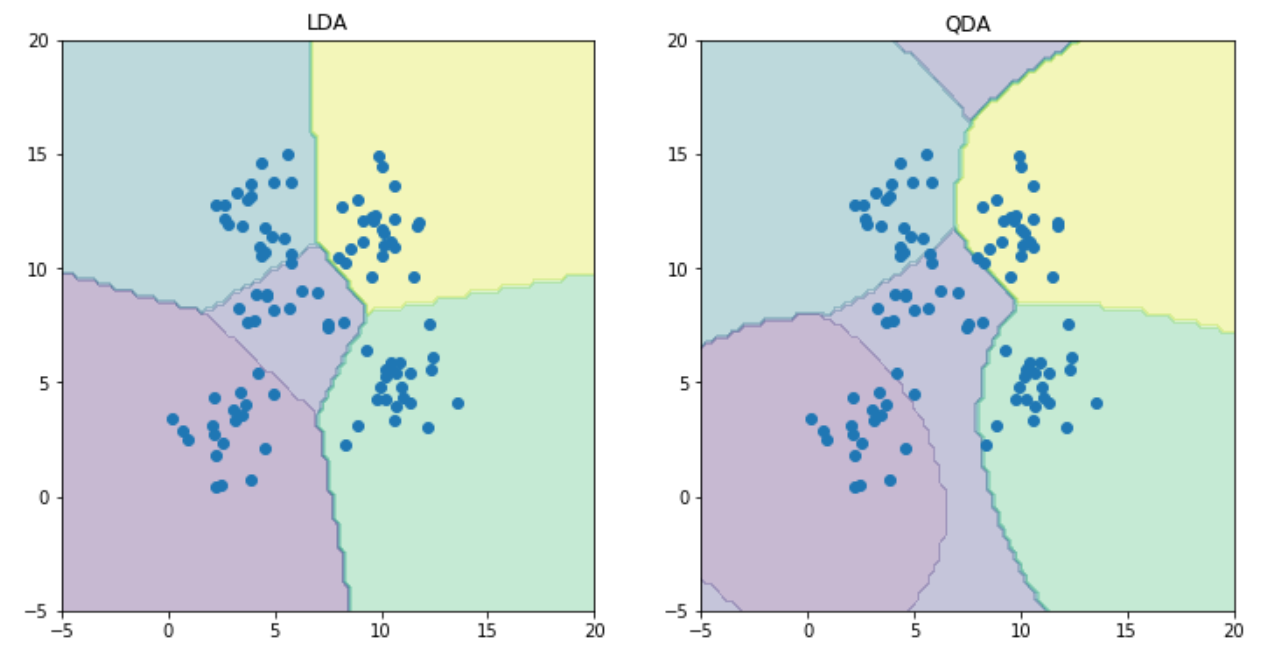
4. However, in case of data provided for the problem the accuracies of LDA and QDA shows that, LDA is better than QDA since we are obtaining a more reliable estimate of variance between the training data. It is because here the covariance calculation takes all the training data into consideration and not just fraction of training data belongs to individual class that’s why giving us a better accuracy.

5. In case of LDA no quadratic term is involved and if the test data is not large enough, then this gives a better variance estimation than QDA and thus lead to higher accuracy on small data. Conversely if the test data is big, then estimating variance would not be much of a concern and hence QDA will have better accuracy in that case.

6. Just because calculations are linear in case of LDA, it has linear boundaries and QDA has quadratic terms, hence its boundaries are quadratic.

LDA Accuracy = 0.97

QDA Accuracy = 0.94



Problem 2:

Observations:

MSE - Mean Squared Error.

Since this is an error estimate, higher the value, higher will be the error and lower will be the accuracy of the linear regression. Conversely, lower the value, lower will be the error and it would mean that the linear regression model fits the data appropriately.

2. Hence looking at the value of MSE with intercept and MSE without intercept, we can say that with intercept it is better. Since this gives an improvised regression line that fits the data very well as opposed to the regression, line that always passes through the origin.

3. This bias that is introduced in the form of intercept so that the line that fits the data points not necessarily have to go through the origin but can intercept the y axis at some point.

Results:

MSE without intercept 106775.361553

MSE with intercept 3707.8401812

Problem 3:

In this problem we need to calculate and report the MSE for training and test data using ridge regression parameters. The training and test data used in this case consists of an intercept which means that the regression line won’t have to necessarily pass through the origin thus generating better accuracy. Errors on test and train data had to be plot for different values of lambda (λ) ranging from 0 to 1 in increments of 0.01.

Let us compare both approaches in terms of error on test and train data:

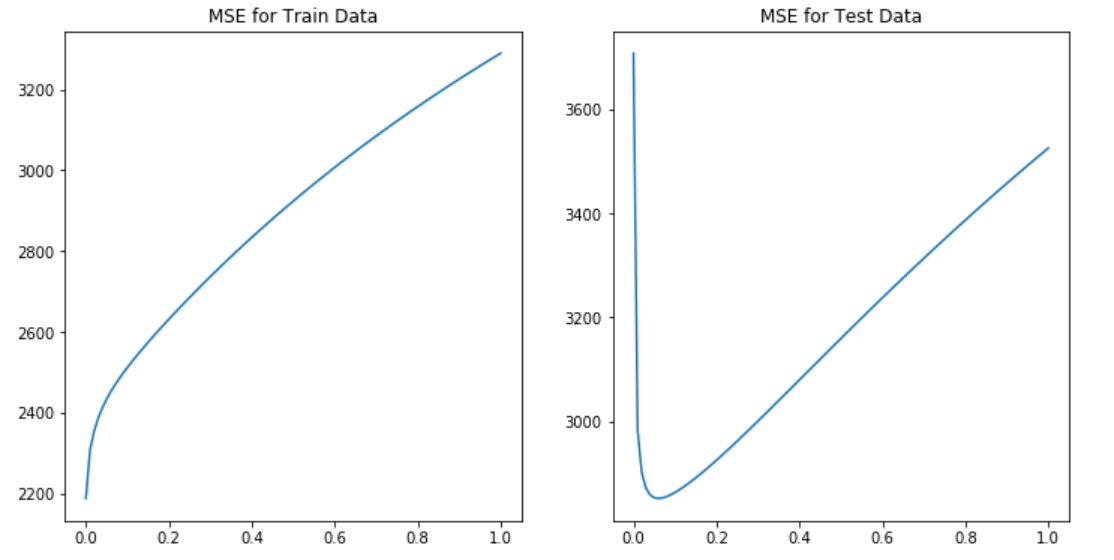
From the observations in the graph it can be clearly seen that for training data the MSE computed using learnOLERegression is the same as MSE computed using learnRidgeRegression for Lambda (λ) = 0, which is pretty obvious since lambda=0 indicates no regularization. However as we increase the lambda the error kept on increasing. I can be observed in the plot for train data.

In case of test data too, the MSE for lambda (λ) =0 when computed using learnRidgeRegression is the same as the MSE which is computed using learnOLERegression.

However, as we increase the lambda the error decreases up to a certain value of lambda, which in our case is 0.06.

After lambda = 0.06 the error gradually increases as we increase the lambda. This has also been shown in the plot for test data.

Thus, the optimal value chosen for lambda (λ) is 0.06 as it gives the least MSE.



Problem 4:

Observations:

The blue curve is the one that is obtained by using scipy.minimize function and the green curve is the one that is obtained using direct minimization.

We can clearly see that both the curves are fairly close to each other but there are some outliers in the curve obtained by using scipy.minimize function.

These outliers are eliminated from the curve of direct minimization as the curve is made smooth by increasing the number of iterations.

For training data, the error increases as the value of lambda increases.

The MSE values for increasing value of lambda using scipy.minimize are shown below:

the given graph we can observe that even though there is some inconsistency in the curve for scipy.minimize the MSE gradually increases as we increase the lambda.

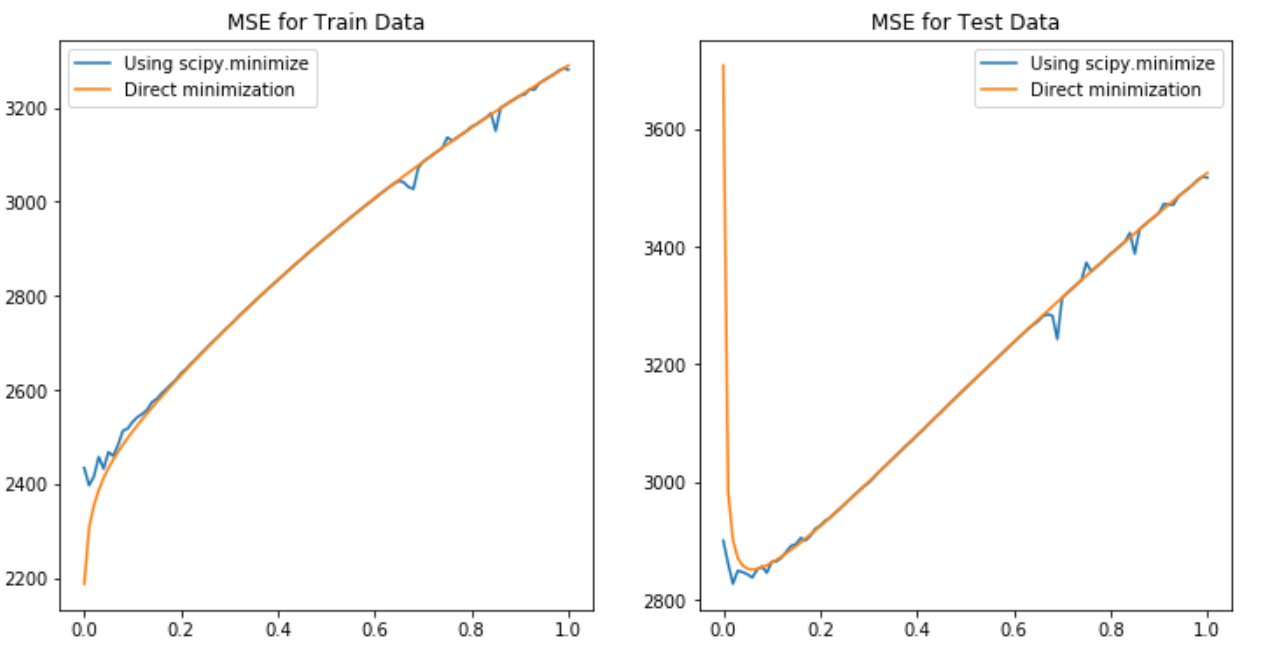
If we consider test data however, the error decreases up to a certain value of lambda and then it increases with lambda. In test data too there is some inconsistency in the smoothness of the curve for MSE when plotted using scipy.minimize compared to direct minimization. MSE values for test data using scipy.minimize have been shown below.

After this point the error gradually increases as we increase the lambda. This has also been shown in the plot for test data.

Comparison with problem 3:

By comparing the curves obtained in Problem 3 and Problem 4, we can clearly see that the curves are fairly the same and we can say that the result obtained by approaches used in Problem 3 and 4 are same.

The only difference is that the curve plotted in Problem 3 is much smoother than the curve which is plotted in Problem 4.



Problem 5:

For this problem we’re using only the third variable as the input variable to study impact of using higher order polynomials for the input features. Weights are computed using two lambda values 0 and optimal value. Optimal lambda value in our case is 0.06 which has been computed in Problem 3.

Regression line has been plotted for various values of p ranging from 0 to 6. For each value of p two lines are plotted corresponding to regression weights generated by the two different lambda values. In this case, p is a number which converts a single attribute x into a vector of p attributes.

Thus as we increase the value of p we get a regression line that tries to pass through each data point. Increasing the value of p beyond a certain limit may lead to an overfitting problem which we may need to avoid.

In the plot for train data the blue line indicates MSE for various values of p ranging from 0 to 6 when regression weights have been calculated using lambda = 0.

The green line indicates the MSE for various values of p when regression weights have been calculated using the optimal lambda value which in our case is 0.06.

This has been done as we want to minimize our error but also want to penalize the complexity of our model. Hence we use regularization. Regularization forces w terms to be closer to 0 thus minimizing the weight vector.

Hence the regression line becomes nonlinear but not too much. Due to this,

the MSE without regularization would decrease as we increase p up to a certain point. This is because the regression line is trying to fit through each data point.

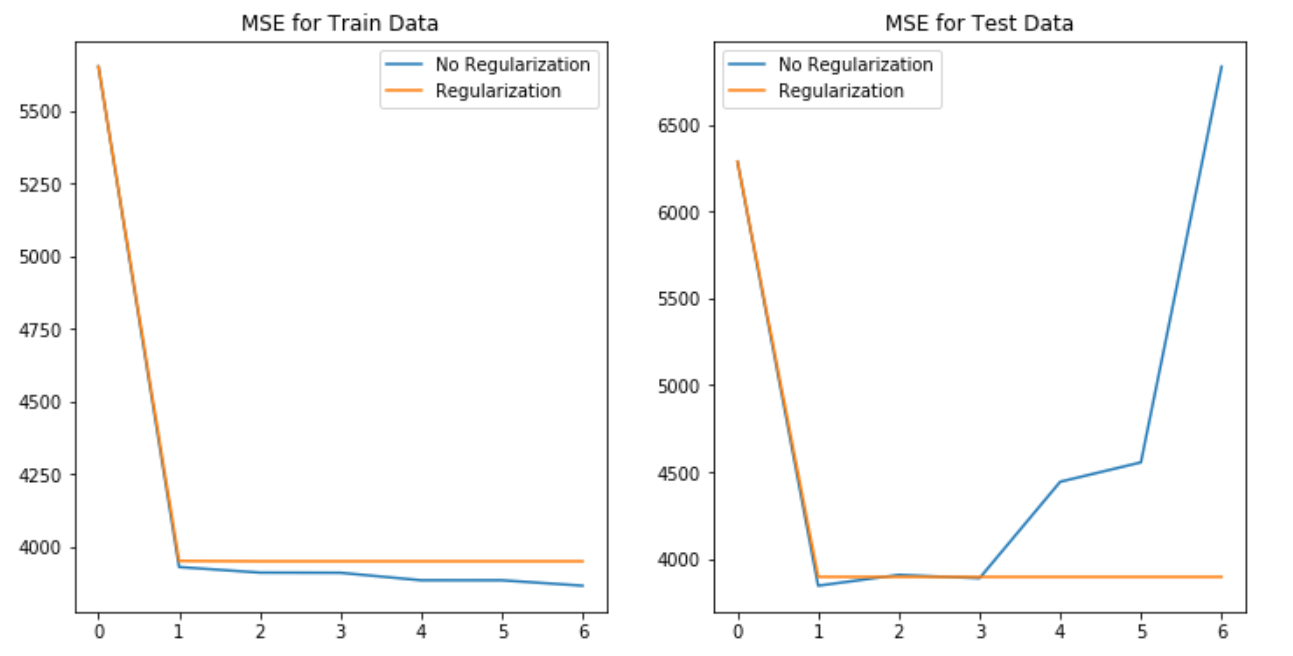
Applying regularization however tries to reduce this non linearity hence we get MSE which is greater than that without regularization which can be seen from the green line.

The effect of this regularization can be seen on the test data where the MSE keeps on increasing as we increase p however if the weight vector has been regularized using optimal value of lambda, the MSE is almost constant and has a low value.

The optimal value of p can be calculated for test error using both regularization and no regularization from the above plot.

If we consider the plots for no regularization, we take a look at the blue line. We can clearly see from the plot that for p = 1 we have the least MSE thus the optimal value of p is 1 for no regularization.

In case of regularization as we increase the p, the MSE slowly but surely decreases in the plot hence the optimal value of p in case of regularization can be stated as p = 6. This is because at p = 6 we have the least MSE.



Problem 6:

Final Observations:

For Train Data:

1) Using Linear Regression: MSE = 2187.16029493

2) Using Ridge Regression: MSE = 2187.16, with lambda = 0

3) Using Gradient Descent: MSE = 2433.66, with lambda = 0

4) Using Non Linear Regression:

MSE = 3895.58, with regularization

MSE = 3845.03, without regularization

For Test Data:

1) Using Linear Regression: MSE = 3707.84

2) Using Ridge Regression: MSE = 2851.33, with lambda = 0.06

3) Using Gradient Descent: MSE = 2826.92, with lambda = 0.02

4) Using Non Linear Regression:

MSE = 3950.68, with regularization

MSE = 3845.03, without regularization

From the observations given above, we can clearly see that the value of MSE is more when we use Non Linear Regression or Simple Linear Regression, however, the value of MSE is comparatively less when we use Ridge Regression and Gradient Descent.

The value of MSE is least when we use Ridge Regression and it takes lesser computations as compared to gradient descent, hence Ridge Regression would the best approach amongst all the other linear models.