Introduction to Machine Learning

Programming Assignment 1

Ankit Mukherjee, 50611335.
Praveen Kumar Gangapuram, 50565977
Rama Rao Vydadi, 50604256

Report 1

Introduction

In this report we trained the sample_train data to train both Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) then we tested it on the sample_test dataset and compared the predicted value with the actual value and calculated its accuracy and plotted the discriminating boundary for linear and quadratic discriminators.

Accuracy of LDA and QDA

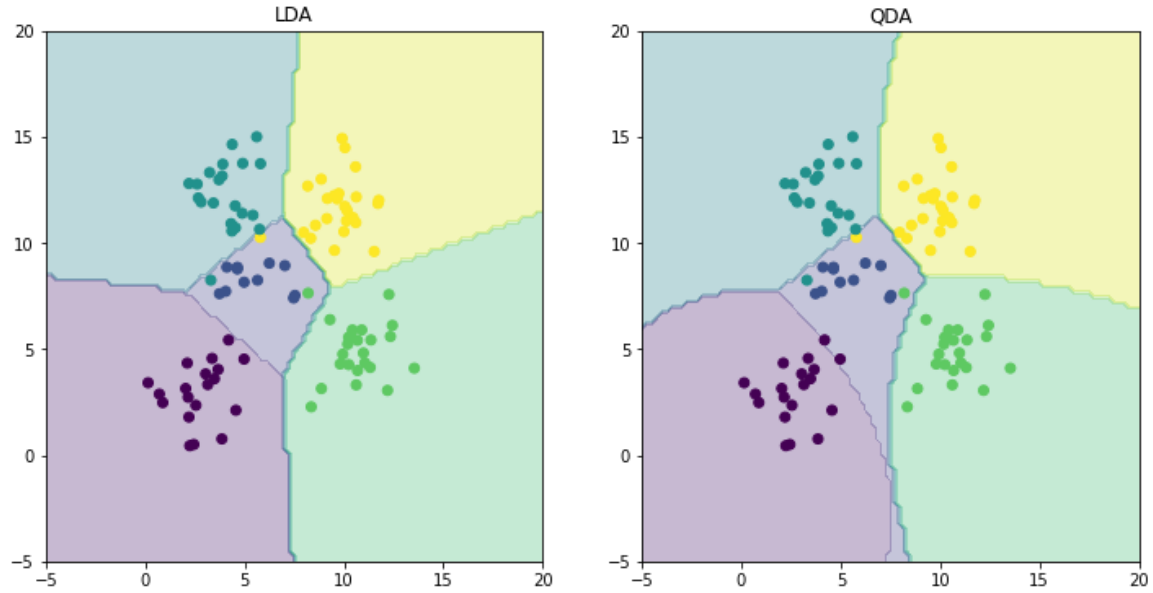
We calculated the accuracy of both the models by comparing the predicted value with the actual labels by flattening the 2d array to 1d and the accuracy we got is as below:

- **LDA Accuracy:** 97.0%
- **QDA Accuracy:** 94.0%

Both the methods perform well but as we can see LDA performs better than QDA for this particular sample_test dataset. This might be due to the linear structure of the dataset.

Decision Boundaries:

We use the code from the main to plot the two discriminating boundaries of LDA and QDA as plotted in the figure below:



The **left plot** shows the discriminating boundaries for LDA. The boundaries are **linear**

The **right plot** shows the discriminating boundaries for QDA. The boundaries are **curved**.

Why is there a difference in boundaries?

In **LDA** all the classes have the same covariance matrix. As a result, there are **linear boundaries** between classes, as the shapes of the data distributions are considered the same (just centered differently).

In QDA all the classes have their own separate covariance matrix which results in quadratic boundaries. This allows QDA to fit in case of more complex datasets. It can result in overfitting if the training dataset is small or contains many outliers.

Report 2:

Mean squared error

- MSE Value **with Intercept** is 3707.840181260657
- MSE Value **without Intercept** is 106775.36150295778

From the above Mean Squared Error Values. The value with intercept is lower than the value without Intercept. This indicates that including intercept in your regression improved the fit of the model.

Which one is better?

With Intercept:
This leads to a better fit, especially the data does not pass through the origin.

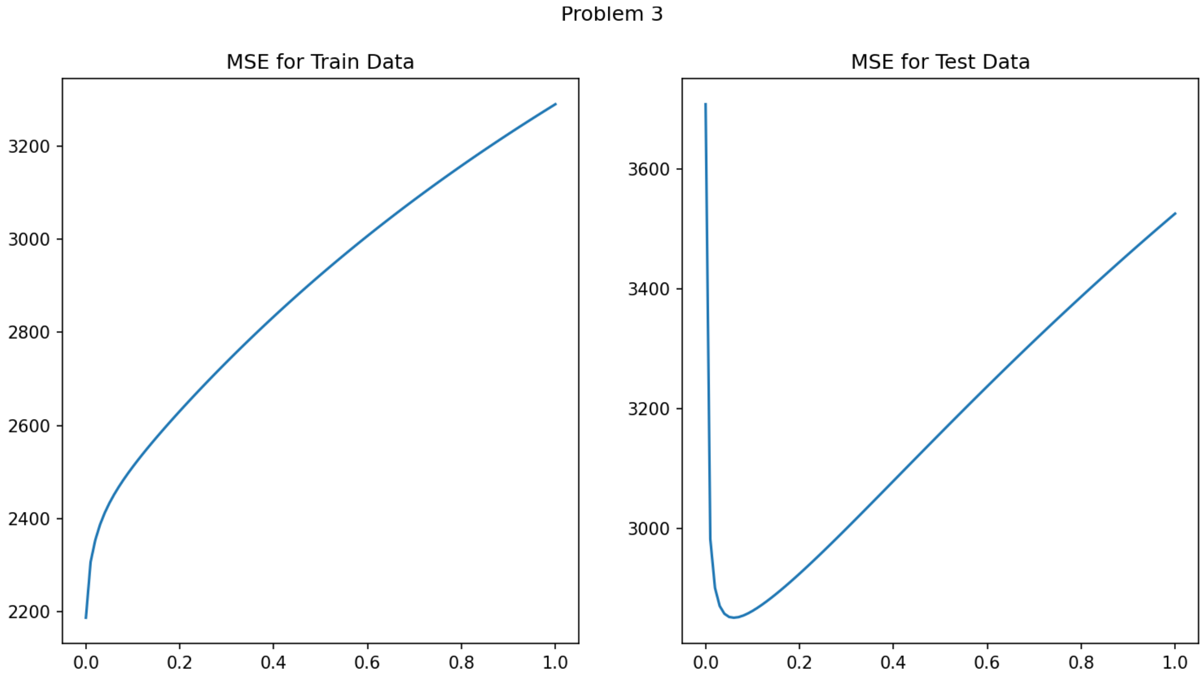
Without Intercept:
The model forces the line to pass through the origin, which might lead to a worse fit if the actual data has a non-zero mean for the dependent variable (y).

Conclusion:
The one with intercept performs better than the value without Intercept.

Report 3:

Plotting the errors

We have calculated the MSE for train and test data using the ridge regression parameters. Please find the Plot of the data for different values of λ from 0 to 1 in steps of 0.01 below



Comparison of relative magnitude of weights learnt using OLE and ridge regression

L2 Norm of OLE Weights: 124,531.53

L2 Norm of Ridge Regression Weights: 1,926.76

The magnitude of weights learnt using OLE is significantly larger than that of the Ridge Regression. The model learned weights with high magnitudes in OLE. This is a typical scenario where the data is prone to overfitting.

The regularization term in ridge regression prevents the weights from becoming too large which helps to prevent overfitting.

Comparison of errors on train and test data

MSE for train data: For the training data the MSE gradually increases with increasing lambda, as the higher values of regularization (lambda) penalizes large weights more, potentially underfitting the data, resulting in higher training error.

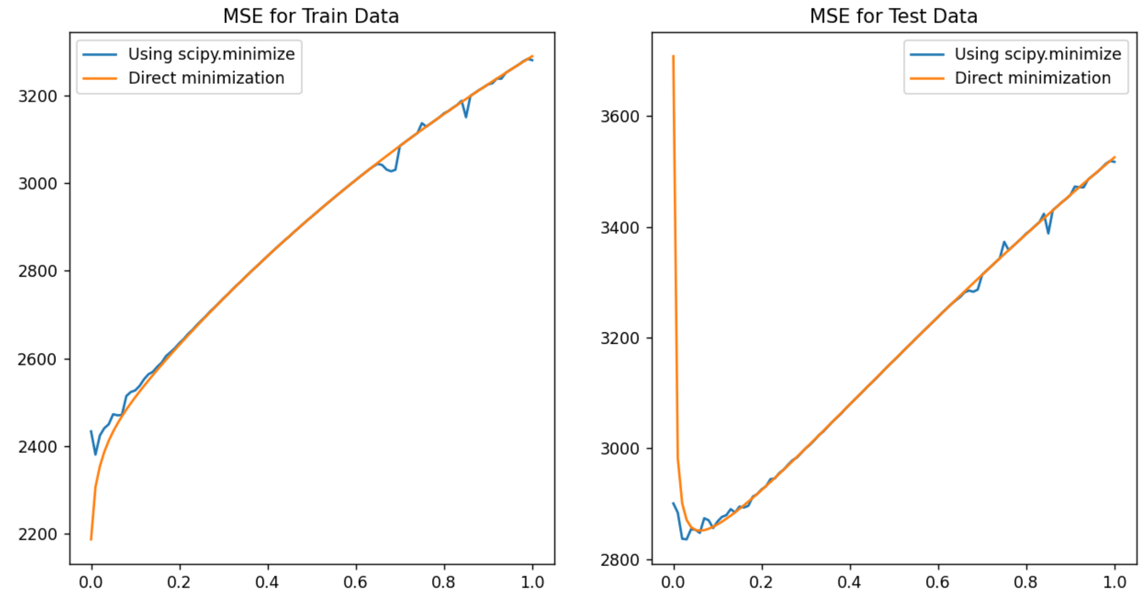
MSE for test data: The MSE for test data is U-shaped curve, which indicates that there is an optimal value of lambda which balances between underfitting and overfitting.

What is the optimal value of lambda?

From the graph we can observe that the **optimal lambda value is 0.06** as the MSE is minimum at this point there is a **balance between overfitting and underfitting**.

Report 4

Plotting the errors



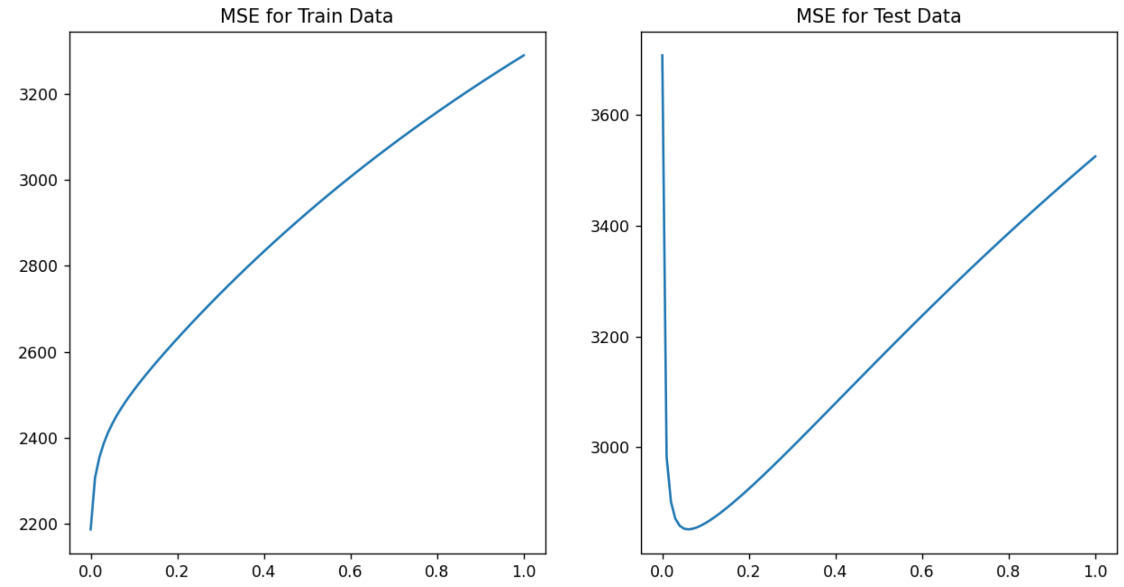
As we can see from the graph above.

Train Data:
The orange line (Direct minimization) is smooth, while the blue line (scipy.minimize) has some fluctuations.

Test Data:
Scipy.minimize shows some minor fluctuations, but the overall trend aligns with the direct minimization results.
The MSE initially drops and then rises, indicating a transition from underfitting to overfitting.

Comparing the results with problem 3

In Problem3:



Train Data:

The MSE on the training set steadily increases as some parameter (possibly complexity or regularization) increases.

The curve is smooth, indicating a clear increase in error as the parameter grows, which suggests a gradual shift towards underfitting.

Test Data:

Shows a classic pattern of overfitting: good performance at some intermediate parameter values but worsening performance when the model becomes too complex or too simple.

Conclusion:

Problem 3 seems to focus on a single method (likely direct minimization) and gives smoother results for both train and test data.

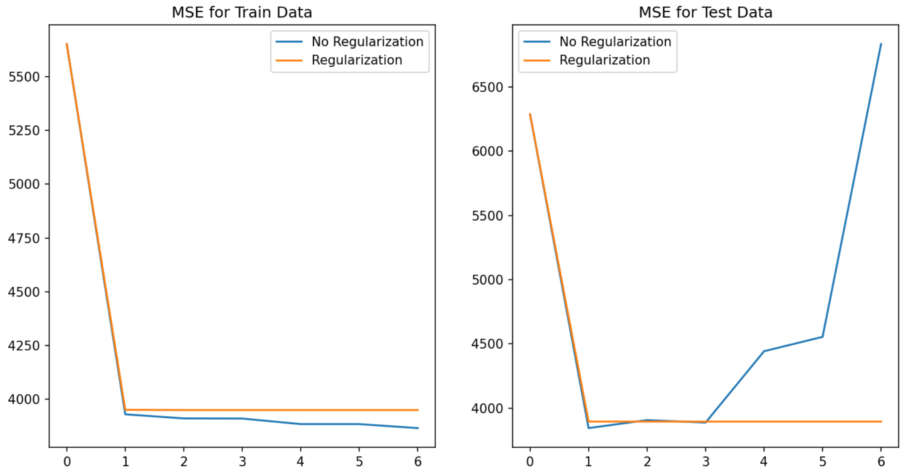
Problem 4 adds a comparison between two methods: scipy.minimize and direct minimization. While scipy.minimize introduces small fluctuations in the results, the overall trends for both training and test data remain consistent with Problem 3.

Report 5

Plotting the errors

Please find the plots of MSE for both training and test data across different polynomial degrees for below scenarios –

- No regularization (λ=0)
- Regularization with optimal λ=0.06



MSE values for p values from 0 to 6

Train data

The test MSE decreases from p=0 to 1, reaching to its lowest value at p=1 which indicates linear model, beyond this error increases significantly especially after p=4 which indicates overfitting.

Test data

The test MSE behaves in the same fashion till p=1, but the error value remains low even after increasing p values which indicates that regularization controls overfitting.

Optimal p value:

- For both cases with and without regularization the optimal p value is 1, as the MSE is the lowest at p=1.
- Without regularization, the model starts overfitting as the polynomial degree increases. The test error increases significantly beyond p=1
- With regularization the test error remains low even at higher degrees, indicating that regularization controls overfitting.

Conclusion:

For the given data, using a regularized model with p=1 and λ=0.06 is the best approach as it prevents overfitting and achieves lowest error.

Report 6

Comparison of the previous 4 regression methods

Based on the findings from the previous reports, here is the comparison of all the regression methods.

Linear Regression (Report 2):

Implementing Ordinary Least Squares Linear regression, The MSE on the test data was high which meant overfitting to the training data

Ridge Regression (Report 3):

Implemented Ridge regression where the regularization parameter λ of 0.06 gave us the best result. Here the MSE on the training data is higher than linear regression but on the test data it is much better

Ridge Regression using gradient descent (Report 4):

It gave us MSE similar to Ridge regression without gradient descent.

Non-linear Regression (Report 5):

By Introducing polynomials, we can handle non linear models and having higher degree polynomials gives us better fitting to the training data but High MSE for Testing data suggesting overfitting.

What metric should be used to choose the best setting?

- In this entire report we have used Mean Squared Error as the metric to evaluate model performance. Given the data, MSE is good with capturing how our model performs by comparing our predicted value to the actual value.Therefore MSE should be used to choose the best setting.
- LDA should used if the dataset is linear, as it achieves high accuracy without overfitting.
- Ridge regression with λ=0.06, should be used in complex datasets which are prone to overfitting.
- OLE without regularization should be avoided as it leads to severe overfitting

In summary, **Ridge Regression** with **λ=0.06** and **polynomial degree p=1** provides the best results for predicting diabetes levels, as it prevents overfitting and maintains low error on both training and testing data.