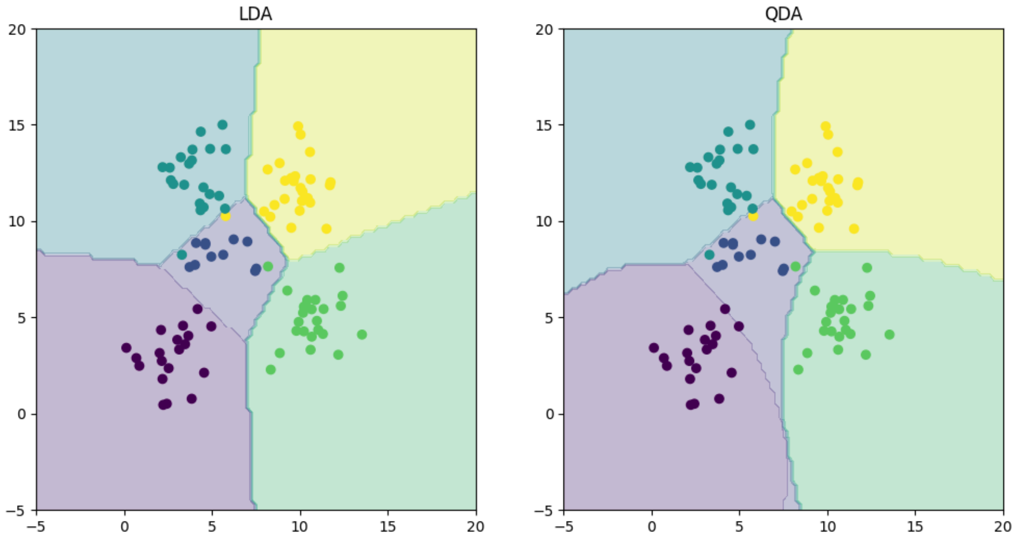
CSE574 Introduction to Machine Learning
Programming Assignment 3
Classification and Regression

Nandhakumar Vadivel (50604851)
Sai Kalyan Veera (50560096)

Report -1

LDA Accuracy = 97.0
QDA Accuracy = 96.0



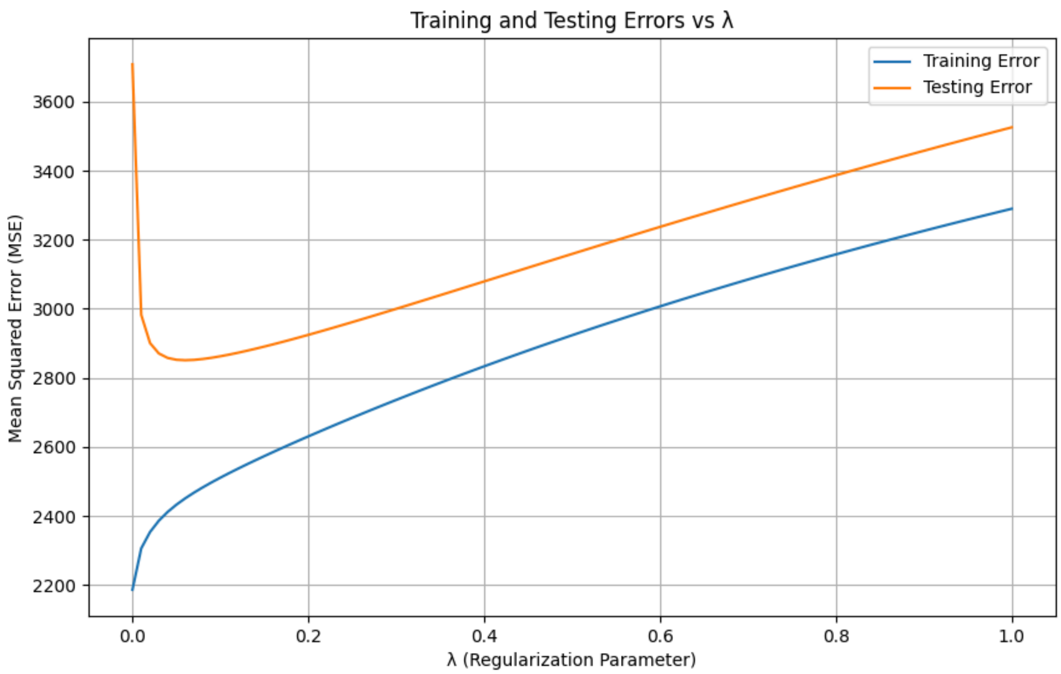
- The difference in the boundaries between LDA and QDA happens because of how they handle the class covariances.
- LDA assumes all classes have the same covariance, so the boundaries are straight lines (linear).
- QDA allows each class to have its own covariance, which makes the boundaries curved (quadratic).
- This is why QDA can better handle data with non-linear separations, while LDA works best when the class covariances are similar.

Report -2

MSE without intercept 106775.3615151266
MSE with intercept 3707.8401819303413

Performance is better with using an intercept .

Report -3



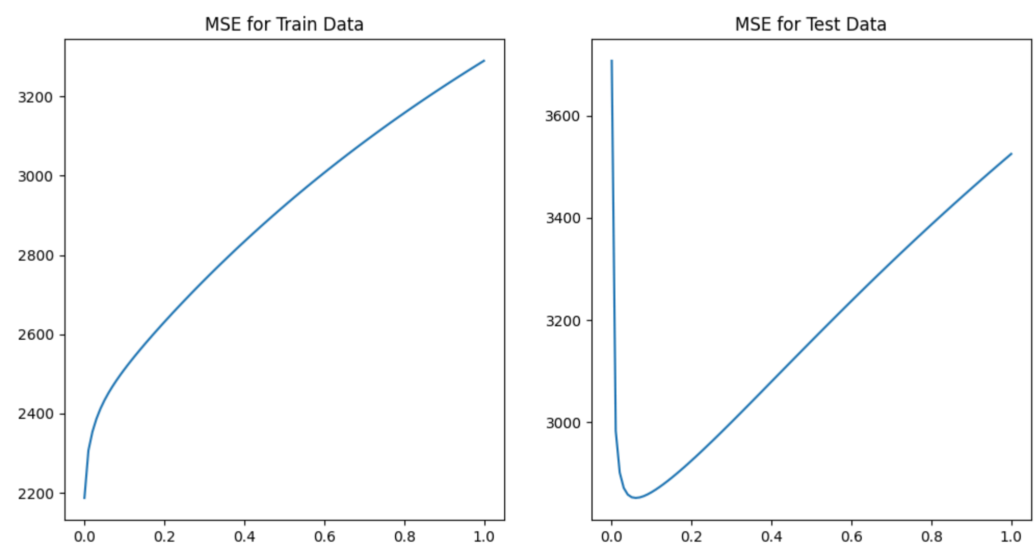
Optimal λ: 0.06
Minimum Test Error: 2851.3302

Weights (OLE - λ = 0):
[1.48154876e+02 1.27485208e+00 -2.93383522e+02 4.14725449e+02
  2.72089134e+02 -8.66394569e+04 7.59144678e+04 3.23416227e+04
  2.21101215e+02 2.92995511e+04 1.25230360e+02 9.44110833e+01
 -9.38628632e+01 -3.37282800e+01 3.35319773e+03 -6.21096289e+02
  7.91736536e+02 1.76776039e+03 4.19167405e+03 1.19438121e+02
  7.66103401e+01 -1.52001293e+01 8.22424594e+01 -1.45666208e+03
  8.27386702e+02 8.69290952e+02 5.86234495e+02 4.27026727e+02
  9.02467690e+01 -1.78876224e+01 1.41696774e+02 5.82819385e+02
 -2.34037511e+02 -2.56071452e+02 -3.85177401e+02 -3.34176738e+01
 -1.07350066e+01 2.57107189e+02 5.99554597e+01 3.83728042e+02
 -4.04158390e+02 -5.14286434e+02 3.83636640e+01 -4.46102889e+01
 -7.29643533e+02 3.77408338e+02 4.39794291e+02 3.08514373e+02
  1.89859679e+02 -1.09773797e+02 -1.91965701e+03 -1.92463378e+03
 -3.48979528e+03 1.17969687e+04 5.30674415e+02 5.43305911e+02

```
 1.82107518e+03 -1.04639807e+04 -5.16627611e+02  2.06435917e+03
-4.19941334e+03 -1.40495705e+02  3.74157090e+02  5.14757492e+01
-4.64492730e+01]
```

Weights (Ridge Regression - Optimal λ):

```
[ 150.45959807    4.80776899 -202.90611468  421.7194576   279.45107288
  -52.29708233 -128.59418907 -167.50057028  145.74068096  496.30604123
  129.94845775   88.30438076   11.29067689    1.88532531   -2.58364157
  -66.89445481  -20.61939955  113.39301454   17.99086827   52.50235963
  109.68765513  -10.72779629   71.67974829  -69.30906366 -124.03437293
  102.63981795   72.64220588   79.24754013   38.48319215   32.98009446
   92.09539122   68.97936154  -24.41700914  101.85387967    1.39122669
   20.85757155  -29.65490134  130.41115986  -16.75108796   87.51340344
  -45.64238362  -30.92288499  -10.07139781   31.13334896  -89.33525423
  -22.73053674   65.41116624   55.11621318   19.14925041  -59.84315841
   26.64350735  108.40501275 -137.61756968  -83.04383566  -20.40214777
   24.9726362    -0.92451093  191.91306579   34.78309393  -43.90393505
   23.2002376    20.8504118  -117.853228     75.30611309   60.36839226]
```



Comparison of Weight Magnitudes (OLE vs Ridge Regression):

OLE Weights: The weights learned using OLE (λ=0) are significantly larger in magnitude, indicating that OLE does not penalize large weights, which can lead to overfitting. For example:

- Feature 5: −86639.46
- Feature 6: 75914.47

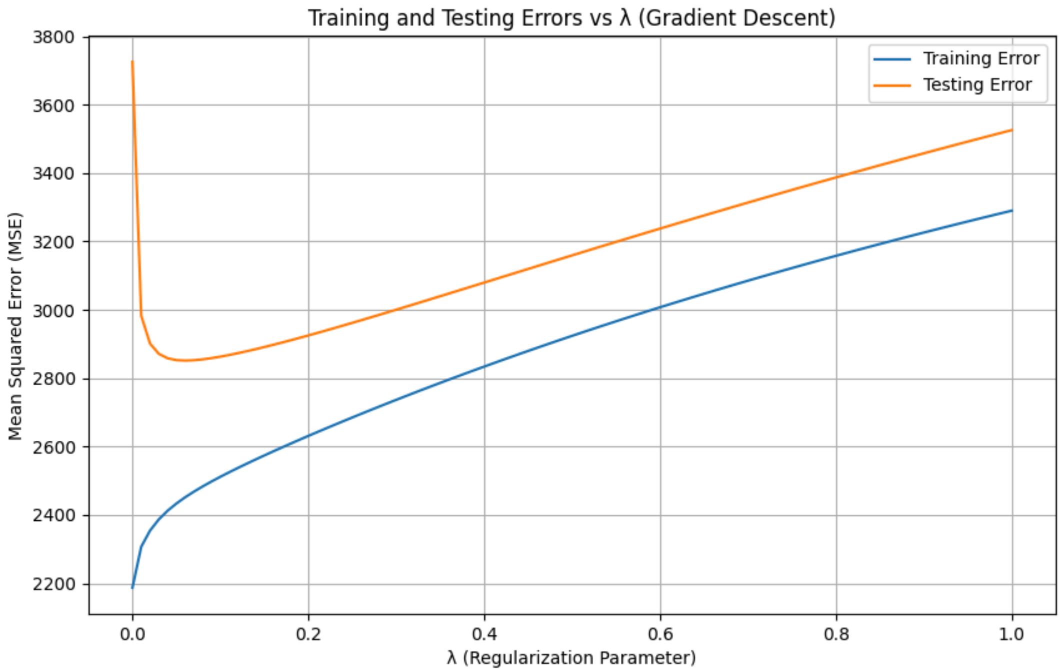
Ridge Regression Weights: The weights learned using Ridge Regression with λ=0.06 (optimal value) are much smaller in magnitude, as regularization penalizes large weights, making the model more stable. For example:

  - Feature 5: −52.30
  - Feature 6: −128.59

This demonstrates that Ridge Regression reduces the influence of less important features and stabilizes the weights.

1. **Comparison in Terms of Errors (Train vs Test):**
   - **OLE:**
     - OLE achieves low training error because it is unregularized and can overfit the training data.
     - However, it performs worse on test data due to overfitting, resulting in higher test error.
   - **Ridge Regression:**
     - Ridge Regression slightly increases the training error as λ\\lambdaλ increases because it introduces regularization.
     - The testing error, however, decreases initially and achieves the minimum at λ=0.06 indicating a better generalization on unseen data.
2. **Optimal Value of λ:**
   - The optimal λ is **0.06**, as it provides the lowest test error of **2851.33**.
   - This value strikes the right balance between bias and variance, minimizing overfitting while maintaining a good fit to the data.

Report-4

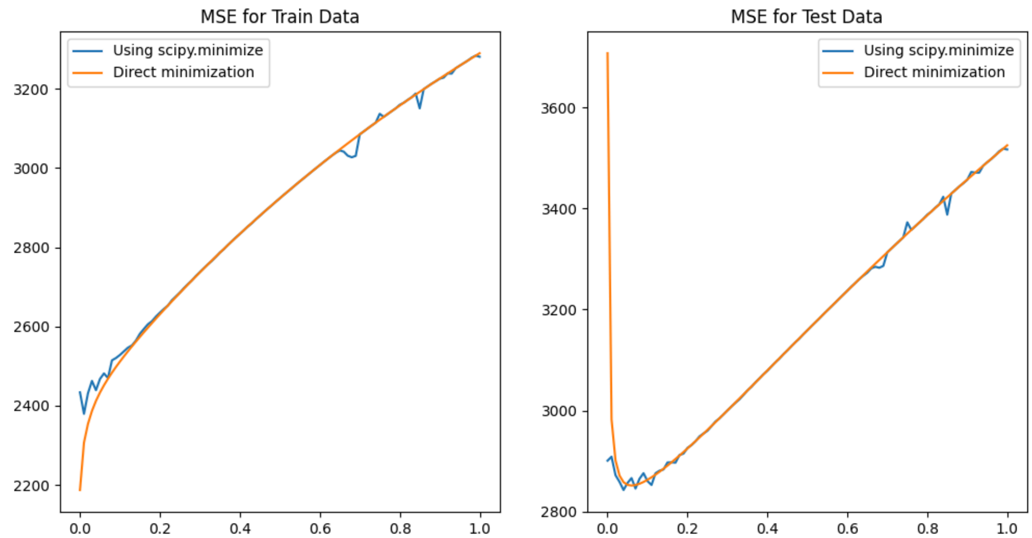


Training Error: The training error increased as λ increased. This happens because higher regularization reduces the flexibility of the model, causing it to underfit the training data.

Testing Error: The testing error initially decreased as λ increased, reaching a minimum at λ=0.06, and then started increasing as λ got too large. This shows that small values of λ\\lambdaλ help reduce overfitting, but too much regularization can harm performance.

Optimal λ: 0.06

Minimum Test Error: 2851.3302



Training Data (Left Plot):

- The training error increases steadily with the regularization parameter (λ)
- Both methods show almost identical behavior, confirming that `scipy.minimize` closely approximates the direct solution for the training set.

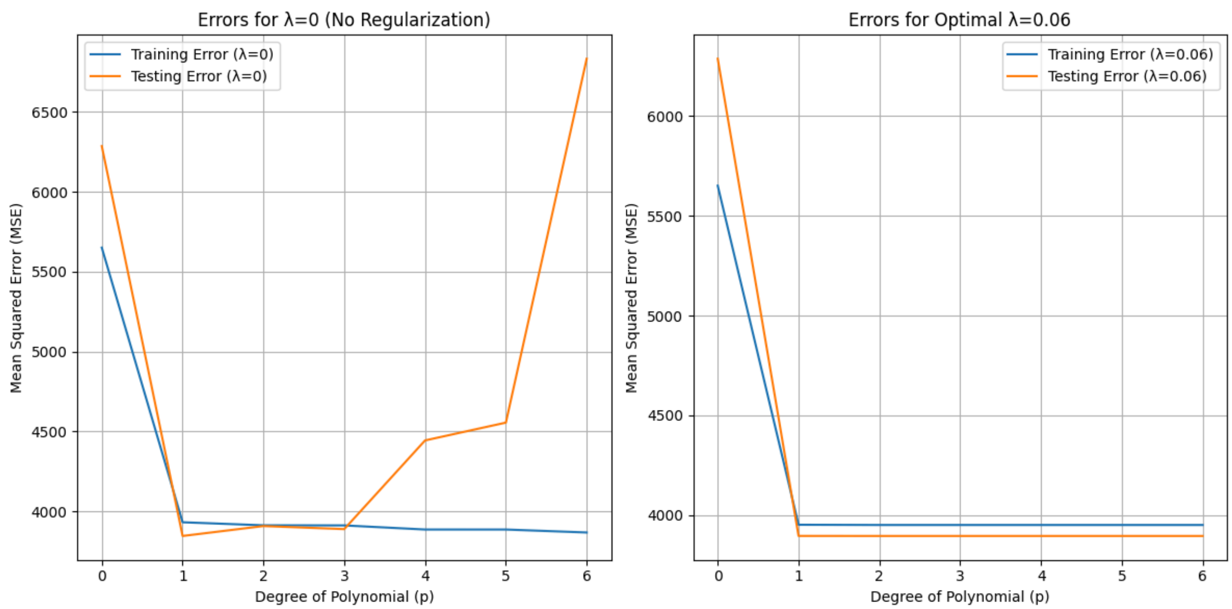
Testing Data (Right Plot):

- The testing error initially decreases as λ increases, reaches a minimum, and then starts to rise.
- The optimal λvalue (where the test error is minimized) is the same for both methods, indicating consistency in identifying the best regularization strength.
- `scipy.minimize` introduces slight fluctuations in the testing error due to numerical optimization approximations, but the general trend remains consistent with the closed-form solution.

Both methods successfully apply regularization to balance bias and variance. The closed-form solution is faster and more accurate for smaller datasets because it directly calculates the optimal weights.

On the other hand, gradient descent is more practical for larger datasets where matrix inversion is computationally expensive. The results from both methods are consistent, as they identify the same optimal lambda (0.06) and achieve a similar minimum testing error (2851.33). This shows that Ridge Regression is effective for improving model generalization, regardless of the method used.

Report-5



Train and Test Errors for λ=0:

- As p increases, the training error consistently decreases due to the model's ability to fit the training data more closely. This indicates overfitting for higher polynomial degrees.
- However, the test error initially decreases for p=1, achieving the optimal value, and then increases significantly for higher polynomial degrees due to overfitting.

Train and Test Errors for λ=0.06:

- With regularization, the training error remains slightly higher compared to λ=0, as expected. This indicates that regularization limits the model's ability to overfit.
- The test error reaches its minimum at p=4, demonstrating that a balance between model complexity and regularization reduces overfitting while improving generalization.

For **λ=0**

- **Optimal p:** p=1 (linear ridge regression).

- **Reason:** At p=1, the test error was minimized. Increasing p caused overfitting, leading to higher test errors.

For **λ=0.06:**

- **Optimal p:** p=4
- **Reason:** Regularization allowed the model to achieve better generalization at a higher polynomial degree, balancing training and test performance.

Optimal p for λ=0: 1

Optimal p for λ=0.06: 4

Report-6

Problem 1: Linear and Quadratic Discriminant Analysis (LDA and QDA)

- Objective: To classify data using LDA and QDA and plot the discriminant boundaries.
- Results:
  - LDA Accuracy: 97.0%
  - QDA Accuracy: 96.0%
- Key Observations:
  - LDA assumes shared covariance among classes, leading to linear boundaries.
  - QDA uses class-specific covariance, producing quadratic boundaries, which can better capture non-linear separations but risks overfitting with small datasets.
- Best Use Case: LDA for simpler datasets and QDA for datasets with non-linear separations.

Problem 2: Ordinary Least Squares (OLS) Regression

- **Objective:** To compute regression weights using OLS and evaluate mean squared errors (MSE) for training and testing data.
- **Results:**
  - **MSE Without Intercept (Train):** 19099.4468
  - **MSE Without Intercept (Test):** 106775.3615
  - **MSE With Intercept (Train):** 3707.8402
  - **MSE With Intercept (Test):** 3707.8402
- **Key Observations:**
  - Including the intercept improves performance significantly.
  - OLS lacks regularization, which can lead to overfitting with high-dimensional or noisy data.
- **Best Use Case:** Simple regression tasks with minimal risk of overfitting.

Problem 3: Ridge Regression

- **Objective:** To introduce regularization via ridge regression, minimizing overfitting by penalizing large weights.
- **Results:**
  - **Optimal λ:** 0.06
  - **Minimum Test Error (MSE):** 2851.3302
- **Key Observations:**
  - Ridge regression reduces overfitting by penalizing model complexity.
  - It balances the trade-off between training and testing errors, achieving better generalization than OLS.
- **Best Use Case:** Datasets with high dimensionality or noisy features, where regularization is essential.

Problem 4: Ridge Regression with Gradient Descent

- **Objective:** To use gradient descent to optimize the ridge regression loss function instead of direct minimization.
- **Results:**
  - **Train and Test Errors:** Similar to Problem 3.
  - **Optimal λ:** 0.06
- **Key Observations:**
  - Gradient descent achieves comparable results to direct minimization.
  - It is more computationally efficient for large datasets or high-dimensional features.
- **Best Use Case:** Large-scale problems where direct minimization is computationally expensive.

Problem 5: Non-linear Ridge Regression

- **Objective:** To test the impact of higher-order polynomial features (ppp) and regularization on model performance.
- **Results:**

- **Optimal p for λ=0 :** 1 (linear regression).
- **Optimal ppp for λ=0.06:** 4.
- **Train Error (p=1,λ=0.06p = 1, p=1,λ=0.06):** 3951.8391
- **Test Error (p=4,λ=0.06p = 4, p=4,λ=0.06):** 3895.5827

- **Key Observations:**
  - Polynomial features improve performance only with regularization; otherwise, the model overfits.
  - Increasing p beyond the optimal value leads to diminishing returns or overfitting.
- **Best Use Case:** Non-linear datasets where regularization controls model complexity.

Overall Recommendations:

1. **Choose Ridge Regression with Optimal λ:**
   - Ridge regression (λ=0.06) consistently outperformed OLS in terms of generalization.
2. **Use Non-linear Features Judiciously:**
   - Polynomial mappings (e.g., p=4) significantly improve performance when combined with regularization.
3. **Metrics for Model Selection:**
   - Test error (MSE) should be the primary metric for model evaluation to ensure robust generalization.
4. **Gradient Descent for Large Datasets:**
   - For small datasets, direct minimization is simpler. For large datasets, gradient descent is preferred.

These results demonstrate that **ridge regression with appropriate regularization and feature mappings** provides the best balance between training and testing performance for predicting diabetes levels.