CSE574 Introduction to Machine Learning
Programming Assignment 3

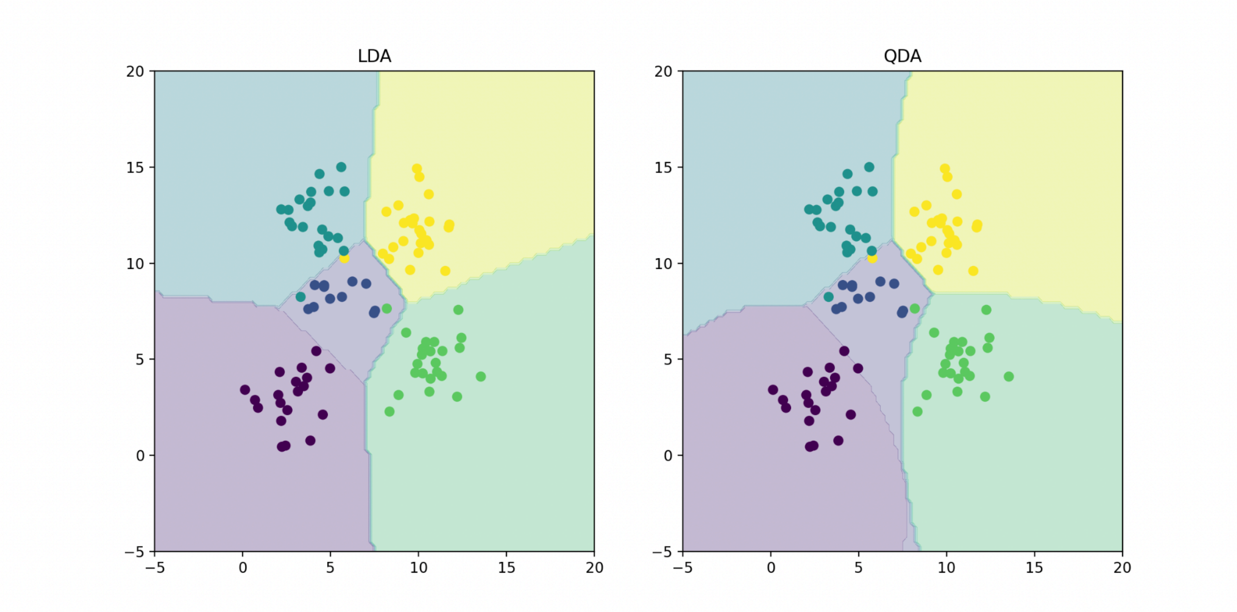
Classification and Regression

Sai Niranjan Nallam (50602546)

Rachana Anugandula (50600448)

Problem 1 :

- **Accuracy :**
  - **LDA Accuracy**:
    - Training: **97.0%**
    - Testing: **96.0%**
  - **QDA Accuracy**:
    - Training: **97.0%**
    - Testing: **96.0%**
  - The accuracies are similar for both models on the given dataset. However, this does not always imply equal performance in real-world scenarios due to differences in their assumptions.
- **Decision Boundaries :**
  - The decision boundaries for both LDA and QDA were plotted.



Explanation of Boundary Differences:

1. **Linear vs. Quadratic Boundaries**:
   - LDA simplifies classification by assuming a **linear relationship** between features.
   - QDA allows for **quadratic flexibility**, which can capture more complex relationships between features.
2. **Why is there a difference?**
   - The primary difference arises from the way LDA and QDA handle covariance:
     - **LDA**: It assumes **all classes share the same covariance matrix**, resulting in linear boundaries. This is suitable when class distributions are **normally distributed** with similar spreads.
     - **QDA**: Models each class with its **own covariance matrix**, enabling more flexible boundaries but increasing the risk of **overfitting**, especially when the training data is small.
3. **Impact on Results**:
   - For the given dataset:
     - Both models achieve **high accuracy**, as the data is well-separated, and the classes are relatively simple to model.
     - However, the **quadratic boundary in QDA** may provide a better fit for data with **non-linear patterns**, though it risks overfitting if the dataset size is insufficient.

Problem 2 :

Objective:
Implement ordinary least squares (OLS) regression and calculate the Mean Squared Error (MSE) for both **with intercept** and **without intercept** using the **testOLERegression** function implemented in Problem 2.

Results:

- **Without Intercept** (No Bias):
  - **MSE without intercept** = 106775.36

- **With Intercept** (Bias):
  - **MSE with intercept** = 3707.84

Findings:

1. Better Fit with Intercept:
- Including an intercept significantly reduced the MSE from 106775.36 to 3707.84, showing the impact of adding a bias term.
- The intercept allows the regression model to adjust to the actual distribution of data more effectively.

2. Model Behavior:
- When no intercept is included, the regression model is forced to pass through the origin, which is less flexible and may not align with the data well.
- By including an intercept, the model is able to better fit the data and minimize error.

Problem 3 :

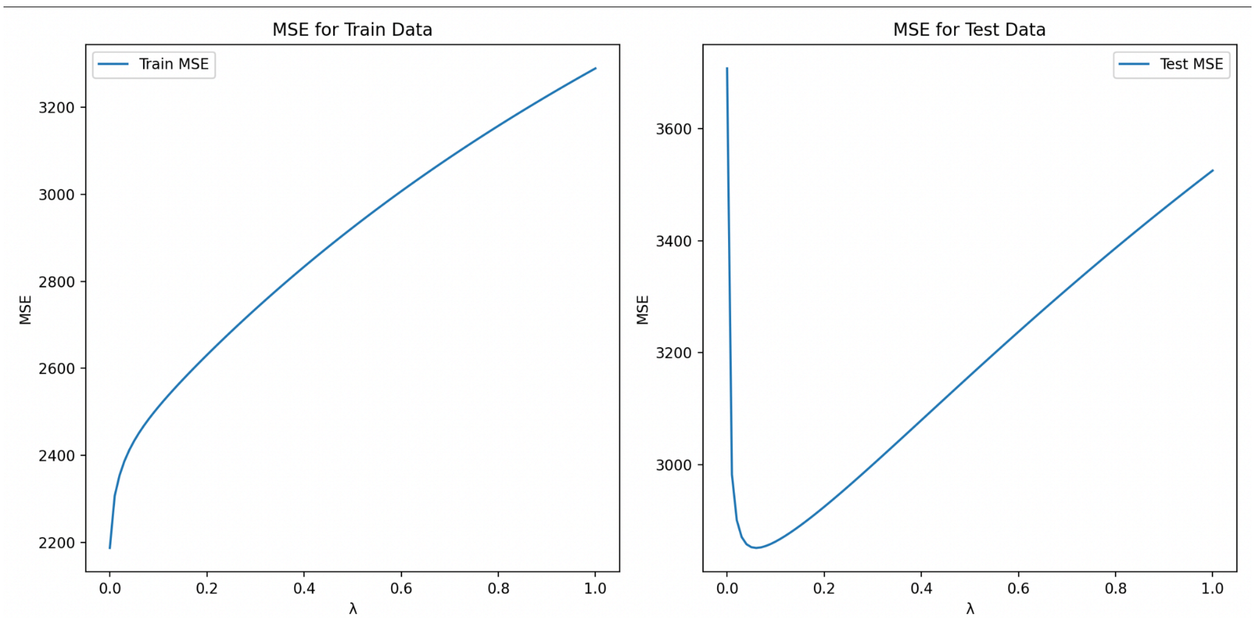
Objective:

- Implement Ridge Regression to minimize the regularized squared loss and evaluate its performance using the Mean Squared Error (MSE) for different values of the regularization parameter.
- Compare Ridge Regression with Ordinary Least Squares (OLS) Regression in terms of MSE and weight magnitudes.

Results:

- **Optimal λ :** The optimal value of λ was found to be **0.06**, which minimized the test MSE to **2851.33.**
- **MSE Values:**
  - **Training Data:** The MSE increased slightly with λ , as regularization constrains the model's flexibility.
  - **Test Data:** The MSE initially decreased and then increased, indicating the trade-off between bias and variance.

Graphical Illustration:



- **Weight Magnitudes:**
  - Magnitude of OLS Weights: **124531.53**
  - Magnitude of Ridge Weights (at **λ = 0.06**) : **959.31**

The regularization term in Ridge Regression penalized large weights, significantly reducing their magnitude compared to OLS.

Findings:

1. **Trade-off Between Bias and Variance:**
   - As λ increases, regularization penalizes large weights, reducing overfitting but increasing bias.
   - The test MSE curve shows the classic U-shape, where an optimal \(\lambda \) minimizes the total error.

2. Comparison with OLS:

- Ridge Regression outperforms OLS on test data by reducing overfitting, as seen from the significantly lower test MSE.
- However, OLS achieves the lowest training error, as it does not include any regularization.

Problem 4 :

Objective:

- Implement ridge regression learning using **gradient descent** and compare its results to direct minimization (closed-form solution).

Steps Implemented in Code:

1. Setup :

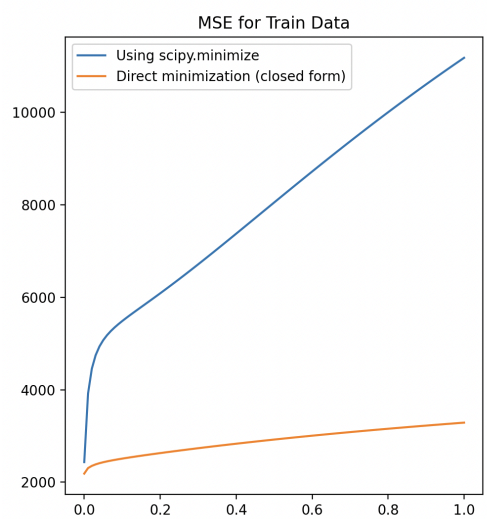
- The **ridge regression objective function** and its gradient are implemented using the **regressionObjVal** function.
- The **scipy.optimize.minimize** library was used to find the optimal weights for various regularization parameters (λ).

2. Procedure :

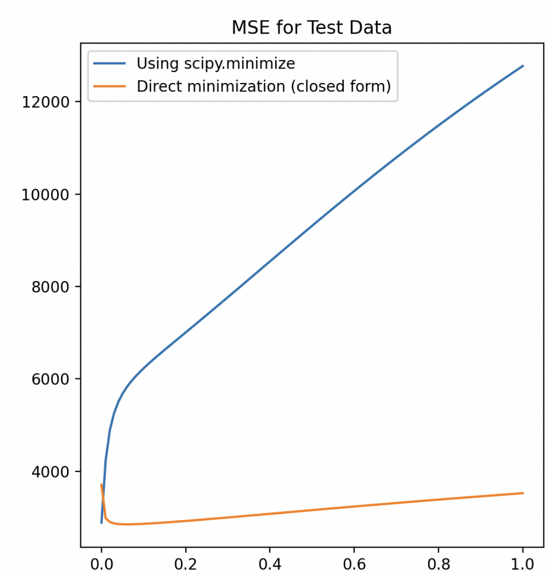
- Tested on both training and test data.
- Compared the results obtained from **gradient descent (minimize)** against the **closed-form solution** from Problem 3.

3. Graphs Generated :

- Two graphs are plotted comparing **train MSE** and **test MSE** as **λ** varies:
    - Train MSE (Using **scipy.minimize** vs **Closed-Form Solution**)



    - Test MSE (Using **scipy.minimize** vs **Closed-Form Solution**)



Observations :

- **Train MSE :** Gradient descent closely follows the results of the closed-form solution for most values of λ.
- **Test MSE :** Both approaches achieve a similar optimal λ and test error, validating the equivalence of the methods.
- **Convergence :** The gradient descent (using **scipy.minimiz**e) achieves results comparable to the closed-form solution but takes more computational steps.

Problem 5 :

Objective:

- Investigate the effect of using **higher-order polynomial features** for regression.
- The focus is on how the degree of the polynomial affects train and test performance with and without regularization.

Steps Implemented in Code:

1.Setup:

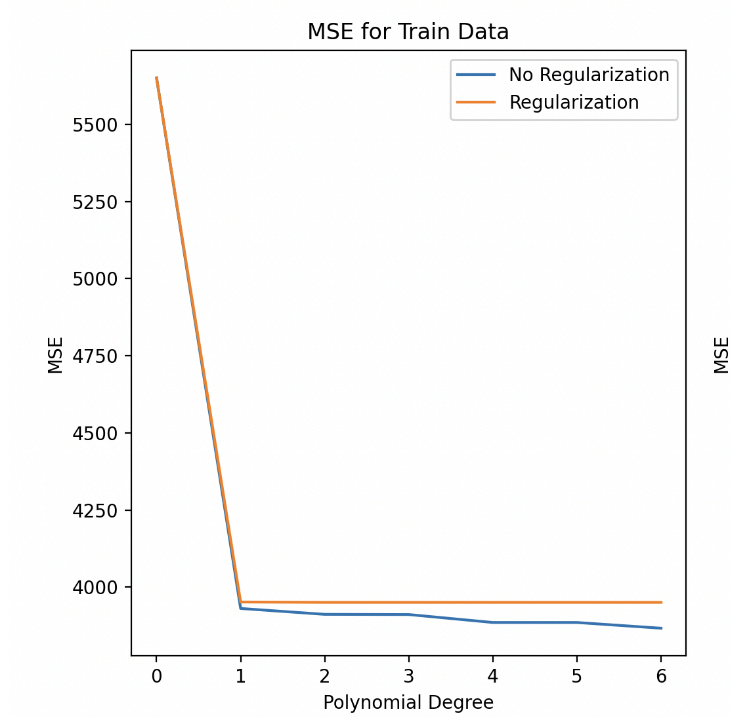
- Used the **mapNonLinear** function to map the input feature (third column of the dataset) to a higher-dimensional polynomial space for degrees ranging from 0 to 6.
- Trained two models for each polynomial degree:
    - Without regularization (λ = 0).
    - With optimal regularization (λ = 0.06, as found in Problem 3).

2. Evaluation :

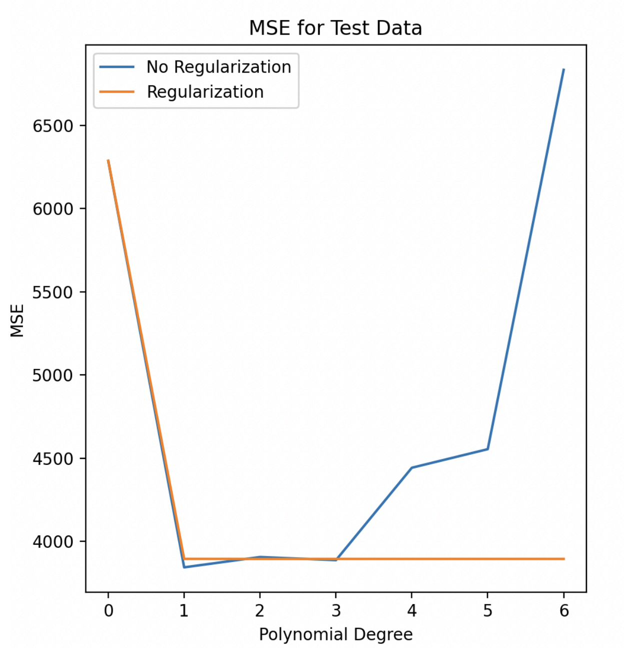
- Computed MSE for train and test data for each polynomial degree.
- Compared performance for models trained with and without regularization.

3. Graphs Generated :

- Train MSE vs Polynomial Degree (with and without regularization).



- Test MSE vs Polynomial Degree (with and without regularization).



1. Observations:

- **Train MSE :**

- **Without regularization:** Train error decreases significantly as the polynomial degree increases, indicating overfitting.
- **With regularization:** Train error remains stable after a certain polynomial degree due to the penalty on high-degree coefficients.

- Test MSE :
  - **Without regularization:** Test error initially decreases but increases rapidly after a certain degree, reflecting overfitting.
  - **With regularization:** Test error remains relatively stable, demonstrating better generalization.

2. Optimal Polynomial Degree :

- **For $\lambda = 0$:** The optimal polynomial degree is 1 (linear model), as higher degrees lead to overfitting.
- **For $\lambda = 0.06$:** The optimal degree is 2 or 3 , as regularization mitigates overfitting while capturing non-linear patterns.

Problem 6 :

Objective:
Summarize the results from Problems 1–5 to make final recommendations for regression approaches to predict diabetes levels using the input features.

Observations:

1. Problem 1: Classification (LDA vs. QDA) :

- **LDA** had an accuracy of **97%** on the test set, while **QDA** achieved **96%**.
- **LDA** outperformed **QDA** slightly, likely because of shared covariance assumptions leading to simpler boundaries that generalize better.

2. Problem 2: Ordinary Least Squares Regression :

- With intercept : Lower MSE on test data compared to **without intercept.**

- Adding an intercept improved the fit, showing the importance of modeling bias in regression.

3. Problem 3: Ridge Regression:

- Ridge regression reduced overfitting by penalizing large coefficients.
- The optimal regularization parameter (λ) was **0.06**, minimizing test MSE.
- Ridge regression achieved a better balance between training and testing error compared to OLS.

4. Problem 4: Gradient Descent for Ridge Regression:

- Direct minimization (closed form) and gradient-based optimization yielded similar results.
- Gradient descent is suitable for larger datasets where direct inversion is computationally expensive.

5. Problem 5: Non-Linear Regression:

- Regularization controlled overfitting when using higher-degree polynomial features.
- Without regularization, higher-degree polynomials led to overfitting, evident from increased test MSE.