Polytechnique Montréal

Department of Computer and Software Engineering

Report TP1

Machine Learning

Missipsa Annane, 2252373

(d) Weighted Ridge Regression Solution

Derive the closed-form solution for weighted ridge regression.

$$w^* = (X^T X + \Lambda)^{-1} X^T y$$

Derivation:

d)

$$\mathcal{L}(\omega) = \|X\omega - Y\|_2^2 + \omega^T \Lambda \omega$$

$$\mathcal{L}'(\omega) = 2X^T(X\omega - Y) + 2\Lambda\omega$$

$$\Rightarrow \omega^* = \arg\min_{\omega} \mathcal{L}(\omega)$$

$$\Rightarrow L'(\omega) = 0$$

$$\Rightarrow \cancel{2}X^T X\omega - \cancel{2}X^T Y + 2\Lambda\omega = 0$$

$$\Rightarrow X^T X\omega + \Lambda\omega = X^T Y$$

$$\Rightarrow (X^T X + \Lambda)\omega = X^T Y$$

$$\Rightarrow \omega^* = (X^T X + \Lambda)^{-1} X^T Y$$

Figure 1: Optimal weight vector for weighted ridge regression

(2) Model Evaluation

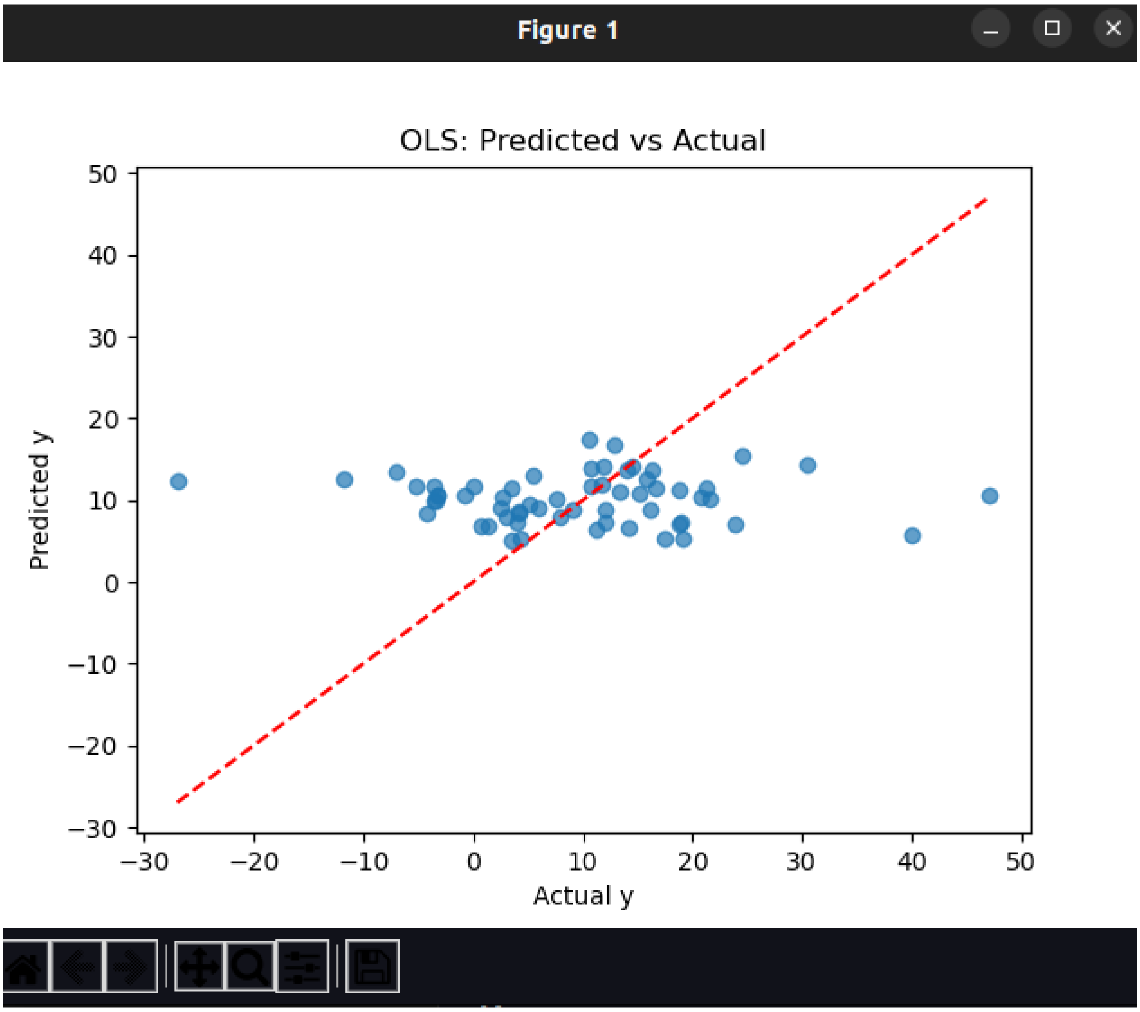


Figure 2: OLS RMSE: 12.3089

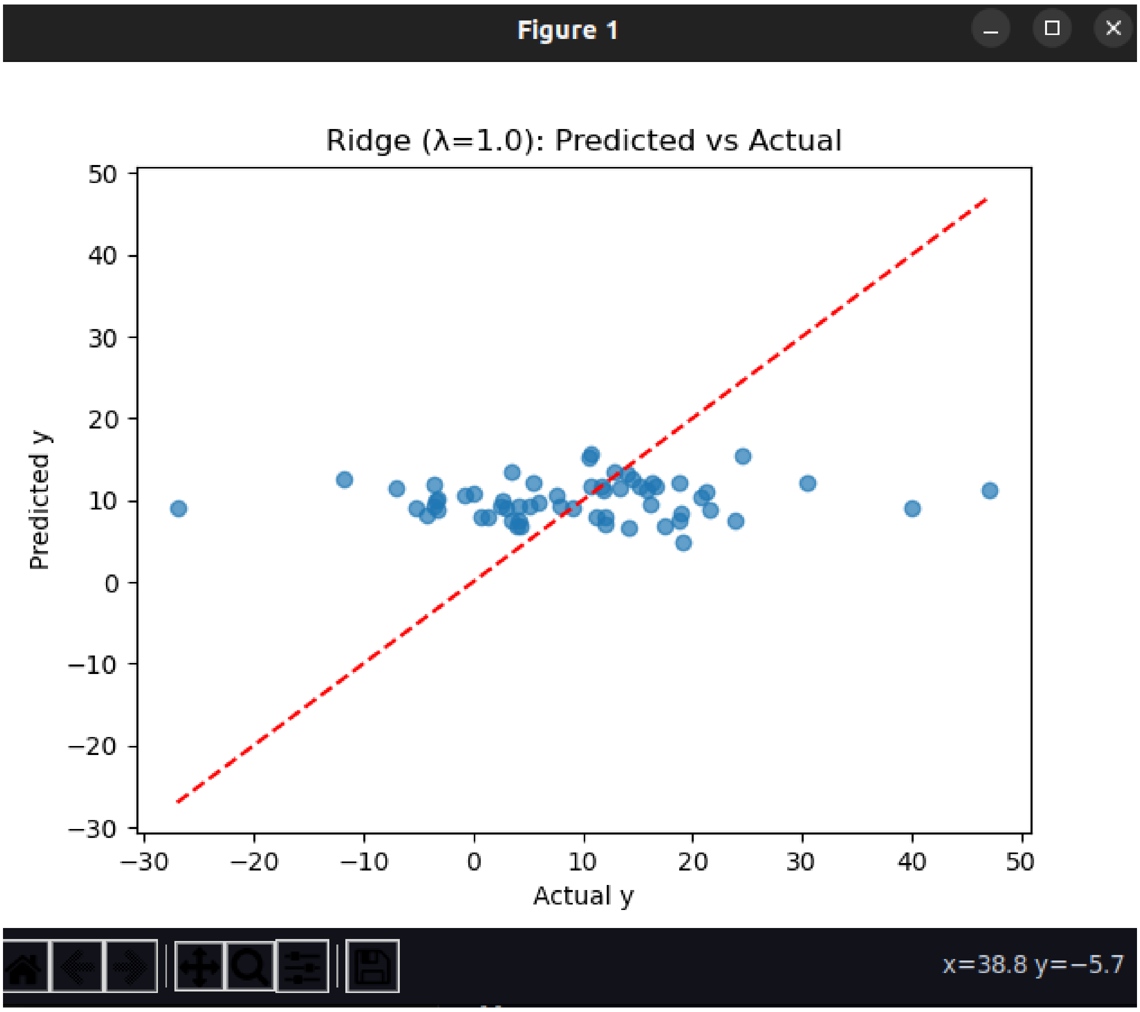


Figure 3: Ridge RMSE: 11.8427

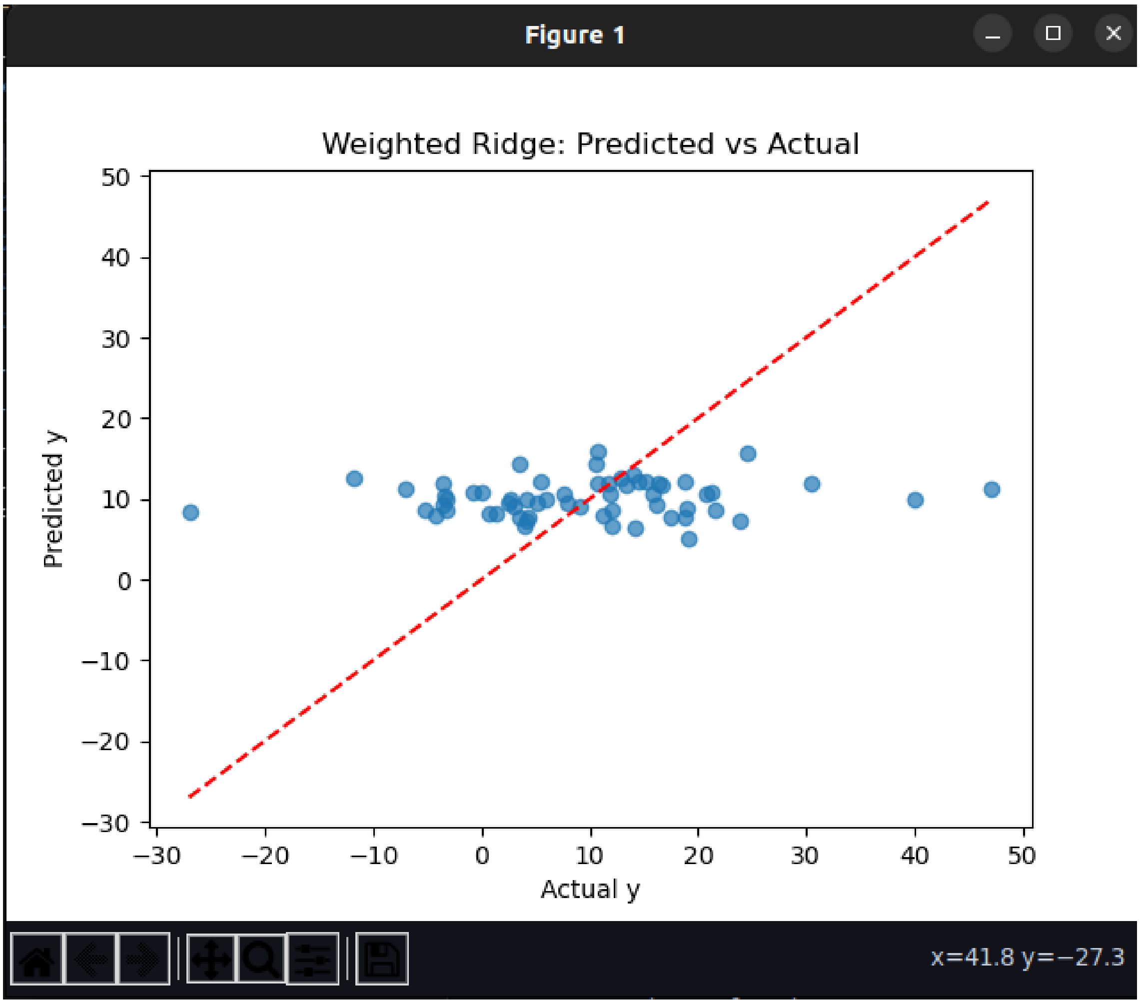


Figure 4: Weighted Ridge RMSE: 11.7615

(3) Cross-Validated Model Selection

```
Best λ for MAE: 10 with score 7.043244466450242
  λ=0.01: MAE=7.3252
  λ=0.1: MAE=7.1967
  λ=1: MAE=7.0606
  λ=10: MAE=7.0432
  λ=100: MAE=7.7759
```

Figure 5: Best lambda and corresponding mean score across folds based on MAE

```
Best λ for MaxError: 10 with score 27.237119347229754
  λ=0.01: MaxError=28.4500
  λ=0.1: MaxError=27.6584
  λ=1: MaxError=29.2785
  λ=10: MaxError=27.2371
  λ=100: MaxError=27.5375
```

Figure 6: Best lambda and corresponding mean score across folds based on MAX-ERROR

```
Best λ for RMSE: 10 with score 9.436892915554608
  λ=0.01: RMSE=9.6671
  λ=0.1: RMSE=9.5124
  λ=1: RMSE=9.5418
  λ=10: RMSE=9.4369
  λ=100: RMSE=10.1447
```

Figure 7: Best lambda and corresponding mean score across folds based on RMSE

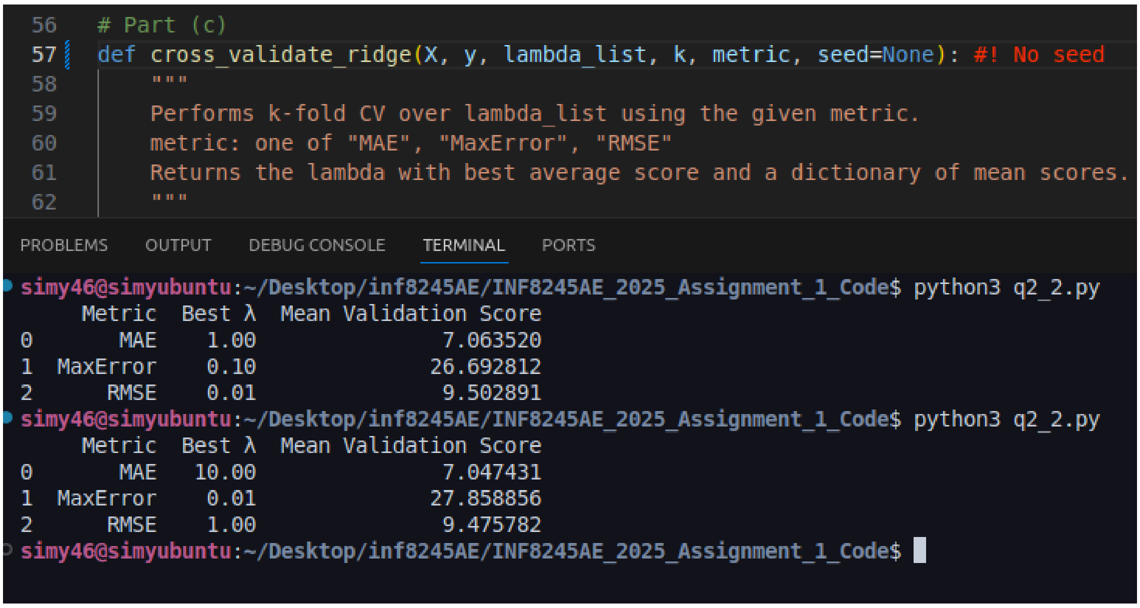
Note: it is important to mention that across two different executions of the code, i may obtain different best lambda. This is largely due to the randomness in the fold generation, combined with the fact that some lambda values yield very similar error metrics. The proof is that if I fix an arbitrary seed, I always get the same folds across different runs, and consequently the same results.

```
# Part (c)
def cross_validate_ridge(X, y, lambda_list, k, metric, seed=46): #! added seed
    """
    Performs k-fold CV over lambda_list using the given metric.
    metric: one of "MAE", "MaxError", "RMSE"
    Returns the lambda with best average score and a dictionary of mean scores.
    """
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$ python3 q2_2.py
     Metric  Best λ  Mean Validation Score
0       MAE      10               7.096259
1  MaxError     100              26.681502
2      RMSE      10               9.439613
simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$ python3 q2_2.py
     Metric  Best λ  Mean Validation Score
0       MAE      10               7.096259
1  MaxError     100              26.681502
2      RMSE      10               9.439613
simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$
```

Figure 8: Two different executions with seed = same best lambdas and metrics



```
# Part (c)
def cross_validate_ridge(X, y, lambda_list, k, metric, seed=None): #! No seed
    """
    Performs k-fold CV over lambda_list using the given metric.
    metric: one of "MAE", "MaxError", "RMSE"
    Returns the lambda with best average score and a dictionary of mean scores.
    """
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$ python3 q2_2.py
     Metric  Best λ  Mean Validation Score
0       MAE    1.00               7.063520
1  MaxError    0.10              26.692812
2      RMSE    0.01               9.502891
simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$ python3 q2_2.py
     Metric  Best λ  Mean Validation Score
0       MAE   10.00               7.047431
1  MaxError    0.01              27.858856
2      RMSE    1.00               9.475782
simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$
```

Figure 9: Two different executions with seed = different best lambdas and metrics

(4) Gradient Descent for Ridge Regression

In this experiment, we compared three learning rate schedules for gradient descent applied to Ridge regression. All schedules converged, but the constant learning rate achieved the lowest test error (RMSE = 13.89). The exponential decay performed similarly (RMSE = 13.93), while cosine annealing resulted in a slightly higher error (RMSE = 14.35). Overall, the constant schedule provided the best generalization in this setting, with exp_decay close behind.

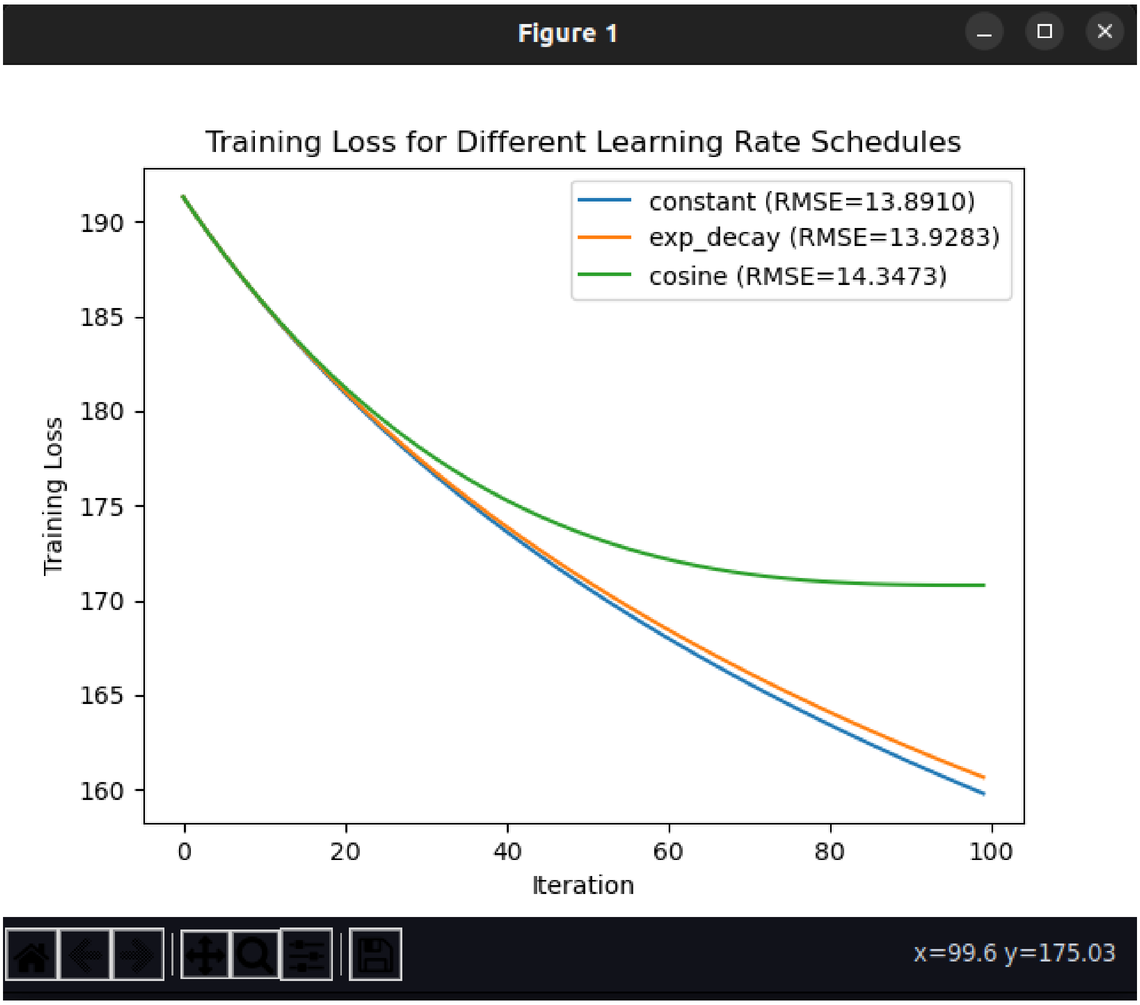


Figure 10: Training loss over iterations for the three learning rate schedules (constant, exponential decay, cosine annealing).

```
● simy46@simyubuntu:~/Desktop/inf8245AE/INF8245AE_2025_Assignment_1_Code$ python3 q3_2.py
  constant → RMSE on test set: 13.8910
  exp_decay → RMSE on test set: 13.9283
  cosine → RMSE on test set: 14.3473
```

Figure 11: Comparison of test RMSE values obtained with each learning rate schedule.