Lab 2

Ruoxin Jia
rj3u25@soton.ac.uk

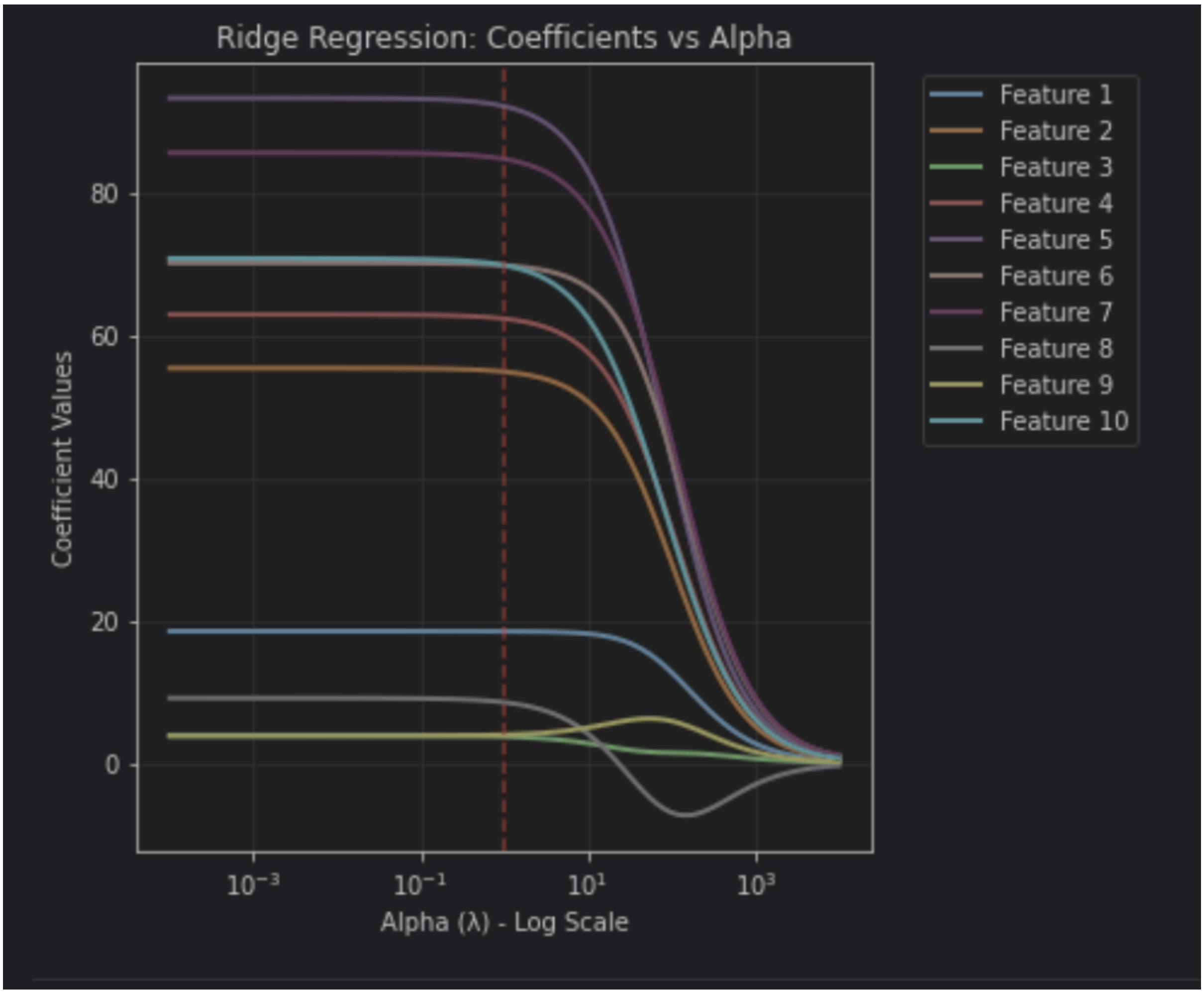
October 19, 2025

1 Part 1: Coefficient Path Analysis

The first part of the experiment involved visualizing how the regression coefficients for Ridge (L_2) and Lasso (L_1) regularization change as the regularization strength, λ (alpha), increases. The generated plots illustrate the fundamental differences in how each method penalizes large coefficients.

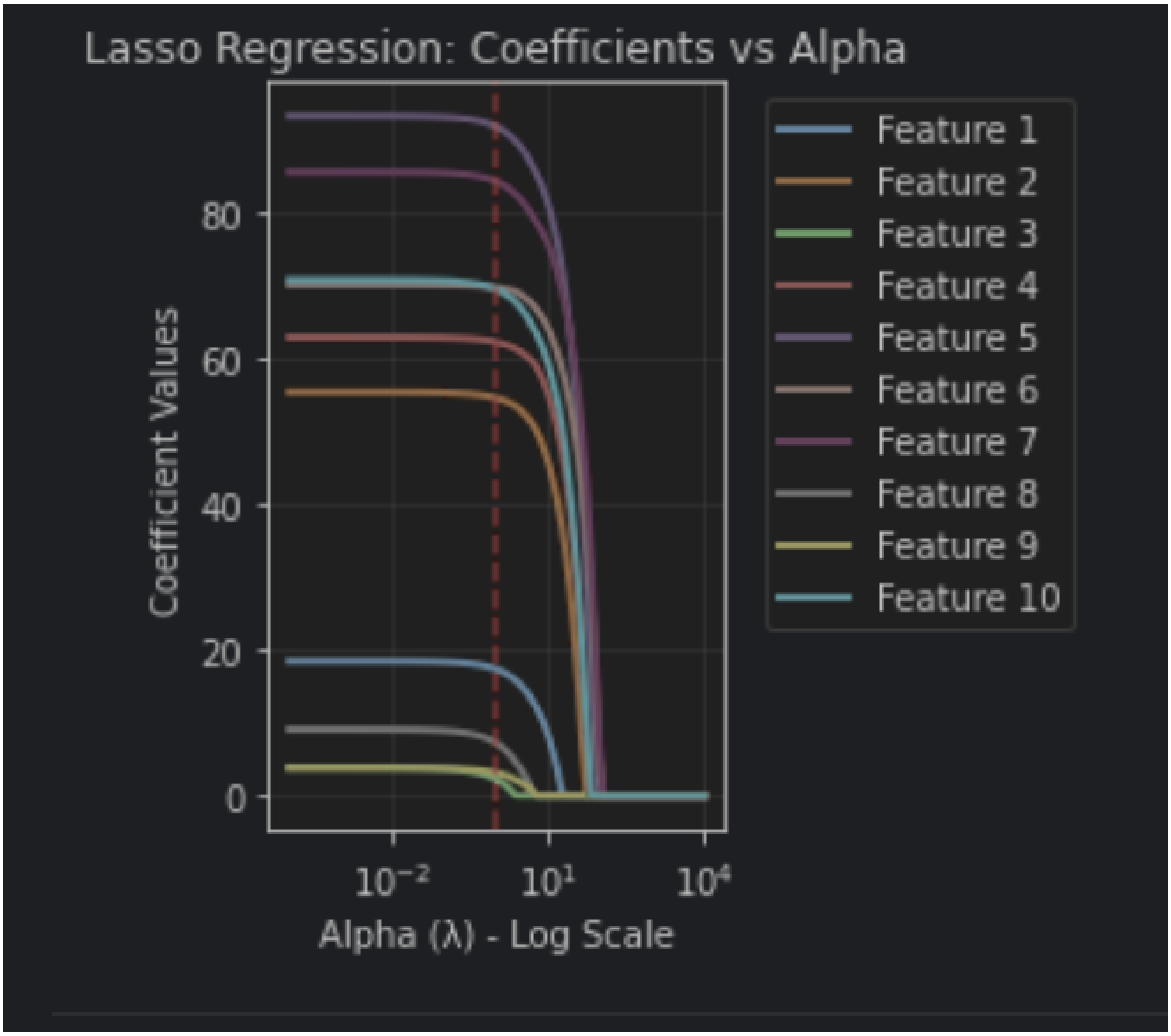
```

plt.figure(figsize=(12, 6))

# Plot Ridge coefficients
plt.subplot(1, 2, 1)
for i in range(ridge_coefs.shape[1]):
    plt.plot(alphas, ridge_coefs[:, i], label=f’Feature {i+1}’, linewidth=2)

plt.xscale(’log’)
plt.xlabel(’Alpha (λ) - Log Scale’)
plt.ylabel(’Coefficient Values’)
plt.title(’Ridge Regression: Coefficients vs Alpha’)
plt.grid(True, alpha=0.3)
plt.legend(bbox_to_anchor=(1.05, 1), loc=’upper left’)

# Add vertical line at alpha=1 for reference
plt.axvline(x=1.0, color=’red’, linestyle=’--’, alpha=0.7, label=’Alpha=1’)
```

Listing 1: Testing different sample sizes and bin configurations



```

# Plot Lasso coefficients
plt.subplot(1, 2, 2)
for i in range(lasso_coefs.shape[1]):
    plt.plot(alphas, lasso_coefs[:, i], label=f'Feature {i+1}', linewidth=2)

plt.xscale('log')
plt.xlabel('Alpha (λ) - Log Scale')
plt.ylabel('Coefficient Values')
plt.title('Lasso Regression: Coefficients vs Alpha')
plt.grid(True, alpha=0.3)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Add vertical line at alpha=1 for reference
plt.axvline(x=1.0, color='red', linestyle='--', alpha=0.7, label='Alpha=1')

plt.tight_layout()
plt.show()
```

Listing 2: Testing different sample sizes and bin configurations



2 Lab Questions and Answers

Q1: Why must we set fit_intercept=False?

The analytical formula $w = (X^TX + \lambda I)^{-1}X^Ty$ assumes no intercept term and expects centered data. When `fit_intercept=True`, `scikit-learn` automatically centers the data and calculates a separate intercept, changing the weight values. Setting `fit_intercept=False` ensures that `scikit-learn` uses the exact same formula without data centering.

Q2: What would happen if X^TX is not invertible and $\lambda = 0$?

If X^TX is singular and $\lambda = 0$:
- The matrix inverse fails completely
- No unique solution exists due to infinite possible weight combinations
- The system becomes numerically unstable

A small positive λ fixes this because:
- $(X^TX + \lambda I)$ is always invertible for $\lambda > 0$
- The λI term makes the matrix positive definite
- All eigenvalues become $\geq \lambda > 0$, ensuring numerical stability
- This guarantees a unique, stable solution

Q3: Looking at your Ridge coefficient plot, what is the general trend?

As the 'alpha' (λ) value increases, all Ridge coefficients are smoothly shrunk progressively closer and closer to zero. However, they never become exactly zero. The penalty (L_2 norm) discourages large coefficients but does not force them to be precisely zero, meaning all features are retained in the final model, albeit with reduced influence.

Q4: Compare the Ridge and Lasso plots. What is the most significant difference?

The most significant difference is:
- Ridge: All coefficients shrink smoothly but remain non-zero.
- Lasso: Coefficients can become exactly zero, especially at large alpha values.

Lasso performs feature selection because it can completely shrink unimportant features' coefficients to zero, automatically removing them from the model. With large alpha values, Lasso may keep only a few most important features, while Ridge retains all features even with very small coefficients.

Q5: Why is it critical to fit the scaler only on the training data?

Fitting the scaler only on training data prevents data leakage and ensures realistic model evaluation. If we fit on the entire dataset, test set information contaminates the training process, leading to over-optimistic performance estimates that don't reflect real-world generalization ability.

Q6: Which model performed better on the test set in terms of MSE?

The Lasso model performed better on the test set. It achieved a lower Test Mean Squared Error (MSE) of 142,996.09 compared to the Ridge model's Test MSE of 146,744.70.

Q7: How many features did the Lasso model select?

The Lasso model selected 6 features out of 19 total features.

This tells us several important things about the dataset:
- Feature Sparsity: Only 31.6% of the original features were deemed important enough to have non-zero coefficients
- Redundancy: Approximately 68.4% of the features were effectively eliminated by Lasso's L1 regularization, suggesting they contribute little to predicting salary
- Efficiency: Salary prediction can be accomplished with fewer variables, making the model more efficient

Q8: Which player statistics seem to be the most important predictors?

Key Salary Predictors:
- Hits
- PutOuts
- CRBI
- Division_W

Model Agreement:
- 4 out of 5 top features are identical in both models
- Same features show consistent direction of influence

- High consensus indicates robust predictors

Conclusion: Both models strongly agree that current hitting, defensive performance, career RBIs, and league division are the most reliable salary predictors.

3 Part 2: Model Performance on Hitters Dataset

3.1 Summary Table: Test MSE Comparison

The table below summarizes the optimal hyperparameters and final test set performance for both models.

```python
# Calculate MSE values
ridge_mse = mean_squared_error(y_test, ridge_pred)
lasso_mse = mean_squared_error(y_test, lasso_pred)

print(f"Ridge Test MSE: {ridge_mse:.2f}")
print(f"Lasso Test MSE: {lasso_mse:.2f}")

# Determine which model performed better
if ridge_mse < lasso_mse:
    better_model = "Ridge"
    mse_difference = lasso_mse - ridge_mse
else:
    better_model = "Lasso"
    mse_difference = ridge_mse - lasso_mse
```

Listing 3: Testing different sample sizes and bin configurations

Table 1: Final Model Performance Comparison on the Test Set

Model	**Best Alpha** (λ)	**Final Test MSE**
Ridge Regression	132.19	146,744.70
Lasso Regression	24.77	142,996.09

4 Conclusion: Ridge vs. Lasso Trade-offs

Based on the findings from this lab, the trade-offs between Ridge and Lasso regression are clear. In our experiment with the Hitters dataset, Lasso yielded a slightly lower test MSE (142,996 vs. 146,745), indicating marginally better predictive accuracy. However, its primary advantage was interpretability and sparsity. By shrinking 13 of the 19 feature coefficients to exactly zero, Lasso provided a parsimonious model that identified a small subset of key salary predictors. This is extremely valuable when the goal is to understand the underlying drivers of a system or to build a simpler, more efficient model. Ridge, in contrast, retained all 19 features, shrinking their coefficients to manage multicollinearity but providing no feature selection. Therefore, Lasso is the superior choice when model simplicity and feature selection are priorities, while Ridge is more appropriate when one believes all features are relevant and the primary goal is to stabilize a model with highly correlated predictors.