CSC4242

Artificial Intelligence - Assignment 01

Registration No.: SC/2020/11730

Student Name: Navinda Hewawickrama

November 2025

Bachelor of Computer Science (Special) Degree
Department of Computer Science, University of Ruhuna

Declaration: I acknowledge the use of Chatgpt to generate contents of the result code parts of the assignment

1 Introduction

The main purpose of doing this assignment was to learn about core regression techniques in Artificial Intelligence (AI), including Linear Regression, Regularized Regression (Ridge/Tikhonov), and Lasso Regression. From each task, step by step, we learn and understand how regression models learn from data, how they generalise to unseen data samples, and how the regularisation prevents the overfitting issue by penalising large weights. We use two datasets: the Diabetes dataset, which is a synthetic dataset, and a real-world dataset like the Huuskonen Solubility dataset.

By using Python, we implemented different regression models, visualised their behaviour, and interpreted the different roles of regularisation parameters. The last part of the assignment focuses on using the learnt methods in the assignment for predicting the aqueous solubility of chemical compounds based on molecular descriptors, which shows about AI in computationalchemistry.

2 Linear Least Squares Regression

2.1 Objective

The first task was given to understand the basic principles of Linear Least Squares Regression using the diabetes dataset from the `sklearn.datasets` library. Modelling the relationship between different features (e.g., BMI, blood pressure, etc.) and a target variable representing a diabetes progression metric, was the goal.

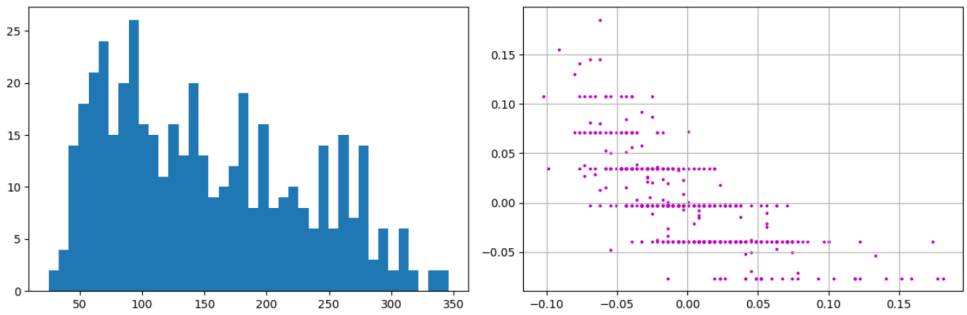

Figure 1: Histogram of Target Values and Scatter Plot of Selected Features

2.2 Implementation

We implemented the regression model in two ways.

- Using the `LinearRegression` class from the `scikit-learn` library.
- Using the analytical solution taken from the pseudo-inverse formula

$$w = (X^T X)^{-1} X^T t$$

where X is the feature matrix and t is the target vector.

In the comparision between the two mthods we see identical predictions, showing that the analytical solution is correct. The scatter plots between the predicted and the actual values demonstrate a relationship near to a linear one, which actually indicates a good model fit (Figure 2).

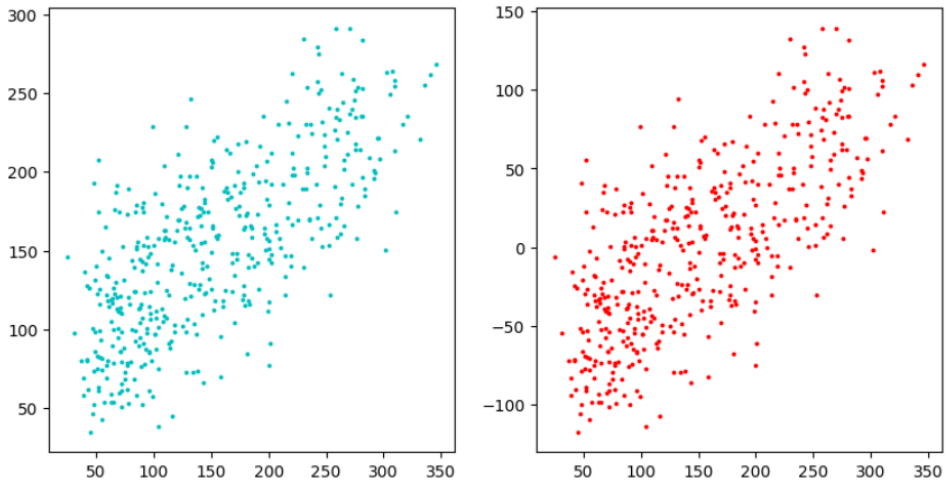

Figure 2: Linear regression in the dataset using two methods

2.3 Discussion

We can see that the linear regression actually estimate the coefficients by minimizing the mean squared error (MSE). The optimal weight vecttor is actually computed correctly by the psuedo inverse method, while `scikit-learn`'s method of implementation uses efficient numerical solvers. When the data matrix X has full column rank, both methods actially yield the same coefficients and R^2 values.

3 Regularization (Ridge/Tikhonov)

3.1 Objective

To control the complexity of the model and to stop the overfitting, regularization actually introduces a penalty term. The Tikhonov (Ridge) regularization modifies the least squares loss function as given below.

$$\min_w \|t - Xw\|_2^2 + \gamma \|w\|_2^2$$

Here γ is the regularization parameter that actually controls the strength of penalization.

3.2 Implementation

The regularized weights are actually computed using:

$$w_R = (X^T X + \gamma I)^{-1} X^T t$$

We can compare the magnitudes of the coefficients taken from the standard and regularized regression. By looking at the two plots we can visualize the effect of regularization. Bar plots of w and w_R show that Ridge regression reduces coefficient variance significantly when compared to the other.

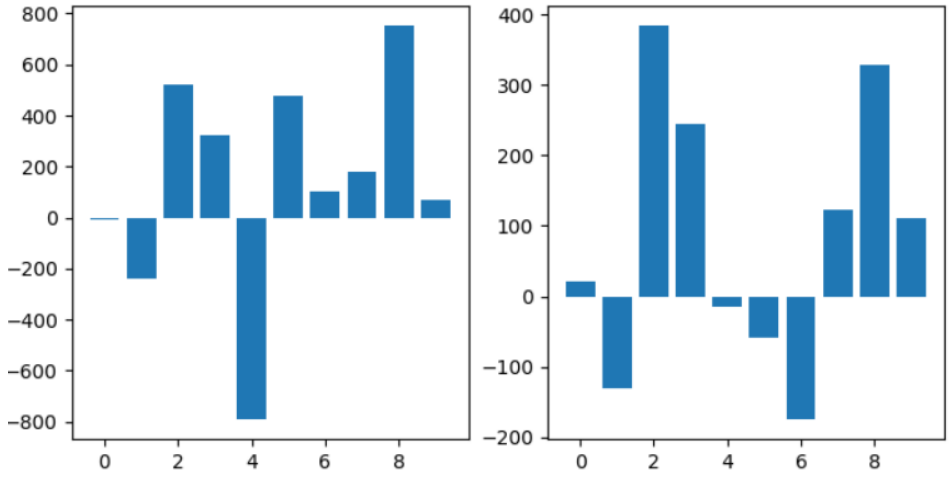

Figure 3: omparison of Weight Magnitudes Before and After Regularization (Least Squares vs Tikhonov)

3.3 Observations

In these bar grpahs we can see that as γ increases, weights shrink towards zero, reducing model variance but slightly increasing the bias. Due to this trade-off, generalization improves, where the feature are correlated or when there is noise in the dataset.

4 Regularization Path and Lasso Regression

4.1 Objective

In this section we examine how coefficients evolve when the regularization strength α changes in Lasso, LARS, and Elastic Net regressions. The `lasso_path`, `lars_path`, and `enet_path` functions from `scikit-learn` are used to compute these paths.

4.2 Implementation

The code for this section was adapted from the official Scikit-learn example titled *"Lasso and Elastic-Net Regularization Paths"*. The implementation computes the evolution of regression coefficients for multiple algorithms (Lasso, LARS, and Elastic Net) as the regularization parameter α changes.

The dataset used is the diabetes dataset, standardized before fitting to ensure comparable scaling across features. Each model computes a **regularization path**, which shows how each coefficient transitions from zero (strong regularization) to its final fitted value (weak regularization). This provides valuable insight into the model's feature selection behaviour.

The full reference implementation can be accessed at:

https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_lasso_lars_elasticnet_path.html

4.3 Results

For Lasso, LARS, and Elastic Net, plots of coefficients trajectories (Regularization Paths) are generated. The evolution of the coefficients as α decreases is represented by each line. The coefficients become non-zero, as α becomes smaller, which leads to more complex models.

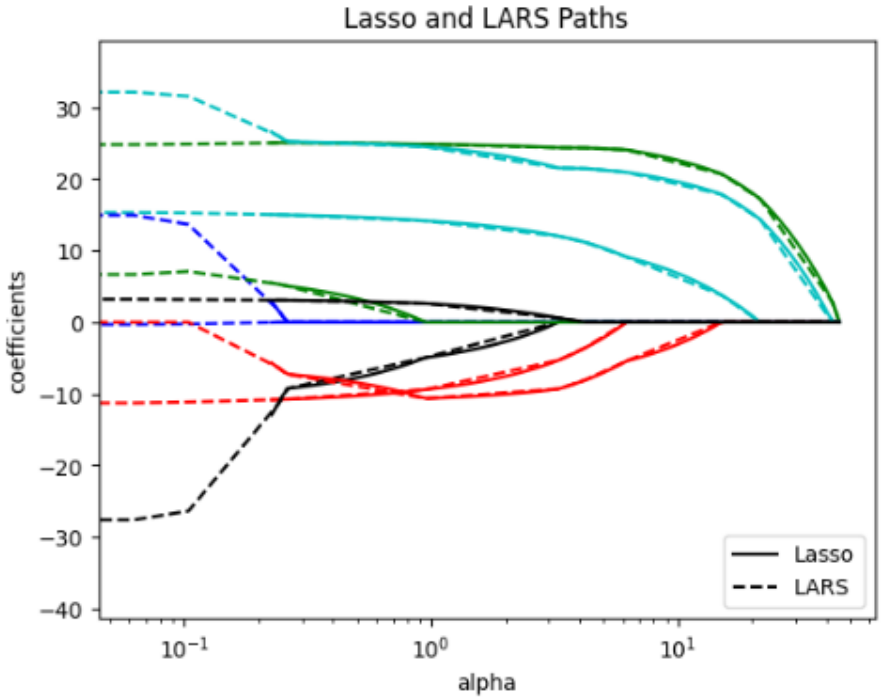

Figure 4: Lasson and Lars Path

4.4 Interpretation

The real ability of lasso is highlighted with its strong regularization, where most coefficients are driven to zero and also shows its ability to perform feature selection. Elastic net, which combines L1 and L2 penalties, shows a smoother transition. Elastic net often retain correlated features together.

5 Comparison Between Ridge and Lasso

When we compare ridge and lasso regressions we can see that ridge tends to shrink all coefficients equally, while lasso drives many coefficients exactly to zero, resulting in a sparse model. This is very risky when it comes to high dimensional data as it helps to identify the most important and relevant features that actually contributes to the predictions.

Wen looking at them we can see that ridge actually minimizes the variance more effectively, but lasso enhances the interpretability by selecting the important features.

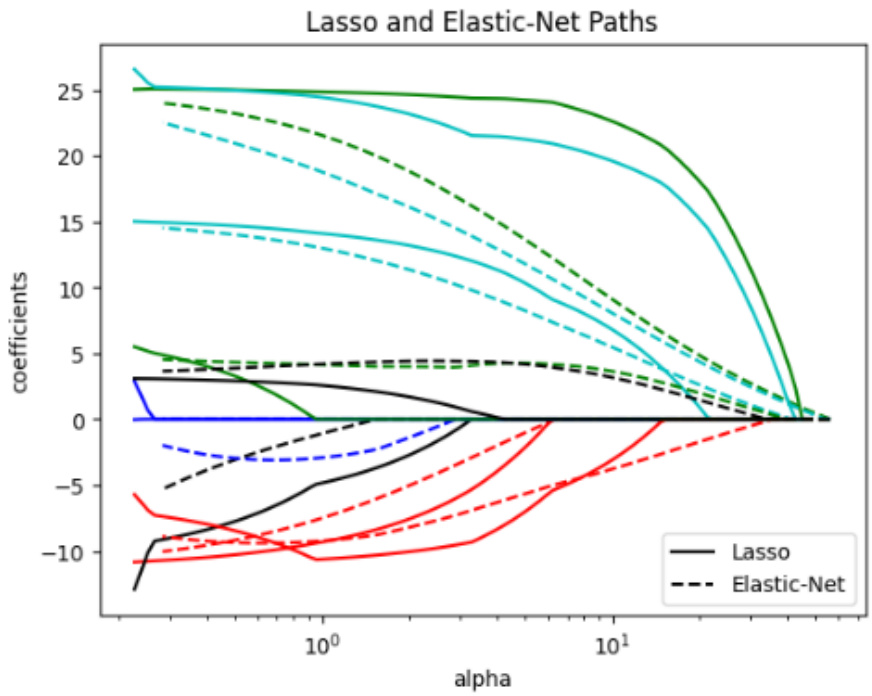

Figure 5: Lasso and Elastic Net Path

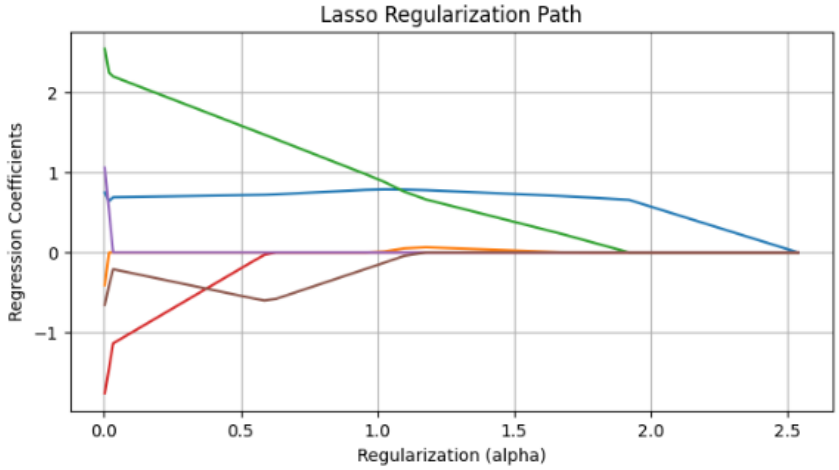

Figure 6: Lasso Regularization path

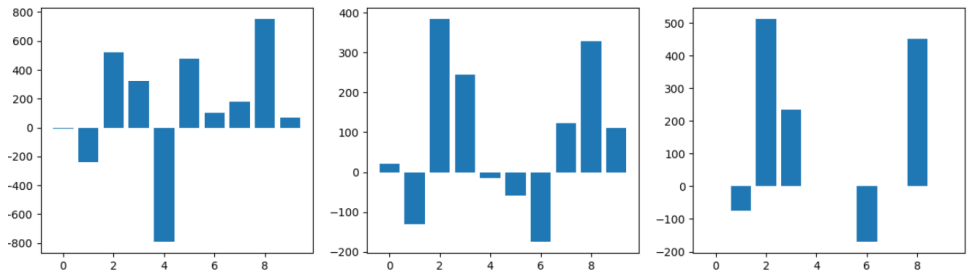

Figure 7: Comparing the effect of regularization with lasso

6 Solubility Prediction of Chemical Compounds

6.1 Objective

In the final taks, the use of regressive techniques to estimate the aqueous solubility of chemical compounds with the help of the Huuskonen Solubility dataset is performed. A large set of molecular descriptors that encapsulates physicochemical properties in each compound has been used to represent that compound in the dataset.

6.2 Data Preparation

The dataset was preprocessed by:

- Removing the first five columns that has no numeric data.
- Selecting descriptors that has only numbers.
- Dropping missing values.
- Splitting the data into 80% training and 20% testing subsets.

A histogram of the dataset values shows a near-normal distribution centered around zero, indicating balanced solubility ranges in the dataset.

6.3 Regularized Regression (Ridge)

A Ridge regression model, as we did earlier, was implemented using:

$$w = (X^T X + \gamma I)^{-1} X^T t$$

where gamma was 2.3. The plots show both the training and the testing set data. They both show a strong correlation between predicted and actual solubility values. The model actually captures the linear relationship between them while avoiding being overfit to the training dataset.

6.4 Lasso Regression and Feature Selection

By using various alpha (α) values the lasso model tested to explore the sparsity and the capabilities of feature selction. The model's test error (MSE) and number of non zero coefficients were noted and tow plots were generated.

- Test MSE vs Regularization Strength (α)

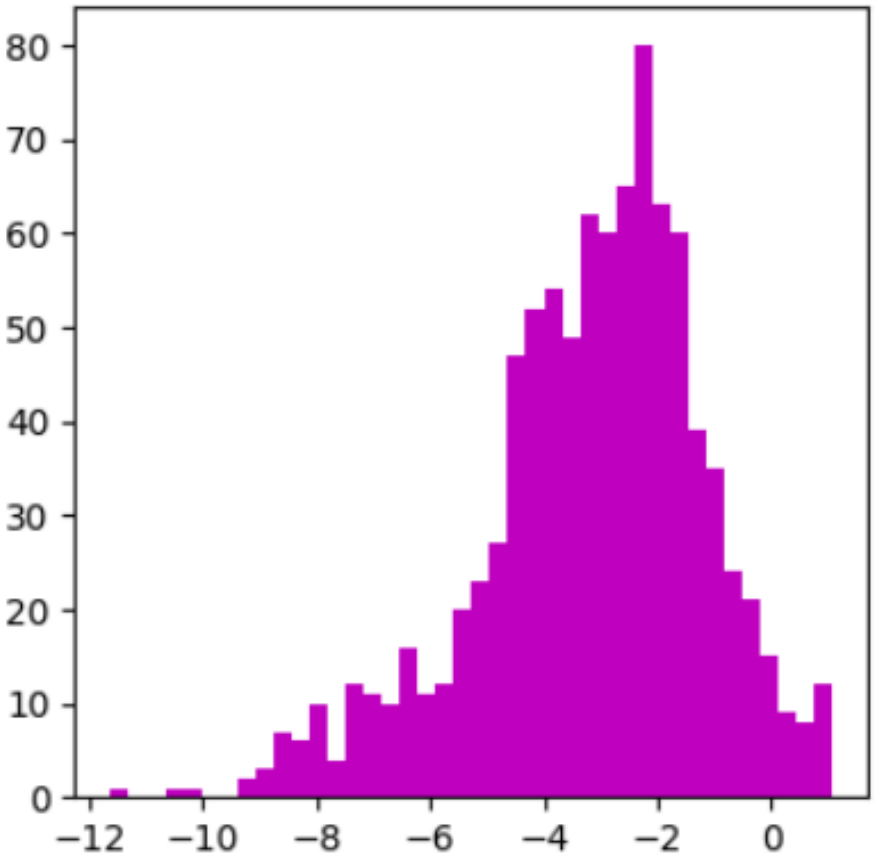

Figure 8: distribution of the logs

- Number of Non-Zero Coefficients vs α

In the results we can see that small values of α yield better accuracy but they include many features. Compared to them, larger α values lead to sparse models with a little bit of reduced performance.

The top ten molecular descriptors of the Lasso model with the best performance were selected as most significant in the prediction of solubility. Re-training the model with these top ten features alone gave a similar value of the $R2$ label as the full Ridge model, which shows that even a small set of descriptors can include most of the predictive information.

6.5 Discussion and Comparison

The final prediction question demonstrate how regularization improves the robust qualities of linear models on high-dimensional chemical data. Stability, low variance estimates are shown in ridge regression while lasso helps to identify key molecular features.

Compared with the findings of Huuskonen (1998) and Pirashvili et al. (2018), our linear models achieve moderate performance ($R^2 \approx 0.75$), whereas neural network and ensemble methods reported in literature reach around $R^2 = 0.9$. Despite all of these results, for future studies in cheminformatics, the interpretability and efficiency of linear models make them valuable for exploratoryanalysis.

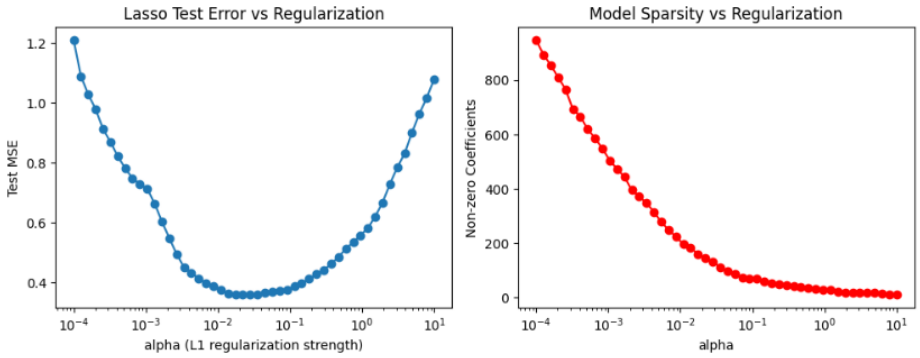

Figure 9: Lasso Regression and Feature Selection

7 Conclusion

This assignment provided a prictical and hands-on understanding of regression techniques actually work and their significance. Through step by step learning we observed how:

- Linear regression models data relationships through least squares fitting.
- Regularization reduces overfitting and enhances model generalization.
- Lasso regression performs feature selection and simplifies models.
- Ridge regression stabilizes solutions in the presence of multicollinearity.
- In real-world data such as solubility prediction, these methods provide interpretable baselines for complex models.

The experiments strengthened theoretical ideas of Artificial Intelligence and Machine Learning through the association of the mathematical derivations with the data-driven outcomes.

Appendix

A. Source Code for Experiments

All scripts were executed using Python 3.10 and `scikit-learn`, `NumPy`, and `Matplotlib` libraries.The full source code and experiment results are available in the interactive Google Colab notebook at:

https://colab.research.google.com/drive/1ki_F14NMSntGXAdEH74MdEY3hNxhX4RR?usp=sharing

B. Non AI references

- https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html
- https://www.geeksforgeeks.org/data-visualization/plotting-histogram-in-python-us
- https://www.w3schools.com/python/matplotlib_histograms.asp
- https://taylorandfrancis.com/knowledge/Engineering_and_technology/Engineering_support_and_special_topics/Tikhonov_regularization/
- https://en.wikipedia.org/wiki/Ridge_regression
- https://www.dataquest.io/blog/regularization-in-machine-learning/
- https://youtu.be/21TgKhy1GY4?si=l04mvRksj-nzHpv2
- https://www.geeksforgeeks.org/machine-learning/regularization-in-machine-learnin
- https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_lasso_lars_elasticnet_path.html
- https://youtu.be/LmpBt0tenJE

C. Dataset Reference

Huuskonen, J. (1998). *Estimation of Aqueous Solubility of Organic Compounds Based on Molecular Structure. Journal of Chemical Information and Computer Sciences*, 40(3), 773–777.

Pirashvili, K., et al. (2018). *Machine Learning Approaches for Solubility Prediction. Computational and Structural Biotechnology Journal*, 16, 104–113.