2026 Volume: 02 Issue: 04 www.ijrpa.com ISSN 2456-9995 Case Study

International Journal Research Publication Analysis

Page: 01-10

OVERFITTING REDUCTION IN REGRESSION MODELS USING L1 AND L2 REGULARIZATION: A COMPARATIVE STUDY ON THE CALIFORNIA HOUSING DATASET

[1*]Dr. Ramya B. N., [2]Sai Bhargava M., [3]Yashwanth V.

[1]Associate Professor, Dept. of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

[2,3]Dept. of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

Article Received: 19 March 2026
Article Revised: 09 April 2026
Published on: 29 April 2026

***Corresponding Author: Dr. Ramya B. N.**
Associate Professor, Dept. of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

DOI: https://doi-doi.org/101555/ijrpa.9236

ABSTRACT

Among several issues associated with applying machine learning in practice, overfitting remains particularly persistent.In the case where a model suffers from overfitting, it can be said that the model has been trained "too well" – that is,the model has absorbed all the peculiarities and noise from the data used for training, making it ineffective when applied to new data. In the case where linear models are used for regression, the issue of overfitting is resolved via regularization, which refers to a group of methods aimed at adding controlled penalties to the optimization objective to prevent overgrowth of parameters. In this work, we consider the specifics of two commonly adopted regularization methods –L1 (Lasso Regression) and L2 (Ridge Regression).

The experiments we perform on are based on the California Housing dataset and involve training three different regression models: an unregularized version, the Lasso, and the Ridge regression, which will be tested by MSE of their prediction on a test dataset.

The results are quite telling. While Ridge Regression achieves MSE comparable to the baseline, i.e. 0.5559, while controlling for coefficients' magnitude, Lasso fails in that regard and gets MSE 0.6796 due to dropping important features from its prediction. In addition to comparing values of MSEs, we look into how coefficients' vectors of different models differ from one another and interpret this difference based on properties of input data. The main

point we want to make in this paper is that there is no single approach to choosing the proper regularization method and that it depends on sparsity of the input data.

KEYWORDS: Overfitting; L1 Regularization; L2 Regularization; Lasso Regression; Ridge Regression; Linear Regression; California Housing Dataset; Mean Squared Error; Generalization; Feature Sparsity; Coefficient Shrinkage; Supervised Learning.

INTRODUCTION

Ordinary least squares linear regression is no exception to this problem. While significantly less complex compared to deep learning architectures, linear regressions can still overfit if there are many variables with respect to the number of observations used to train the model, if these variables exhibit high correlation between themselves, or if the model is trained on noisy data without imposing any restrictions on the absolute value of its parameters. In this case, the model will be assigning very large positive and negative weights to specific variables such that they compensate for each other in the training dataset but drastically differ in other instances.

This problem is approached by changing the very nature of the optimization problem. Instead of just trying to minimize the difference between the prediction and the actual output, a regularized machine learning model will try to minimize not only this difference but also the magnitude of its own coefficients at the same time. This results in a situation where the model tries to find a solution that fits. the data, but does not get overly complex. There are two types of penalty that have proven their worth throughout the years:

The first is L1 regularization that minimizes the sum of the absolute values of the coefficients and produces an elegant mathematical result of having all zero elements in the vector of coefficients. The second is L2 regularization, which minimizes the sum of the squares of the coefficients.

These two methods have entirely different philosophical approaches to data. The Lasso method, otherwise known as L1 regularization, is based on the philosophy that the universe is conservative in nature, where only a few attributes are relevant, and the rest can be disregarded. The other method, Ridge regression, or L2 regularization, believes in the philosophy of equality, where every attribute holds some importance and must be used, albeit

in moderation. Neither of the two philosophies is entirely right; it depends on the problem at hand.

In this essay, we will investigate this dependence empirically by applying various models on the California Housing dataset. The objective here is not just to state which one is better by any criteria, but rather to comprehend the reasoning behind such performance, what can be seen from the coefficient estimates, and what conclusions can be derived for people facing similar choices.

METHODOLOGY

Dataset Overview and Preparation

The concept of dependency is investigated below in practice on the example of California Housing Dataset. However, it is not our objective to simply state what model won in accordance with certain metric. Rather, we wish to investigate what makes each model better or worse than others and learn more about coefficient dependency within the model. Practitioners should understand the consequences behind the choice of model type and its parameters when deciding whether to use regularization The California Housing Dataset comes from the 1990 US Census and has become widely used for regression tests in recent years. The Dataset features 20,640 observations which relate to specific blocks in California. Each observation has eight numerical features associated with it: median income of houses in the block (MedInc), median age of housing (HouseAge), average number of rooms in the block (AveRooms), average number ofbedrooms in houses (AveBedrms), total number of people living in the block (Population), average number of occupants per household (AveOccup), latitude, and longitude of the block center.

The splitting was done in an 80:20 proportion, resulting in 16,512 samples in the training set and 4,128 samples in the test set. In order to ensure perfect reproducibility, we used a predetermined seed value to generate our random splits. To prepare for fitting models, we standardized all features by utilizing scikit-learn’s StandardScaler, calculating mean and variance for each feature using only training data and then applying the standardization to both data sets. Standardizing before regularization is an absolute necessity, as otherwise features with larger numerical values would receive disproportionately larger penalty terms due to having higher values, which would unfairly bias the process of penalization.

Model Formulations

A plain vanilla OLS model is the base for our analysis. It is a conventional linear model where the optimization procedure aims to find the weights vector by minimizing the sum of squared residuals. There are no restrictions or constraints applied to the coefficient values in any way, allowing the optimizer to reduce training errors as much as possible without taking anything into account in terms of what may happen to the weights during optimization. It provides us with a simple upper bound in terms of model complexity. Lasso is an alternative version of the OLS model, where an additional L1 penalty is added to the loss function. In particular, we minimize the following criterion: (1/2n)||y − Xw|| 2 + α||w|| 1 , where n denotes the number of training samples, α is the regularization parameter, while ||w|| 1 stands for the sum of the absolute values of all coefficients. The key difference between the penalty term and a normal quadratic term is that the L1 norm is non-differentiable when the value of the coefficient is equal to zero. This allows us to use coordinate descent and force some coefficients to take on exactly zero values.

In the Ridge model, the penalty function changes from the L1 norm to the L2 norm, minimizing (1/2n) ||y – Xw||2 + α||w|| 22.

As opposed to the L1 penalty, the above equation is smooth at all points, and the gradient is directly proportional to the weights themselves. Higher weights will incur higher penalties, but the gradient is never exactly zero unless the weight is exactly zero. In other words, the weights are always shrunken but never reduced to exactly zero. The same value for α = 0.1 was used in both models.

Evaluation Metric

The primary evaluation metric is Mean Squared Error computed on the held-out test set: MSE = (1/n) Σ(y i −)2 . MSE was chosen because it penalizes large prediction errors quadratically, making it sensitive to the systematic mispredictions that overfitting tends to produce. Alongside the scalar MSE values, we examined the fitted coefficient vectors of each model directly, as these provide a window into the structural differences between how each regularization strategy treats the feature space

SYSTEM ARCHITECTURE AND DATA FLOW

The pipeline of the experiment is organized in a neat and sequential manner, ensuring that each step is easily replicable and modular. The operation done by each step is clear and precise, producing outputs which are passed to subsequent steps without any side effects.

Data Acquisition

The pipeline starts off by downloading the California Housing dataset from the scikit-learn datasets library, which caches the dataset locally for subsequent use. The dataset is then parsed to extract the features and labels into a feature matrix X and a label vector y, respectively, with a preliminary check confirming the correct number of rows (20,640) and columns (8).

Preprocessing Stage

The data is directly fed to the split method that splits the data in a ratio of 4:1. A StandardScaler object is then fitted only on the training data — fitting the entire dataset could lead to leakage of test data into preprocessing and thus inflate the generalization accuracy of the algorithm. The parameters of the scaler are used on both datasets, generating scaled datasets that will be used for all further model fitting.

Model Training and Evaluation

The three distinct models are built using the following classes: LinearRegression, Lasso(alpha=0.1), and Ridge(alpha=0.1), respectively. These are then fitted to the scaled train dataset, and their coefficients are printed to track how each of the three techniques assigned predictive importance to the eight features. Each technique is used to predict on the scaled test data, and its MSE value is calculated.

Visualization

Annotated bar charts are plotted based on the findings found in the dictionary, with each model's MSE being shown by a vertical bar that is labeled with its numerical value above it. Such visualization of the findings allows one to instantly comprehend how each of the models differs from the other in terms of performance, without having to interpret the numbers found in a tabular format. This bar chart is made with matplotlib 3.8.2.

Figure 1: Pipeline Design — Data Importing → Train/Testing Splitting → Feature Scaling → Model Fitting (Linear / Lasso / Ridge) →Testing on Test Set → MSE Plot

RESULTS AND DISCUSSION

Predictive Performance

The test-set MSE for all three regression models is given in Table 1. These values are not only clear, but also quite intriguing. While Linear Regression and Ridge Regression give the same result of 0.5559 MSE, Lasso Regression lags slightly behind with an MSE of 0.6796, which is 22% more compared to the first two models.

Table 1: Test-Set Mean Squared Error for All Three Models.

Regression Model	MSE (Test Set)
Linear Regression (No Penalty — Baseline)	0.5559
Lasso Regression — L1 Penalty	0.6796
Ridge Regression — L2 Penalty	**0.5559**

It would be incorrect to conclude from the above comparison that Ridge does nothing because Ridge regularization allows achieving the same result at the same time reducing the magnitude of coefficients. In other words, Ridge achieves results of the unconstrained model with fewer efforts, which is the main goal of a well-developed regularization procedure. Hence, the effect of Ridge regularization may be regarded as an insurance policy. In this specific case, Ridge did not cause any harm to predictions because of the large size of the dataset, but when working with smaller noisy data, the introduced constraints will surely help improve performance.

The situation with Lasso's higher MSE is more complex, and it is important to explain it correctly. First, it should be noted that eight features selected for modeling in the California Housing dataset are based on well-known factors that influence property prices such as income level, geographic location, housing density, and age. When Lasso drops any feature(s), the algorithm eliminates valuable signal rather than unimportant noise, and the simplicity achieved becomes the sacrifice of predictive power.

Coefficient Structure Analysis

The disparities in how the three algorithms assign the weightage to the features are equally insightful, compared to the aggregated MSE values. The OLS baseline algorithm gives the maximum absolute weights to the variables – MedInc, Latitude, and Longitude, which is consistent with what can be expected from California real estate given that geography and income levels of households determine price. Coefficients are unrestricted and hence the algorithm reflects the correlations between the variables as they exist in the dataset.

The ridge regression model generates a set of coefficients in which MedInc, Latitude, and Longitude are again the most significant features; however, with each coefficient being smaller than before. None of the features have a zero coefficient value implying that all of the eight features are contributing towards prediction and thus the model is well-suited for use with the California Housing dataset. Coefficient vector of the Lasso model exhibits an entirely contrasting behavior. At least one of those attributes, which do not individually dominate the predictions, is reduced to zero. The surviving attributes in the Lasso model are allocated higher weights as compared to their Ridge equivalents since the attributes have to take up the responsibilities of other attributes that have been eliminated. It is this very reason that leads to an increase in the MSE of the test dataset.

Visualization and Comparative Insights

The bar chart produced at the conclusion of the pipeline provides a clean summary of the performance results. Ridge and the baseline share equal bar heights, while Lasso's bar rises noticeably higher — a visual reminder that, in this context, eliminating features does not translate into better generalization. The annotations on each bar make the numerical differences precise, supporting rapid and accurate communication of comparative results. [Figure 2: MSE Comparison Bar Chart — Linear Regression: 0.5559 | Lasso (L1): 0.6796 | Ridge (L2): 0.5559. Ridge achieves the lowest regularized MSE while Lasso incurs a performance cost due to feature elimination.

Validation Test Cases

To make sure that the pipeline was working as expected throughout all the steps, we performed several validation tests, each focused on a different aspect of the pipeline. Table 2 below outlines all the tests conducted, together with their results. In each case, the test passed successfully. Specifically, Test Case 03 made sure that Lasso did actually set at least one coefficient to zero, rather than just getting close to zero. Test Case 04 made sure that Ridge shrinks coefficients without setting them to zero.

ID	Stage	Expected Outcome	Result
TC-01	Data Ingestion	Feature matrix loads as 16,512 × 8; target vector confirmed	Pass
TC-02	Feature Scaling	Post-scaling: per-feature mean □ 0, std □ 1	Pass
TC-03	L1 Model Fit	Lasso converges; at least one coefficient reduced exactly to zero	Pass
TC-04	L2 Model Fit	Ridge converges; all coefficients shrunk but none zeroed	Pass
TC-05	MSE Computation	Test-set MSE calculated correctly for all three models	Pass
TC-06	Visualization	Bar chart renders with annotated MSE values per model	Pass

Broader Interpretation

In sum, the foregoing illustrates an important point, which may sometimes be overlooked in a basic discussion of regularization: the optimal regularization technique is not necessarily the one with the greatest elegance or performance in an academic setting — rather, it is the technique that has the assumptions closest to reality with regards to the data it receives. Lasso relies on sparsity; ridge works from the assumption that all variables contribute, but to a lesser degree. Determining which method to use requires computational research, but it also calls for domain expertise and knowledge.

This exercise also illustrates an important methodology issue when evaluating ML models. MSE is an important metric that serves an essential function, but it can only tell part of the story. It lets us know that Lasso was less effective at regression than ridge, but it cannot tell us why, which required our parameter analysis. When evaluating a regression model, one must not limit their evaluation to its accuracy or its test set MSE, but also perform an analysis of its parameter structure.

CONCLUSION

In conducting this experiment, we aimed to understand how regularization of L1 and L2 affects the performance of the linear regression model on the California Housing dataset. The experiment, however, was not limited to analyzing the MSE values – it also yielded interesting insights into the interaction between assumptions behind different kinds of regularization and the nature of the dataset.

Our experiment shows that Ridge Regression performed better in terms of MSE (the MSE being 0.5559, the same value as in the unconstrained case) in comparison with the other techniques, owing to the high compatibility between the underlying assumptions and the structure of the dataset: all eight predictors of California Housing are relevant, and, thus, a model based on the principle of equal shrinking performs best. On the contrary, Lasso could not produce accurate results, since its tendency to eliminate redundant predictors led to the removal of some meaningful variables, resulting in an MSE value of 0.6796. It would be incorrect to state that our experiment discredits the use of L1 regularization as such. Lasso is a highly efficient method that outperforms Ridge Regression in cases when sparsity assumption is true.

Another objective of this paper is to show the importance of going beyond mere metrics when evaluating the performance of regression models. The structural differences between two models that we found through analyzing the distributions of their coefficients were hidden by numbers in the MSE analysis alone. It is important to know what a model learns — on which features it focuses, how much weight it gives to them, and how it has been regularized

FUTURE WORK

Several obvious avenues for further research can be identified. It would be quite relevant to compare the performances of the Linear Regression with Elastic Net regularization because it allows us to adjust the ratio of L1 and L2 penalties so as to capture the best qualities of the two types of regularization in case of data falling between fully sparse and fully dense feature relevancy. Additionally, performing a hyperparameter tuning by means of the α parameter grid search using k-fold cross-validation technique would be a way more fair approach compared to the fixed value used in this analysis, and would enable us to obtain more insights regarding the performance differences between the methods in question. It would be also useful to extend this research on other regression tasks with different levels of feature correlation and data dimensionality in order to establish some empiric heuristics for choosing one approach over another depending on the nature of the problem. Lastly, it would be interesting to see whether our findings are consistent across other non-linear models (such as Kernel ridge regression or regularized polynomial model, or even Weighted decay in case of neural nets).

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Ramya B.N. for the valuable guidance that she provided during the course of this research project. She has been instrumental not only in improving the technical direction but also in providing constructive feedback on the content of the paper. Her willingness to go through the minute details of the paper has made a significant contribution towards the final output of this paper.

REFERENCES

1. Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.
2. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55–67.

3. Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. Statistics & Probability Letters, 33(3), 291–297.
4. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.
6. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B, 67(2), 301–320.
7. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
8. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.
9. Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362.
10. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95