Comparative Analysis of Multiple Linear Regression with L1 and L2 Regularization for Stock Price Prediction

Paras Mishra
Department of Computer Science
Tribhuvan University, Kathmandu, Nepal.
parasmishra428@gmail.com

Ranjan Pandey
Patan Multiple Campus
Tribhuvan University, Patandhoka, Nepal.
ranjan24.747402@pmc.edu.np

Received: 16 March, 2025 / Accepted: 8 June, 2025 / Published: 9 July, 2025

Abstract

This research explores the application of L1 (Lasso) and L2 (Ridge) regularization techniques within multiple linear regression frameworks for predicting stock prices in the ACLBSL segment of the Nepal Stock Exchange (NEPSE). A key challenge in stock price prediction over fitting is addressed by incorporating regularization methods that penalize model complexity. Through hyper parameter tuning, optimal alpha values of 0.9541 for Lasso and 0.4715 for Ridge were identified. These values led to improved model performance, reducing Mean Squared Error (MSE) to 514.12 and 521.02, respectively.

The study's findings reveal that Lasso regression not only enhances prediction accuracy but also performs effective feature selection by shrinking less significant coefficients to zero. This enables a more interpretable and simplified model without sacrificing performance. In contrast, Ridge regression retains all features with reduced coefficient magnitudes. The results indicate that Lasso regression is more effective in identifying and leveraging key predictors, thereby providing better generalization to unseen stock price data.

This research contributes to the ongoing efforts in financial modeling by demonstrating that regularization techniques can substantially improve the robustness and reliability of predictive models in the context of NEPSE, providing valuable insights for investors and analysts.

Keywords—Stock Price Prediction, Lasso Regression, Ridge Regression, Regularization, NEPSE, Multiple Linear Regression.

Introduction

Stock price prediction is a challenging task with significant implications for financial markets and investors. Various machine learning models have been applied to this problem, each with its advantages and drawbacks. This research discusses the relevant studies and research that have contributed to the understanding of stock price prediction techniques. This research addresses the problem of optimizing regularization parameters for Lasso and Ridge regression models to enhance the predictive performance of stock prices and to identify the optimal alpha values for L1 and L2 regularizations that yield the lowest prediction errors and highest R-squared scores, thereby providing a reliable model for stock price prediction. The findings from this research will guide the application of regularization techniques in financial modeling and contribute to the development of robust predictive models in the stock market domain.

This research introduces the fundamental concept of multiple linear regressions, a powerful tool for modelling the relationships between dependent and independent variables. The results offer insights into the significance of various factors and their influence on stock prices. Performance of L1 and L2 regularization has been accessed with hyper parameter optimization for the ACLBSL dataset. L1 Regularization Adds the absolute values of the coefficients to the loss function, which can shrink some coefficients to zero, effectively performing feature selection and simplifying the model. L2 Regularization Adds the squared values of the coefficients to the loss function, which discourages large coefficients but does not shrink any coefficients to zero, thus keeping all features but with reduced impact.

The main contributions of this research are as follows:

1. **Comparative Evaluation of Regularization Techniques**: This study performs a comparative analysis of L1 (Lasso) and L2 (Ridge) regularization techniques within the framework of Multiple Linear Regression (MLR) for stock price prediction, focusing on their respective strengths in feature selection and model stability.

2. **Hyper parameter Optimization for Predictive Accuracy**: The research applies systematic hyper parameter tuning to identify optimal alpha values for L1 and L2 regularizations, resulting in models with minimized prediction error and maximized R-squared values.

3. **Application to Real-World Financial Data**: Using the ACLBSL stock dataset, this work demonstrates the practical application of regularization techniques to enhance predictive performance in a real-world financial context.

4. **Insights into Feature Importance**: Through L1 regularization, the study highlights the most influential predictors affecting stock prices, offering insights that are valuable for financial analysts and investors.

5. **Contribution to Robust Financial Modeling**: The findings contribute to the broader field of financial modeling by illustrating how regularization can improve generalization, reduce overfitting, and increase the interpretability of stock price prediction models.

Problem Statement

The prediction of stock prices is a complex task due to the volatile nature of financial markets. Over fitting is a common issue in machine learning models, where the model performs well on training data but poorly on unseen test data. Regularization techniques such as L1 (Lasso) and L2 (Ridge) are employed to mitigate over fitting and improve model generalization. This occurs when a model is too complex and captures not only the underlying patterns in the training data but also the noise. This results in high accuracy on the training data but poor generalization to new, unseen data. Regularization techniques like L1 and L2 add a penalty to the loss function based on the magnitude of the coefficients, which discourages overly complex models and helps to prevent over fitting.

L1 and L2 regularization are effective for preventing over fitting by penalizing large coefficients and promoting simpler models. However, they do not directly address under fitting; instead, other strategies must be employed to increase model complexity and improve fit to the data.

Objectives

The objectives of this study is to implement multiple linear regression and evaluate the effectiveness of L1 (Lasso) and L2 (Ridge) regularization techniques by identifying optimal regularization parameters (α).

Background Study

Stock price prediction is a significant area of interest in both academic research and financial practice, driven by its potential economic impact. Traditional approaches like fundamental and technical analysis have been foundational but struggle to capture the dynamic and nonlinear nature of financial markets. The Efficient Market Hypothesis (EMH) historically guided these methods, positing that stock prices follow a random walk and are thus unpredictable. To address these limitations, modern research increasingly adopts machine learning (ML) techniques, leveraging their ability to handle complex patterns in data. Regularization methods such as L1 (Lasso) and L2 (Ridge) are integral in enhancing the robustness and generalization of ML models. L1 regularization aids in feature selection by shrinking coefficients to zero, whereas L2 regularization penalizes large coefficients, mitigating over fitting risks.

This research focuses on applying multiple linear regression (MLR) with L1 and L2 regularization to predict stock prices. Findings demonstrate that optimizing these regularization parameters significantly improves model performance by reducing over fitting and enhancing predictive accuracy. By integrating diverse data sources and sophisticated algorithms, this approach holds promise for advancing stock price prediction in financial analytics.

Literature Review

Stock price prediction has long been a central challenge in financial modeling due to its inherently noisy, nonlinear, and volatile nature. Traditional econometric models, such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH), have been foundational in financial time series forecasting (Hamilton, 1994). However, these models often struggle with capturing nonlinear patterns present in stock market data.

With the rise of machine learning, more advanced models such as decision trees, support vector machines, and ensemble techniques like Random Forests have gained popularity for financial prediction tasks (Yang et al., 2014; Aldridge, 2010). Recent studies have demonstrated that integrating regularization techniques into regression-based models significantly enhances their generalization capabilities by mitigating over fitting.

Saud and Shakya (2021) explored the effects of the L2 regularization parameter in Ridge regression for stock price prediction using historical data. Their findings indicate that the appropriate selection of the regularization parameter significantly improves the model's ability to generalize on unseen data. Similarly, Uniejewski (2024) examined the use of regularization in electricity price forecasting and concluded that both L1 and L2 techniques help control model complexity and enhance predictive stability, particularly in volatile domains like energy and finance.

Moreover, Jia, Anaissi, and Suleiman (2023) introduced a novel deep learning model incorporating regularization layers to forecast stock prices. Their ResNLS model demonstrates how advanced neural architectures, when combined with regularization, can capture complex temporal dependencies in financial datasets. Extending this line of inquiry, Sarkar and Vadivu (2025) proposed an ensemble deep learning framework utilizing Variational Autoencoders (VAE), Transformer, and LSTM networks. The study highlighted that regularization within deep architectures not only prevents over fitting but also contributes to more robust and interpretable financial models.

These studies collectively support the application of L1 (Lasso) and L2 (Ridge) regularization in financial prediction models. L1 regularization is especially effective for feature selection by driving some coefficients to zero, simplifying the model (Wang et al., 2017). On the other hand, L2 regularization provides enhanced stability by shrinking all coefficients uniformly (Xu & Li, 2019). The growing consensus in the literature advocates for hyperparameter tuning to identify optimal regularization strengths, as this significantly affects model performance in terms of both accuracy and robustness.

Methodology

Stock prices are influenced by numerous market factors such as trading volume, price fluctuations, and transaction frequency. In real-world financial datasets like NEPSE's ACLBSL data, these features often exhibit multicollinearity—where independent variables are highly correlated—which can reduce the reliability of predictions in standard Multiple Linear Regression (MLR) models. For instance, in our dataset, "Total Traded Amount" and "Total Traded Shares" tend to move together, leading to unstable coefficient estimates and poor generalization on unseen data.

To address this, the proposed method applies **L1 (Lasso)** and **L2 (Ridge)** regularization to the MLR model, which helps mitigate over fitting and multicollinearity. **Lasso regression** is particularly effective in selecting only the most relevant predictors by shrinking some coefficients to zero, thus simplifying the model. **Ridge regression**, on the other hand, distributes the penalty across all coefficients and is better suited when all input features are potentially useful but suffer from collinearity. In this research, the optimal alpha values for L1 and L2 regularization were determined using **GridSearchCV**, enabling a data-driven approach to tuning.

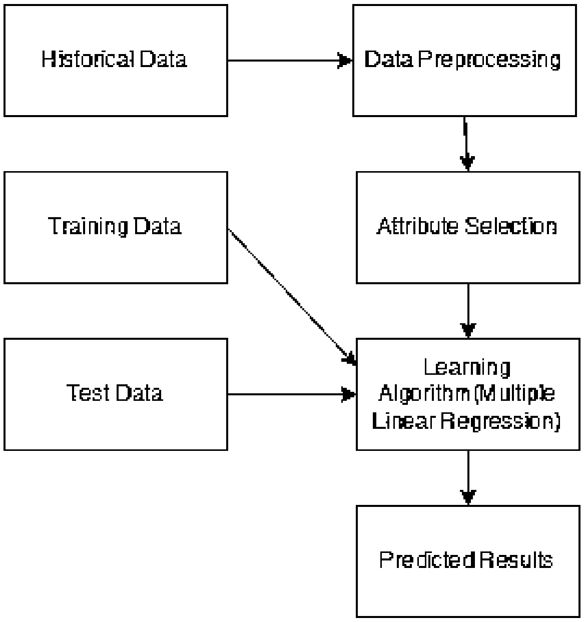


Figure 1: Flowchart detailing the methodology for stock price prediction.

Data Collection

Historical stock data from the Nepal Stock Exchange (NEPSE) is collected for the study. The dataset used during this study is ACLBSL dataset collected from web scrapping and the dataset used for this study is loaded and cleaned to ensure there are no missing or inconsistent values. The independent variables and the dependent variable are identified and separated. For this study, the features include ’Total Transactions’, ’Total Traded Shares’, ’Total Traded Amount’, ’Max. Price’, and ’Min. Price’, while the target variable is ’Close Price’. Instances of two year have been captured for the purpose of analysis from 2019-01-07 to 2021-12-29. The simple architecture of dataset used for analysis is shown below;

Data Preprocessing

This step involves cleaning the data, handling missing values, and normalizing the data to ensure consistency. The methodology for this research follows a structured process as depicted in Figure above.

Attribute Selection

Relevant features, such as total transactions, total traded shares, total traded amount, maximum price, and minimum price, are selected for the analysis. After selecting relevant attributes, the dataset is divided into training and testing sets to validate the model performance. Here, 80

Learning Algorithm

To ensure that the features are on a similar scale, feature scaling is performed. This Multiple Linear Regression (MLR) A Multiple Linear Regression model is trained on the training data. MLR is a simple and interpretable model that helps establish a baseline for the prediction of stock prices. The model coefficients and intercept are extracted to understand the relationship between the features and the target variable. The equation for the regression model is given by:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Where: - y is the dependent variable (closing price). - β_0 is the intercept. - $\beta_1,\beta_2,...,\beta_n$ are the coefficients of different independent variables.

The independent variables in this model include: - Total number of transactions - Total traded shares - Total traded amount - Maximum price - Minimum price Thus, the closing price acts as the dependent variable.

Lasso Regularization

L1 regularization, also known as Lasso (Least Absolute Shrinkage and Selection Operator), is applied to the linear regression model to handle over fitting by penalizing the absolute values of the coefficients. This tends to produce sparse models with fewer non-zero coefficients, effectively performing feature selection.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\text{Cost}(h_\theta(x^i), y^i) + \frac{\lambda}{m}\sum_{j=1}^{n}|\theta_j|\right)$$

Ridge Regularization L2 regularization, or Ridge regression, penalizes the squared values of the coefficients. Unlike Lasso, Ridge regression does not enforce sparsity but can handle collinearity among features more effectively.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\text{Cost}(h_\theta(x^i), y^i) + \frac{\lambda}{m}\sum_{j=1}^{n}\theta_j^2\right)$$

Multiple linear regression models with L1 (Lasso) and L2 (Ridge) regularization are applied. The optimal values for the regularization parameters (α) are determined using GridSearchCV.

Error Calculation

The models are evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Rsquared (R^2) to compare the performance before and after regularization.

Mean Absolute Error (MAE) :

MAE measures the average magnitude of the errors between predicted and actual values, without considering their direction. It provides a straightforward measure of prediction accuracy. The formula for calculating mean absolute error is shown below.

The Mean Absolute Error (MAE) is calculated using the following formula:

$$\text{Mean Absolute Error (MAE)} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where: - y_i are the actual values, - ˆy_i are the predicted values, - n is the number of observations.

Mean Squared Error (MSE): MSE measures the average of the squares of the errors between predicted and actual values. It gives a higher weight to larger errors, thus emphasizing the significance of significant deviations.

The Mean Squared Error (MSE) is calculated using the following formula:

$$\text{Mean Squared Error (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where: - y_i are the actual values, - ˆy_i are the predicted values, - n is the number of observations. Root Mean Squared Error (RMSE): RMSE is the square root of the MSE. It provides an error metric on the same scale as the original data, making it more interpretable in the context of the data.

Root Mean Squared Error (RMSE) =

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

R-squared (R^2) Score:

R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides an indication of the goodness of fit of the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where: - y_i are the actual values, yˆ$_i$ are the predicted values, - ¯y is the mean of the actual values, - n is the number of observations

The performance of each model (MLR, Lasso, and Ridge) is evaluated using Mean Absolute

Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) metrics on the testing data. These metrics provide insights into the accuracy and generalization ability of the models.

Predicted Results

The models’ predictions are compared to actual stock prices to assess their accuracy and reliability. This approach ensures a systematic and rigorous analysis of the stock price prediction models, aiming to reduce over fitting and improve generalization.

Hyper Parameter Tuning

GridSearchCV is employed to perform hyper parameter tuning for both Lasso and Ridge regression models. This involves specifying a range of alpha values and using cross-validation to determine the best alpha that minimizes the error metrics.

Implementation Details

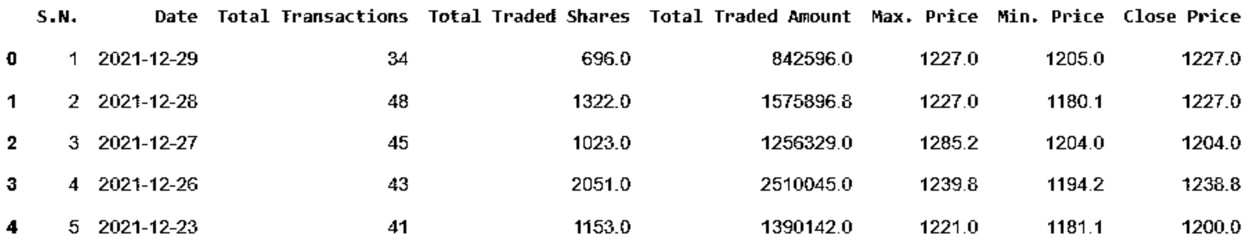
The implementation of this study involved several steps, using Python and relevant libraries such as pandas, scikit-learn, and matplotlib. The detailed description of activities performed at different phases are explained below:

Data Preparation

First, the dataset has been loaded and extracted the relevant and the target variable. Split the data into training and testing sets. The dataset is split into training and testing sets using an 80-20 ratio. This ensures that the model can be trained on one portion of the data and evaluated on another to assess its performance on unseen data.

```
from sklearn.linear_model import Linear Regression, Lasso, Ridge
# Multiple Linear Regression
model = Linear Regression()
model.fit(X_train, y_train)
# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
# Ridge Regression
ridge = Ridge(alpha=0.1)
ridge.fit(X_train, y_train)
```

The head of dataset is shown below;



```
   S.N.        Date  Total Transactions  Total Traded Shares  Total Traded Amount  Max. Price  Min. Price  Close Price
0     1  2021-12-29                  34                696.0             842596.0      1227.0      1205.0       1227.0
1     2  2021-12-28                  48               1322.0            1575896.8      1227.0      1180.1       1227.0
2     3  2021-12-27                  45               1023.0            1256329.0      1285.2      1204.0       1204.0
3     4  2021-12-26                  43               2051.0            2510045.0      1239.8      1194.2       1238.8
4     5  2021-12-23                  41               1153.0            1390142.0      1221.0      1181.1       1200.0
```

Figure 2: Structure of the Dataset

Model Development

Built and trained multiple linear regression, Lasso, and Ridge regression models. The implementation of model development using multiple linear regression, lasso and ridge regularization model is shown below.

```
from sklearn.linear_model import Linear Regression, Lasso, Ridge
# Multiple Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
# Ridge Regression
ridge = Ridge(alpha=0.1)
ridge.fit(X_train, y_train)
```

Hyper Parameter Tuning

The optimal alpha value for Lasso is determined using GridSearchCV, which tests a range of alpha values to find the one that minimizes the Mean Squared Error (MSE) on the validation set.

GridSearchCV is similarly used to find the optimal alpha value for Ridge regression, which balances the bias-variance trade-off to minimize over fitting and improve generalization.

The following Python code demonstrates how to use grid search with cross-validation to find the optimal hyper parameters for Lasso and Ridge regression using ‘scikit-learn‘:

```
# Lasso Regression Grid Search
param_grid_l1 = {'alpha': np.logspace(-4, 1, 50)}
grid_search_l1 = GridSearchCV(Lasso(), param_grid_l1, cv=5,
scoring='neg_mean_squared_error',n_jobs=-1)
grid_search_l1.fit(X_train, y_train)
best_alpha_l1= grid_search_l1.best_params_['alpha']
# Ridge Regression Grid Search
param_grid_l2 = {'alpha': np.logspace(-4, 1, 50)}
grid_search_l2 = GridSearchCV(Ridge), param_grid_l2, cv=5,
scoring='neg_mean_squared_error',n_jobs=-1)
grid_search_l2.fit(X_train, y_train)
best_alpha_l2 = grid_search_l2.best_params_['alpha']
```

Model Evaluation

The performance of the multiple linear regression model was assessed both before and after applying regularization techniques (L1 and L2). Key evaluation metrics included Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). These metrics provided a comprehensive understanding of the model’s accuracy and its ability to generalize to new data, demonstrating significant improvements post-regularization.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
from sklearn.linear_model import Lasso, Ridge
# Predict using the trained Linear Regression model
y_pred = model.predict(X_test)
# Calculate error metrics
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
# Print error metrics
print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}, R²: {r2}')
# Lasso Regression with the best alpha from grid search
lasso_best = Lasso(alpha=best_alpha_l1)
lasso_best.fit(X_train, y_train)
mse_lasso = mean_squared_error(y_test, lasso_best.predict(X_test))
# Ridge Regression with the best alpha from grid search
ridge_best = Ridge(alpha=best_alpha_l2)
ridge_best.fit(X_train, y_train)
mse_ridge = mean_squared_error(y_test, ridge_best.predict(X_test))
```

Result Analysis

The research aimed to compare the effectiveness of L1 and L2 regularization techniques on a predictive model for stock price prediction using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) score.

The regression line obtained after Multiple Linear Regression (MLR) is shown below, where a comparison of actual vs. predicted closing prices using Linear Regression is illustrated. The dots represent the predicted closing prices compared to the actual closing prices, with the line indicating the ideal prediction line. The close alignment of the points to the red line demonstrates the accuracy of the model.

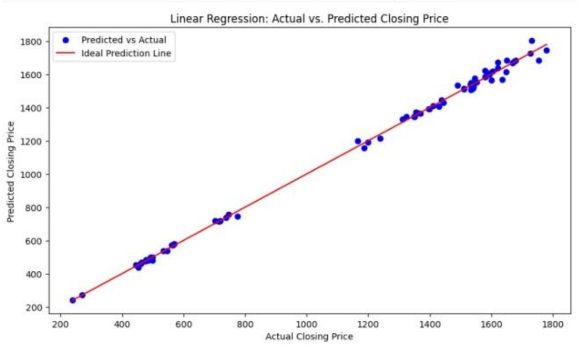
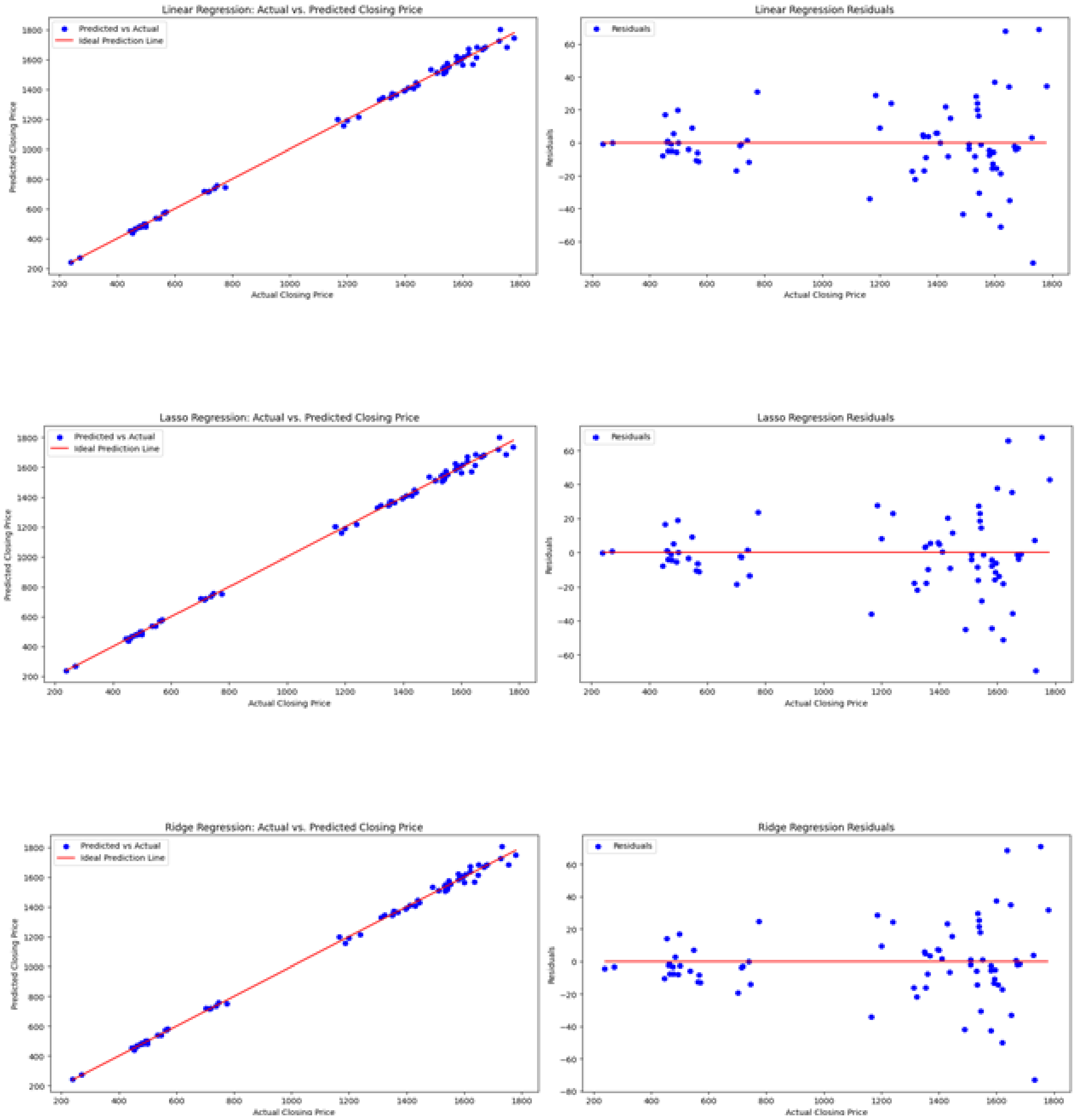


Figure 3: Linear Regression Actual vs Redicted

Stock price prediction is a complex regression task characterized by high variance and potential multicollinearity among predictors. Traditional **Multiple Linear Regression (MLR)** models can fit historical stock price data well but may suffer from overfitting when applied to unseen data, especially in high-dimensional settings. This issue often arises when the model captures noise in the training data rather than the underlying trend.

To address this limitation, **regularization techniques** such as **L1 (Lasso) and L2 (Ridge)** have been employed. These methods introduce a penalty term to the loss function that constrains the magnitude of the coefficients, thereby improving the model's generalization ability. L1 regularization encourages sparsity by shrinking some coefficients to zero, effectively performing variable selection. In contrast, L2 regularization distributes the penalty uniformly, reducing model complexity without eliminating features.

The current analysis presents a comparative evaluation of these three models—standard Linear Regression, Lasso Regression, and Ridge Regression—using actual vs. predicted plots and residual diagnostics. These visualizations provide insights into the predictive accuracy and error distribution of each model. The red lines in the scatter plots represent ideal predictions (where predicted values exactly match actual values), while residual plots highlight how errors are distributed across the prediction space.



The evaluation parameter before and after the hyper parameter optimization for L1 and L2 regularization is shown in table below:

SN	Evaluation Parameter	Before Regularization	After L1 Regularization	After L2 Regularization
1	Mean Absolute Error (MAE)	15.7273	15.5872	15.65948
2	Mean Squared Error (MSE)	523.8600	514.120	521.0197
3	Root Mean Squared Error (RMSE)	22.88798	22.6742	22.826
4	R-squared (R^2) Score	0.99773	0.99777	0.9977495

Table 1: Comparison of Evaluation Parameters before and after Regularization

L1 regularization achieved the lowest MAE, indicating better performance in minimizing the average magnitude of errors compared to L2 regularization. L1 regularization resulted in the lowest MSE, suggesting it effectively reduced the squared differences between predicted and actual values. Similar to MSE, L1 regularization led to the smallest RMSE, indicating better accuracy in predicting the stock prices with smaller deviations from the observed values. All regularization techniques maintained high R-squared scores close to 1, indicating excellent model fit and strong predictive capability.

L1 regularization generally outperformed L2 regularization in terms of MAE, MSE, and RMSE metrics, suggesting it may be more suitable for this predictive modeling task. All regularization methods maintained very high R-squared scores, indicating robust model performance and high explanatory power of the model. The optimized value of alpha after regularization for L1 regularization is 0.95409and for L2 regularization is 0.4714866, indicating that these values minimized prediction error in the stock price prediction model.

Conclusion

Regularization is a critical component in building robust machine learning models, particularly for regression tasks like stock price prediction. By carefully tuning the regularization hyper parameters, such as the alpha values for L1 and L2 regularization, we can significantly improve the model’s performance and generalization. The research provides a framework for optimizing these hyper parameters and demonstrates how to evaluate the effectiveness of each regularization technique using MSE as a performance metric.

Applying L1 and L2 regularization to the multiple linear regression models for stock price prediction resulted in improved performance metrics. Specifically, L1 regularization (Lasso) with an optimal alpha of 0.9541 yielded the lowest Mean Squared Error (MSE) of 514.12, indicating a slight improvement over the model without regularization (MSE: 523.86). The Mean Absolute Error (MAE) and R-squared (R^2) scores also showed slight enhancements with L1 regularization. Similarly, L2 regularization (Ridge) with an optimal alpha of 0.4715 also reduced the MSE to 521.02. These findings demonstrate that regularization techniques can effectively improve model performance by reducing over fitting, leading to more accurate and generalizable predictions.

Future Recommendation

It is also recommended to further explore the tuning of hyper parameters and potentially combining regularization techniques with more complex models like GRU, LSTM, or Transformer networks to see if they can improve the predictive performance even further. To enhance stock price prediction models, future work should integrate alternative data sources like social media sentiment, news feeds, and macroeconomic indicators. Advanced machine learning techniques such as RNNs, LSTMs, GRUs, and Transformer networks can be explored for capturing temporal dependencies and complex market dynamics. Implementing real-time data processing and prediction systems can support high-frequency trading strategies.

References

Aldridge, I. (2010). *High-frequency trading: A practical guide to algorithmic strategies and trading systems*. Wiley.

Hamilton, J. D. (1994). *Time series analysis*. Princeton University Press.

Jia, Y., Anaissi, A., & Suleiman, B. (2023). *ResNLS: An improved model for stock price forecasting* [arXiv preprint].

https://arxiv.org/abs/2312.01020

Saud, A. S., & Shakya, S. (2021). Analysis of L2 regularization hyper parameter for stock price prediction. *Journal of Institute of Science and Technology, 26*(1), 83–88. https://doi.org/10.3126/jist.v26i1.37830

Sarkar, A., & Vadivu, G. (2025). *An advanced ensemble deep learning framework for stock price prediction using VAE, Transformer, and LSTM model* [arXiv preprint]. https://arxiv.org/abs/2503.22192

Uniejewski, B. (2024). *Regularization for electricity price forecasting* [arXiv preprint]. https://arxiv.org/abs/2404.03968

Wang, G., Xie, C., & Li, Y. (2017). Lasso regression modeling based on stock index forecasting in high-dimensional space. *Journal of Finance and Data Science*, *3*(2), 81–91.

Xu, Y., & Li, Q. (2019). Ridge regression in stock prediction: A robust approach in volatile markets. *Financial Innovation*, *5*(1), 15–24.

Yang, H., Min, H., Lee, Y., & Kim, Y. (2014). Stock prediction model using machine learning algorithms. *International Journal of Software Engineering and Its Applications, 8*(12), 117–126.