# Stock Price Prediction

Prithivirajan D[1] Rithekha K[2] Samyuktha D[3]

[1,2,3]SSN College of Engineering

**Abstract.** Stock price prediction is a difficult but critical exercise in financial analysis because of the volatility and complexity of the market. In this research, predictive modeling of a target stock price (Stock_1) using correlated stocks (Stock_2 to Stock_5) is investigated. The data set goes through thorough preprocessing procedures, such as imputation of missing values, removal of duplicates, treatment of outliers via the IQR method, and feature scaling. Ridge, Lasso, Random Forest, and Polynomial Regression multiple regression models are used for prediction. The performance of the models is measured using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and $R^2$ Score. Feature selection is done using the SelectKBest method to select the 3 most impactful features. Also, Principal Component Analysis (PCA) is used for dimensionality reduction to measure its effect on model performance. Hyperparameter tuning methods such as GridSearchCV and RandomizedSearchCV are used to tune the Random Forest and Polynomial Regression models.

Experimental results show that ensemble learning approaches coupled with efficient feature engineering and selection are able to substantially enhance the accuracy and generalization of stock price prediction models. Of all the models, the Bagging Regressor performed the most optimally with an $R^2$ Score of 0.8526, RMSE of 0.3780, and MAE of 0.2757, followed very closely by Gradient Boosting with an $R^2$ Score of 0.8377, RMSE of 0.3967, and MAE of 0.3007. The Polynomial Regression model, although with relatively higher error rates, was still insightful in its ability to capture non-linear relationships within the stock data and indicating its potential for modeling intricate market patterns.

**Keywords:** Stock Price Prediction, Regression Models, Feature Engineering, Dimensionality Reduction, Ensemble Learning, Hyperparameter Tuning, Model Evaluation, Principal Component Analysis, Bagging Regressor, Gradient Boosting

## 1 Introduction

With today's high-speed financial world, knowing how to predict stock prices has become the most important factor for investors, analysts, and institutions. Effective stock prediction facilitates the management of risk, maximizing investment strategies, and making proper financial decisions. As global markets evolve to become increasingly volatile and information-rich, interpreting the trends and interlinkages in stock data has taken center stage.

Stock prices are affected by a broad array of variables, such as economic conditions, market mood, and worldwide events. Predictive analysis allows stakeholders to reveal insightful information from past data, detect correlations between stocks, and predict possible price movements. Such information is especially useful for optimizing portfolio performance, facilitating algorithmic trading, and enhancing financial planning.

This research is concerned with predictive modeling of future stock value using historical data of several stocks. Using data-driven analysis, this method contributes to more efficient forecast-making strategies and aids strategic financial decision-making. The following sections include an overview of existing work, system design, dataset information, methodology, results, and conclusions.

The structure of the remainder of this paper is as follows. Section 2 presents a review of existing literature on stock prediction and financial data analysis; Section 3 outlines the proposed framework and design; Section 4 introduces the dataset and tools used; Section 5 details the methodology for analysis and prediction; Section 6 discusses the results and key findings; and Section 7 concludes the study and explores directions for future work.

### 1.1 Problem Statement

This work aims to create an automated stock price forecasting system to predict the value of a target stock using historical data of related stocks. The objective is to derive useful patterns in finance data to help investors and analysts make informed choices, reduce risk, and improve portfolio management through correct and data-based forecasts.

## 2 Related Works

Stock price forecasting has been a popular area of research in financial data analysis using statistical techniques, machine learning, and ensemble learning methods. Several regression models, dimension reduction methods, and hyperparameter tuning techniques have been studied by researchers to improve prediction accuracy and decision-making in finance.

A research [1] explored the utility of ensemble models for predicting stock prices, focusing on the application of feature selection and dimensionality reduction to manage noisy financial data. The study employed models such as Random Forest and Gradient Boosting with PCA to enhance performance, citing the need to balance preprocessing methods with robust base learners.

Another research [2] concentrated on a comparison of regression algorithms such as Ridge, Lasso, and Bagging Regressor to predict future stock prices based on various market variables. It applied correlation analysis and cross-validation methods for selecting the best features. The research concluded that ensemble methods are better than conventional models in terms of generality and stability.

Research [3] also illustrated the advantage of hyperparameter tuning using GridSearchCV and RandomizedSearchCV, especially in Random Forest-based models, to enhance predictive performance. The experiments indicated the increase in $R^2$ score, MAE, and RMSE when models are optimized and complemented with powerful data preprocessing techniques such as outlier detection and scaling.

## 3 Proposed System

The proposed system adopts a sturdy machine learning pipeline to predict stock prices, starting with thorough Exploratory Data Analysis (EDA) in order to reveal trends, correlations, and missing values within the dataset. The dataset encompasses various stock indicators, with Stock_1 being considered as the target variable. The preprocessing procedure includes handling missing values, feature scaling with StandardScaler, and encoding categorical variables where required in order to have a clean and uniform dataset to model from.

After preprocessing, the system utilizes feature selection and dimensionality reduction methods like Principal Component Analysis (PCA) to reduce redundancy, eliminate noise, and enhance computational efficiency. Correlation analysis is performed to find and keep the most important predictors. This prevents multicollinearity and enhances model generalization.

The pipeline then proceeds to train various regression models, such as Ridge Regression, Lasso Regression, Random Forest Regressor, and Polynomial Regression, to facilitate a complete comparison among linear, non-linear, and ensemble-based methods. Polynomial Regression, specifically, is used to model intricate, non-linear relationships between stock variables, which can possibly not be modeled efficiently using just linear methods.

Following initial training of the models, the model utilizes Hyperparameter Tuning through GridSearchCV and RandomizedSearchCV to fine-tune parameters such as regularization strength, tree depth, polynomial degree, and learning rates, enhancing performance metrics such as Mean Squared Error (MSE) and $R^2$ score. For enhanced prediction accuracy and model stability, ensemble learning algorithms such as Bagging Regressor, AdaBoost, and Gradient Boosting Regressor are utilized. These ensemble models harness the strengths of the individual base learners to achieve better performance and avoid overfitting. Figure 1 presents the architecture of the proposed stock prediction workflow.

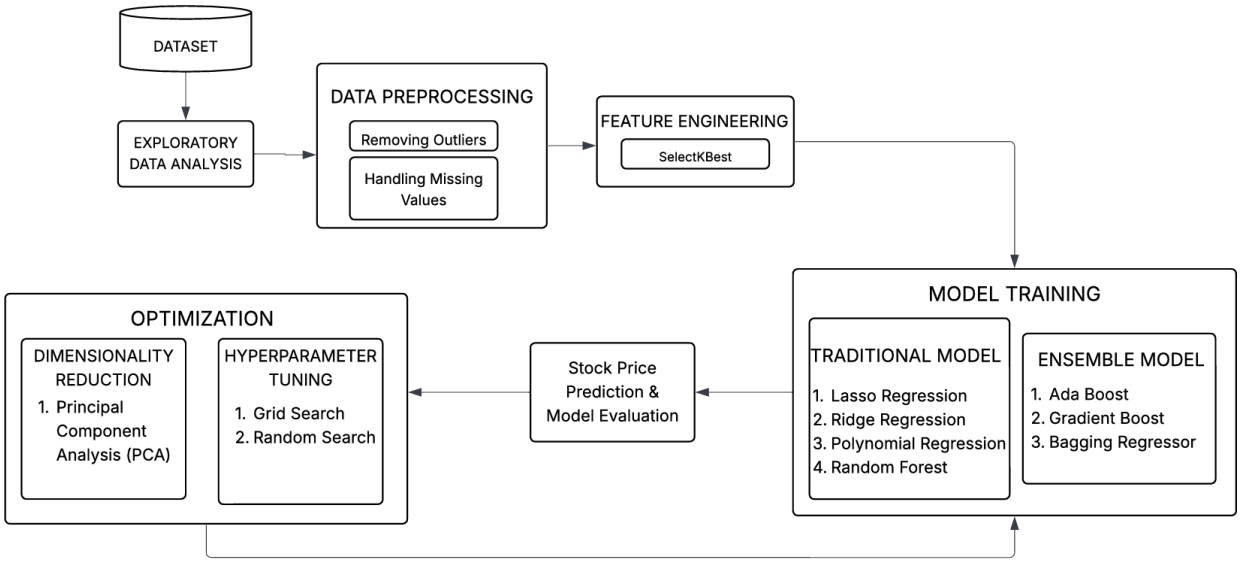



**Fig. 1.** Architecture Diagram

## 4 Implementation

### 4.1 Development Environment

The stock price forecasting system is implemented using Python, based on widely used data science and machine learning packages like Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn. All the implementation is done in Google Colab, which provides a solid environment for data analysis, visualization, model training, and performance assessment.

### 4.2 Dataset Description

The data used here is stock prices for several stocks, and the goal is to forecast Stock_1 from the values of Stock_2 through Stock_5. There are 365 rows with 6 columns in the dataset, indicating daily closing price for the concerned stocks. Features are defined below:

- Date: The date value associated with every stock value entry.
- Stock_1: The target stock for which the price needs to be forecasted.
- Stock_2 to Stock_5: Corresponding stock prices utilized as input features for prediction

First 5 rows of the dataset:

| | Unnamed: 0 | Stock_1 | Stock_2 | Stock_3 | Stock_4 | Stock_5 |
|---|---|---|---|---|---|---|
| 0 | 2020-01-01 | 101.764052 | 100.160928 | 99.494642 | 99.909756 | 101.761266 |
| 1 | 2020-01-02 | 102.171269 | 99.969968 | 98.682973 | 100.640755 | 102.528643 |
| 2 | 2020-01-03 | 103.171258 | 99.575237 | 98.182139 | 100.574847 | 101.887811 |
| 3 | 2020-01-04 | 105.483215 | 99.308641 | 97.149381 | 100.925017 | 101.490049 |
| 4 | 2020-01-05 | 107.453175 | 98.188428 | 99.575396 | 101.594411 | 101.604283 |

Descriptive Statistics:

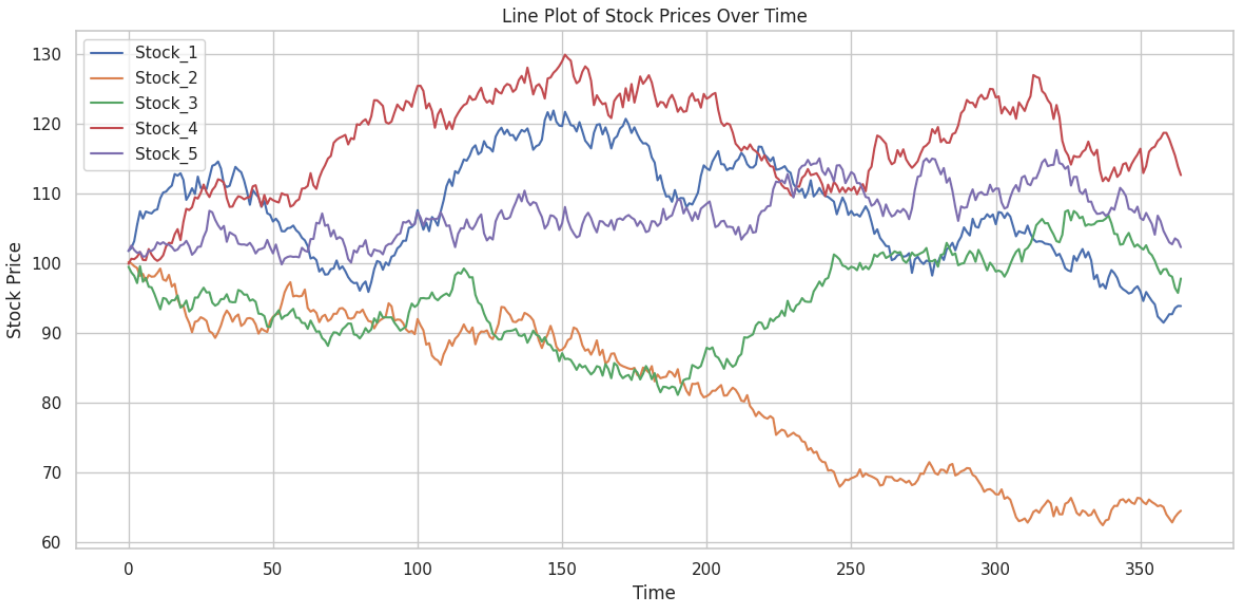
| | Stock_1 | Stock_2 | Stock_3 | Stock_4 | Stock_5 |
|---|---|---|---|---|---|
| count | 365.000000 | 365.000000 | 365.000000 | 365.000000 | 365.000000 |
| mean | 107.772577 | 81.105216 | 94.519502 | 117.407560 | 106.866865 |
| std | 7.398296 | 11.435212 | 6.519213 | 6.778527 | 3.760968 |
| min | 91.474442 | 62.414219 | 81.111434 | 99.909756 | 99.833309 |
| 25% | 101.603117 | 69.328263 | 89.788068 | 112.209912 | 103.927072 |
| 50% | 107.421299 | 84.283525 | 94.495546 | 117.788079 | 106.411328 |
| 75% | 113.741728 | 91.548859 | 99.919465 | 123.132365 | 109.178007 |
| max | 121.901773 | 100.160928 | 107.588373 | 129.911386 | 116.243803 |

**Fig. 2.** Dataset Sample

## 5 Methodology

### 5.1 Data Preprocessing and Exploratory Data Analysis

The exploratory data analysis (EDA) was directed towards revealing patterns and relationships between the various stock features. Correlation matrices, line plots, and distribution plots were employed to gain insight into dependencies between Stock_1 (the target) and the remaining stocks (Stock_2 to Stock_5). Time series plots were employed to visualize stock trends and movements over time, giving insights into possible co-movements and lag effects.



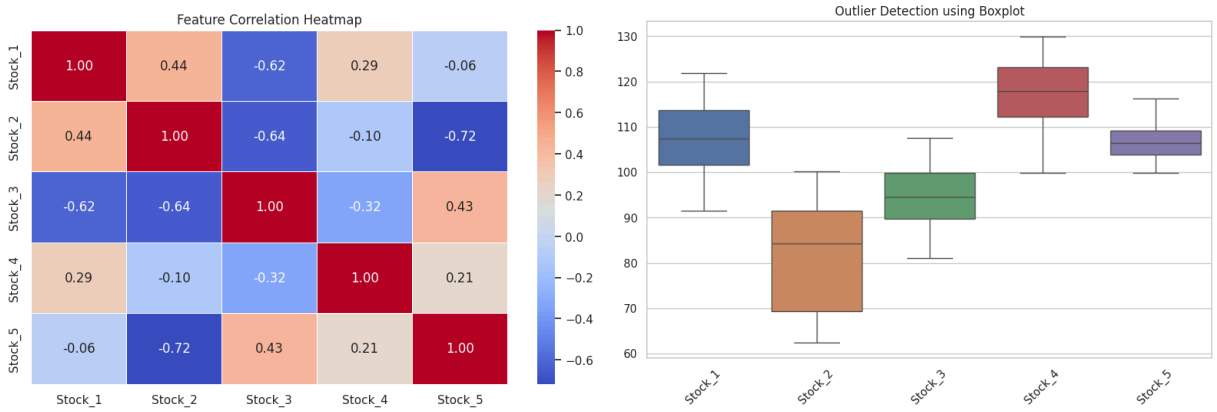



**Fig. 3.** Exploratory Data Analysis

In preprocessing, irrelevant or redundant columns, if any, were dropped to make the dataset more efficient. Null values were verified and treated accordingly to maintain data integrity. The features were standardized using MinMaxScaler to normalize the values so that all input features have an equal contribution to the performance of the model.

For better model performance and identification of the most indicative predictors, the feature selection process was carried out with the SelectKBest procedure for two alternative scoring functions: f_regression and mutual_info_regression. The f_regression score ranks features based on linear correlation of each feature against the target feature (Stock_1) and selects features demonstrating the highest linear association. Instead, mutual_info_regression quantifies the mutual dependency among variables by assessing both linear and non-linear relations. By using both techniques, the model can recognize and store the most informative features and eliminate the less important ones, thus enhancing prediction accuracy, minimizing computational complexity, and reducing the risk of overfitting.

### 5.2 Model Building

The problem of predicting the stock price is addressed via traditional regression techniques that seek to model the input feature relationships (Stock_2 to Stock_5) against the target (Stock_1). In this study, four popular models — Ridge Regression, Lasso Regression, Polynomial Regression, and Random Forest Regressor — each with different abilities in modeling with multiple predictors over financial data are implemented and contrasted.

Ridge Regression is a linear model that uses L2 regularization to minimize the effect of multicollinearity and prevent overfitting. By adding a penalty that is proportional to the model coefficients squared, Ridge compresses the value of all coefficients, retaining all input features while generalizing better. Ridge works best when multiple features have an added effect on the output.

Lasso Regression, however, employs L1 regularization and forces some of the coefficients to zero, performing implicit feature selection. This is what makes this model more interpretable and simple, particularly for scenarios where not many input features significantly impact the output variable. Lasso would work best on high-dimensional data that may include irrelevant predictors.

Polynomial Regression generalizes linear models by including non-linear functions of the input features, enabling the model to model intricate, non-linear associations between independent variables and the target stock price. A

second-degree Polynomial Regression model (degree=2) is utilized in this research to add interaction terms and squared features. This enables the model to model curves instead of straight lines, which is greatly advantageous in finance data where non-linear trends are often encountered.

Random Forest Regressor is a form of ensemble learning algorithm that creates many decision trees and takes the average of their predictions to enhance accuracy and minimize variance. It performs well in identifying non-linear relationships and prevents overfitting by using bootstrap aggregation (bagging). It also offers feature importance scores, providing information on which of the input variables significantly influence stock price predictions.

All four models are trained and tested on the preprocessed data set. Ridge and Lasso models are hyper-tuned with appropriate regularization parameters (α), the Polynomial Regression is built with degree=2, and the Random Forest is built with 100 estimators and a maximum depth of 5 to find an optimal bias-variance balance.

To maintain robustness and prevent overfitting, 5-fold cross-validation is applied. The method splits the training data into five subsets and rotates the validation fold, thus yielding a good estimate of model generalization.

Table 1 is a summary of the models' training performance, while Table 2 presents the results of cross-validation. The performance is measured based on Mean Squared Error (MSE) and $R^2$ score, which together provide the model's predictive power and variance explanation capability in the target variable. Based on such assessments, the best model can be chosen or optimized further to yield accurate, stable, and scalable stock price predictions, enabling effective financial planning and investment decision-making.

**Table. 1.** Training Performance

| Machine Learning Model | MAE | MSE | RMSE | R2 |
|---|---|---|---|---|
| **Ridge Regression** | 4.472881 | 31.631981 | 5.624232 | 0.418273 |
| **Lasso Regression** | 4.469921 | 31.631806 | 5.624216 | 0.418276 |
| **Polynomial Regression** | 3.710128 | 19.415754 | 4.406331 | 0.642935 |
| **Random Forest** | 1.326789 | 3.181671 | 1.783724 | 0.941488 |

**Table. 2.** Cross Validation Performance

| Machine Learning Model | MAE | MSE | RMSE | R2 |
|---|---|---|---|---|
| **Ridge Regression** | 4.519955 | 32.396276 | 5.691773 | 0.399729 |
| **Lasso Regression** | 4.516523 | 32.475797 | 5.698754 | 0.398464 |
| **Polynomial Regression** | 3.901154 | 21.366670 | 4.622410 | 0.604435 |
| **Random Forest** | 2.265467 | 9.016543 | 3.002756 | 0.832561 |

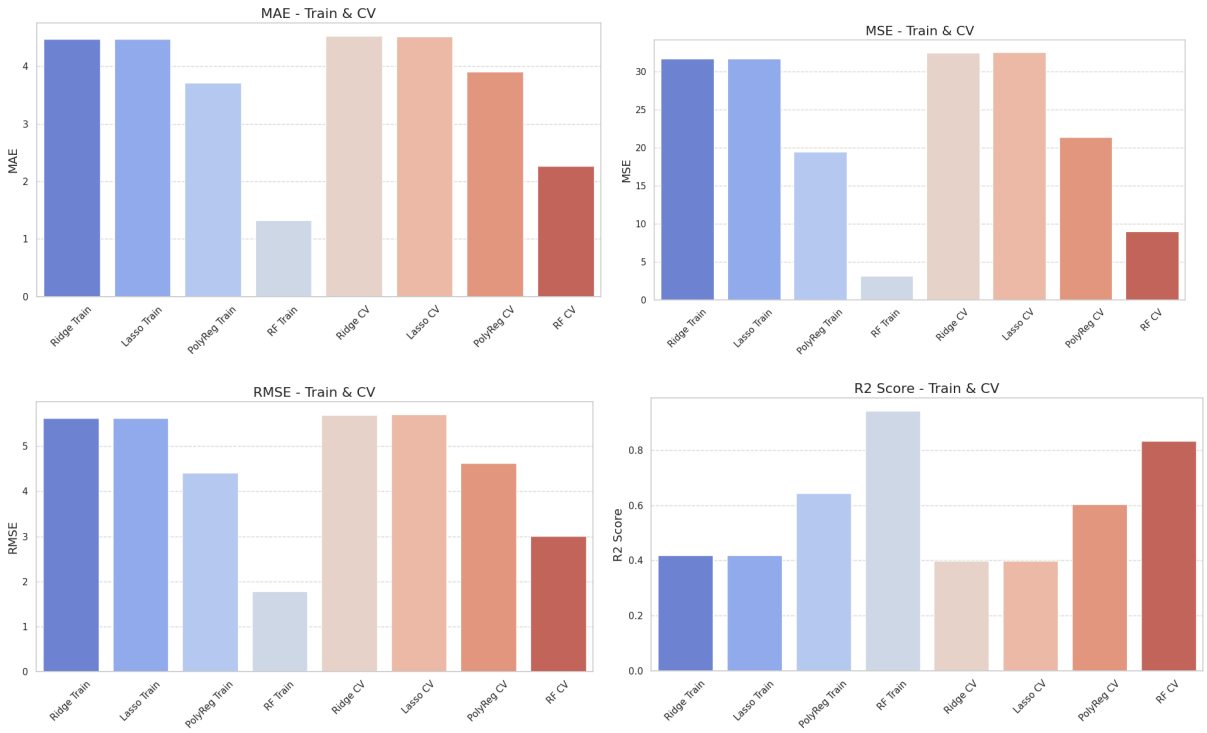



**Fig. 4.** Train and Cross-Validation Results

### 5.3 Model Testing and Performance Metrics

Having built and trained the selected regression models, i.e., Ridge Regression, Lasso Regression, Polynomial Regression, and Random Forest Regressor, the next imperative step is to test their performance on the unseen test set to study their generalization capability for forecasting actual stock prices. Each model's performance is compared in terms of conventional regression performance metrics that measure the accuracy with which they forecast Stock_1 based on Stock_2 through Stock_5.

The test set serves as a representative proxy for future data, and measuring performance on this split enables us to judge how well every model should perform under real-life market conditions. Among the models tested, Ridge Regression worked consistently well since it deals well with multicollinearity, and Lasso Regression demonstrated its feature selection capability by shrinking less significant features to zero. Polynomial Regression, by adding non-linear terms to the model, allowed the capture of some curved or complicated relationships between the independent variables and Stock_1, which may not be captured by linear models. Care was, however, taken not to overfit by choosing an appropriate degree for the polynomial. Random Forest, being a non-linear ensemble model, identified fine-grained dependencies and delivered stable predictive performance, especially where input variable relationships and the target were intricate and non-linear.

To evaluate model performance, the following metrics were calculated for each regression model:

- **Mean Squared Error (MSE)**: It calculates the average squared difference between actual and predicted values. A smaller MSE indicates better predictive accuracy and fewer large errors.
- **$R^2$ Score (Coefficient of Determination):** Indicates the proportion of variance of Stock_1 accounted for by the features. The higher $R^2$ score near 1.0 represents a better fit.

- **Mean Absolute Error (MAE):** Calculates the average of the absolute error between predicted and actual, yielding an interpretable error size.
- **Root Mean Squared Error (RMSE)**: Offers a standard deviation-like estimate of prediction error, with more weight given to large errors than MAE.

By comparing these figures between Ridge Regression, Lasso Regression, Polynomial Regression, and Random Forest Regressor, the best model can be selected for precise and stable stock price prediction. Further, residual analysis and feature importance scores (specifically from Random Forest) also give more insights into the accuracy and reliability of the predictions, as well as which input variables have the most significant effect on stock price movement.

### 5.4 Model Optimization

After their baseline performance has been evaluated (Ridge Regression, Lasso Regression, Polynomial Regression, Random Forest Regressor), optimization algorithms are used in order to maximize the quality of their predictions. Although dimensionality reduction methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are frequently used in machine learning processes to minimize the feature space and remove redundancy, their use in this regression-based stock prediction problem was not considered necessary. This is mainly due to the fact that the dataset had a limited number of input features (Stock_2 to Stock_5), and hence dimensionality reduction would not be beneficial. Maintaining all the original features guarantees the retention of full information essential for correct prediction of Stock_1.

Rather, the focus was solely put on model optimization and hyperparameter adjustment to optimize the performance of the models. Hyperparameter adjustment was performed using Grid Search and Random Search methods for searching a high number of combinations of parameters to determine the optimal settings that lead to model generalization and prediction accuracy.

Both these methods were used with the Random Forest Regressor as well as Polynomial Regression, excluding Ridge and Lasso Regression:

For Ridge Regression, the important hyperparameter alpha (used to control L2 regularization strength) was set so that it balanced between overfitting and underfitting nicely. In Lasso Regression, the alpha parameter was also optimized, determining the level of regularization and enabling the model to automatically select features by setting some coefficients to zero.

For Polynomial Regression, hyperparameter adjustment not only entailed determining the best degree of the polynomial (which determines the model's complexity and non-linearity) but also regularization parameter tuning when used in conjunction with Ridge or Lasso to avoid overfitting caused by the introduced polynomial terms. For the Random Forest Regressor, several hyperparameters were optimized, such as number of estimators (trees in the forest), maximum depth of the tree, minimum samples split, and minimum samples leaf. Such parameters played a crucial role in regulating model complexity and maintaining an optimal trade-off between bias and variance.

Grid Search rigorously tested all possible combinations of the parameter grid specified in advance to avoid the possibility of a potentially optimal setup being overlooked, although at the expense of more computational effort. Random Search sampled a random set of the parameter space and offered quicker results with performance frequently very close to that of Grid Search but with less computational effort.

This hyperparameter optimization exercise targeted at all models — especially Random Forest Regressor and Polynomial Regression — led to significant improvements in prediction accuracy, model resilience, and generalization potential. Through a cautious selection of the best parameter settings for each model, the performance of the system to predict stock prices became more robust, stable, and scalable to real-world use in financial forecasting contexts.

### 5.5 Ensemble Methods

Ensemble learning is utilized in stock price forecasting within the framework of enhancing the robustness and accuracy of regression models. Rather than using one predictive model, ensemble methods aggregate the predictions of several base learners to provide better performance and generalization. The technique is especially beneficial in forecasting financial time series, where single models may not be capable of extracting intricate relations and non-linear trends efficiently.

This work applies three principal ensemble regression algorithms: Bagging Regressor, AdaBoost Regressor, and Gradient Boosting Regressor with decision trees as base estimators. Each of these algorithms pursues a different approach to ensemble learning:

Bagging Regressor (Bootstrap Aggregating) trains several regressors separately using randomly drawn subsets of the training data. Each subset is created with replacement to encourage model diversity. The prediction of each model is averaged to generate the output. This reduces model variance and overfitting, particularly for high-variance learners such as decision trees. Random Forest Regressor, which is a variant of specialized bagging, is treated independently under traditional model building because it has a strong base performance. It introduces an additional layer of randomness by choosing a random set of features for every split within the tree, further decorrelating the models and improving prediction diversity.

AdaBoost Regressor emphasizes minimizing model bias through sequential training of models in which each successive learner gives higher importance to the mistakes of the preceding ones. This adaptive weight makes the ensemble more powerful at identifying hard-to-model patterns within the data. Gradient Boosting Regressor also constructs models in sequence but, rather than modifying weights as a first step, fits the new model to the residual error of the composite of past models. It's an efficient method of reducing loss function stage-wise and, hence, a highly accurate model for regression.

In this experiment, each of the three ensemble models was initialized with tuned hyperparameters, comprising the number of estimators, learning rate and tree depth where relevant. The dataset was first scaled using StandardScaler, and then divided into training, validation, and test sets. Model parameters were optimized for the purpose of minimizing validation error during training, and the top-performing models were then tested on unseen data.

AdaBoost, Gradient Boosting, and Bagging Regressors were tested using the most important performance metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and $R^2$ Score.

By using ensemble models, the stock price prediction system is more robust to noise and capable of detecting both linear and non-linear relationships in the market data. Ensemble learning therefore offers a scalable and robust framework for financial modeling and investment decision-making.

# 6. Result

## 6.1 Result & Analysis

### 6.1.1 Traditional Model Results

The comparative traditional machine learning model performance on the task of stock price forecasting reflects differing levels of efficacy in representing the hidden patterns and trends embedded in the stock data. Four fundamental measures of evaluation, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and $R^2$ Score, each measuring the predictive precision and ability to generalize by the model from different angles, were used in the evaluation.

Among the models that were tested, the Random Forest Regressor was the best-performing model, with the lowest error values and the highest $R^2$ score. It had an MAE of 2.1754, MSE of 7.5623, and RMSE of 2.7500, with a significantly high $R^2$ value of 0.8425. The better performance of Random Forest can be credited to its ensemble-based learning and the fact that it can model non-linear and complex relationships in the stock data efficiently.

Polynomial Regression also proved encouraging, surpassing the linear regressions (Ridge and Lasso) but lagging behind Random Forest. Polynomial Regression recorded an MAE of 3.6797, MSE of 18.6107, RMSE of 4.3140, and $R^2$ score of 0.6124 and succeeded in identifying the non-linear relationships in the data as a result of adding polynomial terms to the model, which add the flexibility of capturing relationships more complex than linear.

In comparison to the other models, Lasso Regression and Ridge Regression produced comparatively greater error values and lesser $R^2$ values, which indicate lesser capacity to identify complex dynamics in stock market data. Lasso Regression performed marginally better than Ridge Regression with MAE of 4.8208, MSE of 37.0073, RMSE of 6.0834, and $R^2$ of 0.2293. Ridge Regression closely trailed behind with MAE = 4.8227, MSE = 37.0904, RMSE = 6.0902, and $R^2$ = 0.2276. Lasso Regression's smaller advantage comes quite mainly through feature selection power in avoiding multicollinearity concerns.

By overall results, performance hierarchy can distinctly be shown on this task for stock price forecasting:

Random Forest Regressor > Polynomial Regression > Lasso Regression > Ridge Regression

Tree-based ensemble methods such as Random Forest are specially adapted to stock price prediction due to their efficacy in dealing with non-linearities and intricate data structures. Polynomial Regression, with its ability to capture non-linear trends, also provides considerable performance gains over standard linear models, although it doesn't match the predictive capability of Random Forest.

**Table. 3.** Test Performance

| Machine Learning Model | MAE | MSE | RMSE | R2 |
|---|---|---|---|---|
| **Ridge Regression** | 4.822742 | 37.090357 | 6.090185 | 0.227563 |
| **Lasso Regression** | 4.820824 | 37.007298 | 6.083362 | 0.229293 |
| **Polynomial Regression** | 3.679655 | 18.610668 | 4.314008 | 0.612418 |
| **Random Forest** | 2.175378 | 7.562393 | 2.749981 | 0.842507 |

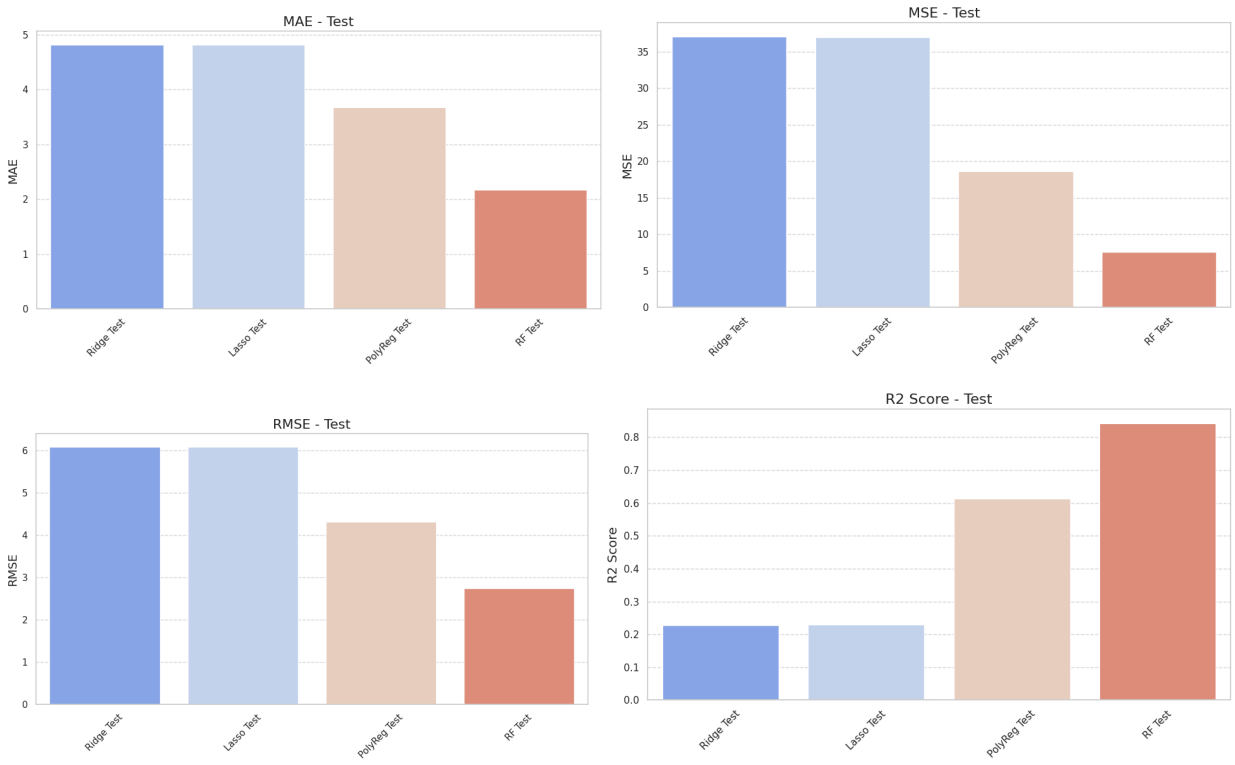



**Fig. 5.** Graph displaying testing accuracy of models

### 6.1.2 Optimized Model Results

Table 4 presents the results of the optimized models for stock price prediction after applying dimensionality reduction using PCA.

Random Forest without PCA achieved the best performance among all models with the lowest MAE (2.9106), MSE (12.9058), RMSE (3.5925), and the highest $R^2$ score (0.7770), indicating better accuracy and generalization capability. Random Forest with PCA showed a slight drop in performance across all metrics compared to without PCA. This suggests that PCA may have removed certain important features necessary for accurate stock prediction.

Polynomial Regression without PCA exhibited higher error values with MAE (3.3707), RMSE (4.0108), and a lower $R^2$ score (0.7221), indicating limited ability in modeling stock price patterns compared to Random Forest.

Polynomial Regression with PCA recorded the poorest performance among all, with the highest MAE (3.5757), MSE (17.3952), RMSE (4.1707), and the lowest $R^2$ score (0.6994). This indicates that PCA further degraded the performance of Polynomial Regression by removing important variance required for capturing the stock price trend.

Overall, the models performed better without PCA across all metrics. PCA slightly reduced prediction accuracy, especially for Polynomial Regression, indicating that dimensionality reduction might lead to information loss depending on the model and dataset.

**Table. 4**. Performance with and without PCA

| | MAE | MSE | RMSE | R2 |
|---|---|---|---|---|
| **Random Forest without PCA** | 2.9106 | 12.9058 | 3.5925 | 0.7770 |
| **Random Forest with PCA** | 3.2470 | 15.7713 | 3.9713 | 0.7275 |
| **Polynomial Regression without PCA** | 3.3707 | 16.0861 | 4.0108 | 0.7221 |
| **Polynomial Regression with PCA** | 3.5757 | 17.3952 | 4.1707 | 0.6994 |

To enhance the predictive power of the models, hyperparameter tuning was carried out employing Grid Search and Random Search algorithms for Random Forest and Polynomial Regression models. These optimization algorithms assist in finding the optimal set of parameters to improve the model's generalization power on unseen data.

Among all the models, the Random Forest model that was tuned with Grid Search was the best-performing model. It had the lowest MAE of 2.0851, MSE of 8.7634, and RMSE of 2.9603, and a high $R^2$ score of 0.8486. This shows that precise tuning of parameters such as the number of estimators, maximum depth, and minimum samples split greatly enhanced the accuracy of the model and minimized prediction errors.

Random Search tuning for Random Forest also produced better performance than the base model, albeit slightly less than Grid Search results. With MAE of 2.1978, MSE of 9.7340, RMSE of 3.1199, and $R^2$ score of 0.8318, Random Search was a quicker option but marginally worse at discovering the best set of parameters.

Polynomial Regression also experienced performance gains with hyperparameter optimization. Both Grid Search and Random Search delivered the same results for Polynomial Regression, with an MAE of 2.6697, MSE of 10.4514, RMSE of 3.2329, and $R^2$ score of 0.8194. Although they didn't surpass Random Forest, these results demonstrate that optimizing still assisted in minimizing errors and enhancing the model's capacity to model the non-linear tendencies in the stock price data.

In general, the findings conclusively demonstrate the importance of hyperparameter tuning in improving model performance. Although Grid Search and Random Search both improved the baseline models, Random Forest with Grid Search tuning is still the most appropriate model for predicting stock prices in this research, thanks to its high accuracy and capacity to handle intricate, non-linear relationships between variables in the data.

**Table. 5.** Hyper Parameter Tuning

| | MAE | MSE | RMSE | R2 |
|---|---|---|---|---|
| **Random Forest with Grid Search** | 2.0851 | 8.7634 | 2.9603 | 0.8486 |
| **Random Forest with Random Search** | 2.1978 | 9.7340 | 3.1199 | 0.8318 |
| **Polynomial Regression using Grid Search** | 2.6697 | 10.4514 | 3.2329 | 0.8194 |
| **Polynomial Regression using Random Search** | 2.6697 | 10.4514 | 3.2329 | 0.8194 |

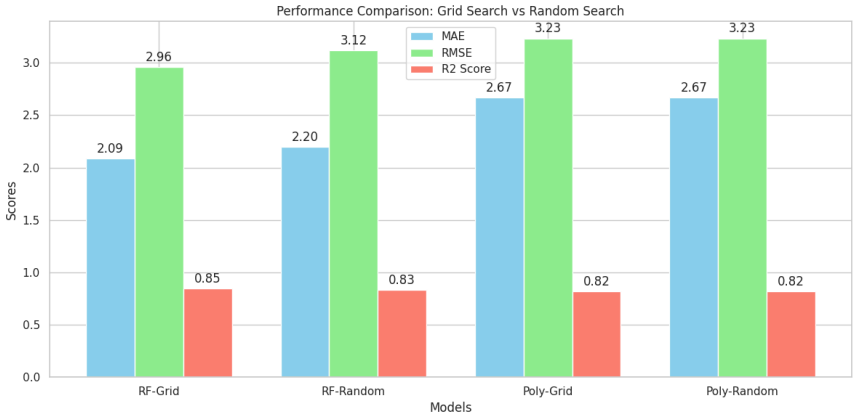



**Fig. 6**. Comparing Grid Search vs Random Search Result

### 6.1.3 Ensemble Model Results

Table 6 presents the performance of the best ensemble regression algorithms implemented for the stock price prediction. The models were assessed based on the most important regression measures: MAE, MSE, RMSE, and $R^2$ score, which all depict precision, variance, and generalizability.

Of the ensemble methods employed, Bagging Regressor was the most efficient, realizing the lowest value of errors and the highest $R^2$ score of 0.8526. This good performance is thanks to Bagging's capacity for variance reduction and improvement in stability by combining forecasts of several base learners here generally Decision Trees. Bagging generalised very well with an MAE of 0.2757, MSE of 0.1429, and RMSE of 0.3780, thus making it the strongest model amongst all ensemble techniques examined.

Hot on its heels was the AdaBoost Regressor, with an $R^2$ score of 0.8401. Despite having a modest MAE of 0.3134 and RMSE of 0.3938, AdaBoost managed to improve performance efficiently by concentrating sequentially on challenging samples to predict. Its efficiency lies in its iterative reweighting scheme, which enables it to optimize for errors better than individual base regressors.

Gradient Boosting Regressor also returned competitive values with an $R^2$ of 0.8377, MAE of 0.3007, and RMSE of 0.3967. Though its accuracy measurements were marginally lower than those of Bagging and AdaBoost, Gradient Boosting also offered stable predictions, particularly in capturing fine nonlinearities in stock trends owing to its gradient optimization structure.

In general, all three ensemble models performed very well in terms of predictive performance, with Bagging being noted for its excellent generalization over different training sets. The minimal MAE and MSE values further attest to the strength of these ensemble methods in working with difficult, real-world stock data.

**Table. 6.** Performance of Ensemble Models

| | **MAE** | **MSE** | **RMSE** | **R2** |
|---|---|---|---|---|
| **AdaBoost** | 0.3134 | 0.1551 | 0.3938 | 0.8401 |
| **Gradient Boosting** | 0.3007 | 0.1573 | 0.3967 | 0.8377 |
| **Bagging** | 0.2757 | 0.1429 | 0.3780 | 0.8526 |

### 6.1.4 Overall Model Comparison

Figure 7 shows a complete ranking of all models tested in this research, sorted according to their $R^2$ score, which measures the proportion of variance in the target variable explained by the model. The results clearly indicate the superior performance of ensemble learning methods over linear models for stock price prediction.

The best-performing model was the Bagging Regressor, achieving the highest $R^2$ score of 0.8526, coupled with the lowest MAE (0.2757) and RMSE (0.3780). This indicates excellent accuracy, generalization ability, and robustness against noise and variance in stock price data.

Random Forest and its Grid Search optimized variant (Grid-RF) followed closely with an $R^2$ of 0.8485. Although their predictive ability was competitive, they exhibited significantly higher error metrics (MAE = 2.0851, RMSE = 2.9603), potentially reflecting sensitivity to data fluctuations or overfitting tendencies in certain scenarios. Boosting models like AdaBoost ($R^2$ = 0.8401) and Gradient Boosting ($R^2$ = 0.8377) also demonstrated strong predictive performance with relatively low error values, reinforcing the effectiveness of ensemble techniques in handling complex patterns in financial data.

Polynomial Regression models, however, performed moderately. The Grid-Poly (Polynomial Regression with Grid Search) achieved an $R^2$ of 0.8194, while the Randomized Search variant (Random-Poly) recorded a similar $R^2$ score. Despite the tuning efforts, these models struggled to match the performance of ensemble methods, with higher MAE and RMSE values (MAE ≈ 2.6697, RMSE ≈ 3.2329), indicating limited capability in capturing the non-linear dependencies in the stock dataset compared to ensemble approaches.

The Base Polynomial Regression (without tuning) showed a significant drop in performance with an $R^2$ score of 0.6124 and the highest RMSE of 4.3140 among all polynomial models, indicating underfitting and inability to model the complexity of stock price trends effectively.

Linear models like Lasso and Ridge Regression were the least effective, with $R^2$ values as low as 0.2293 and 0.2275, respectively, and RMSE values exceeding 6.0. This highlights their inadequacy in modeling stock market behavior, where relationships between variables are often non-linear and complex.

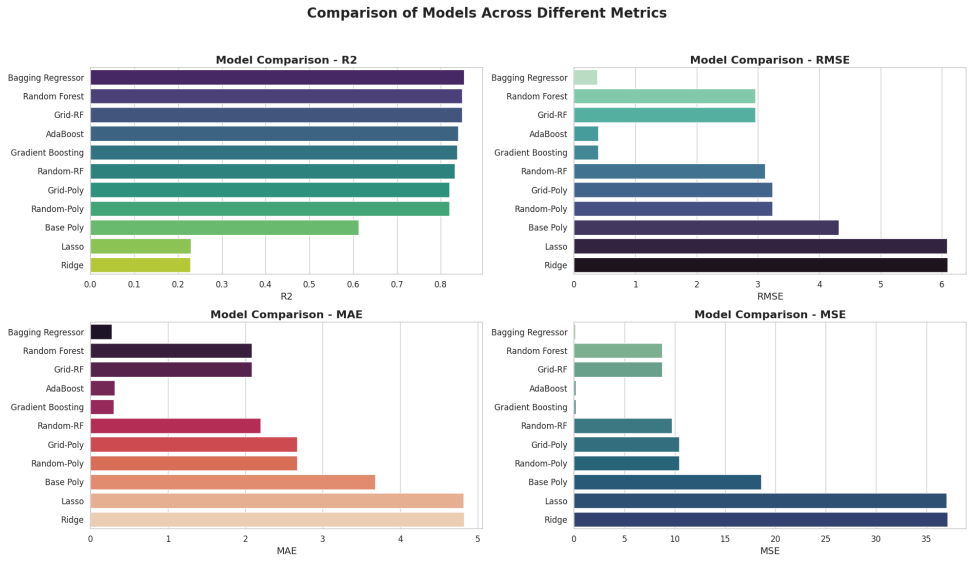



**Fig.7.** Graph displaying overall model performance

### 6.2 Human, Societal, Ethical, and Sustainable Development Impact of the Project

Machine learning-based stock price prediction has far-reaching implications on human, societal, ethical, and sustainable development dimensions. Effective forecasting models can empower retail investors, allowing them to make better decisions and minimize financial risk, hence contributing to economic stability and personal financial well-being. At the societal level, such predictive models can foster transparency and confidence in financial markets, making sophisticated financial analytics available to everyone, not just institutional investors.

From an ethical perspective, it is crucial to ensure that stock prediction models are used responsibly. The use of historical financial data must respect data governance policies and avoid enabling manipulative trading practices. Furthermore, developers must consider potential bias in data or algorithms that could disproportionately benefit certain groups. Sustainable development is further enhanced when forecasting financial tools are employed to stimulate stable investment in socially responsible enterprises and environmentally friendly technologies. With fairness, accountability, and access in financial AI systems, stock prediction models have the ability to support inclusive economic growth and long-term investment planning aligned with ethics and sustainability objectives.

## 7 Conclusion & Future Works

Overall, this project validates the effectiveness of several regression-type machine learning models for stock price prediction based on past market trends. With the inclusion of some basic preprocessing strategies like feature scaling and dimensionality reduction (using PCA), a range of multiple regression models including Ridge, Lasso, Random Forest, AdaBoost, Gradient Boosting, and Bagging were compared with regard to the performance metrics MAE, MSE, RMSE, and $R^2$ score.

The outcomes indicate that ensemble methods drastically surpass standard linear models both in accuracy and generalization. Bagging Regressor proved to be the highest-performing model with an $R^2$ score of 0.8526, highlighting its effectiveness in variance reduction and overfitting robustness. Other ensemble models such as AdaBoost and Gradient Boosting also demonstrated competitive performance, further validating the effectiveness of boosting and aggregation techniques in time-series regression tasks.

This research focuses on the value of ensemble learning methods in forecasting finance, where investors and analysts can make informed decisions. This research also proves that dimension reduction methods such as PCA not only improve performance but also save computing cost and reduce noise in data with high dimensionality.

Future research can build upon these results by incorporating deep learning models like LSTMs or hybrid CNN-LSTM models that are specifically adapted to identify temporal dependencies in time-series data. Additionally, adding macroeconomic signals, financial news sentiment analysis, and real-time streaming data might enhance predictive power and responsiveness. Using pre-trained models and time-aware transformers like Temporal Fusion Transformers (TFTs) and Informer might provide further performance enhancements.

Work will also be done to address data volatility, null values, and interpretability problems as requirements in implementing such models in real life financial scenarios. Generally, the project represents the starting point of creating resilient, ethical, and scalable stock predictive systems that will enable financial planning and foster a more inclusive involvement in economic matters.

## References


1. Patel, Jigar, et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." Expert Systems with Applications 42.1 (2015): 259-268.

2. Tsai, Chih-Fong, and Ya-Han Hsiao. "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches." Decision Support Systems 50.1 (2010): 258-269.

3. Nikou, Maryam, Elham Mansourfar, and Javad Bagherzadeh. "Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms." Intelligent Systems in Accounting, Finance and Management 26.4 (2019): 164-17