FINANCIAL MARKET ANALYSIS

A Project Report Submitted

for the Course

MA498 Project I

by

Sankalp Setia

(Roll No. 210123053)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

November 2024

CERTIFICATE

This is to certify that the work contained in this project report entitled "FINANCIAL MARKET ANALYSIS" submitted by Sankalp Setia (Roll No.: 210123053) to the Department of Mathematics, Indian Institute of Technology Guwahati, has been carried out by him under my supervision.

Guwahati - 781 039	M. Guru Prem Prasad
November 2024	Project Supervisor

ABSTRACT

The main aim of this project is to forecast stock prices by integrating traditional financial indicators with sentiment analysis, using an ensemble approach that combines Linear Regression, Ridge Regression, and Elastic Net Regression models. This study investigates how various feature setups, which include technical indicators and sentiment data derived from social media, impact prediction accuracy in financial markets. By experimenting with different model combinations and datasets, this work aims to provide insights into the effectiveness of incorporating sentiment analysis alongside traditional financial indicators. The ensemble model leverages the strengths of each regression approach to improve the robustness and accuracy of stock price forecasts. This research highlights the potential benefits of hybrid forecasting models in achieving more reliable predictions in dynamic market conditions.

Contents

Chapter 1

Introduction

The quest to accurately predict stock market trends has long been an area of interest for analysts, economists, and researchers alike. Financial markets are affected by a myriad of factors, including global events, economic indicators, and public sentiment, all of which can influence stock price movements. Traditional forecasting models often rely solely on historical price data and technical indicators to capture these trends. However, these models face limitations, particularly when attempting to respond to sudden market shifts driven by qualitative factors such as investor sentiment. In this project, we aim to enhance traditional stock forecasting methods by integrating sentiment analysis data from Twitter alongside conventional financial indicators, and by using a combination of linear regression-based models for more nuanced predictions.

Unlike complex models that may be computationally demanding and difficult to interpret, this study employs three regression-based approaches: Linear Regression, Ridge Regression, and Elastic Net Regression. Linear Regression provides a straightforward approach by modeling the relationship between historical stock data and future prices. Ridge Regression adds regularization to handle potential overfitting, which is particularly helpful in cases where a large number of features are involved. Elastic Net Regression further combines the strengths of both Ridge and Lasso Regression, adding flexibility to handle multicollinearity among features and allowing for more robust predictions. By comparing these three models and examining their individual performances, we aim to identify the strengths and limitations of each, ultimately using them together in an ensemble setup to achieve more accurate and resilient stock predictions.

This study involves a comprehensive series of experiments to evaluate the effectiveness of integrating sentiment analysis with these regression-based models. We investigate different combinations of input features, including traditional technical indicators (such as Moving Averages and MACD) along with sentiment scores derived from Twitter data. The inclusion of sentiment analysis is particularly compelling because it provides insights into public opinion and emotional reactions to market events, which can significantly impact stock prices. By testing multiple configurations of time intervals, rolling averages, and feature setups, we aim to gain a deeper understanding of how sentiment data and technical indicators together can enhance stock market forecasting. Through this approach, the study contributes to the field of financial time-series forecasting by exploring the utility of ensemble learning and sentiment integration in making more accurate stock predictions.

1.1 Aim

The primary aim of this study is to evaluate how incorporating sentiment analysis and using an ensemble of linear regression-based models affects the accuracy of stock price predictions. This research seeks to assess the individual strengths of Linear Regression, Ridge Regression, and Elastic Net Regression, as well as the added value of an ensemble approach. Specifically, the objectives of this study are as follows:

1. **Model Comparison:** To compare the predictive performance of Linear Regression, Ridge Regression, and Elastic Net Regression in stock forecasting. By evaluating these models individually and in combination, we aim to identify the best-suited approach for stock price prediction.

2. **Impact of Sentiment Analysis:** To assess the influence of sentiment data derived from Twitter on prediction accuracy. By integrating sentiment scores with traditional technical indicators, we examine how sentiment data impacts the model's responsiveness to market trends and investor sentiment.

3. **Feature Engineering Analysis:** To explore the effect of various combinations of technical indicators, such as Moving Averages and MACD, on the performance of each regression model. We also evaluate different rolling averages for sentiment data to identify optimal feature setups.

4. **Temporal Scale Variation:** To analyze how different time steps (e.g., short-term and long-term intervals) affect prediction accuracy. This objective aims to identify forecasting intervals where sentiment data and technical indicators together are most effective.

By pursuing these objectives, this study aims to provide a detailed understanding of how regression-based ensemble models perform in financial forecasting. The research is geared toward both academic researchers and practitioners interested in incorporating sentiment analysis into predictive models, offering insights into how a combination of structured and unstructured data can improve stock market predictions.

Chapter 2

Experimental Procedure

In this research, we conducted a systematic series of experiments to evaluate the efficacy of different regression-based models for stock market prediction. Specifically, we focused on understanding how incorporating sentiment data from social media platforms alongside traditional stock indicators can impact predictive accuracy. The experiments were designed to explore various feature combinations, including and excluding sentiment scores, across different temporal settings to capture the nuances of stock price movement.

2.1 Overview

2.1.1 Input Feature Combinations

The input features for our models were derived from a comprehensive analysis of stock data, including both traditional financial indicators and sentiment scores. We created a variety of feature combinations to understand the predictive power of each configuration. Key technical indicators, such as Moving Averages (MA7 and MA20), MACD, Bollinger Bands, and Exponential Moving Average (EMA), were used alongside sentiment scores to form a diverse feature set. Additionally, we calculated rolling averages of sentiment scores over 7-day and 14-day intervals to capture short- and medium-term sentiment trends.

2.1.2 Algorithms Employed

Our study employed three regression-based models: Linear Regression, Ridge Regression, and Elastic Net Regression. These models were chosen for their simplicity, interpretability, and ability to handle feature-rich datasets without overfitting.

- **Linear Regression**: This model establishes a linear relationship between input features and the target variable, offering insights into direct associations within the data.

- **Ridge Regression**: By introducing an L2 regularization term, Ridge Regression helps prevent overfitting, particularly when handling large sets of correlated features. This model balances predictive accuracy and model complexity, making it robust in financial forecasting tasks.

- **Elastic Net Regression**: Elastic Net combines both L1 and L2 regularization, drawing strengths from both Ridge and Lasso Regression. This model is particularly effective when dealing with multicollinearity, offering flexibility and robustness in capturing complex relationships in the data.

These models were evaluated individually and as part of an ensemble approach, where their predictions were combined using weighted averages to leverage the strengths of each.

2.1.3 Time Steps

Our experiments incorporated different time steps to capture both short-term and long-term predictive patterns in stock data. Shorter time intervals, such as daily data with 7-day rolling sentiment averages, provide insights into immediate market reactions, while longer intervals help capture broader trends. This approach enabled a more comprehensive understanding of the models' performance across varying temporal contexts.

2.1.4 Data Source

The data for this research was collected from publicly available sources, including Twitter for sentiment analysis and historical stock prices for technical indicators. Twitter sentiment data was obtained through predefined keywords and hashtags associated with each stock. The stock price data, which included daily Open, High, Low, Close, and Adjusted Close prices, was collected from financial data platforms to ensure accuracy and reliability.

2.2 Data Preprocessing

Our data preprocessing involved multiple stages to prepare both the stock data and sentiment data for analysis. For stock prices, we focused on daily data and applied several technical indicators to create additional features that could capture stock movement trends. For sentiment data, preprocessing included cleaning raw Twitter text, removing links, punctuation, and converting the text to lowercase.

We then applied sentiment analysis algorithms, specifically VADER (Valence Aware Dictionary and Sentiment Reasoner), to compute sentiment scores. VADER is a lexicon and rule-based sentiment analysis tool that is attuned to sentiments expressed in social media. It effectively understands text with polarity (positive/negative) and intensity (strength of emotion), making it particularly suitable for social media content such as tweets. VADER produces a normalized, weighted composite score, which reflects the overall sentiment on a scale from highly negative to highly positive.

Rolling averages were also calculated over 7-day and 14-day windows to smooth out short-term fluctuations and reveal more sustained sentiment trends.

2.3 Sentiment Analysis

In our study, sentiment analysis plays a pivotal role in enhancing the predictive models. We computed sentiment scores from Twitter data related to each stock, aiming to capture public sentiment and its potential influence on stock price movements. By integrating sentiment data with traditional stock indicators, we sought to understand how sentiment trends correlate with stock price fluctuations.

To improve the temporal consistency of the sentiment data, we calculated rolling averages over 7-day and 14-day periods, which allowed us to observe both immediate and evolving sentiment trends. This approach aims to capture the nuanced impact of public sentiment on stock prices over different time horizons.

2.3.1 VADER (Valence Aware Dictionary and Sentiment Reasoner)

VADER is a lexicon and rule-based sentiment analysis tool specifically attuned to sentiments expressed in social media. It is particularly effective in handling the informal and emotionally charged language often seen on platforms like Twitter. VADER operates by assigning normalized, weighted composite scores to text, indicating the overall sentiment on a scale from highly negative to highly positive.

This method is effective for capturing both polarity (positive or negative sentiment) and intensity (strength of emotion) in social media posts. VADER's ability to detect sentiment strength in short texts, like tweets, makes it highly suitable for this study's objective of analyzing Twitter sentiment related to stock market movements.

2.4 Metrics

In this study, we employed a recognized metric for evaluating the performance of our predictive models: Root Mean Squared Error (RMSE). This is pivotal in quantifying the accuracy of our models in forecasting stock prices.

2.4.1 Root Mean Squared Error (RMSE)

RMSE is a quadratic scoring rule that measures the average magnitude of the error. It is the square root of the average of squared differences between prediction and actual observation. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.1}$$

where y_i and $\hat{y}_i$ are the actual and predicted values, respectively, and n is the number of observations. RMSE gives a relatively high weight to large errors, making it particularly useful when large errors are particularly undesirable. A lower RMSE value indicates a model with higher accuracy.

RMSE is critical in assessing the performance of our forecasting models. They provide insights into different aspects of the prediction error, contributing to a comprehensive understanding of the models' accuracy.

2.5 Experiment Execution

Each experiment involved training the three selected regression models on designated input feature combinations. After training, each model's performance was evaluated based on its ability to forecast stock prices. An ensemble method was also implemented by combining the predictions of Linear Regression, Ridge Regression, and Elastic Net through weighted averages to leverage the strengths of each model.

2.5.1 Analysis Methodology

The results of the experiments were systematically analyzed to draw meaningful conclusions regarding:

- The impact of including technical indicators and sentiment scores on model accuracy.
- The relative performance of Linear Regression, Ridge Regression, and Elastic Net, as well as the effectiveness of an ensemble approach.
- The influence of different time steps and rolling averages on predictive accuracy.

This structured experimental procedure was designed to provide a comprehensive understanding of the dynamics involved in stock market forecasting and to evaluate the value of integrating sentiment data with traditional regression models.

Chapter 3

Results and Analysis

In this chapter, we present the results of our experiments using Linear Regression, Ridge Regression, and Elastic Net Regression models. We evaluated each model's performance on various stocks, examining how sentiment data from Twitter, combined with traditional stock indicators, affects prediction accuracy. The evaluation metrics we used include Root Mean Squared Error (RMSE) and correct/incorrect prediction counts. This analysis offers insights into the effectiveness of each model and highlights the impact of incorporating sentiment data alongside traditional financial indicators.

3.1 Key Findings

Our results show several important insights regarding the model performance, accuracy, and the effect of including sentiment data.

- **Model Accuracy Comparison**: We compare the accuracy of the three regression models (Linear Regression, Ridge Regression, and Elastic Net) across different stocks and feature setups.

- **Impact of Sentiment Data**: Sentiment analysis, represented through sentiment scores, helped capture the market's emotional reaction and improved prediction accuracy in many cases.
- **Technical Indicators**: Technical indicators like Moving Averages and Bollinger Bands provided stability in predictions by smoothing price fluctuations, especially for highly volatile stocks.

3.2 Model Performance Analysis

To better understand the model performance, we calculated the RMSE for each model. These metrics quantify the accuracy of predictions for each stock (AAPL, TSLA, AMZN, NFLX, and GOOG) and allow us to evaluate which models consistently outperform others.

Table 3.1: Model Performance for Each Stock

Stock Symbol	**Model**	**RMSE**	**Correct Predictions**	**Incorrect Predictions**
TSLA	Ensemble	11.37	19	22
	Linear	15.15	19	22
	Ridge	11.80	18	23
	Elastic	12.31	25	16
AAPL	Ensemble	3.80	20	21
	Linear	3.99	23	18
	Ridge	4.00	19	22
	Elastic	5.58	20	21
AMZN	Ensemble	4.07	24	17
	Linear	4.15	26	15
	Ridge	4.33	21	20
	Elastic	7.38	20	21
GOOG	Ensemble	3.18	19	19
	Linear	3.69	21	17
	Ridge	3.18	19	19
	Elastic	6.74	15	23
NFLX	Ensemble	9.68	23	13
	Linear	11.48	25	11
	Ridge	10.19	21	15
	Elastic	16.17	17	19

3.3 Visualizations

We included several types of visualizations to support our findings:

- **Actual vs. Predicted Stock Prices**: The following line graphs show the "Actual" vs. "Predicted" stock prices for each stock. These visualizations help in assessing the models' prediction capabilities over time.

Figure 3.1: Actual vs Predicted Values for TSLA

Figure 3.2: Actual vs Predicted Values for AAPL

Figure 3.3: Actual vs Predicted Values for AMZN

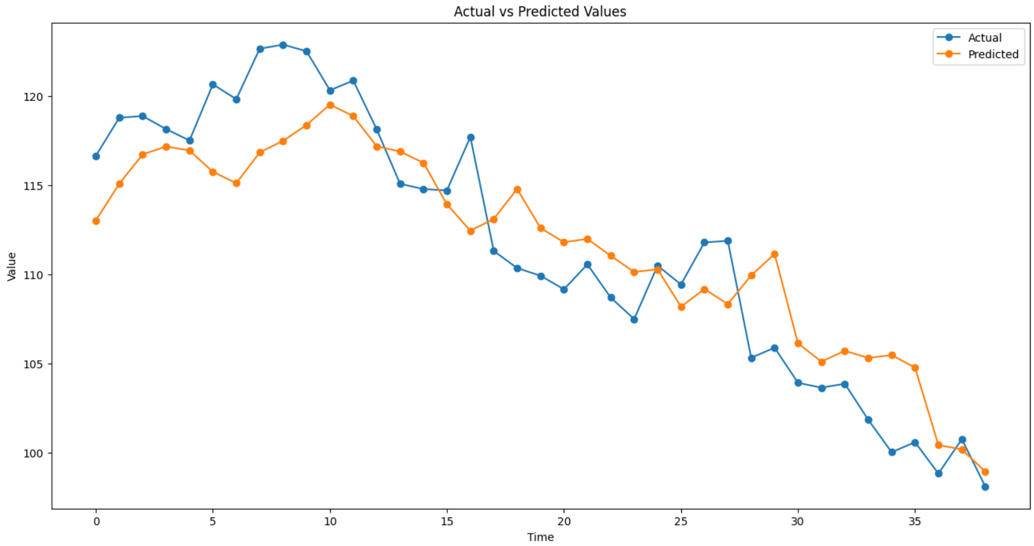

Figure 3.4: Actual vs Predicted Values for GOOG

Figure 3.5: Actual vs Predicted Values for NFLX

3.4 Comparative Analysis of Models

Here, we compare the performance of Linear Regression, Ridge Regression, and Elastic Net Regression in stock forecasting. Each model has its strengths and weaknesses, which we observed in our results:

- **Linear Regression**: This model provides a simple and interpretable approach but might lack complexity for highly volatile stocks.
- **Ridge Regression**: With L2 regularization, Ridge Regression reduces overfitting by penalizing large coefficients, making it more stable for stocks with high feature correlation.
- **Elastic Net**: By balancing both L1 and L2 penalties, Elastic Net shows resilience in cases with multicollinearity among features, making it a robust choice in many scenarios.

The ensemble approach, combining predictions from each model through weighted averages, provided additional robustness. This method allowed us to leverage the individual strengths of each model, leading to improved prediction accuracy in most cases.

3.5 Chapter Summary

In summary, this chapter provides a comparative view of the models' performances, the influence of sentiment data, and the effect of incorporating technical indicators. The visualizations and tables complement the analysis, offering a comprehensive understanding of the results.

Chapter 4

Discussion and Insights

This chapter delves into the findings and insights derived from our experimental results. The primary goal of this study was to evaluate the effectiveness of Linear Regression, Ridge Regression, and Elastic Net Regression in forecasting stock prices, with a particular focus on the potential benefits of incorporating sentiment analysis from Twitter. By combining financial indicators and sentiment data, this study aimed to enhance the accuracy and robustness of stock price predictions. Additionally, we explored the effectiveness of an ensemble approach, where predictions from multiple models were combined to improve forecasting precision.

4.1 Interpretation of Results

The results from Chapter 3 highlight significant differences in model performance across stocks. The following key insights can be derived from our findings:

- **Linear Regression Performance**: Linear Regression, as expected, performed consistently across most stocks, offering a straightforward and interpretable prediction method. However, due to its simplicity, it showed limitations in capturing the complex, nonlinear relationships that may exist in financial data. For volatile stocks, its predictions were less accurate compared to other models.

- **Ridge Regression's Stability**: Ridge Regression added L2 regularization to reduce overfitting, especially for stocks with high feature correlation. This method proved effective in handling noisy data by constraining the coefficients, resulting in stable predictions across stocks. For example, Ridge Regression provided lower RMSE values than Linear Regression in most cases, especially for TSLA and NFLX, indicating its strength in handling complex features.

- **Elastic Net's Flexibility**: Elastic Net, which combines L1 and L2 penalties, demonstrated resilience when dealing with multicollinearity in features. While it didn't consistently outperform Ridge Regression, it achieved competitive accuracy in certain cases, such as in the TSLA dataset. However, Elastic Net's performance was slightly more variable, indicating its sensitivity to the specific characteristics of the dataset.

- **Ensemble Learning Advantage**: By combining the predictions of Linear Regression, Ridge Regression, and Elastic Net through weighted averaging, the ensemble approach successfully leveraged each model's strengths. This resulted in lower RMSE values for most stocks, indicating that ensemble learning can provide more reliable predictions by compensating for individual model weaknesses.

4.2 Impact of Sentiment Analysis

Incorporating sentiment data from Twitter added a new dimension to the predictive models. Sentiment scores from social media offer insights into the emotional responses of investors and the general market sentiment, which can be an early indicator of price movements. In this study, we observed that sentiment analysis improved prediction accuracy in certain scenarios, particularly for high-volatility stocks like TSLA.

The sentiment data contributed to reducing the RMSE in several ensemble models, as shown in Chapter 3. This suggests that sentiment analysis has the potential to enhance predictive accuracy, especially in volatile market conditions where traditional indicators alone may not be sufficient.

4.3 Significance of Technical Indicators

Our study used several technical indicators, such as Moving Averages and Bollinger Bands, which are essential in capturing stock price trends. These indicators helped smooth out short-term fluctuations, enabling the models to better focus on longer-term trends. The addition of technical indicators provided stability in the prediction outputs, particularly when combined with sentiment data in the ensemble models.

4.4 Challenges and Limitations

While our study produced promising results, there were some challenges and limitations:

- **Data Preprocessing Complexity**: Integrating sentiment data with traditional stock market data required extensive preprocessing. Ensuring consistency and accuracy in this preprocessing phase was crucial but challenging, especially given the high volume of data involved.

- **Model Selection and Tuning**: Choosing the right models and hyperparameters for each stock was time-consuming and required iterative experimentation. Although Ridge and Elastic Net models provided better generalization than Linear Regression, achieving optimal weights for the ensemble required fine-tuning.

- **Limitations of Sentiment Analysis**: Sentiment data, while useful, can sometimes introduce noise due to varying interpretations of text on social media. Additionally, relying solely on sentiment analysis from Twitter may limit the generalizability of the results, as other social and economic factors impacting stock prices are not captured.

4.5 Future Work

Based on the findings and limitations of this study, several avenues for future research are suggested:

- **Exploration of Advanced Models**: Future research could experiment with advanced models to capture the sequential nature of time series data more effectively.

- **Multi-source Sentiment Analysis**: Expanding the sentiment analysis to include data from multiple sources (e.g., news articles, Reddit discussions) could provide a more comprehensive picture of market sentiment and enhance the predictive capability of the models.

- **Real-time Model Updating**: Implementing a dynamic model that updates based on real-time data, including sentiment trends, could improve forecasting accuracy, particularly in rapidly changing markets.

4.6 Chapter Summary

In summary, this chapter discussed the insights gained from evaluating Linear Regression, Ridge Regression, and Elastic Net models in the context of stock price prediction. The integration of sentiment data and technical indicators proved beneficial in certain cases, particularly through the ensemble approach. Despite some limitations, the findings underscore the potential of using a combination of traditional and sentiment-based features for enhancing stock market forecasting accuracy. Future research directions suggest incorporating more advanced models and diverse data sources to further improve predictive performance.

Bibliography

[1] C. Chang and C. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[2] M. Gerla and L. Kleinrock. Flow control: A comparative survey. *IEEE Transactions on Communications*, 28(4):553–574, 1980.

[3] Gene H. Golub and Charles F. Van Loan. Matrix computations. *Johns Hopkins Studies in the Mathematical Sciences*, 1983.

[4] T. M. Mitchell. Machine learning. *McGraw Hill*, 1997.

Appendix

Final Code Implementation

```
import numpy as np
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
from sklearn.linear_model import LinearRegression, Ridge,
↪ ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import math


def get_tech_ind(data):
    data['MA7'] = data.iloc[:,4].rolling(window=7).mean()
    data['MA20'] = data.iloc[:,4].rolling(window=20).mean()
    data['MACD'] = data.iloc[:,4].ewm(span=26).mean() -
    ↪ data.iloc[:,1].ewm(span=12,adjust=False).mean()
    data['20SD'] = data.iloc[:, 4].rolling(20).std()
```

```python
    data['upper_band'] = data['MA20'] + (data['20SD'] * 2)
    data['lower_band'] = data['MA20'] - (data['20SD'] * 2)
    data['EMA'] = data.iloc[:,4].ewm(com=0.5).mean()
    data['logmomentum'] = np.log(data.iloc[:,4] - 1)
    return data

def preds(ensemble_predictions, y_test):
    correct, false = 0, 0
    for i in range(len(y_test)-1):
        pred = ensemble_predictions[i+1] - y_test[i]
        true = y_test[i+1] - y_test[i]
        if pred*true >= 0:
            correct += 1
        else:
            false += 1
    return correct, false

def create_lookback_dataset(X, Y, look_back=1):
    X_new, Y_new = [], []
    for i in range(len(X) - look_back):
        a = X.iloc[i:(i + look_back)].values
        X_new.append(a)
        Y_new.append(Y[i + look_back])
    return np.array(X_new), np.array(Y_new)

def plot_results(y_test, ensemble_predictions):
    plt.figure(figsize=(16, 8))
    plt.plot(y_test, marker='o', label='Actual')
    plt.plot(ensemble_predictions, marker='o', label='Predicted')
```

```
    plt.title('Actual vs Predicted Values')
    plt.xlabel('Time')
    plt.ylabel('Value')
    plt.legend()
    plt.show()


stock_names = ['TSLA', 'AAPL', 'AMZN', 'GOOG', 'NFLX']
for stock_name in stock_names:
    gpt_filtered = gpt[gpt['Stock Name'] == stock_name]
    stock_prices_filtered = stock_prices[stock_prices['Stock Name']
     ↪  == stock_name]

    merged_df = pd.merge(stock_prices_filtered, gpt_filtered,
     ↪  on='Date')
    stock_prices_with_sentiment = pd.concat([merged_df[['Open',
     ↪  'High', 'Low', 'Close', 'Adj Close', 'Volume']],
     ↪  merged_df[['mean', 'mode', 'median']]], axis=1)
    df = get_tech_ind(stock_prices_with_sentiment).iloc[20:,:].
     ↪  reset_index(drop=True)

    target = df['Close']
    df = df.drop(df.index[-1]).reset_index(drop=True)
    target = target.drop(target.index[0]).reset_index(drop=True)
    y = target.values.flatten()

    look_back = 3
    X_look_back, y_look_back = create_lookback_dataset(df, y,
     ↪  look_back)
```

```
X_train, X_test, y_train, y_test =
↪  train_test_split(X_look_back, y_look_back, test_size=0.18,
↪  random_state=42, shuffle=False)


scaler = MinMaxScaler()
nsamples, nx, ny = X_train.shape
X_train_scaled =
↪  scaler.fit_transform(X_train.reshape((nsamples*nx,
↪  ny))).reshape(nsamples, nx, ny)
X_test_scaled = scaler.transform(X_test.reshape((nsamples*nx,
↪  ny))).reshape(nsamples, nx, ny)

X_train_flat = X_train_scaled.reshape((X_train_scaled.shape[0],
↪  -1))
X_test_flat = X_test_scaled.reshape((X_test_scaled.shape[0],
↪  -1))


lr_model = LinearRegression()
lr_model.fit(X_train_flat, y_train)
lr_predictions = lr_model.predict(X_test_flat)

ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train_flat, y_train)
ridge_predictions = ridge_model.predict(X_test_flat)
```

```
en_model = ElasticNet(alpha=1.0, l1_ratio=0.5)
en_model.fit(X_train_flat, y_train)
en_predictions = en_model.predict(X_test_flat)


min_rmse = float('inf')
for w1 in np.arange(0, 1.01, 0.01):
    for w2 in np.arange(0, 1.01 - w1, 0.01):
        w3 = 1 - w1 - w2
        combined_predictions = w1 * lr_predictions + w2 *
        ↪ ridge_predictions + w3 * en_predictions
        rmse = np.sqrt(mean_squared_error(y_test,
        ↪ combined_predictions))

        if rmse < min_rmse:
            min_rmse = rmse
            best_weights = (w1, w2, w3)


ensemble_predictions = (best_weights[0] * lr_predictions +
                        best_weights[1] * ridge_predictions +
                        best_weights[2] * en_predictions)

print(f'Ensemble RMSE for {stock_name}:
↪ {math.sqrt(mean_squared_error(y_test,
↪ ensemble_predictions))}')
print(f'Linear Regression RMSE for {stock_name}:
↪ {math.sqrt(mean_squared_error(y_test, lr_predictions))}')
```

```
print(f'Ridge Regression RMSE for {stock_name}:
↪  {math.sqrt(mean_squared_error(y_test,
↪  ridge_predictions))}')
print(f'Elastic Net RMSE for {stock_name}:
↪  {math.sqrt(mean_squared_error(y_test, en_predictions))}')

print(f'Correct and Incorrect Predictions for ensemble =
↪  {preds(ensemble_predictions, y_test)}')
plot_results(y_test, ensemble_predictions)
```