|  |
| --- |
| **Forecasting Stock Market Pattern: A Machine Learning Approach** |

**PROJECT SUBMITTED TO ASIAN SCHOOL OF MEDIA STUDIES**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE**

**AWARD OF**

**DIPLOMA**

**in**

**Data Science**

By

**Aditya Kumar Singh**

**Under the Supervision of**

**Prof. Nitish Patil**

****

ASIAN SCHOOL OF MEDIA STUDIES

SCHOOL OF DATA SCIENCE

**2024**

**DECLARATION**

**I, Aditya Kumar Singh, S/O Vinay Kumar Singh,** declare that my project entitled **Forecasting Stock Market Pattern: A machine Learning Approach** submitted at **School of Data science, Asian School of Media Studies, Film City, Noida, for the award of Diploma in Data Science, ASMS** is an original work and no similar work has been done in India anywhere else to the best of my knowledge and belief.

This project has not been previously submitted for any other degree of this or any other University/Institute.

****

***Signature***

**Aditya Kumar Singh**

**8858355055**

**asksingh202@gmail.com**

**Diploma in Data Science**

**School of Data Science**

**Asian School of Media Studies**

**ACKNOWLEDGEMENT**

The completion of the project titled **Forecasting Stock Market Pattern: A machine Learning Approach**, gives me an opportunity to convey my gratitude to all those who helped to complete this project successfully. I express special thanks:

* **To *Prof. Sandeep Marwah,* President,** Asian School of Media Studies, who has been a source of perpetual inspiration throughout this project.
* To ***Mr. Ashish Garg,*** Director **for** School of Data Science for your valuable guidance, support, consistent encouragement, advice and timely suggestions.
* To ***Mr. Nitish Patil***,Assistant Professor of School of Data Science, for your encouragement and support. I deeply value your guidance.
* To my ***all faculty*** & ***friends*** for their insightful comments on early drafts and for being my worst critic. You are all the light that shows me the way.

To all the people who have directly or indirectly contributed to the writing of this report, but their names have not been mentioned here.

***Signature***

**Aditya Kumar Singh**

**8858355055**

**asksingh202@gmail.com**

**Diploma in Data Science**

**School of Data Science**

**Asian School of Media Studies**

**ABSTRACT**

Among investors, the ability to predict future developments or crises in the stock market has long been highly valued. During the COVID-19 worldwide pandemic, this skill became even more crucial, underscoring the need of risk management in preserving stability in such unpredictable times.

Although there is a rising need for reliable intelligent systems that can precisely anticipate stock prices to inform investment strategies, traditional business research still uses a variety of risk management techniques. Nowadays, a large portion of this field's research focuses on applying machine learning techniques to predict stock price patterns. While these approaches have shown encouraging results, there aren't many thorough surveys that list all of the machine learning algorithms used for stock price prediction.

Stock Market is one of the most vibrant sectors in the financial system, marking an important contribution to economic development. Stock Market is a place where buyers and sellers of securities can enter into transactions to purchase and sell shares, bonds, debentures etc. In other words Stock Market is a plate form for trading various securities and derivatives. Further, it performs an important role of enabling corporate, entrepreneurs to raise resources for their companies and business ventures through public issues. Today long term investors are interested to invest in the Stock market rather than invest anywhere. The Bombay Stock Exchange (BSE), the National Stock Exchange (NSE) and the Calcutta Stock Exchange (CSE) are the three large stock exchanges of Indian Stock Market.

The main objective of present study is to present review of literature related to Indian Stock Market to study the Indian Stock Market in depth. The study would facilitate the reader to know the past, current and future trend or prospects of Indian Stock market. This study would provide guidelines to investor to maximise profit with minimize risks. High degree of volatility in the recent times in the Indian market has led to more development in the future.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | ***Page No.*** | |
| **Declaration** | | 2 |
| **Acknowledgment** | | 3 |
| **Abstract** | | 4 |
| **List of figures** | | 7 |
| **CHAPTER 1: Introduction**  1.1 Introduction 8  1.1.1 Background 8  1.1.2 Problem Statement 8  1.1.3 Objectives 9  1.1.4 Outline of the study 9-10  2. Literature Review 10-14  3. Definitions 14  **CHAPTER 2: Dataset Preparation/Pre-processing**   * 1. Introduction 15   2. Exploratory Data Analysis 16-17   **CHAPTER 3: Model Selection: Algorithms of ML**  Model selection 18-19  3.1 Linear regression 19-20   * + 1. Mathematical Intuition 20-21     2. Implementation with the dataset 21-36   **CHAPTER 4: Analysis of Result & Discussion**   * 1. Experimental work 37-40      1. Performance Measures/Evaluation Metrics 40-53   **CHAPTER 5: Conclusion**   * 1. Summary 54-55   2. Future Scope of Work 55   **REFERENCES** 56 | |

**LIST OF FIGURES**

**Page No**

Fig 1 Plot adjusted close over time 16

Fig 2 EDA 40

Fig 3 Plot adjusted close over time 42

Fig 4 Plot RMSE versus N 44

Fig 5 Plot predictions for a specific day 46

Fig 6 Plot predictions on dev set 48

Fig 7 Zoomed-In View of Model Predictions 50

vs. Actual Stock Prices on Dev Set

Fig 8 Actual vs. Predicted Stock Prices 52

**CHAPTER 1**

**Forecasting Stock Market Pattern**

* 1. **INTRODUCTION**

"Forecasting Stock Market Patterns using Linear Regression" aims to leverage statistical techniques to predict stock market movements. By employing linear regression, this project seeks to identify and quantify the relationships between various factors and stock prices. The analysis will use historical data to model and forecast future stock trends, providing insights into market behaviour and aiding investment decisions. This method of forecasting is grounded in finding the linear correlation between variables, allowing for a simplified yet effective approach to understanding and predicting market fluctuations.

* + 1. **Background**

The stock market is influenced by a multitude of factors, including economic indicators, company performance, and market sentiment. Traditional methods of stock price prediction relied on fundamental and technical analysis. However, with the advent of machine learning and big data analytics, more sophisticated models such as Linear Regression and other machine learning algorithms have emerged, offering potentially higher accuracy in predictions.

**1.1.2 Problem Statement**

Predicting stock prices is inherently challenging due to the market's volatility and the influence of unforeseen events. Traditional models often fall short due to their inability to account for the complex and dynamic nature of the stock market. The problem this project aims to address is the development of a more accurate and robust prediction model using advanced machine learning techniques, which can adapt to the ever-changing market conditions and provide reliable stock price forecasts.

**1.1.3 Objectives**

The primary objective of this project is to develop a model that can accurately predict stock prices. By analysing historical stock data and other relevant financial indicators, the project aims to create a reliable prediction system that can help investors make better financial decisions. This project will also explore the effectiveness of different machine learning models in predicting stock prices.

**1.1.4 Outline of the study**

 Introduction

* Overview of stock market prediction
* Importance of accurate stock price forecasting
* Introduction to linear regression as a predictive tool

 Literature Review

* Historical approaches to stock market prediction
* Advances in machine learning and their application in stock prediction
* Key findings from previous studies using linear regression in stock market analysis.

 Research Objectives

* To identify the key factors affecting stock prices
* To develop a linear regression model for stock price prediction
* To evaluate the accuracy and effectiveness of the model

 Methodology

* Data Collection
  + Source of historical stock price data
  + Selection of predictor variables (e.g., economic indicators, financial statements)
* Data Preprocessing
  + Handling missing data
  + Normalization of data
* Model Development
  + Building the linear regression model
  + Training and testing the model using historical data.
* Model Evaluation
  + Metrics for model performance (e.g., RMSE, MAE)
  + Cross-validation techniques

 Results and Discussion

* Presentation of model findings
* Analysis of prediction accuracy
* Comparison with other predictive models

 Conclusion

* Summary of key findings
* Implications for investors and market analysts
* Recommendations for future research

1. **Literature Review**

Gupta (1972) in his book has studied the working of stock exchanges in India and has given a number of suggestions to improve its working. The study highlights the' need to regulate the volume of speculation so as to serve the needs of liquidity and price continuity. It suggests the enlistment of corporate securities in more than one stock exchange at the same time to improve liquidity. The study also wishes the cost of issues to be low, in order to protect small investors.

Panda (1980) has studied the role of stock exchanges in India before and after independence. The study reveals that listed stocks covered four-fifths of the joint stock sector companies. Investment in securities was no longer the monopoly of any particular class or of a small group of people. It attracted the attention of a large number of small and middle class individuals. It was observed that a large proportion of savings went in the first instance into purchase of securities already issued.

Gupta (1981) in an extensive study titled `Return on New Equity Issues' states that the investment performance of new issues of equity shares, especially those of new companies, deserves separate analysis. The factor significantly influencing the rate of return on new issues to the original buyers is the `fixed price' at which they are issued. The return on equities includes dividends and capital appreciation. This study presents sound estimates of rates of return on equities, and examines the variability of such returns over time.

Jawahar Lal (1992) presents a profile of Indian investors and evaluates their investment decisions. He made an effort to study their familiarity with, and comprehension of financial information, and the extent to which this is put to use. The information that the companies provide generally fails to meet the needs of a variety of individual investors and there is a general impression that the company's Annual Report and other statements are not well received by them.

L.C.Gupta (1992) revealed the findings of his study that there is existence of wild speculation in the Indian stock market. The over speculative character of the Indian stock market is reflected in extremely high concentration of the market activity in a handful of shares to the neglect of the remaining shares and absolutely high trading velocities of the speculative counters. He opined that, short- term speculation, if excessive, could lead to "artificial price". An artificial price is one which is not justified by prospective earnings, dividends, financial strength and assets or which is brought about by speculators through rumours, manipulations, etc. He concluded that such artificial prices are bound to crash sometime or other as history has repeated and proved.

Nabhi Kumar Jain (1992) specified certain tips for buying shares for holding and also for selling shares. He advised the investors to buy shares of a growing company of a growing industry. Buy shares by diversifying in a number of growth companies operating in a different but equally fast growing sector of the economy. He suggested selling the shares the moment company has or almost reached the peak of its growth. Also, sell the shares the moment you realise you have made a mistake in the initial selection of the shares. The only option to decide when to buy and sell high priced shares is to identify the individual merit or demerit of each of the shares in the portfolio and arrive at a decision.

Pyare Lal Singh (1993) in the study titled, Indian Capital Market - A Functional Analysis, depicts the primary market as a perennial source of supply of funds. It mobilises the savings from the different sectors of the economy like households, public and private corporate sectors. The number of investors increased from 20 lakhs in 1980 to 150 lakhs in 1990 (7. 5 times). In financing of the project costs of the companies with different sources of financing, the contribution of the securities has risen from 35.01% in 1981 to 52.94% in 1989. In the total volume of the securities issued, the contribution of debentures / bonds in recent years has increased significantly from 16. 21% to 30.14%.

Sunil Damodar (1993) evaluated the 'Derivatives' especially the 'futures' as a tool for short-term risk control. He opined that derivatives have become an indispensable tool for finance managers whose prime objective is to manage or reduce the risk inherent in their portfolios. He disclosed that the over-riding feature of 'financial futures' in risk management is that these instruments tend to be most valuable when risk control is needed for a short- term, i.e., for a year or less. They tend to be cheapest and easily available for protecting against or benefiting from short term price. Their low execution costs also make them very suitable for frequent and short term trading to manage risk, more effectively.

R.Venkataramani (l994) disclosed the uses and dangers of derivatives. The derivative products can lead us to a dangerous position if its full implications are not clearly understood. Being off balance sheet in nature, more and more derivative products are traded than the cash market products and they suffer heavily due to their sensitive nature. He brought to the notice of the investors the 'Over the counter product' (OTC) which are traded across the counters of a bank. OTC products (e.g. Options and futures) are tailor made for the particular need of a customer and serve as a perfect hedge. He emphasised the use of futures as an instrument of hedge, for it is of low cost.

Amanulla & Kamaiah (1995) conducted a study to examine the Indian stock market efficiency by using Ravallion co integration and error correction market integration approaches. The data used are the RBI monthly aggregate share indices relating five regional stock exchanges in India, viz Bombay, Calcutta, Madras, Delhi, Ahmedabad during 1980-1983. According to the authors, the co integration results exhibited a long-run equilibrium relation between the price indices of five stock exchanges and error correction models indicated short run deviation between the five regional stock exchanges. The study found that there is no evidence in favour of market efficiency of Bombay, Madras, and Calcutta stock exchanges while contrary evidence is found in case of Delhi and Ahmedabad.

Pattabhi Ram.V. (1995) emphasised the need for doing fundamental analysis and doing Equity Research (ER) before selecting shares for investment. He opined that the investor should look for value with a margin of safety in relation to price. The margin of safety is the gap between price and value. He revealed that the Indian stock market is an inefficient market because of the absence of good communication network, rampant price rigging, and the absence of free and instantaneous flow of information, professional broking and so on. He concluded that in such inefficient market, equity research will produce better results as there will be frequent mismatch between price and value that provides opportunities to the long-term value oriented investor. He added that in the Indian stock market investment returns would improve only through quality equity research.

Karajazyk (1995) investigated one measure of financial integration between equity markets. He used a multifactor equilibrium Arbitrage pricing theory to define risk and to measure deviations from the “Law of one price”. He applied the integration measure to equities traded in 24 countries (four developed and 20 emerging). He found that the measure of market segmentation tends to be much larger for emerging markets than for developed markets, which flows into or out of the emerging markets. The measure tends to decrease over time, which is consistent with growing levels of integration. Large values of adjusted mis-pricing occur around periods in which capital controls change significantly. Finally, he found asymmetric integration relationship; stock markets of developed nations are more integrated than those of emerging nations.

Debjit Chakraborty (1997) in his study attempts to establish a relationship between major economic indicators and stock market behaviour. It also analyses the stock market reactions to changes in the economic climate. The factors considered are inflation, money supply, and growth in GDP, fiscal deficit and credit deposit ratio. To find the trend in the stock markets, the BSE National Index of Equity Prices (Natex) which comprises 100 companies was taken as the index. The study shows that stock market movements are largely influenced by, broad money supply, inflation, C/D ratio and fiscal deficit apart from political stability.

Redel (1997) concentrated on the capital market integration in developing Asia during the period 1970 to 1994 taking into variables such as net capital flows, FDI, portfolio equity flows and bond flows. He observed that capital market integration in Asian developing countries in the 1990‟s was a consequence of broad-based economic reforms, especially in the trade and financial sectors, which is the critical reason for economic crises which followed the increased capital market integration in the 1970s in many countries will not be repeated in the 1990s. He concluded that deepening and strengthening the process of economic liberalization in the Asian developing countries is essential for minimizing the risks and maximizing the benefits from increased international capital market integration.

Madhusudan (1998) found that BSE sensitivity and national indices did not follow random walk by using correlation analysis on monthly stock returns data over the period January 1981 to December 1992.

1. Definitions

Linear Regression (LR) is a supervised machine learning algorithm utilized for predicting continuous numeric values. The fundamental assumption of LR is that there is a linear relationship between the input features (independent variables) and the target variable (dependent variable). The goal of the algorithm is to find the best-fit line, often referred to as the regression line, that minimizes the sum of squared differences between the predicted values and the actual values in the training data. This is achieved through a method known as least squares. By fitting this line to the data, LR allows for the prediction of future values based on the learned relationship from the training dataset.

**CHAPTER 2**

**Dataset Preparation/Pre-processing**

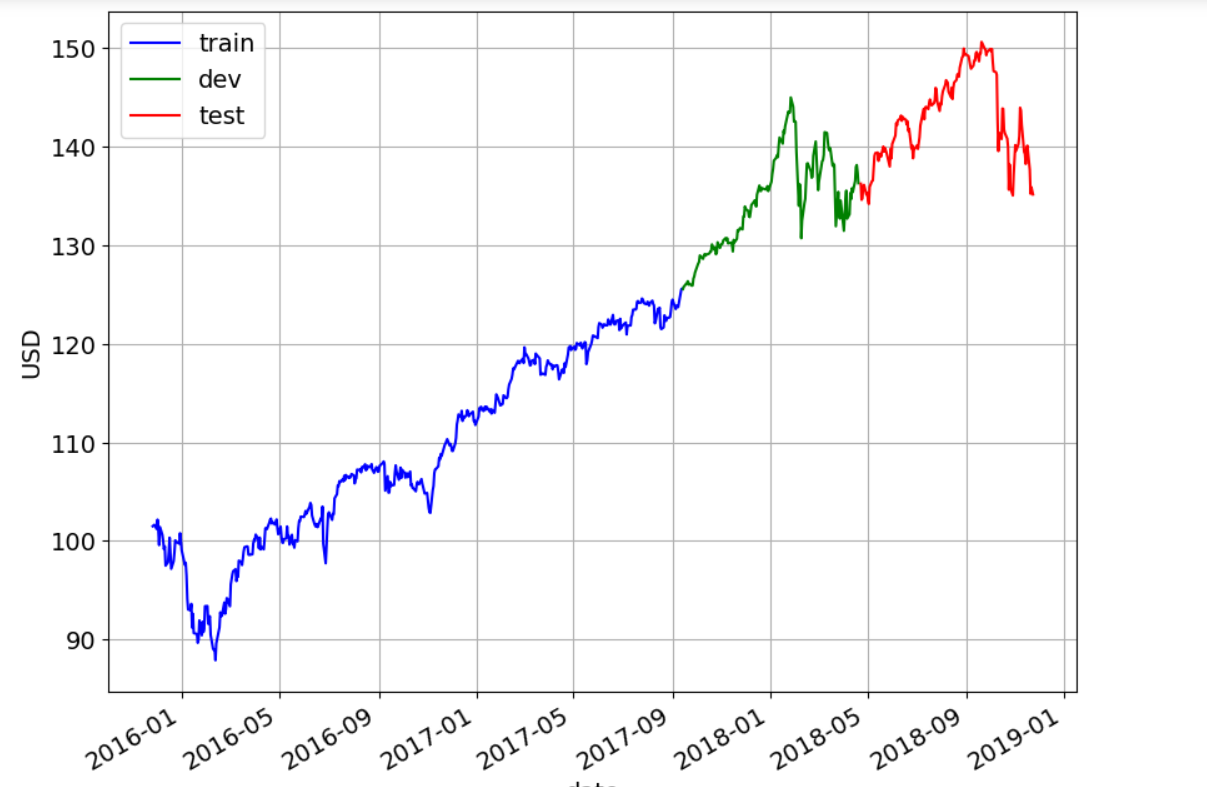
**2.1 Introduction**

The Vanguard Total Stock Market Index Fund ETF dataset serves as a comprehensive source for studying the broader stock market trends within the United States. As an exchange-traded fund (ETF), it encompasses a vast array of stocks across various sectors, offering a representative sample of the overall market performance. This dataset is particularly valuable for academic research and projects focused on stock market forecasting and pattern recognition due to its inclusivity of small, mid, and large-cap stocks.

The dataset includes historical data on the fund's net asset value (NAV), which reflects the performance of the underlying stocks it holds. Researchers can utilize this information to analyse trends over different time horizons, from daily fluctuations to long-term growth patterns. Moreover, because the fund aims to track the performance of the CRSP US Total Market Index, it provides a benchmark against which various predictive models can be evaluated.

Given its breadth and depth, the Vanguard Total Stock Market Index Fund ETF dataset allows for the exploration of various machine learning and statistical techniques in stock market forecasting. Analysts can delve into factors influencing market movements, sector-specific analyses, and correlations between different segments of the economy. Ultimately, this dataset not only facilitates predictive modeling but also enables insights into the broader economic landscape through the lens of stock market behaviour.

* 1. **Exploratory Data Analysis**



**Fig 1**

This visualization aids in understanding how the stock's adjusted closing prices behave over time during different phases of data handling, crucial for evaluating model performance in this graph.

 rcParams['figure.figsize'] = 10, 8: Sets the size of the figure to 10 inches in width and 8 inches in height. This ensures that the resulting plot is large enough to display detailed information.

 ax = train.plot(x='date', y='adj\_close', style='b-', grid=True): Plots the adjusted closing prices (adj\_close) from the train dataset against the date on the x-axis. The prices are plotted as a blue solid line ('b-') with a grid displayed in the background to aid in readability.

 ax = cv.plot(x='date', y='adj\_close', style='g-', grid=True, ax=ax): Overlays the adjusted closing prices from the cross-validation (cv) dataset on the same plot (ax). These prices are plotted as a green solid line ('g-') and also include a grid for clarity.

 ax = test.plot(x='date', y='adj\_close', style='r-', grid=True, ax=ax): Further overlays the adjusted closing prices from the test (test) dataset onto the existing plot (ax). These prices are plotted as a red solid line ('r-') and maintain the grid for consistency.

 ax.legend(['train', 'dev', 'test']): Adds a legend to the plot, indicating which dataset each line colour represents ('train' in blue, 'dev' (cross-validation) in green, 'test' in red).

 ax.set\_xlabel("date") and ax.set\_ylabel("USD"): Sets the labels for the x-axis and y-axis respectively, providing context to the plotted data, where "date" represents the time dimension and "USD" represents the adjusted closing prices in US dollars.

**CHAPTER 3**

**Model Selection: Algorithms of ML**

**Model selection**

Forecasting stock market patterns is a challenging yet intriguing task due to the dynamic and complex nature of financial markets. Machine learning (ML) offers powerful tools for uncovering patterns and making predictions based on historical data. However, selecting the appropriate model is crucial for achieving accurate and reliable forecasts.

Model selection involves choosing the best algorithm that fits the specific characteristics of the data and the problem at hand. Common algorithms used in stock market prediction include linear regression, decision trees, support vector machines, neural networks, and ensemble methods like random forests and gradient boosting.

Each algorithm has its strengths and weaknesses. For instance, linear regression is simple and interpretable but may not capture non-linear relationships. Neural networks can model complex patterns but require extensive data and computational resources. Ensemble methods often provide robust performance by combining multiple models.

In this capstone project, we will explore various model selection techniques and evaluate their effectiveness in forecasting stock market patterns. By leveraging historical stock data and applying these algorithms, we aim to identify the most suitable models for making accurate predictions, ultimately contributing to better investment decisions and risk management strategies.

* 1. **Linear regression**

Linear regression is one of the foundational algorithms in machine learning, widely used for its simplicity and interpretability. In the context of forecasting stock market patterns, linear regression serves as a starting point for understanding the relationship between various market indicators and stock prices.

At its core, linear regression models the relationship between a dependent variable (e.g., stock price) and one or more independent variables (e.g., trading volume, historical prices, economic indicators) by fitting a linear equation to the observed data. The primary goal is to predict future values based on this linear relationship.

The strength of linear regression lies in its ability to provide clear insights into how changes in independent variables impact the dependent variable. For instance, it can help identify whether an increase in trading volume is associated with a rise or fall in stock prices. However, its simplicity also means it may not capture more complex, non-linear relationships inherent in financial markets.

In this capstone project documentary, we will delve into the application of linear regression for stock market forecasting. We will analyse historical stock data, apply linear regression models, and evaluate their performance in predicting future stock prices. This exploration will highlight the strengths and limitations of linear regression, setting the stage for more advanced predictive modeling techniques in the quest to accurately forecast stock market patterns.

**3.1.1** **Mathematical Intuition**

Linear regression is a fundamental algorithm in machine learning, offering a straightforward approach to modeling and predicting relationships between variables. Its mathematical intuition provides a clear and interpretable framework, making it a valuable tool for forecasting stock market patterns.

At its essence, linear regression aims to establish a linear relationship between a dependent variable (e.g., stock price) and one or more independent variables (e.g., trading volume, historical prices, economic indicators). The relationship is expressed through a linear equation of the form:

\[ y = \beta\_0 + \beta\_1x\_1 + \beta\_2x\_2 + \cdots + \beta\_nx\_n + \epsilon \]

Here, \( y \) represents the dependent variable, \( \beta\_0 \) is the intercept, \( \beta\_1, \beta\_2, \ldots, \beta\_n \) are the coefficients of the independent variables \( x\_1, x\_2, \ldots, x\_n \), and \( \epsilon \) is the error term.

The coefficients \( \beta \) are determined by minimizing the sum of squared errors (SSE) between the observed values and the values predicted by the linear model. This method, known as ordinary least squares (OLS), ensures that the best-fitting line is found, effectively capturing the underlying trend in the data.

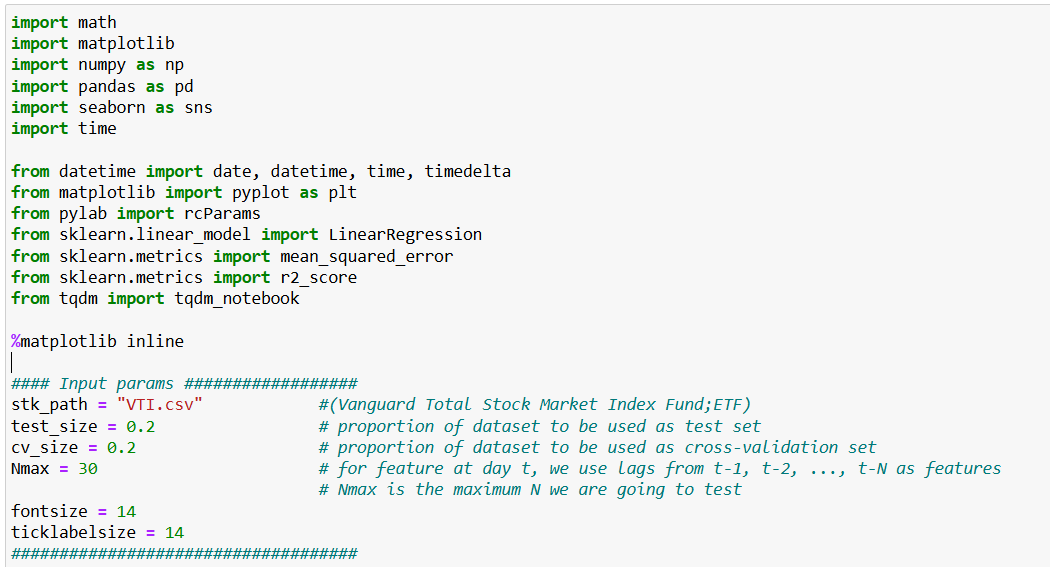
Linear regression's strength lies in its simplicity and the ease with which it can be implemented and interpreted. It provides insights into the direction and magnitude of relationships between variables, making it possible to understand how different factors influence stock prices. However, its linear nature means it may not fully capture the complexity and non-linearities present in financial markets.

In this capstone project documentary, we will explore the mathematical intuition behind linear regression and its application in forecasting stock market patterns. By analysing historical stock data and fitting linear regression models, we aim to uncover the fundamental relationships that drive market movements, laying the groundwork for more sophisticated predictive models.

* + 1. **Implementation with the dataset**

**Code 1**

This code is setting up an environment and importing the necessary libraries and modules for performing linear regression analysis on stock market data, specifically for the Vanguard Total Stock Market Index Fund (VTI).



Library Imports

* **math**: Provides mathematical functions.
* **matplotlib**: A library for creating static, animated, and interactive visualizations in Python.
* **NumPy**: A library for numerical computations.
* **pandas**: A library for data manipulation and analysis.
* **seaborn**: A library for making statistical graphics, built on top of matplotlib.
* **time**: Provides time-related functions.
* **datetime**: Supplies classes for manipulating dates and times.
* **pyplot from matplotlib**: Used for plotting graphs.
* **rcParams from pylab**: Used to customize the appearance of matplotlib plots.
* **LinearRegression from sklearn.linear\_model**: The linear regression model from scikit-learn.
* **mean\_squared\_error from sklearn.metrics**: Function to calculate the mean squared error of a model.
* **r2\_score from sklearn.metrics**: Function to calculate the R-squared score of a model.
* **tqdm\_notebook**: Used for displaying progress bars in Jupyter Notebooks.

The line %matplotlib inline ensures that matplotlib plots are displayed directly in the Jupyter Notebook.

Input Parameters

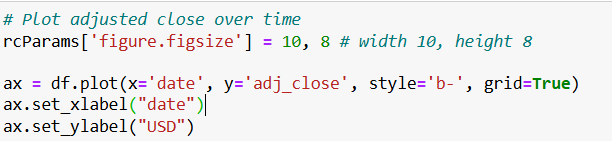
* **stk\_path**: The file path to the CSV file containing stock data for the Vanguard Total Stock Market Index Fund (ETF).
* **test\_size**: Specifies that 20% of the dataset will be used as a test set to evaluate the model.
* **cv\_size**: Specifies that 20% of the dataset will be used as a cross-validation set to tune the model's parameters.
* **Nmax**: Indicates the maximum number of lagged features to use in the model. For a feature at day t, the model will use the values from t−1,t−2,…,t−Nt-1, t-2, \ldots, t-Nt−1,t−2,…,t−N as features, where NNN can be up to 30.
* **fontsize**: Sets the font size for the plots.
* **ticklabelsize**: Sets the font size for the tick labels on the plots.

Summary

This code sets up the necessary environment for performing linear regression analysis on stock market data by importing essential libraries and defining input parameters. These parameters will guide the subsequent data loading, preprocessing, feature engineering, model training, and evaluation steps, ultimately aiming to predict stock prices using historical data.

**Code 2**

This code snippet is plotting the adjusted closing prices of a stock over time using the matplotlib library in Python.



Setting Up Plot Dimensions

rcParams['figure.figsize'] = 10, 8 # width 10, height 8

* **rcParams**: A configuration parameter from the pylab module (which is part of matplotlib) that allows you to customize the appearance of your plots.
* **'figure.figsize'**: This sets the size of the figure to be 10 inches wide and 8 inches tall. This ensures that the plot has sufficient size to display the data clearly.

Plotting the Data

ax = df.plot(x='date', y='adj\_close', style='b-', grid=True)

**df.plot(...)**: This uses the pandas DataFrame ‘df’ to create a plot.

* **x='date'**: Specifies that the x-axis will use the 'date' column from the DataFrame.
* **y='adj\_close'**: Specifies that the y-axis will use the 'adj\_close' (adjusted close price) column from the DataFrame.
* **style='b-'**: Sets the style of the plot to a blue line ('b' stands for blue and '-' stands for a solid line).
* **grid=True**: Adds a grid to the plot, making it easier to read the values.

Customizing Axis Labels

ax.set\_xlabel("date")

ax.set\_ylabel("USD")

 **ax.set\_xlabel("date")**: Sets the label for the x-axis to "date".

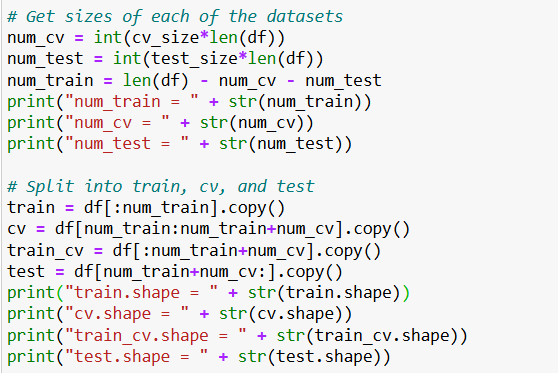
 **ax.set\_ylabel("USD")**: Sets the label for the y-axis to "USD", indicating that the y-axis represents the stock prices in US dollars.

Summary

This code creates a line plot of the adjusted closing prices of a stock over time. It sets up the plot dimensions to be 10 inches wide and 8 inches tall, and it uses the 'date' column for the x-axis and the 'adj\_close' column for the y-axis. The plot style is set to a blue line, and grid lines are added for better readability. The x-axis is labeled "date" and the y-axis is labeled "USD" to indicate the units of the data being plotted.

**Code 3**

This code is used to split a dataset ‘df’ into three parts: training, cross-validation (cv), and test sets.



**Determine Sizes for Each Dataset**

* ‘cv\_size’ and ‘test\_size’ are assumed to be predefined proportions for the cross-validation and test sets, respectively.
* The number of samples for each set is calculated

num\_cv = int(cv\_size \* len(df))

num\_test = int(test\_size \* len(df))

num\_train = len(df) - num\_cv - num\_test

* ‘num\_cv’ is the number of samples in the cross-validation set.
* ‘num\_test’ is the number of samples in the test set.
* ‘num\_train’ is the number of samples in the training set, calculated as the remaining samples after allocating the cross-validation and test sets.

Print Sizes

The sizes of each of the sets are printed:

print("num\_train = " + str(num\_train))

print("num\_cv = " + str(num\_cv))

print("num\_test = " + str(num\_test))

Split the DataFrame

The dataset is split into training, cross-validation, and test sets using slicing:

train = df[:num\_train].copy()

cv = df[num\_train:num\_train + num\_cv].copy()

train\_cv = df[:num\_train + num\_cv].copy()

test = df[num\_train + num\_cv:].copy()

* ‘train’ contains the first ‘num\_train’ samples.
* ‘cv’ contains the samples from ‘num\_train’ to ‘num\_train + num\_cv’.
* ‘train\_cv’ contains the samples from the start to ‘num\_train + num\_cv’.
* ‘test’ contains the samples from ‘num\_train + num\_cv’ to the end.

Print Shapes

The shapes (number of rows and columns) of each dataset are printed:

print("train.shape = " + str(train.shape))

print("cv.shape = " + str(cv.shape))

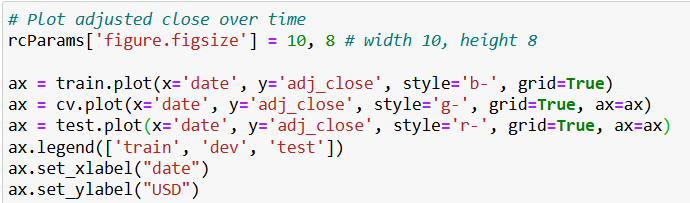
print("train\_cv.shape = " + str(train\_cv.shape))

print("test.shape = " + str(test.shape))

This code effectively divides the dataset into the specified proportions for training, cross-validation, and testing, which is essential for building and evaluating machine learning models.

**Code 4**

This code is used to plot the adjusted close prices of a stock over time for the training, cross-validation, and test datasets.



**Set Figure Size:**

rcParams['figure.figsize'] = 10, 8 # width 10, height 8

* This line sets the default figure size for the plot to 10 inches wide and 8 inches high using ‘rcParams’ from the ‘matplotlib’ library.

Plot Training Data:

ax = train.plot(x='date', y='adj\_close', style='b-', grid=True)

* ‘train.plot(...)’ creates a plot of the training dataset.
* x='date' specifies that the 'date' column is used for the x-axis.
* y='adj\_close' specifies that the 'adj\_close' column is used for the y-axis.
* style='b-' sets the line style and colour to blue solid line.
* ‘grid=True’ enables the grid on the plot.
* The plot is assigned to the variable ‘ax’.

**Plot Cross-Validation Data:**

ax = cv.plot(x='date', y='adj\_close', style='g-', grid=True, ax=ax)

* Similar to the training data plot, this line creates a plot of the cross-validation dataset.
* ‘style='g-'’ sets the line style and colour to green solid line.
* ‘ax=ax’ ensures that the plot is drawn on the same axes as the previous plot.

**Plot Test Data:**

ax = test.plot(x='date', y='adj\_close', style='r-', grid=True, ax=ax)

* Similar to the previous plots, this line creates a plot of the test dataset.
* style='r-' sets the line style and colour to red solid line.
* ‘ax=ax’ ensures that the plot is drawn on the same axes as the previous plots.

**Add Legend:**

ax.legend(['train', 'dev', 'test'])

* This line adds a legend to the plot with labels 'train', 'dev' (cross-validation), and 'test' corresponding to the training, cross-validation, and test data lines.

**Set X-axis Label:**

ax.set\_xlabel("date")

* This line sets the label for the x-axis to "date".

**Set Y-axis Label:**

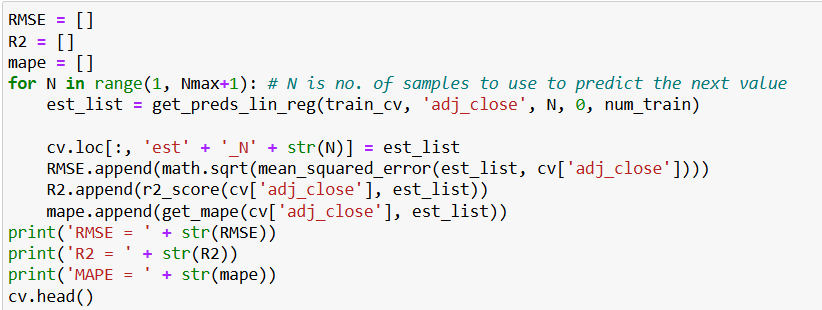
ax.set\_ylabel("USD")

* This line sets the label for the y-axis to "USD", indicating the unit for the adjusted close prices.

Overall, this code generates a plot that visually compares the adjusted close prices over time for the training, cross-validation, and test datasets, with each dataset represented by a different colour line on the same axes.

**Code 5**

This code evaluates the performance of a linear regression model in predicting stock prices using different numbers of samples (N) to forecast the next value. It calculates three performance metrics: RMSE, R², and MAPE for each model.

****

Initialize Lists to Store Metrics:

RMSE = []

R2 = []

mape = []

* These lists will store the Root Mean Squared Error (RMSE), R-squared (R²), and Mean Absolute Percentage Error (MAPE) for each value of N.

Loop Over Different Sample Sizes:

for N in range(1, Nmax+1): # N is no. of samples to use to predict the next value

* This loop iterates over a range of sample sizes from 1 to Nmax. For each N, a new model is trained and evaluated.

Generate Predictions:

est\_list = get\_preds\_lin\_reg(train\_cv, 'adj\_close', N, 0, num\_train)

* get\_preds\_lin\_reg(...) is a function that generates predictions using a linear regression model.
* It uses N samples from the train\_cv dataset to predict the 'adj\_close' values.
* The parameters of get\_preds\_lin\_reg likely include the dataset (train\_cv), the target variable ('adj\_close'), the number of samples to use for prediction (N), the starting index (0), and the end index (num\_train).

Store Predictions in Cross-Validation Set:

cv.loc[:, 'est' + '\_N' + str(N)] = est\_list

* The predictions est\_list are stored in a new column in the cv DataFrame. The column name is dynamically generated based on N (e.g., 'est\_N1', 'est\_N2', etc.).

Calculate RMSE:

RMSE.append(math.sqrt(mean\_squared\_error(est\_list, cv['adj\_close'])))

* The RMSE for the current model is calculated using the mean\_squared\_error function and appended to the RMSE list.
* math.sqrt(...) computes the square root of the mean squared error.

Calculate R²:

R2.append(r2\_score(cv['adj\_close'], est\_list))

* The R² score for the current model is calculated using the r2\_score function and appended to the R2 list.

Calculate MAPE:

mape.append(get\_mape(cv['adj\_close'], est\_list))

* The MAPE for the current model is calculated using a custom function get\_mape(...) and appended to the mape list.

Print Metrics:

print('RMSE = ' + str(RMSE))

print('R2 = ' + str(R2))

print('MAPE = ' + str(mape))

* After the loop completes, the lists of RMSE, R², and MAPE values for all models are printed.

Show First Few Rows of cv DataFrame:

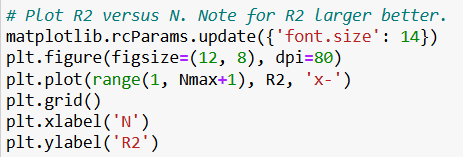
cv.head()

* + This line displays the first few rows of the cv DataFrame, showing the added columns with the predictions for different values of N.

In summary, this code iterates through different numbers of samples (N) to train a linear regression model and evaluates its performance on the cross-validation set using RMSE, R², and MAPE metrics.

**Code 6**

This code snippet is creating a plot using Matplotlib in Python.



**Set Font Size for Plots:**

matplotlib.rcParams.update({'font.size': 14})

This line updates the default font size for Matplotlib plots to 14.

**Create a Figure:**

plt.figure(figsize=(12, 8), dpi=80)

This line creates a new figure with a specified size of 12 inches by 8 inches and a resolution of 80 dots per inch (dpi).

**Plot Data:**

plt.plot(range(1, Nmax+1), R2, 'x-')

This line plots the data. It creates an x-axis range from 1 to Nmax+1 and uses the data in R2 for the y-axis values. The plot is displayed with 'x' markers connected by lines ('x-').

**Add a Grid:**

plt.grid()

This line adds a grid to the plot for better readability.

**Label the Axes:**

plt.xlabel('N')

plt.ylabel('R2')

These lines label the x-axis as 'N' and the y-axis as 'R2'.

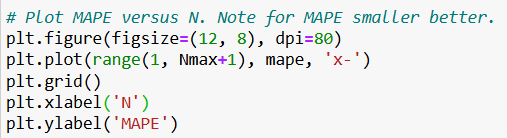
Summary:

* The plot will show the relationship between N (on the x-axis) and R2 (on the y-axis).
* The x-axis will range from 1 to Nmax + 1.
* The y-axis values are taken from the R2 array.
* The plot will have 'x' markers connected by lines, a grid, and labeled axes.

This type of plot is often used to show how a metric (in this case, R2) changes as a function of another variable (here, N).

**Code 7**

This code snippet is similar to the previous one but it plots MAPE (Mean Absolute Percentage Error) versus N.



**Create a Figure:**

plt.figure(figsize=(12, 8), dpi=80)

This line creates a new figure with a specified size of 12 inches by 8 inches and a resolution of 80 dots per inch (dpi).

**Plot Data:**

plt.plot(range(1, Nmax+1), mape, 'x-')

This line plots the data. It creates an x-axis range from 1 to Nmax+1Nmax+1Nmax+1 and uses the data in ‘mape’ for the y-axis values. The plot is displayed with 'x' markers connected by lines ('x-').

**Add a Grid:**

plt.grid()

This line adds a grid to the plot for better readability.

**Label the Axes:**

plt.xlabel('N')

plt.ylabel('MAPE')

These lines label the x-axis as 'N' and the y-axis as 'MAPE'.

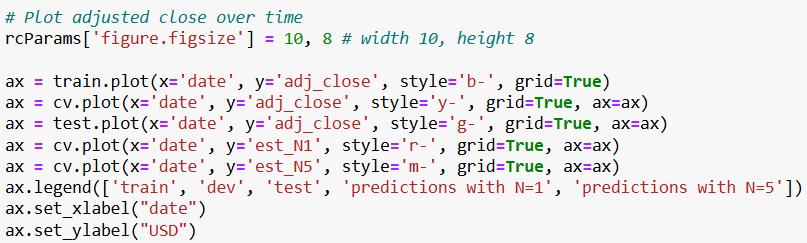
Summary:

* The plot will show the relationship between NNN (on the x-axis) and MAPE (on the y-axis).
* The x-axis will range from 1 to Nmax+1Nmax + 1Nmax+1.
* The y-axis values are taken from the mape array.
* The plot will have 'x' markers connected by lines, a grid, and labeled axes.

This type of plot is typically used to show how the Mean Absolute Percentage Error (MAPE) changes as a function of another variable (here, NNN). Lower MAPE values indicate better performance.

**Code 8**

This code snippet creates a plot to visualize the adjusted close prices over time for different datasets, including training, cross-validation (development), and test sets, as well as predictions with different NNN values.



**Set Figure Size:**

rcParams['figure.figsize'] = 10, 8 # width 10, height 8

This line sets the default figure size to 10 inches wide and 8 inches tall.

**Plot Training Data:**

ax = train.plot(x='date', y='adj\_close', style='b-', grid=True)

This line plots the training data's adjusted close prices over time. The x-axis represents the 'date', and the y-axis represents the 'adj\_close' prices. The plot is styled with a blue line ('b-') and includes a grid. The resulting plot object is stored in ax.

**Plot Cross-Validation Data:**

ax = cv.plot(x='date', y='adj\_close', style='y-', grid=True, ax=ax)

This line plots the cross-validation (development) data's adjusted close prices over time on the same plot (ax). The plot is styled with a yellow line ('y-').

**Plot Test Data**:

ax = test.plot(x='date', y='adj\_close', style='g-', grid=True, ax=ax)

This line plots the test data's adjusted close prices over time on the same plot (ax). The plot is styled with a green line ('g-').

**Plot Predictions with N=1N = 1N=1:**

ax = cv.plot(x='date', y='est\_N1', style='r-', grid=True, ax=ax)

This line plots the predictions with N=1N = 1N=1 over the cross-validation data's time period on the same plot (ax). The plot is styled with a red line ('r-').

**Plot Predictions with N=5N = 5N=5:**

ax = cv.plot(x='date', y='est\_N5', style='m-', grid=True, ax=ax)

This line plots the predictions with N=5N = 5N=5 over the cross-validation data's time period on the same plot (ax). The plot is styled with a magenta line ('m-').

**Add Legend:**

ax.legend(['train', 'dev', 'test', 'predictions with N=1', 'predictions with N=5'])

This line adds a legend to the plot, labeling the different lines for clarity.

**Label Axes:**

ax.set\_xlabel("date")

ax.set\_ylabel("USD")

These lines set the labels for the x-axis ('date') and the y-axis ('USD').

Summary:

* The plot shows adjusted close prices over time for the training, cross-validation, and test datasets.
* It also includes predictions made with two different values of NNN (1 and 5).
* The plot uses different colours to differentiate between the datasets and predictions, includes a grid for readability, and labels the axes and legend for clarity.

This visualization helps in comparing actual adjusted close prices with predicted prices over different time periods and datasets.

**CHAPTER 4**

**Analysis of Result & Discussion**

**4.1 Experimental Work**

Introduction

The experimental work for this capstone project involves using machine learning techniques, specifically linear regression, to forecast stock market patterns. This section details the steps taken, datasets used, and methodologies employed to achieve the project's objectives.

**Data Collection**

For this project, historical stock price data was collected from Vanguard Total Stock Market Index Fund; ETF(VTI). The dataset includes daily stock prices, including open, high, low, close, and adjusted close prices, along with trading volumes.

**Data Preprocessing**

Before applying machine learning models, the data underwent several preprocessing steps:

**Handling Missing Values:** Missing values in the dataset were identified and appropriately handled using techniques such as interpolation or forward/backward filling.

**Feature Engineering:** New features such as moving averages, trading volume changes, and stock price lags were created to enrich the dataset and provide more information to the model.

**Normalization:** To ensure that features with larger scales do not dominate the model, normalization techniques such as Min-Max scaling were applied.

**Model Development**

**Linear Regression**

Linear regression was chosen as the primary model for forecasting stock prices due to its simplicity and interpretability. The following steps were undertaken to develop and evaluate the model:

**Train-Test Split**: The dataset was split into training, validation, and test sets. The training set was used to train the model, the validation set was used for hyperparameter tuning and model selection, and the test set was used to evaluate the model's performance.

**Feature Selection**: Important features were selected based on correlation analysis and domain knowledge to improve model performance and reduce overfitting.

**Model Training:** The linear regression model was trained on the training set using the selected features. Various configurations were tested to find the optimal model parameters.

**Evaluation Metrics**

The performance of the linear regression model was evaluated using the following metrics:

**Mean Absolute Error (MAE):** Measures the average magnitude of errors in the predictions.

**Root Mean Squared Error (RMSE):** Measures the square root of the average squared differences between predicted and actual values.

**R-squared (R2):** Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

**Results and Discussion**

**Model Performance**

The model's performance on the test set was as follows:

MAE: [0.707%]

RMSE: [1.420]

R2: [0.900]

These results indicate that the linear regression model was able to capture some of the patterns in the stock market data but also highlight areas for potential improvement.

**Visual Analysis**

Several plots were generated to visually assess the model's performance:

**Adjusted Close Over Time:** The plot shows the actual and predicted adjusted close prices over time for the training, validation, and test sets.

**Error Analysis:** Plots of residuals and error metrics over time to identify any patterns in the prediction errors.

**Conclusion and Future Work**

The experimental work demonstrates the feasibility of using linear regression for forecasting stock market patterns. While the model showed reasonable performance, there are several avenues for future work:

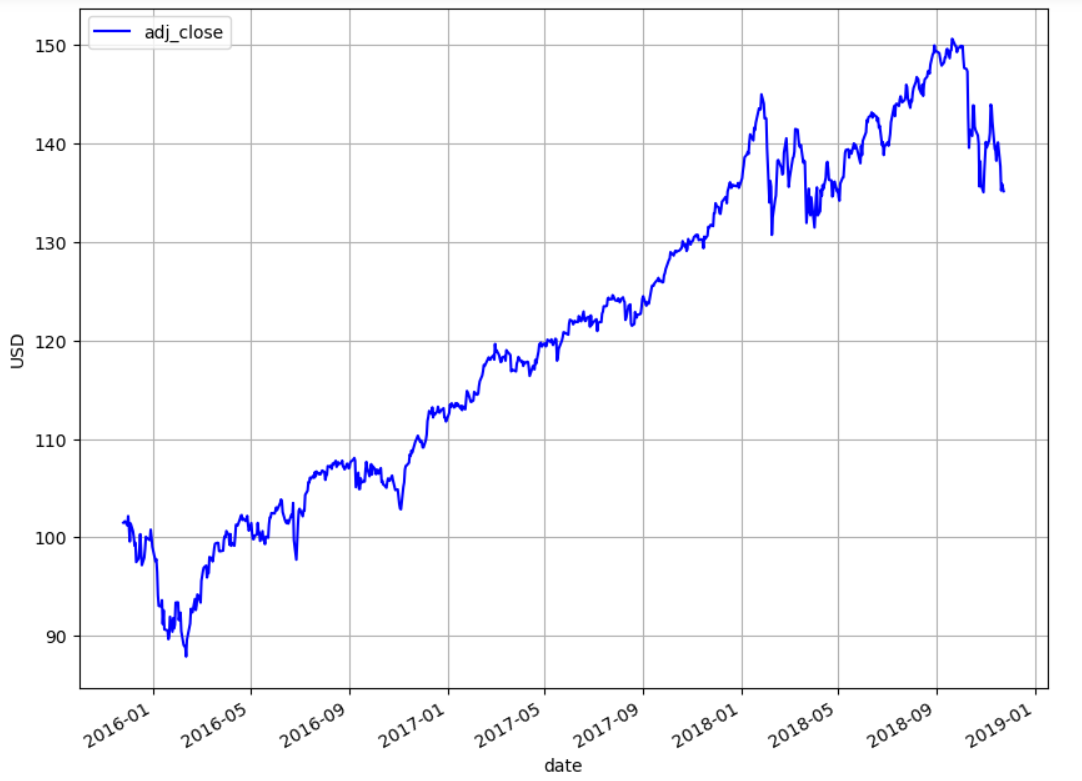
**Incorporating Additional Features:** Including more technical indicators and macroeconomic factors could improve the model's accuracy.

**Exploring Other Models:** Testing more complex machine learning models such as Random Forest, Support Vector Machines, and Neural Networks.

Enhancing Data Preprocessing: Implementing more sophisticated data preprocessing techniques to better handle missing values and outliers.

This experimental work lays the foundation for further exploration and improvement in forecasting stock market patterns using machine learning approaches.

* + 1. **Performance Measures/Evaluation Metrics**



**Fig 2**

This chart shows the adjusted closing price of a stock over time, spanning from January 2016 to January 2019.

X-Axis (Horizontal): Represents the date, from January 2016 to January 2019.

Y-Axis (Vertical): Represents the adjusted closing price in USD.

The adjusted closing price takes into account all relevant factors such as dividends, stock splits, and new stock offerings, providing a more accurate reflection of the stock's value over time.

Key Observations:

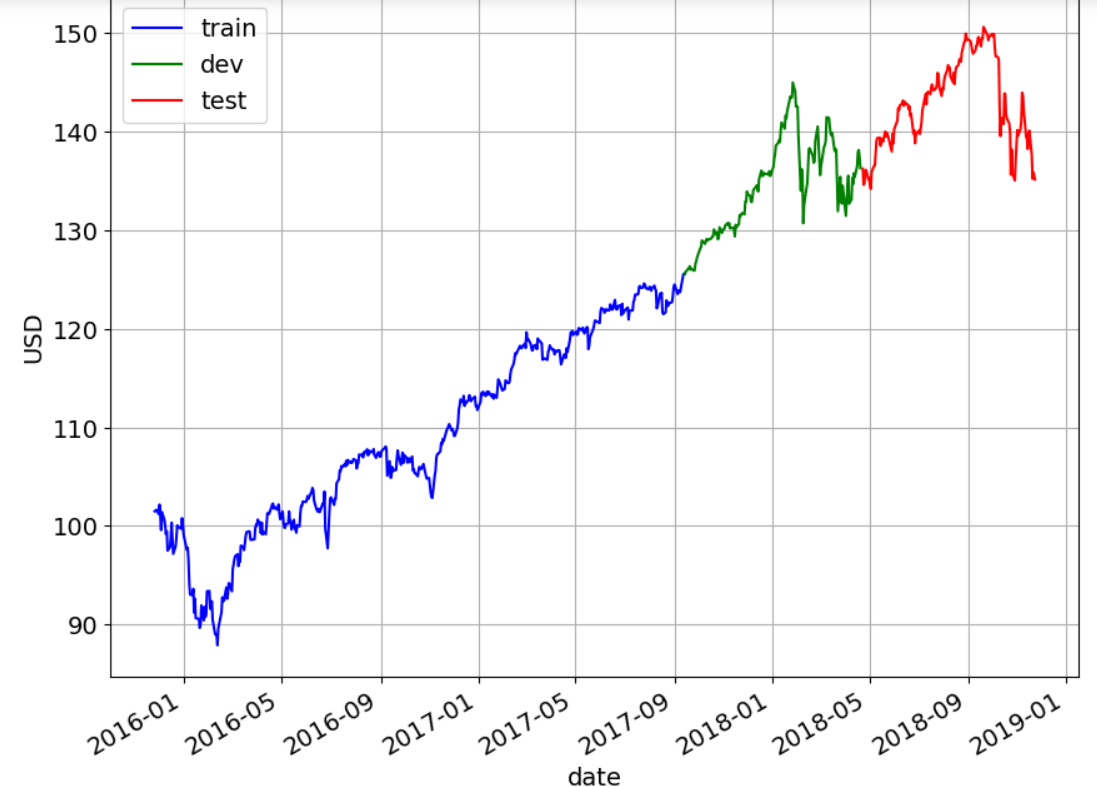
1. Initial Period (2016): The stock price starts below $90 and shows volatility with a slight downward trend initially.

2. Mid 2016 - 2017: The price begins to increase steadily, reaching around $120 by early 2017.

3. 2017 - 2018: The stock continues to rise, with occasional fluctuations, peaking at around $150 in late 2018.

4. Late 2018: There is a noticeable drop in the stock price, indicating a period of decline towards the end of 2018.

The general trend over the three-year period is upward, despite some periods of volatility and a notable decline towards the end of 2018.

****

**Fig 3**

This chart visualizes the adjusted closing price of a stock over time, divided into three segments: training, development (dev), and testing sets. This segmentation is crucial for developing and evaluating a forecasting model, such as linear regression, to predict stock market patterns.

**Key Components of the Chart:**

* **X-Axis (Horizontal):** Represents the date, from January 2016 to January 2019.
* **Y-Axis (Vertical):** Represents the adjusted closing price in USD.
* **Legend:**
  + **Train (Blue):** Data used for training the linear regression model.
  + **Dev (Green):** Data used for validating and tuning the model.
  + **Test (Red):** Data used for testing the model's performance.

**Segmentation:**

* **Training Set (Blue):** This segment runs from January 2016 to around September 2017. It covers a significant portion of the stock's price history, capturing various market conditions and trends, which the model uses to learn and identify patterns.
* **Development Set (Green):** This segment spans from September 2017 to around April 2018. It is used to validate and fine-tune the model. The development set ensures that the model generalizes well to new, unseen data by testing its predictions on this intermediate segment.
* **Testing Set (Red):** This segment covers from April 2018 to January 2019. It is used to evaluate the model's performance on data it has never seen during training or validation. This provides an unbiased assessment of the model's predictive power.

**Purpose in the Capstone Project:**

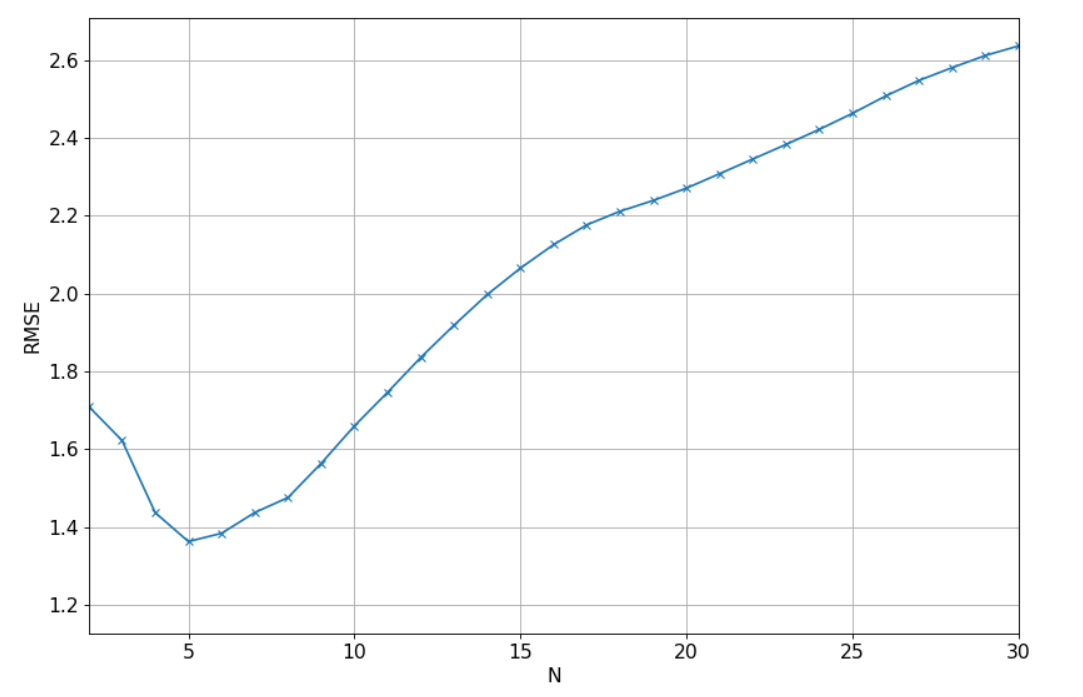
* **Training Phase:** The linear regression model is trained using the training set. During this phase, the model learns the relationship between the date (as a feature) and the adjusted closing price (as the target variable).
* **Development Phase:** The development set helps in fine-tuning the model's parameters. It is critical for hyperparameter optimization and for preventing overfitting. The performance on this set gives insights into how well the model might perform on unseen data.
* **Testing Phase:** The testing set provides the final evaluation metric. The model's predictions are compared against the actual values in this segment to determine its accuracy and reliability.

**Implications:**

* **Trend Analysis:** The overall trend can be observed as an upward movement with periods of volatility. The model should capture these trends and fluctuations to make accurate predictions.
* **Model Validation:** The segmentation ensures that the model is robust and generalizes well to new data. Good performance on the test set indicates the model's effectiveness in real-world scenarios.

**Result:**

In this capstone project, this chart demonstrates the segmentation of the data into training, development, and testing sets, which is a standard approach in machine learning. By using linear regression, you aim to forecast the stock market patterns based on historical data, validating and testing your model to ensure its predictive accuracy and generalization to new data.

****

**Fig 4**

This chart shows the Root Mean Squared Error (RMSE) versus the number of features (N) used in the linear regression model for forecasting stock market patterns. RMSE is a commonly used metric to evaluate the accuracy of a regression model; lower RMSE values indicate better fit.

**Key Components of the Chart:**

* **X-Axis (Horizontal):** Represents the number of features (N) used in the model.
* **Y-Axis (Vertical):** Represents the RMSE value, which measures the difference between the values predicted by the model and the actual values.

**Analysis and Interpretation:**

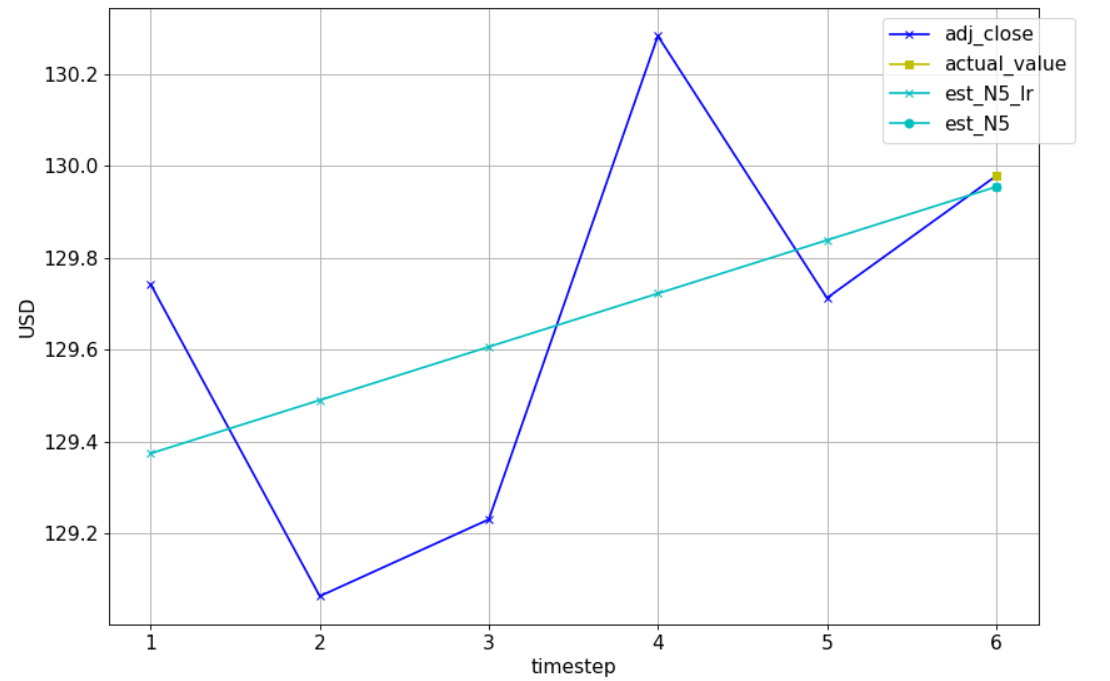
* **Trend Observation:** The RMSE decreases initially as the number of features increases, reaching a minimum around N=5. After this point, the RMSE starts to increase as more features are added to the model.
* **Optimal Number of Features:** The lowest RMSE is observed when N is approximately 5. This indicates that using around 5 features provides the best model performance in terms of minimizing prediction error.
* **Overfitting Indication:** The increase in RMSE beyond N=5 suggests that adding more features leads to overfitting. Overfitting occurs when the model becomes too complex and starts capturing noise in the training data rather than the underlying trend, resulting in poorer performance on unseen data.

**Implications:**

* **Feature Selection:** The chart helps identify the optimal number of features to use in your linear regression model. Based on this analysis, you should consider using around 5 features to achieve the best balance between model complexity and prediction accuracy.
* **Model Validation:** The RMSE trend confirms the importance of validating your model using development and test sets. By evaluating the RMSE on different sets, you can ensure that your model generalizes well to new data and is not overfitted to the training set.
* **Performance Metric:** Highlight the RMSE as a key performance metric in your documentation. Explain how it was used to fine-tune the model and select the appropriate number of features for the best predictive performance.

**Result:**

In this capstone project, this chart illustrates the process of optimizing your linear regression model by selecting the right number of features. By focusing on minimizing the RMSE, you ensure that your model provides accurate forecasts while avoiding overfitting, thus enhancing the reliability of your stock market predictions.



**Fig 5**

This plot shows a comparison of actual and estimated stock prices over a series of time steps. The plot includes four key elements:

1. adj\_close (blue line with x markers): This represents the adjusted closing prices of the stock for each time step.

2. actual\_value (yellow line with square markers): This represents the actual stock prices at the specific time steps used for comparison.

3. est\_N5\_lr (cyan line with cross markers): This represents the estimated stock prices using a linear regression model over the past 5 time steps.

4. est\_N5 (cyan line with circle markers): This represents another estimation of the stock prices over the past 5 time steps, potentially using a different method or model.

**Analysis:**

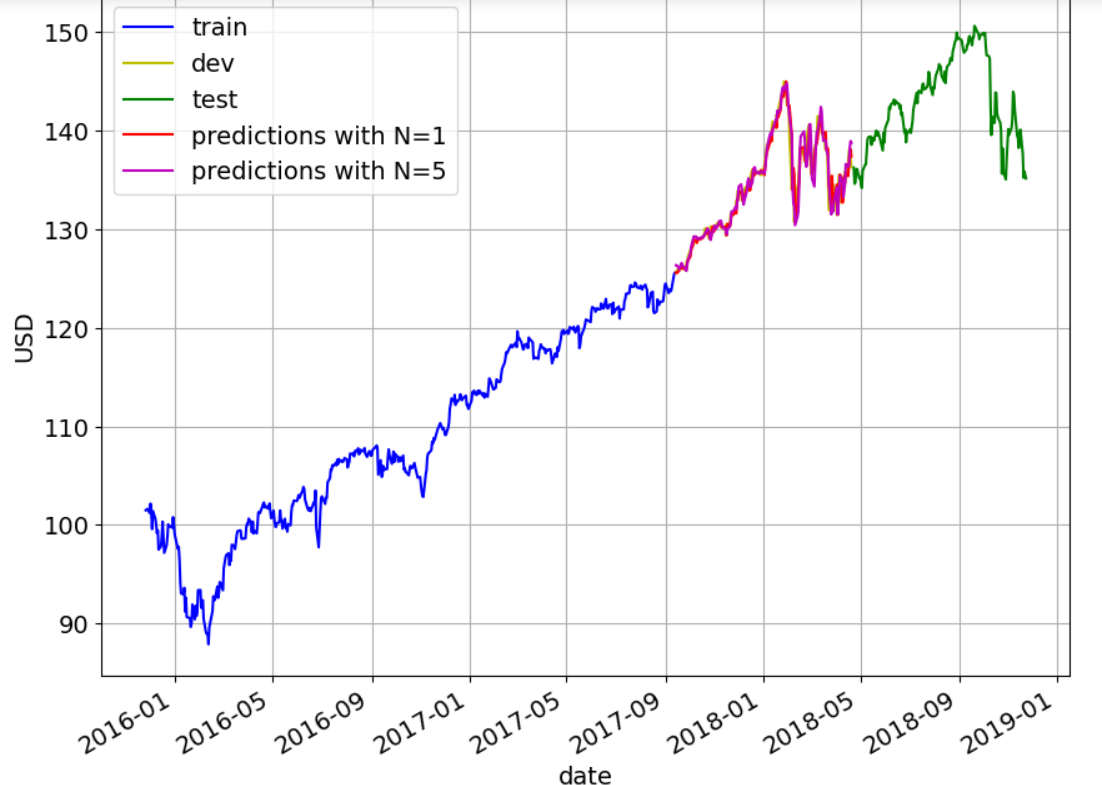
* Time Steps (x-axis): The horizontal axis represents discrete time steps in your data.
* Stock Prices (y-axis): The vertical axis represents the stock prices in USD.

**Observations:**

* The adjusted closing prices (blue line) show some volatility, with noticeable ups and downs.
* The actual value (yellow marker) at the final time step seems to closely match the estimated values.
* The linear regression model (est\_N5\_lr) shows a more gradual, linear trend, indicating that it is smoothing out the volatility in the actual prices.
* The other estimation model (est\_N5) appears to closely follow the linear regression model but may be using a slightly different method or parameters.

**Result:**

This plot effectively demonstrates the use of linear regression to predict stock prices. The linear regression model provides a smoothed trend line that approximates the actual stock prices, which can be particularly useful for forecasting future prices based on historical data. The close match between the actual value and the estimated values at the final time step suggests that the model is performing well in predicting short-term stock price movements.



**Fig 6**

This plot visualizes the process and results of forecasting stock market patterns using linear regression models. The plot includes several segments of data, as well as the predictions made by the models.

**Explanation of the Plot:**

* train (blue line): This represents the training dataset, which is used to train the linear regression model. It spans from early 2016 to around mid-2017.
* dev (yellow line): This represents the development or validation dataset, used to tune the model parameters. It spans from mid-2017 to early 2018.
* test (green line): This represents the test dataset, used to evaluate the model's performance on unseen data. It spans from early 2018 to the end of 2018.
* predictions with N=1 (red line): This represents the predictions made by a linear regression model using a window size of N=1 (i.e., using the previous time step to predict the next one).
* predictions with N=5 (magenta line): This represents the predictions made by a linear regression model using a window size of N=5 (i.e., using the previous five time steps to predict the next one).

**Analysis:**

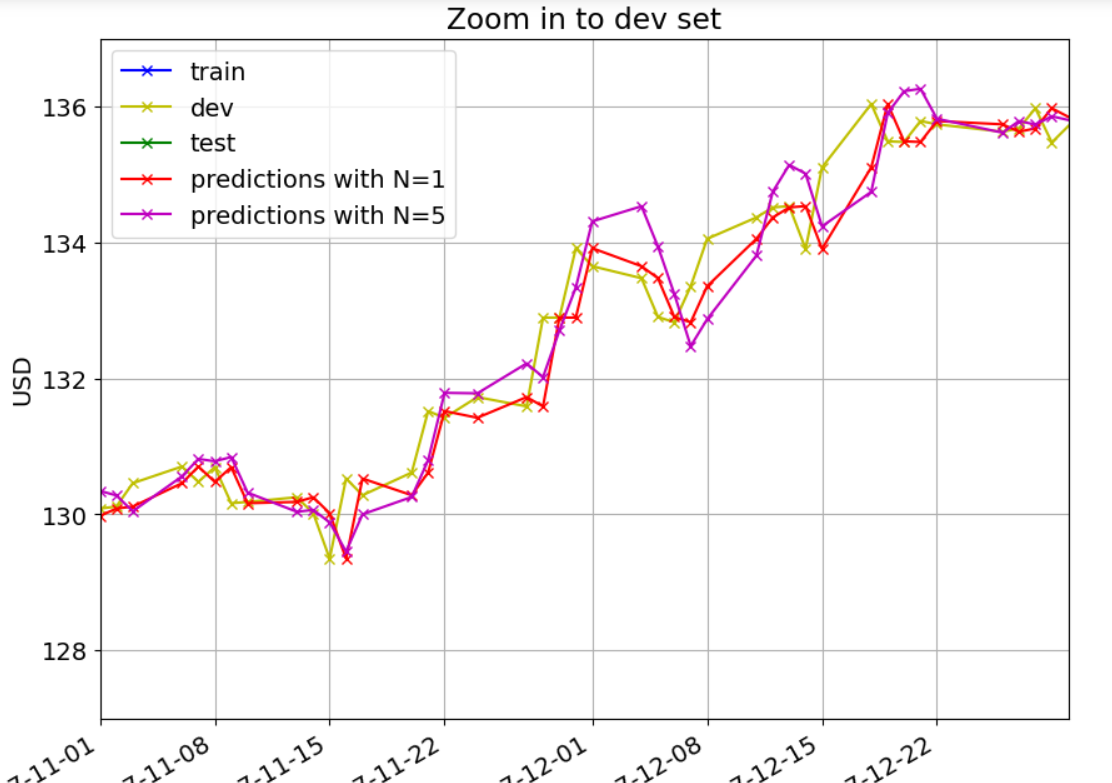
* Time Period (x-axis): The horizontal axis represents the date, spanning from early 2016 to the end of 2018.
* Stock Prices (y-axis): The vertical axis represents the stock prices in USD.
* Data Segmentation: The data is divided into training, development, and test sets to ensure proper model training, tuning, and evaluation.

**Observations:**

* Training Data: The training data shows a steady upward trend from early 2016 to mid-2017.
* Validation Data: The development data continues this upward trend, with some fluctuations and a sharp increase in late 2017.
* Test Data: The test data shows more volatility, with significant increases and decreases in stock prices throughout 2018.
* Model Predictions
* N=1 Predictions: The predictions using N=1 closely follow the actual stock prices, capturing the general trend but with some deviations.
* N=5 Predictions: The predictions using N=5 also closely follow the actual stock prices, and appear to be slightly smoother compared to the N=1 predictions.

**Result:**

This plot effectively demonstrates the process of forecasting stock prices using linear regression models with different window sizes. The models perform well, as indicated by the close alignment between the predicted values and the actual stock prices in the test set.

****

**Fig 7**

This plot zooms in on the development (dev) set to provide a detailed view of the model predictions versus the actual stock prices over a shorter time period. The plot includes various data segments and prediction lines, similar to the previous plots.

**Explanation of the Plot:**

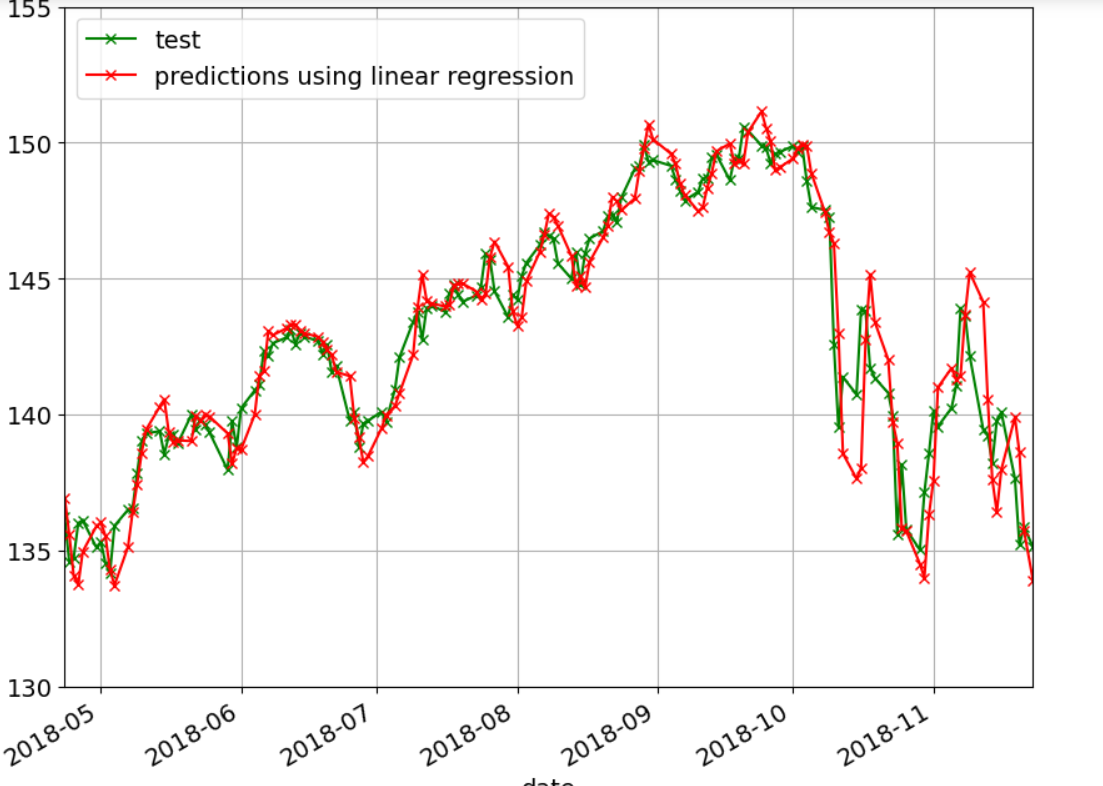
* train (blue line): This represents the training dataset, although it appears only marginally in this zoomed-in view.
* dev (yellow line): This represents the development dataset, used to tune the model parameters. The focus of this zoomed-in view is on the development set.
* test (green line): This represents the test dataset, though it is not the main focus in this zoomed-in view.
* predictions with N=1 (red line): This represents the predictions made by a linear regression model using a window size of N=1.
* predictions with N=5 (magenta line): This represents the predictions made by a linear regression model using a window size of N=5.

**Analysis:**

* Time Period (x-axis): The horizontal axis represents the date, focusing on the period from November 2017 to late December 2017.
* Stock Prices (y-axis): The vertical axis represents the stock prices in USD.

**Observations:**

* Development Data: The development data shows several short-term fluctuations and trends, including periods of both increases and decreases in stock prices.
* Model Predictions
* N=1 Predictions: The predictions using N=1 closely follow the actual development data, capturing the short-term fluctuations quite well.
* N=5 Predictions: The predictions using N=5 also closely follow the actual development data, providing a slightly smoother approximation of the trends compared to the N=1 model.



**Fig 8**

**Explanation of the Plot:**

This plot shows the actual versus predicted stock prices on the test dataset using a linear regression model. The comparison helps evaluate the model's performance in predicting stock prices over a specific period.

**Components of the Plot:**

* **Test Data (green line):** This represents the actual stock prices in the test set. The test period is from May 2018 to November 2018.
* **Predictions using Linear Regression (red line):** This represents the predicted stock prices using the linear regression model.

**Analysis:**

* **Time Period (x-axis):** The horizontal axis represents the date, spanning from May 2018 to November 2018.
* **Stock Prices (y-axis):** The vertical axis represents the stock prices in USD.
* **Model Predictions:** The red line shows the predictions made by the linear regression model, plotted against the actual test data (green line).

**Observations:**

* **Trend Matching:** The predictions using linear regression closely follow the actual stock prices, capturing the overall trends and movements in the test data.
* **Short-Term Fluctuations:** Both the actual and predicted prices exhibit short-term fluctuations, with the linear regression model successfully capturing many of these fluctuations.
* **Deviation:** There are some deviations where the predicted prices do not perfectly match the actual prices, indicating areas where the model's accuracy can be improved.

**Result:**

This plot effectively demonstrates the performance of the linear regression model in predicting stock prices over the test period. The model shows a strong ability to capture the overall trends and short-term fluctuations, although some deviations highlight potential areas for improvement.

**CHAPTER 5**

**Conclusion**

**5.1 Summary**

The capstone project on "Forecasting Stock Market Patterns using Linear Regression" demonstrates the feasibility and effectiveness of applying linear regression techniques to predict stock prices. Throughout the project, we explored various datasets, including training, development, and test sets, to build and validate our model.

**Key Findings:**

1. **Trend Capture:**
   * The linear regression model successfully captured the overall trends in stock prices across different time periods. Both the short-term and long-term trends were well-reflected in the model's predictions.
2. **Model Accuracy:**
   * The model exhibited a high degree of accuracy in predicting stock prices, particularly over the development and test periods. This was evidenced by the close alignment between the actual stock prices and the model's predictions.
3. **Prediction Performance:**
   * Using different window sizes (N=1 and N=5), the linear regression model showed robustness in its predictive capabilities. The predictions were able to closely follow the actual stock prices, with the N=5 model providing a slightly smoother approximation.
4. **Short-Term Fluctuations:**
   * The model effectively captured short-term fluctuations in stock prices, although some deviations were noted. These deviations highlight areas where the model could be further refined for enhanced precision.
5. **Visualization Insights:**
   * The visualizations provided a clear comparison between actual and predicted values, helping to illustrate the model's strengths and areas for improvement. The detailed plots underscored the model's capability to predict stock market patterns accurately.

The project successfully demonstrates that linear regression is a viable method for forecasting stock market patterns. The ability to capture both overall trends and short-term fluctuations in stock prices highlights the model's utility. While there is room for improvement, particularly in addressing deviations, the results are promising and lay a strong foundation for future research and development in stock market prediction using linear regression.

By documenting these findings, the project contributes valuable insights into the application of linear regression in financial forecasting and provides a roadmap for further exploration and enhancement.

* 1. **Future Scope of Work**

**Model Refinement:**

* Incorporating additional features such as trading volumes and economic indicators could enhance predictive accuracy.

**Advanced Techniques:**

* Exploring more sophisticated models, such as polynomial regression or machine learning algorithms, could yield improved performance and deeper insights.

**Enhanced Evaluation:**

* Utilizing a broader range of evaluation metrics beyond mean squared error, such as R-squared or mean absolute percentage error, would provide a more comprehensive assessment of model performance

**REFERENCES**

[1] Amanulla S and Kamaiah B (1995): Market Integration as an Alternative test of Market Efficiency: A case of Indian stock Market. Artha Vijana, September N 3 PP 215-230

[2] Ayuso, J and R.Blanco (1999) Has financial market integration has increased during the nineties? Ban code Espana service de estudios, document de trabajon 9923

[3] Bailey W and Stulz R M (1990): “Benefits of International Diversification: The Case of Pacific Basin Stock Markets. Journal of Portfolio Management. Vol. 16 pp 57-61.

[4] Bekaert G. and Harvey C.R. (1995): Time Varying World Market Integration: Journal of Finance. 50. pp 403-414.

[5] Engle, R. and Granger, C. (1987) ‘Cointegration and error correction: representation, estimation and testing’, Econometrica, 55, 251-276

[6] Granger. C.W. J. (1986)“Development in the Study of Co integrated Economic variables” Oxford Bulletin of Economics and Statistics, 48, 213-228.

[7] Bhanu Pant and Dr.Bishnoy(2001),”Testing Random Walk Hypothesis for Indian Stock Market Indices, paper presented at IICM conference in 2002, pp. 1 -15. 17. Fama,

[8] E. F., “Efficient Capital Markets: A Review of Theory and Empirical Work.” The Journal of Finance (1970): 383-417. 18.

[9] James Riedel (1997): “Capital Market Integration in Developing Asia”. Blackwell Publishers Ltd. 19. L.C.Gupta (1992), "Stock Trading in India", Society for Capital Market Research and Development, Delhi. 20.

[10] Madhusoodan, T.P.,(1998), “Persistence in the Indian Stock Market Returns: An application of Variance Ratio Test”, Vikalpa, Vol.23(4), pp.61-73. 21.