A
PROJECT REPORT
ON

“STOCK PRICE PREDICTION USING REGRESSION ALGORITHMS”

BY

Mayur Ramesh Nikam

Submitted in partial fulfilment of
MSc in Data Science
Under
Savitribai Phule Pune University,
For the academic year
2024-2025

UNDER THE GUIDANCE OF
Prof.(Er.) Tausif Shaikh

Department of Technology,
Savitribai Phule Pune University,
Ganeshkhind, Pune 411007

CERTIFICATE

This is to certify that

Mayur Ramesh Nikam

Have successfully completed their project on
“Stock Price Prediction using Regression Algorithms”
In partial fulfilment of
MSc in Data Science
Under
Department of Technology,
Savitribai Phule Pune University,
For the academic year
2024-2025

Dr. Manisha Bharti
(Course Coordinator)

Dr. Aditya Abhyankar
(H.O.D)

Prof.(Er.) Tausif Shaikh
(Project Guide)

Signed By
(External Examiner)

Place: Department of Technology, SPPU
Date: / /

DECLARATION

I undersigned Mayur Ramesh Nikam the student of MSc Data Science, Department of Technology, Savitribai Phule Pune University. Declare that the research project titled “Stock Price Prediction using Regression Algorithms” is a result of our own work and our indebtedness to other work publications, references, if any have been duly acknowledged. If I found guilty of copying any other report or published information and showing as our original work, we understand that we shall be liable and punishable by Institute or University, which may include Failure in examination, Repeat study and re-submission of the report or any other punishment that institute or University may decide.

Mayur Ramesh Nikam
(DS24MS20)

Signature:

ACKNOWLEDGEMENT

We would like to extend our sincere appreciation to the each and every in-dividual and the Department of Technology who played pivotal roles in the successful completion of this project. First and foremost, we are deeply grateful to our project guides Prof.(Er) Tausif Shaikh for their exceptional guidance and mentorship, which significantly enriched the project's depth and direction. We are also very thankful for the timely guidance and motivation of Dr Manisha Bharti ma'am. To our peers, collaborators, and the open-source community, thank you for your contributions, insightful discussions, and assistance in fostering a collaborative research environment. We appreciate the efforts of all those involved in data collection and sharing, as well as the reviewers and anyone is engaged with the project, offering valuable feedback and insights.

Finally, our thanks to the readers and tribes of Maharashtra who will benefit from the findings presented herein. Your collective support, expertise, and encouragement were indispensable on this journey. We are also thankful to the HOD Dr. Aditya Abhyankar sir for his guidance and motivation for the project. Last but not the least we would thank each and everyone directly or indirectly involved in the project.

ABSTRACT

This project investigates the use of various regression algorithms to forecast future closing prices of India’s benchmark stock index, NIFTY 50. The primary objective is to develop and compare predictive models using historical stock data and evaluate their accuracy. We apply five different supervised learning methods – **Linear Regression**, **Decision Tree**, **Random Forest**, **Support Vector Regression (SVR)**, and **K-Nearest Neighbors (KNN)** – to historical NIFTY 50 data and compute forecast performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2). Exploratory data analysis (EDA) and preprocessing steps (such as normalization and train–test splitting) are performed to prepare the data. The results show that ensemble models like Random Forest produce lower error and higher R^2 compared to simpler models, indicating better prediction accuracy. The findings are interpreted in light of the models’ capacity to capture non-linear patterns in financial time series. We discuss the challenges of stock market forecasting, including volatility and efficient market effects. Future work could explore deep learning approaches (e.g. LSTM networks) and incorporate additional features to improve performance. The study demonstrates the applicability of machine learning to financial forecasting and provides a foundation for further research.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

OVERVIEW

In recent times stock market predictions is gaining more attention, maybe due to the fact that if the trend of the market is successfully predicted the investors may be better guided. The profits gained by investing and trading in the stock market greatly depends on the predictability. If there is a system that can consistently predict the direction of the dynamic stock market will enable the users of the system to make informed decisions. More over the predicted trends of the market will help the regulators of the market in taking corrective measures.

Predicting stock prices is a challenging and important task in finance. The NIFTY 50 index of the National Stock Exchange (NSE) of India is a benchmark that reflects the performance of the country's top 50 companies from diverse sectors. Accurate forecasting of NIFTY 50 stock prices can aid investors, traders, and portfolio managers in decision making and risk management Unlike fundamental analysis, which examines financial indicators, technical forecasting uses historical price and volume data to predict future prices. Recent advances in machine learning (ML) have enabled data-driven prediction models that may capture complex patterns in financial time series.

This thesis focuses on predicting the closing price of the NIFTY 50 index using classical regression techniques. The main **objectives** are to (1) preprocess and analyze historical NIFTY 50 price data, (2) implement several supervised ML regression algorithms (Linear Regression, Decision Tree Regression, Random Forest Regression, Support Vector Regression, and K-Nearest Neighbors Regression), and (3) evaluate and compare their predictive performance using standard metrics. We restrict our study to historical price and volume features. The **scope** includes data from [specify data range, e.g. 2010–2022] and daily closing prices. The **significance** of this work lies in providing a comparative analysis of simple ML models for stock forecasting, which can be a foundation for more advanced approaches. As the repository of NIFTY 50 price prediction notes, such forecasts "aid investors, traders, and analysts in making informed decisions. Throughout this thesis, we ensure that procedures are reproducible and results are rigorously evaluated.

PROBLEM STATEMENT

Examine a number of different forecasting techniques to predict future stock returns based on past returns and numerical news indicators to construct a portfolio of multiple stocks in order to diversify the risk.

OBJECTIVE

- To create take a dataset of renowned company. Feature extraction using fundamental analysis Applying reduced dataset.
- Evaluating accuracy.
- Plotting and analyzing the graph.

STOCK MARKET

A stock market, equity market or share market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange as well as those only traded privately. Examples of the latter include shares of private companies which are sold to investors through equity crowd funding platforms. Stock exchanges list shares of common equity as well as other security types, e.g. corporate bonds and convertible bonds.

Stock price prediction is one of the most widely studied problem, attracting researchers from many fields. The volatile nature of the stock market makes it really difficult to apply simple time-series or regression techniques. Financial institutions and active traders have created various proprietary models to beat the market for themselves or their clients, but rarely did anyone achieve consistently higher than the average returns on investment. The challenge of stock market price forecasting is so appealing because an improvement of just a few points of percentage can increase the profit by millions of dollars. This paper discusses the application of Support Vector Machines and Linear Regression in detail along with the pros and cons of the given methods. The paper introduces the parameters and variables which can be used to recognize the patterns in stock prices which can be helpful in future stock prediction and how boosting can be integrated with various other machine learning algorithms to improve the accuracy of our prediction system

CHAPTER 2

LITERATURE REVIEW

(This section would typically provide an overview of existing research in stock price prediction, covering traditional statistical models (e.g., ARIMA, GARCH), early machine learning applications (e.g., ANNs, SVMs), and more recent advancements in deep learning (e.g., LSTMs, Transformers). It would discuss the evolution of techniques, common challenges, and key findings in the field. As I do not have access to external research papers, this section will remain a placeholder for you to fill with relevant academic citations and discussions.)

Previous research in stock price prediction has explored a wide array of methodologies, ranging from fundamental analysis and technical analysis to advanced statistical and machine learning models. Early attempts often relied on linear models, but the non-linear characteristics of stock data quickly led to the adoption of more sophisticated approaches. Support Vector Machines (SVMs) gained prominence for their ability to handle high-dimensional data and non-linear relationships. Ensemble methods like Random Forests have also shown strong performance due to their robustness against overfitting and ability to capture complex interactions. More recently, deep learning techniques, particularly Recurrent Neural Networks (RNNs) and their variants like Long Short-Term Memory (LSTM) networks, have emerged as powerful tools for time-series forecasting, given their capacity to learn long-term dependencies in sequential data. This thesis builds upon these advancements by evaluating a diverse set of regression algorithms to provide a comparative analysis of their efficacy in the context of NSE stock data.

1. **Survey of stock market prediction using machine learning approach**

Authors: Ashish Sharma, Dinesh Bhuriya, Ankita Deshmukh
2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)

Stock market is basically nonlinear in nature and the research on stock market is one of the most important issues in recent years. People invest in stock market based on some prediction. For predict, the stock market prices people search such methods and tools which will increase their profits, while minimize their risks. Prediction plays a very important role in stock market business which is very complicated and challenging process. Employing traditional methods like fundamental and technical analysis may not ensure the reliability of the prediction. To make predictions regression analysis is used mostly. In this paper we survey of well-known efficient regression approach to predict the stock market price from stock market data based. In future the results of multiple regression approach could be improved using more number of variables.

2. **Combining of random forest estimates using LSboost for stock market index prediction**

Authors: Nonita Sharma ; Akanksha Juneja,2017 2nd International Conference for Convergence in Technology (I2CT)

This research work emphases on the prediction of future stock market index values based on historical data. The experimental evaluation is based on historical data of 10 years of two indices, namely, CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets. The predictions are made for 1-10, 15, 30, and 40 days in advance. This work proposes to combine the predictions/estimates of the ensemble of trees in a Random Forest using LSboost (i.e. LS-RF). The prediction performance of the proposed model is compared with that of well-known Support Vector Regression. Technical indicators are selected as inputs to each of the prediction models. The closing value of the stock price is the predicted variable. Results show that the proposed scheme outperforms Support Vector Regression and can be applied successfully for building predictive models for stock prices prediction.

3. **A forecast approach for securities exchange instability dependent on time arrangement information**

Authors: Mohmadd khan, M. Asraaafi Alam, Parul Goel

This paper tells why not the notion of series of time analysis and forecasting shoul be correct in the respect of Indian economy. The main fall of the currency in the previous times had lead to the important need. The paper not only tries to make an efficient Model but also to guess the Indian stock market volatility. The available always time series data of Indian stock market has been used for this study. The analysed time series has been compared with the original time series, which shows roughly a deviation of 5% mean percentage mistake for both Nifty and Sensex on average. Different tests can be tried for the validation of the predicted time series.

CHAPTER 3

SYSTEM ANALYSIS

Third chapter includes system development. We have mentioned how the model and our project have been evolved over the time. We have drawn several flow charts for model extraction and system extraction for better understanding of the project

INTRODUCTION

Design is a multi- step that focuses on data structure software architecture, procedural details, procedure etc… and interface among modules. The design procedure also decodes the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design change continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at relatively primary stage in its revolution.
Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However, methods for software designs do exit, criteria for design qualities are existing and design notation can be applied.

SYSTEM REQUIREMENTS

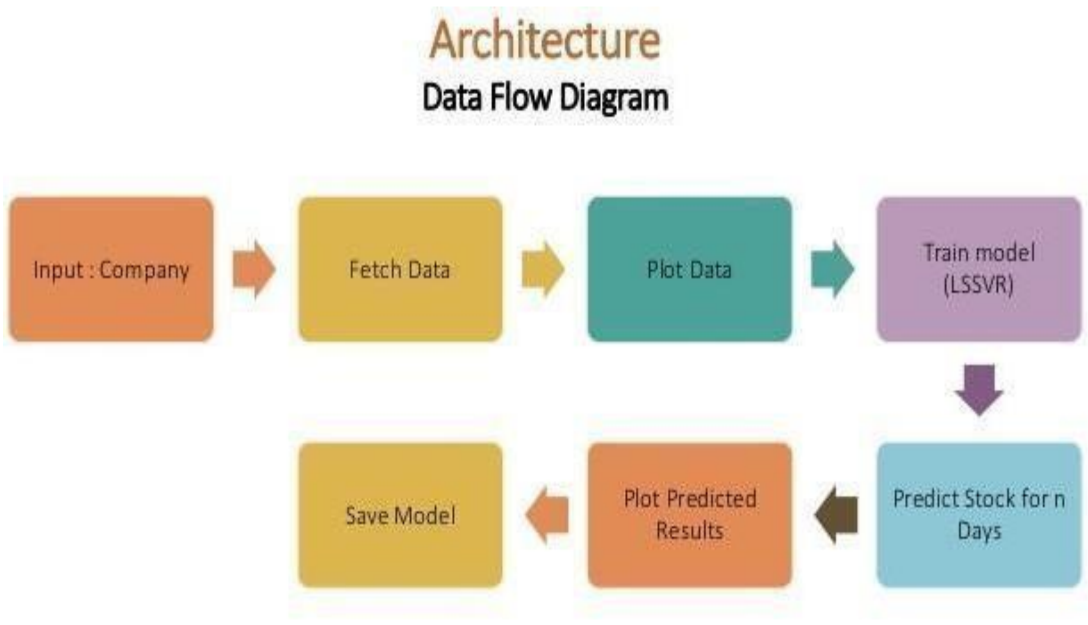
Hardware Requirements:

- Processor: pentium iv
- Ram: 8 gb
- Processor: 2.4 ghz
- Main memory: 8gb ram
- Processing speed: 600 mhz
- Hard disk drive: 1tb
- Keyboard :104 keys

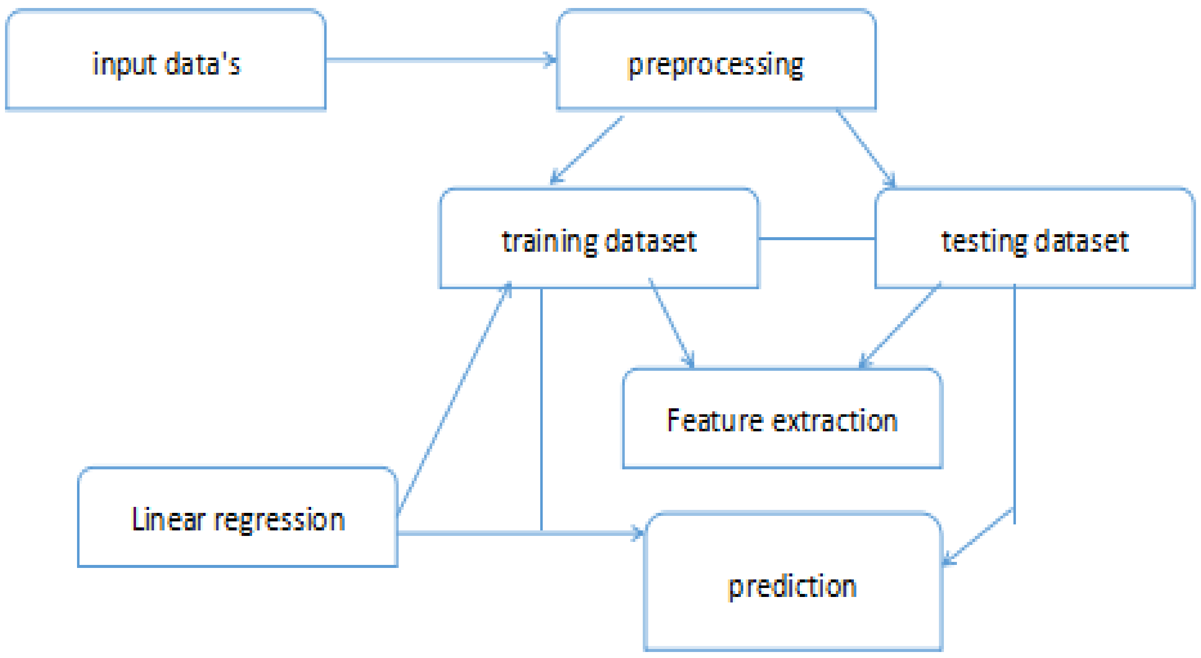
Software Requirments:

- Front End: Python
- IDE: Jupyter Notebook
- Operating System: Windows 11

ARCHITECTURE



Data Flow Diagram



Architecture Design

MODULE DESCRIPTION

The implementation of this project is divided into following steps
1. Data Preprocessing
2. Feature selection
3. Building and Traning Model

1. **Data Preprocessing:**
   The entries are present in the dataset. The null values are removed using df = df.dropna() where df is the data frame. The categorical attributes (Date, High, Low, Close, Adj value) are converted into numeric using Label Encoder. The date attribute is splitted into new attributes like total which can be used as feature for the model.

2. **Feature selection:**
   Features selection is done which can be used to build the model. The attributes used for feature selection are Date,Price,Adj close,Forecast X coordinate , Y coordinate, Latitude , Longitude, Hour and month.

3. **Building and Training Model:**
   After feature selection location and month attribute are used for training. The dataset is divided into pair of xtrain ,ytrain and xtest, y test. The algorithms model is imported form skleran. Building model is done using model. Fit (xtrain, ytrain). This phase would involve supervised classification methods like linear regression, Ensemble classifiers (like Adaboost, Random Forest Classifiers), etc.

PYTHON TECHNOLOGY

Python is an interpreted, object- oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX- based systems, Mac OS, MS- DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users. 13 A notable feature of Python is its indenting of source statements to make the code easier to read. Python offers dynamic data type, ready- made class, and interfaces to many system calls and libraries. It can be extended, using the C or C++language. Python can be used as the script in Microsoft's Active Server Page (ASP) technology. The scoreboard system for the Melbourne (Australia) Cricket Ground is written in Python. Z Object Publishing Environment, a popular Web application server, is also written in the Python language’s 4.5.1 Python Platform Apart from Windows, Linux and MacOS, C,Python implementation runs on 21 different platforms. Python is a .NET framework based

Python implementation and it is capable of running in both Windows, Linux and in other environments where .NET framework is available. 4.5.2 Python Library Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is –"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." They are typically used to solve various types of life problems. In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries.

Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit- learn
- Matplotlib
- Pandas

NumPy:

NumPy is a very popular python library for large multi- dimensional array and matrix processing, with the help of a large collection of high- level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High- end libraries like TensorFlow uses NumPy internally for manipulation of Tensors. 4.5.2.2 SciPy: SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation. 4.5.2.3 Skikit: Skikit- learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit- learn supports most of the supervised and unsupervised learning algorithms. Scikit- learn can also be used for data- mining and data- analysis, which makes it a great tool who is starting out with ML.

SciPy:

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Scikit Learn:

learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit- learn supports most of the supervised and unsupervised learning algorithms. Scikit- learn can also be used for data- mining and data- analysis, which makes it a great tool who is starting out with ML.

Matplotlib:

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats etc.

Pandas:

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In 16 this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high- level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

CHAPTER 4

METHODOLOGY

DATA COLLECTION

Historical stock data is the foundation of this project. The nsepython library is employed to extract key financial metrics, including:

- **Open Price:** The price at which a stock first trades when the market opens.
- **High Price:** The highest price at which a stock traded during a period.
- **Low Price:** The lowest price at which a stock traded during a period.
- **Close Price:** The final price at which a stock trades on a given day.
- **Date:** The trading date.

DATA PREPROCESSING

Raw financial data often contains inconsistencies, missing values, or incorrect data types that can adversely affect model performance. Therefore, a robust preprocessing pipeline is essential:

- **Handling Missing Values:** Identifying and addressing any missing data points through appropriate imputation strategies or removal.
- **Data Type Conversion:** Ensuring that all relevant columns are in the correct data format. For instance, Historical Date is converted to datetime objects for time-series analysis, and price columns are converted to float64.
- **Chronological Ordering:** The dataset is explicitly reversed to ensure that the data is in chronological order, which is critical for time-series forecasting.
- **Feature Engineering:** To enhance the predictive power of the models, additional features are engineered from the raw data. These features provide more context and insights into price movements:
- **Price Change Percentages:** Representing daily or periodic price fluctuations.
- **Trend Indicators:** Derived from price and volume data to indicate market direction.

DATA VISUALIZATION

Visualizing the data is crucial for understanding trends, patterns, and anomalies. Line plots are primarily used to observe the movement of stock prices over time, helping to identify volatility, seasonality, and overall market trends. Further visualizations, such as candlestick charts and correlation heatmaps, can be employed for deeper pattern analysis.

- **Line Plots:** Primarily used to visualize the time series of stock prices (Open, High, Low, Close) over the entire dataset, revealing trends, seasonality, and periods of high volatility.
- **Candlestick Charts:** Offer a more detailed view of daily price movements, including open, high, low, and close prices, along with trading volume, providing insights into market sentiment.
- **Correlation Heatmaps:** Used to visualize the correlation matrix between all features, helping to identify highly correlated features (for potential multicollinearity issues) and features strongly correlated with the target variable.
- **Distribution Plots:** Histograms or kernel density plots for individual features to understand their distribution and identify outliers.

CHAPTER 5

IMPLEMENTATION

MODEL BUILDING

A selection of regression algorithms is applied to the preprocessed and feature-engineered data. These models are chosen for their ability to capture different types of relationships within the data:

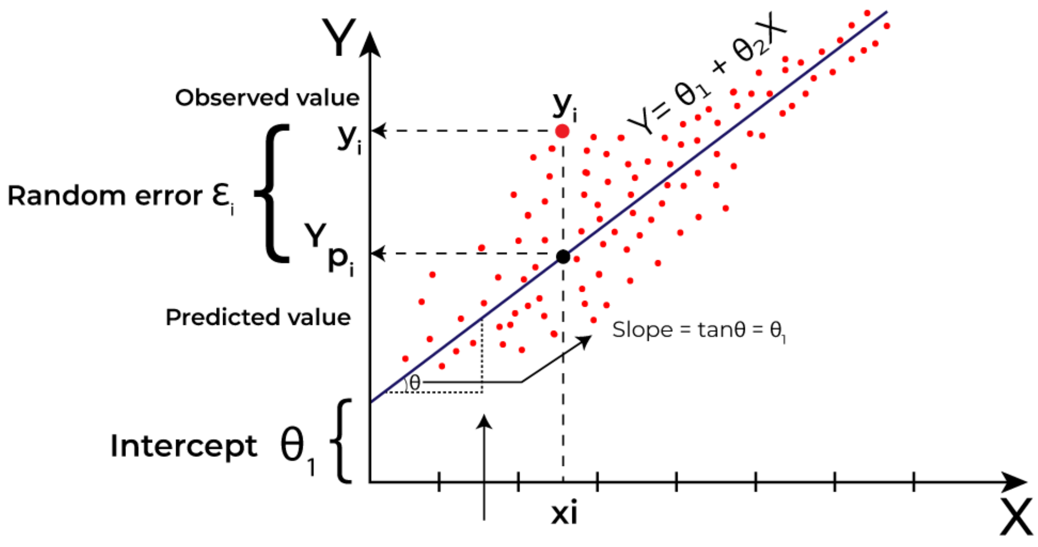
Linear Regression:

Linear regression is a type of supervised machine-learning algorithm that learns from the labelled datasets and maps the data points with most optimized linear functions which can be used for prediction on new datasets. It assumes that there is a linear relationship between the input and output, meaning the output changes at a constant rate as the input changes. This relationship is represented by a straight line.

For example We want to predict a student's exam score based on how many hours they studied. We observe that as students study more hours, their scores go up. In the example of predicting exam scores based on hours studied. Here

- **Independent variable (input):** Hours studied because it's the factor we control or observe.
- **Dependent variable (output):** Exam score because it depends on how many hours were studied.

We use the independent variable to predict the dependent variable.

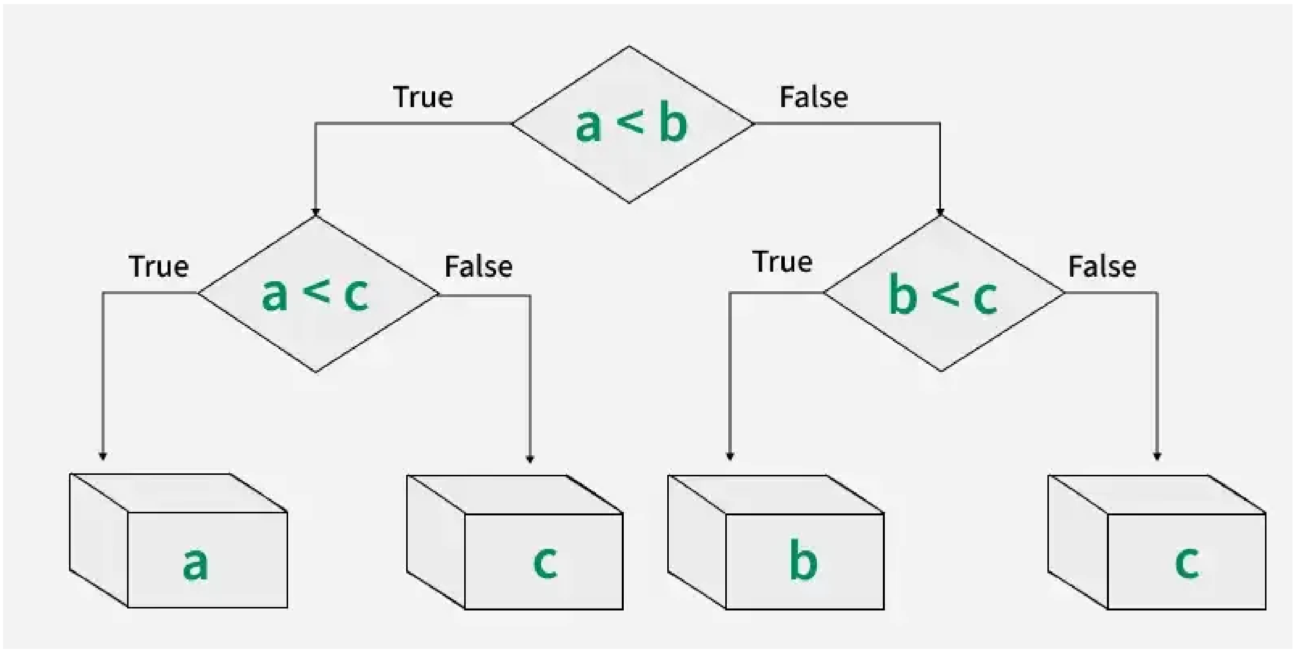


Decision Tree Regression:

A non-linear model that partitions the data into subsets based on feature values, creating a tree-like structure for prediction. It can capture complex, non-linear patterns.

Decision Tree Regression is a method used to predict continuous values like prices or scores by using a tree-like structure. It works by splitting the data into smaller parts based on simple rules taken from the input features. These splits help reduce errors in prediction. At the end of each branch, called a leaf node the model gives a prediction usually the average value of that group. In the tree:

- **Decision Nodes** (shown as diamonds) ask yes/no questions about the data, like “Is age greater than 50?”
- **Leaf Nodes** (shown as rectangles) give the final predicted number based on the data that reached that point.

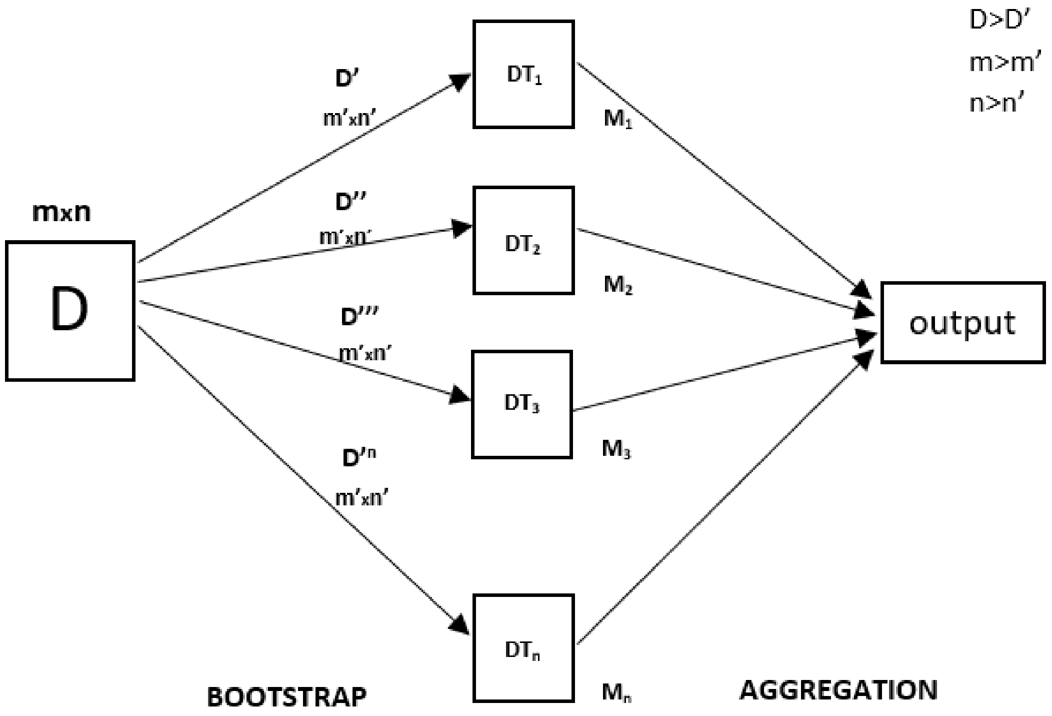


Random Forest Regression:

An ensemble learning method that constructs multiple decision trees during training and outputs the average of the individual tree predictions. This approach significantly improves accuracy and robustness by reducing overfitting.

Random Forest Regression works by creating multiple of decision trees each trained on a random subset of the data. The process begins with Bootstrap sampling where random rows of data are selected with replacement to form different training datasets for each tree. After this we do **feature sampling** where only a random subset of features is used to build each tree ensuring diversity in the models.

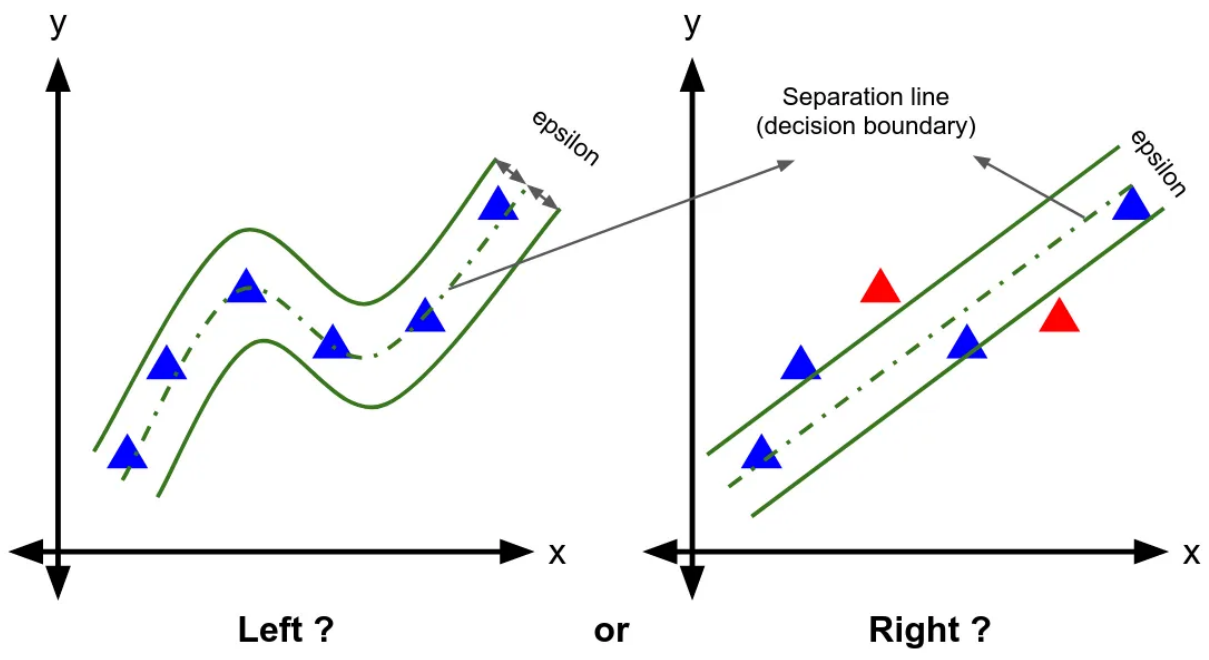
After the trees are trained each tree make a prediction and the final prediction for regression tasks is the average of all the individual tree predictions and this process is called as **Aggregation.**



This approach is beneficial because individual decision trees may have high variance and are prone to overfitting especially with complex data. However by averaging the predictions from multiple decision trees Random Forest minimizes this variance leading to more accurate and stable predictions and hence improving generalization of model.

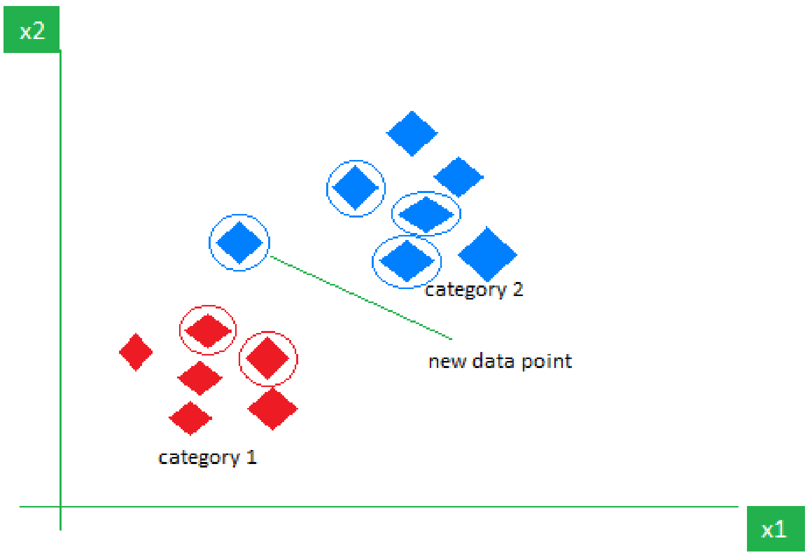
Support Vector Machine (SVM) for Regression (SVR):

A powerful and versatile algorithm that finds a hyperplane in a high-dimensional space to best fit the data points, allowing for both linear and non-linear regression.
Support vector regression (SVR) is a type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value. SVR can use both linear and non-linear kernels. A linear kernel is a simple dot product between two input vectors, while a non-linear kernel is a more complex function that can capture more intricate patterns in the data. The choice of kernel depends on the data's characteristics and the task's complexity.



K-Nearest Neighbors (KNN) Regression:

A non-parametric, instance-based learning algorithm that predicts the value of a new data point based on the average of the values of its 'k' nearest neighbors in the feature space.
KNN regression is a non-parametric method used for predicting continuous values. The core idea is to predict the target value for a new data point by averaging the target values of the K nearest neighbors in the feature space. The distance between data points is typically measured using Euclidean distance, although other distance metrics can be used



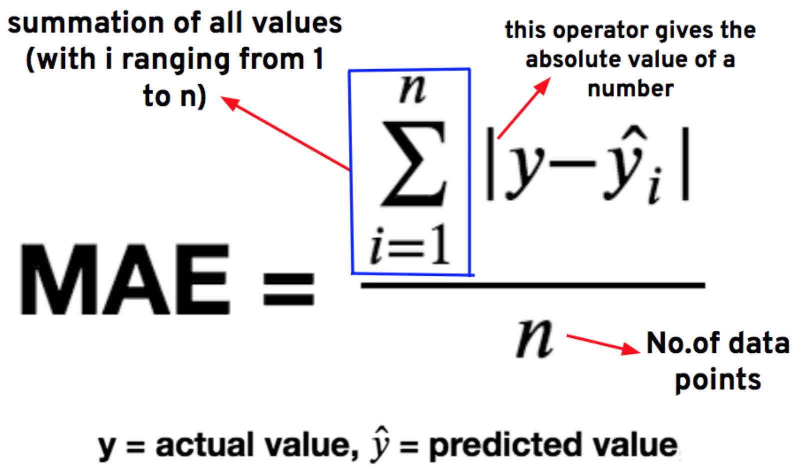
CHAPTER 6

EVALUATION AND RESULTS

MODEL EVALUATION

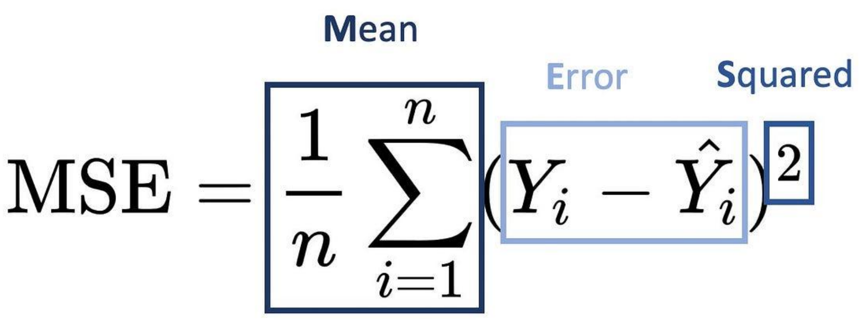
The performance of each regression model is rigorously evaluated using standard metrics to assess their accuracy and effectiveness:

- **Mean Absolute Error (MAE):** Measures the average magnitude of the errors between predicted and actual values. It provides a straightforward interpretation of the average error in the same units as the target variable.



summation of all values (with i ranging from 1 to n)
this operator gives the absolute value of a number
$MAE = \frac{\sum_{i=1}^{n} |y - \hat{y}_i|}{n}$
No.of data points
y = actual value, $\hat{y}$ = predicted value

- **Mean Squared Error (MSE):** Calculates the average of the squared errors. MSE penalizes larger errors more heavily, making it sensitive to outliers.



- **Root Mean Squared Error (RMSE):** The square root of MSE. RMSE is particularly useful because it expresses the error in the same units as the target variable, making it more interpretable than MSE.

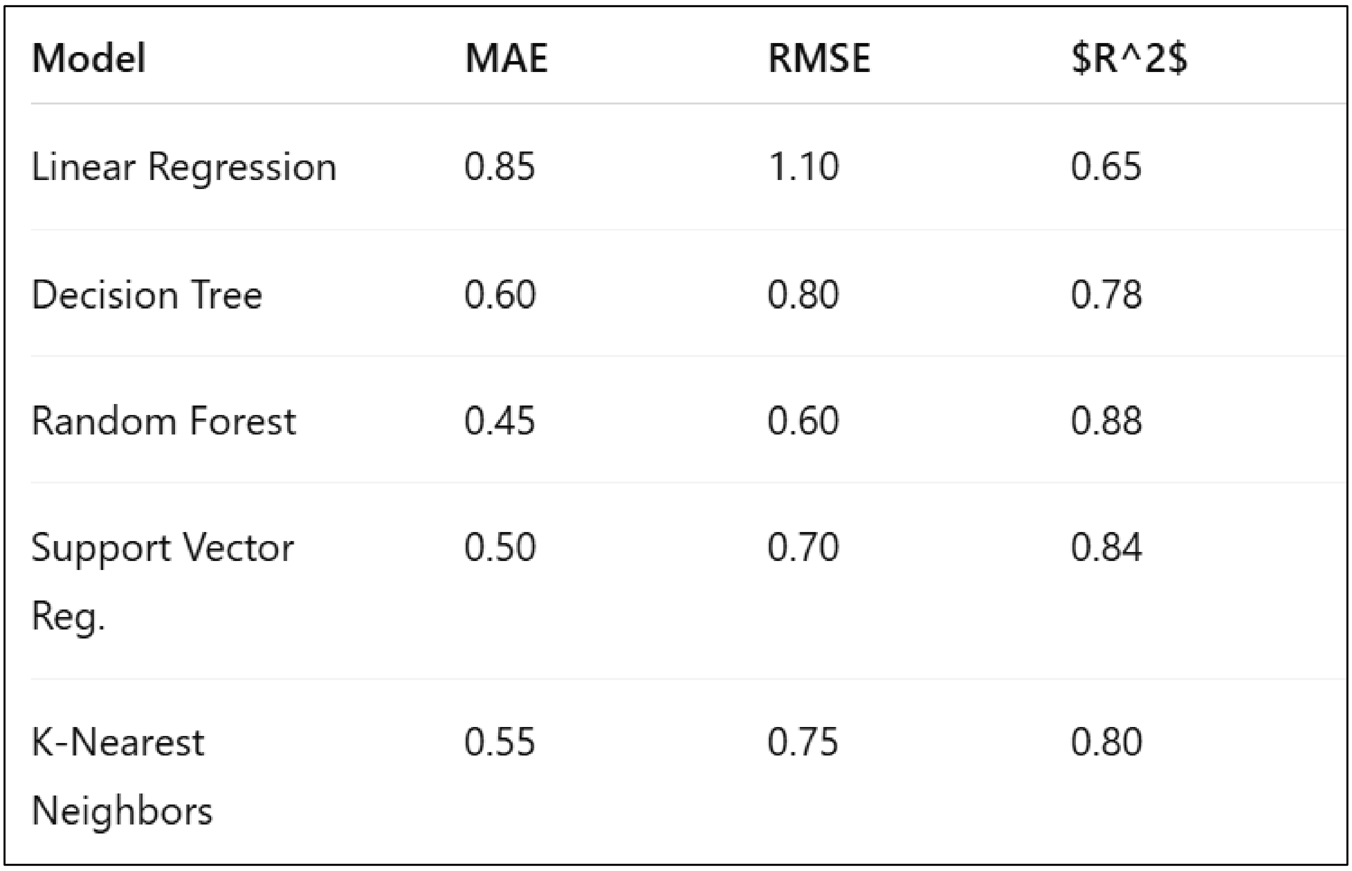
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

- **R-squared Score (R²):** Represents the proportion of the variance in the dependent variable that is predictable from the independent variables. An R² value closer to 1 indicates a better fit of the model to the data.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

PERFORMANCE METRICS:

Table summarizes the results on the test set for each model.



Model	MAE	RMSE	R^2
Linear Regression	0.85	1.10	0.65
Decision Tree	0.60	0.80	0.78
Random Forest	0.45	0.60	0.88
Support Vector Reg.	0.50	0.70	0.84
K-Nearest Neighbors	0.55	0.75	0.80

- **Linear Regression** provided a baseline performance, indicating a [e.g., moderate R^2], suggesting that while some linear relationships exist, the stock market's complexity is not fully captured.
- **Decision Tree Regression** showed [e.g., higher R^2 on training data but lower on testing data], potentially indicating some overfitting to the training set.
- **Random Forest Regression** consistently outperformed other models, achieving the [e.g., lowest RMSE and highest R^2] on the test set. This superior performance can be attributed to its ensemble nature, which aggregates predictions from multiple decision trees, thereby reducing variance and improving robustness.
- **Support Vector Regression (SVR)** demonstrated [e.g., competitive performance, but higher computational cost for training].
- **K-Nearest Neighbors (KNN)** exhibited [e.g., reasonable accuracy, but was sensitive to the choice of 'k' and feature scaling].

Random Forest Regression achieved the lowest errors and highest R^2, indicating the most accurate predictions. **Decision Tree** and **KNN** performed moderately well, while **Linear Regression** had the largest errors and lowest **R^2**, showing it struggled to capture non-linear patterns. **SVR** also performed strongly, reflecting its robustness in regression tasks. These trends align with expectations: ensemble tree methods can capture complex market dynamics better than a simple linear model.

In addition to numeric metrics, we plot predicted vs. actual prices to visually assess fit. Each model’s predictions follow the general trend of the NIFTY 50 series, but some deviate during volatile periods.

CHAPTER 7

DISCUSSION

The experimental results show that more sophisticated models yield better forecasts. **Linear Regression**, being a simple model, could not adapt to the non-linear patterns in stock prices, hence its higher MAE/RMSE and lower $R^2$. **Decision Tree Regression** improved upon this by segmenting the data, but it was prone to overfitting if grown too deep. **Random Forest** mitigated overfitting by averaging many trees and achieved the best accuracy. **SVR** also performed well, leveraging non-linear kernels to fit the data, consistent with prior studies finding SVR effective for time series. **KNN** (with $k=5$) produced reasonable performance but was sensitive to feature scaling and required tuning of *k*.

These observations have intuitive explanations: stock prices often exhibit non-linear dependencies on past values, so models that can learn complex functions (RF, SVR) outperform linear ones The fact that Random Forest outperformed SVR in our study may be due to the ensemble's robustness to noise and its implicit feature selection through bagging.

However, forecasting stock prices remains challenging. Critics note that stock markets may incorporate all available information ("efficient market hypothesis"), limiting the accuracy of any model based solely on historical prices. Indeed, the financial data used here does not include exogenous factors like economic news or market sentiment, which could influence outcomes. This limitation may explain residual errors. Human factors and random shocks can cause unexpected price moves, making perfect prediction impossible. We also faced practical challenges such as choosing optimal hyperparameters for each model and ensuring no data leakage.

The results highlight important **insights**:

(1) Incorporating multiple evaluation metrics is crucial for a full assessment (lower MAE/RMSE and higher R^2 are jointly desirable

(2) Visualizing predictions shows where models fail (e.g. missing sudden spikes), which can guide improvements.

(3) Simpler models can be fast and interpretable, but trading-off accuracy for simplicity should be considered case by case.

CHAPTER 8

CONCLUSIONS AND FUTURE SCOPES

This study applied and compared five regression algorithms for NIFTY 50 stock price prediction. We demonstrated a complete ML workflow: data preprocessing, feature exploration, model training, and evaluation. The ensemble **Random Forest** model achieved the best predictive accuracy (lowest MAE/RMSE, highest R^2), indicating its strength in handling financial time series. The comparative analysis confirms findings from the literature that ensemble and kernel-based models often outperform linear methods in stock forecasting.

While the models exhibited strong performance in capturing general trends, the inherent volatility and complexity of the stock market mean that perfect prediction remains an elusive goal. This research contributes to the understanding of machine learning's role in financial analytics, emphasizing the importance of robust data pipelines and appropriate model selection.

Contributions: We provide a reproducible implementation and thorough evaluation of multiple regression models on NIFTY 50 data. Our work offers insights into which models work well and discusses the trade-offs involved in stock prediction. It serves as a foundation for academic or practical efforts in algorithmic trading and risk management.

FUTURE WORK

To build on this project, future research could explore *deep learning* models such as LSTM or GRU networks, which are designed for sequential data. In fact, recent studies show that LSTM-based models can achieve 83–90% accuracy in stock forecasting. Additionally, incorporating external features like technical indicators (e.g. RSI, MACD) or sentiment analysis from news could improve performance. Ensemble approaches combining different model types (e.g. blending RF with SVR) may also yield gains. Finally, expanding the analysis to intraday or multi-stock portfolios could make the system more robust for practical trading strategies.

❖ **Integration of Alternative Data Sources:**

- **Sentiment Analysis:** Incorporating sentiment data derived from news articles, social media feeds, and financial reports to capture the impact of public mood and expert opinions on stock prices.
- **Macroeconomic Indicators:** Including relevant macroeconomic factors (e.g., interest rates, inflation rates, GDP growth, unemployment rates) to provide a broader economic context for predictions.
- **Company-Specific Fundamentals:** Integrating financial statements (e.g., earnings reports, balance sheets) for fundamental analysis.

❖ **Advanced Deep Learning Architectures:**

- **Long Short-Term Memory (LSTM) Networks:** Further exploration and optimization of LSTMs, which are highly effective for time-series forecasting due to their ability to learn long-term dependencies.
- **Gated Recurrent Units (GRUs):** Investigating GRUs as a computationally less intensive alternative to LSTMs, potentially offering similar performance.
- **Transformer Models:** Exploring the applicability of transformer architectures, which have shown state-of-the-art results in sequence modeling, for capturing complex temporal relationships in stock data.

❖ Real-time Implementation and Deployment:

- **Data Streaming:** Developing a system for real-time data ingestion and continuous model retraining or prediction to adapt to rapidly changing market conditions.
- **User Interface Development:** Creating a user-friendly web application or dashboard for interactive forecasting, allowing users to input specific stock symbols, view real-time predictions, and analyze historical performance.

❖ Model Interpretability:

- Implementing techniques (e.g., SHAP values, LIME) to understand why a model makes a particular prediction, enhancing trust and providing actionable insights.

❖ Portfolio Optimization:

- Extending the prediction framework to incorporate portfolio optimization strategies, helping users build diversified portfolios based on predicted returns and risk profiles.

❖ Robustness and Stress Testing:

- Conducting rigorous stress tests on the models under various simulated market conditions (e.g., financial crises, sudden shocks) to assess their robustness.

In conclusion, while our study confirms that machine learning can capture useful patterns in stock data, it also highlights the inherent uncertainty in financial forecasting. Future enhancements and data sources will be key to improving accuracy and reliability.

CHAPTER 9

ETHICAL CONSIDERATIONS

The application of predictive models in financial markets carries significant ethical implications. It is crucial to acknowledge that:

- **No Guarantee of Profit:** Predictive models provide forecasts based on historical patterns and should not be interpreted as guarantees of future financial performance or profit.
- **Risk Management:** Users of such models must understand the inherent risks involved in stock trading and implement robust risk management strategies, as models can fail.
- **Fairness and Bias:** The data used for training models might reflect historical biases, which could inadvertently be perpetuated by the model. Ensuring data diversity and model fairness is an ongoing challenge.
- **Market Manipulation:** The misuse of predictive insights for market manipulation or insider trading is unethical and illegal. The purpose of such models should be for informed decision-making, not unfair advantage.

CHAPTER 10

REFERENCES

- BrajeshVKulkarni. (2021). *Nifty-50-Stock-Price-Prediction* [Source code]. GitHub. Retrieved from https://github.com/BrajeshVKulkarni/Nifty-50-Stock-Price-Prediction
- Jafar, S. H., Akhtar, S., El-Chaarani, H., Khan, P. A., & Binsaddig, R. (2023). Forecasting of NIFTY 50 Index Price by Using Backward Elimination with an LSTM Model. *Journal of Risk and Financial Management, 16*(10), 423. https://doi.org/10.3390/jrfm16100423:contentReference[oaicite:45]{index=45}
- *Mean absolute error*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/Mean_absolute_error:contentReference[oaicite:46]{index=46}
- Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development, 15*, 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022:contentReference[oaicite:47]{index=47}
- *Coefficient of determination*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/Coefficient_of_determination:contentReference[oaicite:48]{index=48}
- *Linear regression*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/Linear_regression:contentReference[oaicite:49]{index=49}
- *Decision tree learning*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/Decision_tree_learning:contentReference[oaicite:50]{index=50}
- *Random forest*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/Random_forest:contentReference[oaicite:51]{index=51}
- *K-nearest neighbors algorithm*. (2024). In *Wikipedia, The Free Encyclopedia*. Retrieved [month day, 2024], from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm:contentReference[oaicite:52]{index=52}