Stock Market Prediction

Submitted for

Statistical Machine Learning CSET211

Submitted by:

(E23CSEU0288) **Paarangat Rai Sharma**

(E23CSEU0282) **Anusha Pundir**

Submitted to

DR. ASHIMA YADAV

July-Dec 2024

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX

ABSTRACT

In this project, we explore machine learning techniques to predict trends in the stock market, with a specific focus on the NIFTY 50 index. Using historical stock data, we developed, trained, and evaluated multiple regression models—including Linear Regression, Support Vector Regression (SVR), K-Nearest Neighbours (KNN), and Polynomial Regression—to forecast stock prices. Our objective is to determine the most effective model for accurate stock price prediction by comparing each model's performance based on metrics like Mean Squared Error (MSE) and R^2 score.

By identifying the most promising model, we aim to contribute a foundational framework for future advancements in stock market forecasting, demonstrating the potential of machine learning in finance. Through this project, we seek to bridge financial data analysis with advanced machine learning methodologies, creating valuable tools for investors, financial analysts, and other stakeholders.

INTRODUCTION

Predicting stock market movements has always intrigued us due to the dynamic and multifactorial nature of financial markets. We recognize that market fluctuations are influenced by various factors—ranging from economic conditions to investor sentiment and global events—making accurate predictions highly challenging yet valuable.

Traditional approaches often fall short in capturing these complex, nonlinear patterns, so we decided to leverage machine learning to tackle this task. By analysing historical stock data and utilizing predictive modelling techniques, we aim to contribute to the growing body of research on stock market predictions. Our goal is to develop a tool that not only enhances the accuracy of price predictions but also provides investors and analysts with data-driven insights that aid decision-making.

RELATED WORK

Stock market prediction has been a popular topic in the field of finance, data science, and machine learning. Researchers have used various techniques and methods to predict stock prices and trends. Below is an overview of related work and how the current project is different:

1. Linear Regression for Stock Prediction

- Previous projects often use simple linear regression to model the relationship between stock features such as Open, High, Low, and Close prices. Linear regression is easy to understand but cannot capture the complex patterns in stock data.

- Difference: This project uses additional models like Support Vector Regression (SVR) and Random Forest to handle non-linear relationships in the data. Combining regression and classification models helps create a more complete prediction approach.

2. Time-Series Forecasting with ARIMA

- Many studies use ***ARIMA (Auto-Regressive Integrated Moving Average)*** to forecast stock prices based on past data. ARIMA models focus on time trends but may miss other important factors like volume and seasonality.

- Difference: In this project, we include features like Volume, Month, Day, and Year to capture more factors that can influence stock price movements. This helps to model the data more effectively.

3. Machine Learning Classifiers for Price Movement

- Some projects use classification algorithms like Logistic Regression or Random Forest to predict if the stock price will go up or down. These models give a yes/no answer based on previous closing values.

- Difference: In this project, we use a mix of classifiers, including Naive Bayes, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). We also use different metrics and visualizations, such as confusion matrices, to evaluate the performance of each model.

4. Neural Networks for Stock Prediction

- Recently, deep learning models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have become popular for stock prediction. These models are powerful for handling sequences of data and perform well with large datasets.

- Difference: Instead of deep learning, this project uses traditional machine learning models to make it easier to understand and implement. The focus is on using simpler models that are computationally less expensive and more accessible.

5. Sentiment Analysis-Based Predictions

- Some projects also use sentiment analysis on news articles or social media to understand market sentiment and predict stock price movements. These methods use natural language processing to add extra information.

- Difference: This project focuses only on structured financial data, such as historical prices and volume. This makes the project easier to reproduce and avoids the complexity of dealing with text data.

METHADOLOGY

We approached this project through a structured methodology that includes the following steps:

- **Data Collection**: We collected historical data for the NIFTY 50 index from Yahoo Finance, covering a period of five years to ensure a comprehensive dataset.

- **Data Preprocessing**: To prepare the data, we addressed missing values, outliers, and inconsistencies. We also carried out feature engineering by extracting features like day, month, and year from the 'Date' column, which provided us with temporal insights.

- **Model Development**: We developed both regression and classification models to predict stock prices and their directional movements. Our regression models included Linear Regression, SVR, KNN, and Polynomial Regression, while our classification models consisted of Logistic Regression, SVM, KNN, Naive Bayes, and Random Forest.

- **Model Evaluation**: Each model was evaluated based on relevant metrics. For regression models, we used MSE and R^2 score to gauge performance, while for classification models, we assessed accuracy, precision, recall, and F1 score. This comparative evaluation helped us pinpoint the most suitable models for predicting stock market trends.

HARDWARE / SOFTWARE REQUIREMENTS

To execute this project, we required both specific hardware and software resources:

- Hardware: A computer with a minimum of 8GB RAM and a modern processor to handle data processing and model training efficiently.
- Software:
    - Python 3.x for the programming environment.
    - Google Colab as the primary platform for coding and running the models.
    - Libraries used include NumPy, Pandas, Matplotlib, Seaborn, and Scikit-Learn, which supported data manipulation, visualization, and model development throughout the project.

EXPERIMENTAL RESULTS

Regression Models:

- **Linear Regression and Polynomial Regression (degree 2)** both show excellent fits to the data, with **high R^2** values (~0.999) and **low MSEs**. This indicates these models are **highly accurate** and capture the underlying trends effectively.
- **K-Nearest Neighbors (KNN) Regression** performs **moderately well,** with an R^2 of 0.87, but its **MSE is higher**, indicating it doesn't fit the data as closely as Linear or Polynomial Regression.
- **Support Vector Regression (SVR)** performs **poorly**, with a **high MSE** and an **R^2 close to 0**, suggesting it struggles to capture any meaningful relationships in the data.

Classification Models:

- **Logistic Regression** has the **highest accuracy** at 55.6%, showing a strong bias toward correctly identifying class 1 (recall of 0.80) but struggling with class 0.
- **SVM and Random Forest** have **moderate accuracy** (~52%), with Random Forest showing better balance between class 0 and class 1.
- **KNN and Naive Bayes** have the **lowest accuracy** (around 50-51%) and struggle to achieve balanced classification across classes.

Across all models, class 1 is generally easier to classify correctly than class 0, leading to skewed performance and lower overall reliability.

CONCLUSIONS

Our study demonstrates the varying effectiveness of different machine learning models for stock price prediction, with regression and classification models showing distinct strengths and limitations.

In the regression analysis, both Linear and Polynomial Regression (degree 2) models achieved near-perfect R^2 values (~0.999) and low Mean Squared Errors (MSEs), indicating their high accuracy in capturing underlying stock price trends. These models' strong performance suggests they can effectively identify patterns within the NIFTY 50 data, providing reliable predictions of stock prices. However, K-Nearest Neighbors (KNN) Regression, while moderately effective with an R^2 of 0.87, had a higher MSE, which reflects a less precise fit compared to the linear models. Support Vector Regression (SVR) performed poorly in this dataset, with a high MSE and an R^2 near zero, suggesting it failed to capture any meaningful relationships, likely due to its limited suitability for this specific stock data.

For classification tasks, Logistic Regression achieved the highest accuracy at 55.6%, showing an effective bias towards correctly identifying class 1 instances, with a high recall of 0.80 for this class. However, it struggled with accurately predicting class 0, leading to an imbalanced performance. Support Vector Machine (SVM) and Random Forest classifiers performed moderately well (~52% accuracy) and showed better balance between classes, which may offer a more generalized solution for classification tasks. Conversely, KNN and Naive Bayes classifiers achieved the lowest accuracy scores (around 50-51%) and encountered significant challenges in maintaining balanced classification across both classes.

Across all classification models, class 1 proved generally easier to classify correctly than class 0, resulting in skewed performance. This bias underscores a limitation in model reliability for classification tasks, which could be addressed in future studies by experimenting with class-balancing techniques or incorporating additional features to improve overall accuracy.

In summary, Linear and Polynomial Regression models stand out as the most effective for this stock price prediction task, offering highly accurate fits to the data. Logistic Regression provides some value for directional classification but requires improvement for balanced performance. Future work could explore model tuning and additional feature engineering to address these limitations and enhance the models' reliability across both classes.

FUTURE SCOPE

Future work could involve:

- Implementing hyperparameter tuning techniques, such as GridSearchCV, to optimize model performance.
- Incorporating additional financial indicators like Moving Averages (MA) and Relative Strength Index (RSI) to enhance prediction accuracy.
- Exploring deep learning models, such as Long Short-Term Memory (LSTM) networks, for capturing temporal dependencies in stock data.

GITHUB LINK FOR COMPLETE PROJECT

https://github.com/paarangat/simple-stock-market-prediction