Stock Predictor

|  |  |  |
| --- | --- | --- |
| Daniel Hodges  Knoxville, USA  Dhodge12@vols.utk.edu | Robert King  Knoxville, USA  rking69@vols.utk.edu | Kush Patel  Knoxville, USA  pkx959@vols.utk.edu |
|  |  |  |

*Abstract*—Predicting next-day stock prices can help investors make better choices in the fast-paced stock market. This project introduces a Stock Next Day Closing Price Predictor that uses a Random Forest Regressor, trained on Berkshire Hathaway’s past stock data. Key data points like Open, High, Low, Close, and Volume help the model predict closing prices and classify stock trends. While the model works well when the market is stable, it struggles with big market changes. To improve, we propose adding features for testing on specific dates, boosting trend classification accuracy, and including external financial indicators. These upgrades aim to make the model more reliable and helpful for users facing various market conditions.

*Index Terms*—Analysis, classification, data, finance, prediction

# I. INTRODUCTION

# In the world of stock trading, accurately predicting stock price movements is a critical component of successful investment strategies. Traditional stock price forecasting is challenging due to the complex and fluctuating nature of financial markets, where prices are influenced by many different factors. For investors and analysts, understanding the potential trajectory of stock prices can provide a significant advantage, helping them make informed decisions and mitigate financial risks. This project addresses the need for an effective and accessible tool to forecast stock trends by developing a Stock Next Day Closing Price Predictor. By harnessing historical stock data and employing a machine learning model, specifically a Random Forest Regressor, this project aims to provide accurate next-day stock price predictions. The goal is to give traders and analysts a reliable prediction model that not only forecasts the closing prices but also classifies expected stock trends, ultimately aiding in better decision making and strategy development in the financial market.

# II. Data Exploration

## A. Dataset Summary

The dataset used in this project is made up of 2,408 daily entries of Berkshire Hathaway stock data, containing many important financial attributes: Date, Open, High, Close, Adjusted Close, and Volume. Each of these attributes is important in trend analysis and model training. The Date captures the trading date, which allows us to predict stock behavior over time. Open represents the initial stock price at market open, while High and Low indicate the maximum value and minimum price movements throughout any given day. The Close value marks the price of the stock at market close for the day. Adjusted Close accounts for dividends and stock splits, offering a more accurate reflection of value. Volume tracks the number of shares traded, which often has to do with stock volatility.

## B. Data Cleaning and Preprocessing

In order to have an effective training process, there were several steps taken to preprocess the data to be used. First, the Date field was converted from a string to a datetime format, helping in chronological sorting and easy trend extraction. Although our dataset was complete and was not missing any values, we have implemented missing data handling strategies such as row removal or interpolation to deal with any future data inconsistencies. To create a target variable for the next-day price prediction, the Close column was shifted by one day to produce a new column called Next Day Close. This shift allows our model to use today’s prices to predict tomorrow’s closing prices. We remove and filter out any null values tht result from this shift to make sure our data is complete to properly train the model.

## C. Exploratory Data Analysis

Exploratory Data Analysis was conducted in order to uncover trends, correlations, and significant patterns. Visualizing the Close and Adjusted Close prices over time reveals stock price trends and potential seasonal cycles in the market. These patterns can provide insight into stock behavior in response to annual events or shifts in the economy. Volume analysis highlights days with higher trading activity, which could be seen to correspond to important news or an increase in investor interest. Examining the correlations between Open, High, Low, and Close helps to assess the relationships within the price movements throughout the day, which could help in the model’s ability to capture daytime volatility.

# III. Baseline Solution

1. Introduction

For our baseline solution, we implemented a Random Forest Regressor to predict the next day’s closing price of the Berkshire Hathaway stock. Random Forest is an ensemble learning method that makes multiple decision trees during training and average their predictions to provide a more efficient and accurate forecast. This approach is effective for regression tasks like predicting stocks.

1. Existing Solutions

Predicting stock prices has traditionally been

approached using different methods, ranging from simple statistical techniques like Moving Averages (MA) and Exponential Moving Averages (EMA) to more complex machine learning models such as Long Short-Term Memory (LSTM) networks and Support Vector Machines (SVM). While statistical models focus on historical price trends and smoothing fluctuations over time, “moving averages smooth the data, but they may noy capture more complex patterns or sudden changes in a time series” (). Machine learning models, such as LSTM, are made to capture time dependencies but “are prone to overfitting and require large datasets to generalize well” (). Given the structured, tabular nature of stock data and the need for a model that balances accuracy and interpretability. Random Forest Regression was selected as the baseline model due to its robustness, ability to handle non-linear data, and resistant to overfitting.

1. Baseline Selection

The Random Forest Regressor (RFR) was chosen as the

baseline model for multiple reasons. First, Random Forest is an effective method that fits several decision trees and average their predictions, making it more robust and accurate than a single decision tree. Random forest is particular good at handing non-linear data between variables without requiring extensive tuning of the hyperparameters. This makes the RFR an ideal candidate for capturing the relationships between stock price variables such as “highs” and “lows”. Unlike some models like SVM or Neural Networks, which may require extensive tuning and is prone to overfitting with small datasets. Random Forest offers a strong balance between performance and simplicity.

1. Implementation

The implementation of the Random Forest model has

several steps. First, the historical stock data for Berkshire, consisting of open, close, adjacent close, and low prices, was preprocessed. We then created a target variable, the Next Day’s Closing Price, by shifting the “Close” column down by one day. The dataset was then split into a testing (8%) and training (92%) sets, ensuring that the model could be tested on unknown data.

The RFR was configured with 100 trees (n\_estimators = 100), with default hyperparameters for maximum depth and minimum sample splits. The random forest classifier is a combination of tree classifiers such that each tree depends on the value of the random vector sampled independently and with the same distribution for all trees in the forest. The model was on trained on the training set, using features like open, close, adjacent close, high, low to predict the next day’s closing price. The random state was set to 24 to ensure reproducibility.

1. Baseline Performance

The model was tested on its ability to predict the next

Day’s closing price and classify stock trends as either “Uptrend” or “Downtrend”. The predicted closing prices were evaluated against the actual next-day prices. We also implemented a prediction difference to see how the model’s output aligned with the real data. Generally, the prediction differences were modest, suggesting that the model is effectively capturing the stock price movements. However, for some days there was more of a deviation between the predicted and actual closing prices. This behavior is common among models that rely on long term data, because the model doesn’t account for sudden external situations, such as market news or economic changes.

In addition to predicting closing prices, the model’s trend classification performance analyzed by comparing the “Predicted Trend” with the “Actual Trend” for each day. While the model aligned with the actual trend direction during stable periods, sometimes there was a misclassification trends in volatile conditions, predicting a downtrend trend when the actual trend was uptrend, and vice versa. For example, on 2015-07-31, the model predicted a Uptrend movement, however, it was a downtrend movement. These misclassifications indicate that while the baseline model provides valuable insight into stock price movement trends, it may benefit from additional features, such as technical indicators or sentiment data, to improve its performance during unpredictable market conditions. Overall, the model’s ability to track trends during stable periods demonstrates its potential, yet enhancements could make it more robust in dynamic market environments.

IV. Proposed Extension

1. Implementing a Date-Specific Testing Capability

A valuable extension to the Stock Predictor is to implement a functionality that allows the user to test predictions for specific dates. This addition would enable users to evaluate the model’s accuracy on days of particular market interest, such as during economic policy changes, earning announcements, or periods of when the market movement is volatile. By examining these specific dates, users can gain insight into how well predictor adapts to the market’s conditions. Additionally, this feature would allow for more granular market performance evaluation by pinpointing the specific dates where the model succeeded or struggled. Ultimately, the feature would enable more targeted improvements in model architecture, potentially increasing the model’s reliability across every economic condition (simple or volatile stock movements).

1. Enhance Stock Movement Classification Accuracy

Accurate classification of stock movement-determining between “Uptrend”, “downtrend”, and “neutral” patterns – is important for having a reliable predictor. A potential improvement would be in refining the model’s architecture to effectively capture quick stock price shifts, especially during the transitional phases between trends. Strategies like oversampling underrepresented movement classes, implementing ensemble methods, or implementing feature engineering to make trend-specific indicator could improve the classification outcomes. Additionally, setting a confidence threshold for each day’s prediction could enhance classification accuracy by only flagging the predictions that were above a certain confidence level, thereby figuring out what is happening on the flagged dates. Did something happen in the market that day? Is it a coding/systems error? That’s what we need to figure out. By fixing that problem, the model can reduce misclassifications. This threshold approach could be useful in a volatile market, where when minor misclassification could lead to major financial impacts. Improving classification accuracy would make the model more useful, reliable for users that could aim for informed trading strategies.

C. Possible Implement of External Financial Indicator

Adding an external financial indicator, such as market indices (S&P 500), trading volume data, or macroeconomic indicators (interest rates, inflation rates), could provide important information that could enhance the model’s prediction accuracy. These indicators often influence market sentiments and can have an indirect or direct impact on individual stock movements. By adding these variables into the model, the predictor could make more accurate predictions that account for broader market. This provides user with a well-rounded analysis rather than relying on stock-specific historical data. This context-rich model would likely perform better period of economic uncertainty or market-wide shifts, where external factors play an impacting role in shaping stock trends. Incorporating this data could lead to adaptative predictions across sectors.

V. Conclusion

In conclusion, this project successfully developed a Next Day Closing Price Stock Predictor using a Random Forest Regressor, illustrating the potential of machine learning for financial forecasting. Predicting stock prices is a challenging task due to the complex, dynamic nature of financial markets, where prices are impacted by a multitude of predictable and unforeseen factors. By balancing historical data and employing preprocessing techniques. This project established a reliable baseline model capable of providing accurate next day closing price predictions and trend classifications, especially under stable market conditions. Through exploratory data analysis, we identified trends and relationships that enhanced the model’s performance, providing a foundation for a reliable tool. However, while the baseline model is a good model, it also highlights areas for improvement, especially when the market is volatile and unpredictable. The proposed extensions, including the implementations of date-specific testing, enhanced trend classification accuracy, and the integration of external financial indicators, will aim to address these limits. Data specific testing would enable users to assess the model’s accuracy during important market events. Additionally, the integration of external indicators, such as market indices and economic data, would help with the model predictions and economic trends that influences stock behavior. Theses proposed extensions aims to refine the models accuracy and also create a more adaptive tool for users navigating the financial markets. Overall, this project demonstrates machine learning in stock price predictions.

Distribution of Work

Daniel Hodges:

* Helping on implementation of Baseline Model
* Wrote First half of Midterm Report
* Understanding the data and reading it in

Robert King

* Implemented the main Baseline Model
* Edited the Paper
* Wrote code to read in data

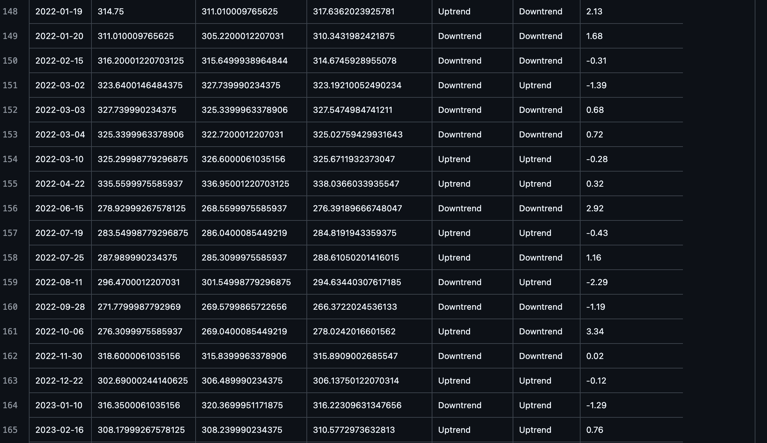
Kush Patel

* Understood and helped coding reading in the data
* Wrote 2nd half of Midterm Report
* Helped on Implementation of Baseline Model

A screenshot of a computer program

Description automatically generated

### Fig 1. Part of the Baseline Model



### Fig 2. Part of the output .csv