CS663 Group I Project

Portfolio Prediction Model

Rahul Saini (sainir@uab.edu) Manisha Manchana (manchana@uab.edu)
Karunakar Nuvvula (knuvvula@uab.edu) Slesha Bucchireddy Gari (sbucchir@uab.edu)
Dinesh Kumar Barla (dbarla@uab.edu)

We declare that we have completed this assignment in accordance with the UAB Academic Integrity Code and the UAB CS Honor Code. We have read the UAB Academic Integrity Code and understand that any breach of the Code may result in severe penalties.

We also declare that the following percentage distribution ***faithfully*** represents individual group members' contributions to the completion of the assignment

Name	Overall Contribution (%)	Major Work items completed by me	Initials	Date
Rahul Saini	30%	• LSTM model training and hyperparameter tuning. • Evaluation of LSTM performance metrics. • Development of custom classes for predictions and workflows.	RS	12/05/24
Manisha Manchana	25%	• Data collection and preparation (handling missing data, normalization). • Feature engineering for time-series forecasting.	MM	12/05/24
Karunakar Nuvvula	15%	• Random Forest implementation and comparative analysis. • Visualization of Random Forest vs. LSTM results.	KN	12/05/24
Slesha Bucchireddy Gari	15%	• Portfolio-level profit/loss calculations and visualizations.	SB	12/05/24
Dinesh Kumar Barla	15%	• Integration of models into the portfolio prediction system. • Drafting discussion, conclusions, and future work recommendations.	DK	12/05/24

Portfolio Prediction Model

ABSTRACT

This project develops a Portfolio Prediction Model to forecast stock prices and provide actionable insights for portfolio management. By leveraging Long Short-Term Memory (LSTM) networks, the model captures sequential dependencies in stock market data. The focus is on a portfolio comprising Apple (AAPL), Tesla (TSLA), and Google (GOOGL). Key steps include data preprocessing, feature engineering, model training, and evaluation of profit/loss metrics. The findings highlight the model's capability to deliver accurate predictions, assisting investors in decision-making.

CONCEPTS

Data Collection, Data Preprocessing, Feature Engineering, Time-Series Analysis, Long Short-Term Memory (LSTM) Networks, Sliding Window Approach, Model Training and Evaluation, Hyperparameter Optimization, Data Normalization, Correlation Analysis, Visualization of Data and Predictions, Sequential Dependency Modeling.

Keywords

Portfolio Management, Normalization, Profit/Loss Analysis, Sequential Dependencies, Sliding Window Approach, AAPL (Apple), TSLA (Tesla), GOOGL (Google), Financial Forecasting.

1. Introduction

Financial markets exhibit high volatility, and predicting stock price movements is essential for investors to optimize their portfolios. This project aims to integrate data mining, machine learning, and visualization techniques to build a robust prediction model. By analyzing historical data, the project forecasts stock trends for a portfolio comprising AAPL, TSLA, and GOOGL (*sample portfolio*). The model also evaluates profit/loss metrics to provide insights into portfolio performance. The objective is to enable data-driven decisions in portfolio management.

In today’s fast-paced financial markets, the ability to predict stock prices accurately has become a critical skill for investors and financial analysts. Stock market movements are influenced by a complex interplay of factors such as company performance, economic conditions, and investor sentiment, making accurate forecasting a challenging yet rewarding endeavor.

This project aims to address this challenge by developing a Portfolio Prediction Model that utilizes machine learning, specifically Long Short-Term Memory (LSTM) networks, to forecast stock prices and trends. The primary focus is on a portfolio comprising three major stocks: Apple (AAPL), Tesla (TSLA), and Google (GOOGL). These stocks represent technology giants with significant market influence, making them ideal candidates for an impactful study.

The model's objectives are twofold:

1. **Stock Price Prediction**: Forecast the next-day prices of any given stock using historical stock data.
2. **Portfolio Performance Analysis**: Evaluate the financial outcomes of predicted prices by calculating profit/loss metrics for the entire portfolio.

To achieve these objectives, the project integrates advanced data mining techniques, feature engineering, and sequential data modeling. The application of LSTM networks enables the model to capture the temporal patterns inherent in stock market data, providing a robust framework for prediction.

This study highlights the potential of machine learning in financial decision-making and lays the groundwork for more sophisticated predictive systems. By combining cutting-edge algorithms with rigorous analysis, this project contributes to the growing body of research on data-driven portfolio management and optimization.

2. Data Collection and Preparation

Data Collection

Data collection is the foundation of any predictive modeling project. For this project, historical stock price data for Apple (AAPL), Tesla (TSLA), and Google (GOOGL) was sourced from reliable financial data providers such as APIs (e.g., Yahoo Finance). The data included essential features like:

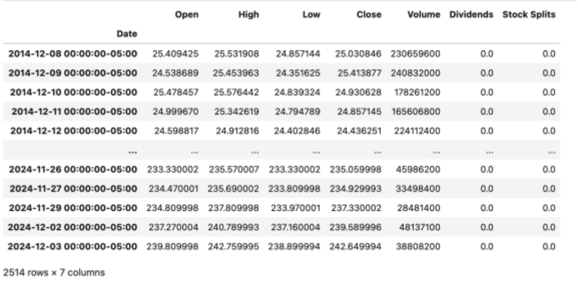
- **Open Price**: The price at which a stock starts trading at the beginning of the trading day.
- **Close Price**: The final trading price of a stock at the end of the day.
- **High and Low Prices**: The highest and lowest prices were reached during the trading session.
- **Trading Volume**: The number of shares traded during a particular time frame.

Data Preparation

Data preparation ensures that the dataset is clean, consistent, and ready for machine learning model training. Key steps included:

1. **Data Source:**

   Data is extracted using the '**yfinance**' library, which provides historical stock data.



	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2014-12-08 00:00:00-05:00	25.409425	25.531908	24.857144	25.030846	230659600	0.0	0.0
2014-12-09 00:00:00-05:00	24.538689	25.453963	24.351625	25.413877	240832000	0.0	0.0
2014-12-10 00:00:00-05:00	25.478457	25.576442	24.839324	24.930628	178261200	0.0	0.0
2014-12-11 00:00:00-05:00	24.999670	25.342619	24.794789	24.857145	165606800	0.0	0.0
2014-12-12 00:00:00-05:00	24.598817	24.912816	24.402846	24.436251	224112400	0.0	0.0
...	...	...	...	...	...	...	...
2024-11-26 00:00:00-05:00	233.330002	235.570007	233.330002	235.059998	45986200	0.0	0.0
2024-11-27 00:00:00-05:00	234.470001	235.690002	233.809998	234.929993	33498400	0.0	0.0
2024-11-29 00:00:00-05:00	234.809998	237.809998	233.970001	237.330002	28481400	0.0	0.0
2024-12-02 00:00:00-05:00	237.270004	240.789993	237.160004	239.589996	48137100	0.0	0.0
2024-12-03 00:00:00-05:00	239.809998	242.759995	238.899994	242.649994	38808200	0.0	0.0

2514 rows × 7 columns

Figure 1: Dataset information of 'AAPL' stock

2. **Handling Missing Data**:

   Missing values were identified and treated.

   - Missing Values: Forward-fill method was used to handle missing data points.

3. Removing Outliers or Anomalies:

- Filter the dataset to include only rows where the 'Close' price is greater than 0, ensuring no negative or zero values are present.

4. Dropping Unwanted Columns

- Remove irrelevant columns such as 'Dividends', 'Stock Splits', and 'Volume' from the dataset to focus on the relevant features for the analysis.

5. Feature Selection:

A heatmap illustrated a strong correlation among features like `Open`, `High`, `Low`, and `Close`, validating the focus on `Close` for predictions

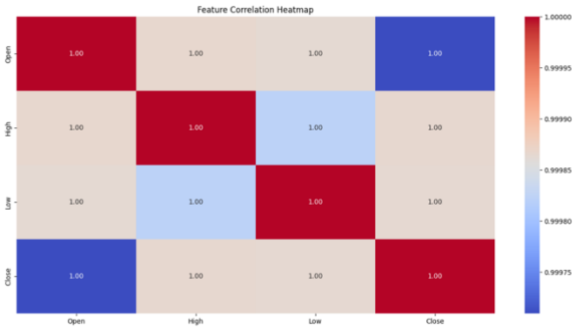


Figure 2: Heatmap representing the Correlation between the features

6. Dataset Splitting:

- The data was divided into training and testing sets. A typical split allocated 80% of the data for training and 20% for testing to evaluate the model's performance on unseen data.

7. Temporal Data Transformation:

- A sliding window approach was applied to convert the time-series data into supervised learning data. For example, a window of past 60 days' prices was used to predict the next day's price.

Features of the Dataset:

Features	Explanation
Date	Trading Date
Open	Starting price of the stock for the trading date
High	Highest Stock price for the trading date
Low	Lowest Stock price for the trading date
Close	Final price of the stock for the trading date
Volume	Number of shares sold for the trading date
Dividends	Share of company’s profits
Stock Splits	Increasing the current stock numbers by dividing it into multiple stocks

3. Prediction Models and Results

To train the portfolio prediction models, we investigated and compared several methods, including Long Short-Term Memory (LSTM) networks, Random Forest Regressors, and other potential machine learning approaches such as Gradient Boosting and Support Vector Machines. Each approach was chosen for its ability to address unique issues in stock price forecasting, such as capturing temporal dependencies, handling noisy data, and making strong forecasts for non-linear trends. The major purpose of this comparative investigation was to discover which model produced the most accurate and dependable results when predicting stock prices and portfolio trends.

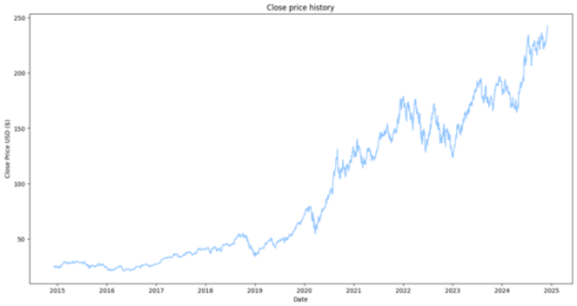


Figure 3: Closing Price History of 'AAPL' stock over the past 10 years.

Random Forest Regressor

The Random Forest Regressor, a powerful ensemble learning method, builds numerous decision trees during training and averages their outputs to provide predictions. This technique reduces overfitting while ensuring forecast stability. Random Forest is very good at handling non-linear correlations and outliers, hence it's a popular choice for many regression problems.

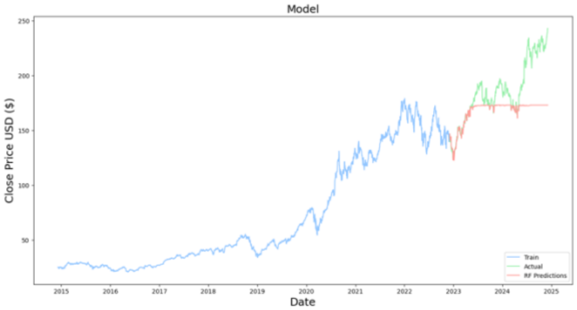


Figure 4: 'AAPL' Stock price prediction using Random Forest Model

Long Short-Term Memory(LSTM)

LSTM networks, a type of Recurrent Neural Network (RNN), are specifically developed to represent sequential input. This makes them ideal for time-series data such as stock prices, where present values are significantly reliant on prior trends. LSTM has a unique structure that incorporates forget gates, input gates, and output gates, allowing the model to choose store or reject information. This property allows LSTMs to successfully capture long-term dependencies, which are important for simulating stock price variations over time.

Model Architecture:

- Two LSTM layers for feature extraction.
- Dense layers to produce the final output.

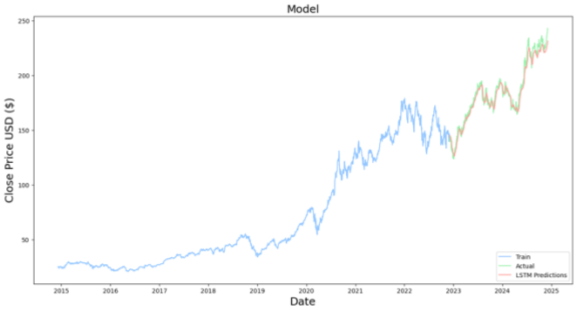


Figure 5: 'AAPL' Stock price prediction using LSTM Model

We choose LSTM for its potential to:

- Recognize and learn from temporal patterns.
- Handle data with variable time intervals or lags.
- Adapt to shifting trends, such as unexpected market fluctuations.

During training, LSTM networks displayed extraordinary effectiveness when compared to Random Forest in capturing the stock data's inherent sequence. Learning from historical price swings and trends.

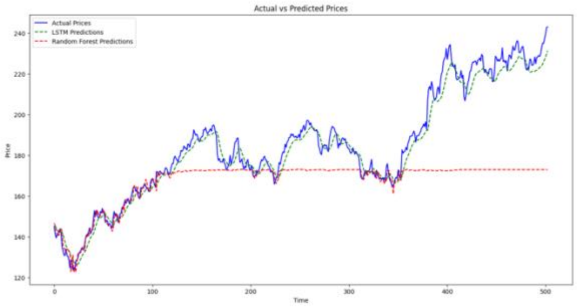


Figure 6: Random Forest vs LSTM Model comparison

Evaluation Metrics and Dataset Splits

The dataset was split into an 80% training set and a 20% testing set. Metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were used to evaluate the models. These metrics quantify the prediction error in both relative and absolute terms.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

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

Evaluation Results:

Metric	Random Forest	LSTM
MAE	18.22	5.07
MSE	735.97	39.14
RMSE	2.7e+01	6.3

Portfolio Prediction System

The portfolio prediction system is a comprehensive framework that automates the processes of developing, training, and testing prediction models for individual stocks and portfolios. This system employs custom-built classes that contain the critical components of data management, predictive modeling, and portfolio-level analysis. It streamlines the financial forecasting workflow by integrating tools for gathering data, preparing it, developing machine learning models, and analyzing results.

Custom Classes

There are three main custom classes form the framework of the system, each of which oversees an independent component of portfolio prediction:

1. StockPredictionModel

The LSTM model is the fundamental machine learning method for sequential data, and this class manages its design, training, and prediction features. Important features include of:

- **Model Design:** Builds a sequential neural network with dense layers for final predictions after LSTM layers for modeling time-series data.
- **Training:** Manages the model's training with user-specified parameters, including the number of hidden units, epochs, and batch size.
- **Prediction:** Using inverse transformations, it creates predictions

for test data and scales them back to the original range.

- **Evaluation:** Offers ways to compute performance metrics that are essential for evaluating model accuracy, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). For our Model we have used MAE as it is more robust to outliers.

2. Stock

The prediction system and the stock data are interfaced by the Stock class. Data retrieval, cleansing, and training preparation are all automated. Important features include of:

- **Data Fetching:** Using the Yahoo Finance API, historical stock data is retrieved, making sure the data covers the necessary time period.
- **Data preprocessing** involves addressing missing values, scaling features, and separating the data into training and testing groups in order to clean and analyze the raw data.
- **Feature engineering:** Produces the lag features needed for forecasting time series.
- **Integration with Models:** For smooth integration, data is prepared in the format required by the StockPredictionModel.

3. PortfolioPredictor

This class manages the overall prediction workflow for a portfolio consisting of multiple stocks.

- **Model Management:** Creates and maintains models for each stock in the portfolio.
- **Profit/Loss Calculation:** Estimates profit or loss based on predicted vs. actual stock prices, aiding in portfolio performance evaluation.
- **Visualization Tools:** Generates comparative plots for individual stocks and the entire portfolio, providing insights into prediction accuracy and stock trends.
- **Next-Day Predictions:** Calculates the expected prices for the next trading day to support investment decision-making.

Profit/Loss Estimation

One of the critical features of the system is its ability to compute the profit or loss for the portfolio. The PortfolioPredictor class calculates this by:

- Fetching the current and predicted prices for each stock in the portfolio.
- Multiplying the price difference by the weight of each stock in the portfolio to determine individual contributions to profit or loss.
- Summing these contributions to compute the overall portfolio performance.

This feature provides actionable insights for optimizing portfolio allocation and rebalancing based on predicted trends.

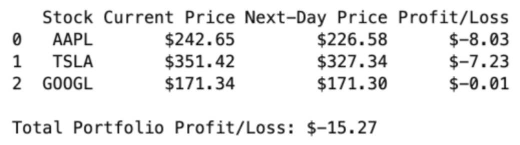
Portfolio {

"AAPL": 50%,

"TSLA": 30%,

"GOOGL": 20%

}



```
   Stock Current Price Next-Day Price Profit/Loss
0   AAPL       $242.65        $226.58      $-8.03
1   TSLA       $351.42        $327.34      $-7.23
2  GOOGL       $171.34        $171.30      $-0.01

Total Portfolio Profit/Loss: $-15.27
```

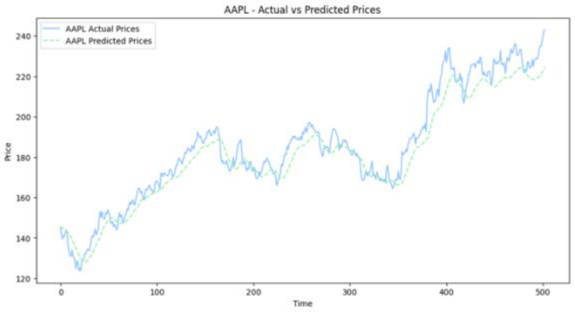


Figure 7: Actual vs Predicted Prices – ‘AAPL’

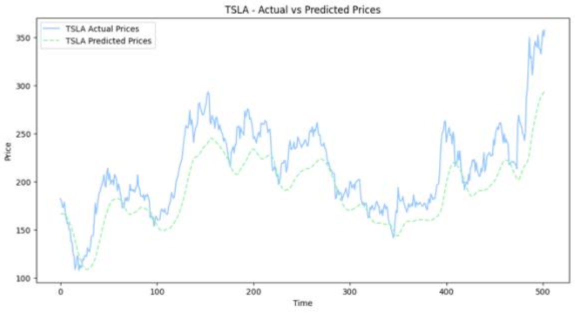


Figure 8: Actual vs Predicted Prices – ‘TSLA’

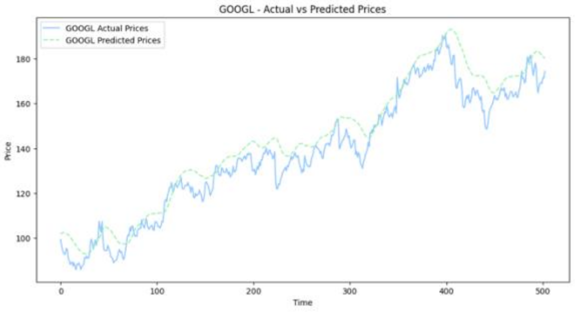


Figure 9: Actual vs Predicted Prices – ‘GOOGL’

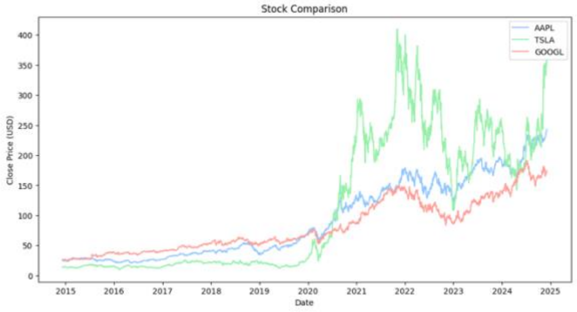


Figure 10: ‘AAPL’, ‘TSLA’ & ‘GOOGL’ stock history comparison

4. Discussion and Conclusions

The portfolio prediction models' outcomes provided a number of significant new insights into how well various machine learning techniques performed. We assessed several models, such as Random Forest Regressor and Long Short-Term Memory (LSTM) networks, and took volatility into account when making predictions. We also investigated the possible advantages of ensemble approaches. The main conclusions and their ramifications are summed up as follows:

- **LSTM** is the most effective model for stock price forecasting, particularly for volatile stocks, as it captures sequential dependencies and adapts to changing trends.
- **Random Forest**, while strong in certain scenarios, does not perform as well as LSTM for time-series data and volatile stock predictions.
- **High volatility** remains a challenge for all models, underscoring the need for advanced techniques to better handle market fluctuations and extreme price movements.

5. Future Work

These future improvements aim to make the system more accurate, efficient, and practical for real-world investment decision-making.

- **Advanced Models:** Transformers and Hybrid Approaches: Exploring **Transformers**, which excel at capturing long-range dependencies, could improve predictions over traditional LSTM models. Additionally, **Hybrid Models** combining LSTM with attention mechanisms or reinforcement learning could better capture complex patterns in stock price movements, enhancing prediction accuracy.
- **Portfolio Dashboard:** Real-Time Tracking: A Portfolio Dashboard will allow users to manage their stocks in real-time, view stock forecasts, and track portfolio performance. Interactive features will help users assess whether to add or remove stocks based on predicted future prices.

6. References

1. **GeeksforGeeks**. (n.d.). *Introduction to Long Short-Term Memory (LSTM)*. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
2. **Keras**. (n.d.). *Sequential Model*. https://keras.io/guides/sequential_model/
3. **Eurico Paes**. (2020, May 20). *Extracting Data from Yahoo Finance with yfinance*. Medium. https://medium.com/@euricopaes/extracting-data-from-yahoo-finance-with-yfinance-96798253d8ca
4. **Data Science Tutorials**. (2020, May 7). *Stock Price Prediction using Python*. YouTube. https://youtu.be/QIUxPv5PJOY?si=PKt44abYBiXmbjg1