Project Report

Team 57

Shawn Dai, Diane Guan, Taehui Kim, William Li, Sarvesh Vishnu Muhundu Parthiban

Motivation and Data Understanding

The basis of economic analysis and investment strategy in financial markets is the capacity to assess and forecast changes in stock prices. This project focuses on Apple Inc. (AAPL), a significant global technology participant whose stock performance gauges investor mood and broader market trends and reflects the company's health. Apple's position as a technology innovator and a major component of stock indices like the S&P 500 makes its stock a critical subject of study. Due to its vast supply chain and market reach, the company's influence on global technology trends, consumer behavior, and geopolitical dynamics makes its stock movements particularly significant.

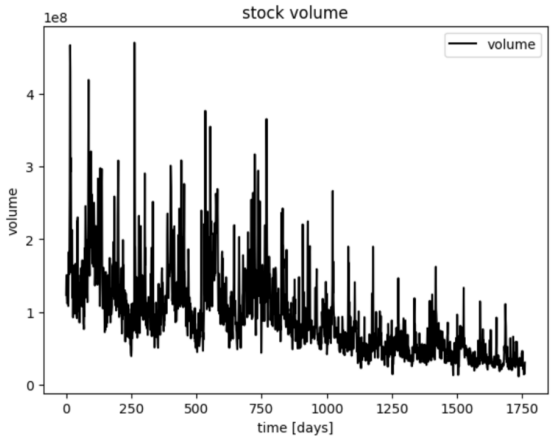
Predicting the closing prices can offer firms several advantages. Increased trading volume, higher profits, improved risk management, and protection of profits are some significant benefits. Accurate predictions can also enhance the firm’s reputation, attract more clients, and foster the development of new products and services. Ultimately, this gives the ability to navigate economic indicators effectively. Besides market-making firms, predicting the values is advantageous to various financial entities. Hedge Funds: Utilize strategies like short selling and leverage for higher returns. Mutual Funds: Diversifying portfolios enhances asset allocation decisions. Pension Funds: Managing assets for retirement income, ensuring informed decisions for meeting obligations. Insurance Companies: Support asset management and risk mitigation.

They can also enhance overall market efficiency and develop new predictive models to provide a competitive edge, leading to increased profits and market share.

The dataset has raw, as-is daily prices of the Apple stock. Most of the data spans from 2010 to the end of 2016, and the data have been added adjustments for splits.

Data Preparation

Time series data analysis requires indexing by date or time. The 'Date' column is used as an index in our dataset, facilitating time-based sorting and operations. This approach allows for chronological analysis, maintains the sequence integrity crucial for time series forecasting, and enables better visualization. For data normalization, we used Min-Max Scaling. It is a normalization technique that rescales the range of features to scale the range in [0, 1] or [-1, 1]. This method is particularly beneficial for the AAPL stock price dataset because stock market data often encompasses various magnitudes. For example, the closing price might range in hundreds of dollars, whereas trading volume could be in millions. Min-Max Scaling brings all these different variables to a comparable and uniform scale, preventing any one feature from dominating the model due to its scale. Min-Max Scaling also does not alter the distribution of the data. It preserves the original distribution of the dataset while scaling the values. This is crucial for financial data, where the distribution of prices and volumes can contain valuable market information. While Min-Max Scaling is sensitive to outliers, we completed data cleaning and integrity checks; we ensured there were no missing or incomplete entries, any NA value would be replaced with the column median, and we also checked for extreme outliers that could impact the data analysis.



Modeling

Linear Regression

A fundamental statistical technique in predictive modeling is linear regression. Its goal is to simulate the linear relationship between one or more independent variables and a dependent variable. We used it as a base model because of its simplicity and ease of interpretation. It helps understand the impact of individual variables on the stock price. However, it cant not capture the complex, non-linear relationships often present in financial data.

Multi-layer Perceptron Base Model

A Multi-layer Perceptron is a class of feedforward artificial neural networks. MLPs can model complex relationships between input features, like the highs and lows of stock price, and outputs closing prices, going beyond linear relationships. It is more capable of capturing non-linear relationships than linear models and is more flexible in terms of architecture to fit various data patterns. However, this model can be prone to overfitting.

Recurrent Neural Network (RNN)

This model is ideal for modeling time-series data like stock prices due to their strong ability to capture sequence and time-dependent patterns. Unlike traditional feedforward neural networks, RNNs have a unique feature: they maintain a form of internal memory or state that captures information about previous inputs in the sequence. This makes them ideal for applications where the sequence of data points is crucial, such as time series analysis, natural language processing, and more. Since the financial data is dynamic and influenced by numerous factors. RNNs can adapt to new patterns as they emerge, updating their internal state with each new data point, making them adept at handling the fluid nature of stock prices. However, traditional RNNs might struggle to learn long-term dependencies due to the vanishing gradient problem.

Long Short-Term Memory (LSTM)

Long Short-Term Memory networks is a subtype of RNNs, it is designed to overcome the challenges of traditional RNNs, particularly the vanishing gradient problem. It is capable of learning long-term dependencies in sequential data, which is critical in this case.

Implementation

Linear Regression

The criterion is set to MSELoss because it emphasizes larger errors, smooth differentiability, interpretability, and statistical consistency. It aligns well with the objectives of accurately predicting stock prices and minimizing large deviations, which are critical in financial forecasting. For the optimizer, we choose Adam because it is an iterative optimization algorithm used to minimize the loss function during the training of neural networks. Is is also known for

being robust and efficient regarding computational complexity. We set the learning rate to 0.001 because it is generally considered a good starting point. It is neither too small to significantly slow down the learning rate, nor too large to cause instability.

Multi-layer Perceptron Base Model

We used the same optimizer and criterion as the linear regression model. Since it requires careful tuning, we tried 2 and 3 hidden layers with different numbers of nodes and picked the one that had the lowest validation error.

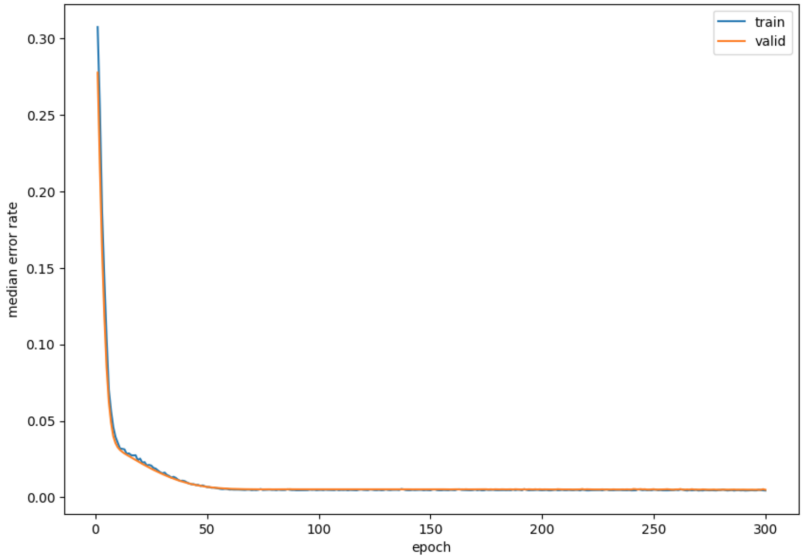
RNN Model & LSTM Model

In order to properly manage the job of predicting univariate time series data, the architecture of our RNN and LSTM models has been particularly built to suit the requirements of the task. The capacity of the model to acquire and internalize data associations is directly impacted by the hidden_dim parameter, which has a value of thirty. We increased the number of layers in the model by three (num_layers) in order to improve its capacity to study and comprehend more complex functions and relationships as time goes on. A total of two hundred epochs of training are performed on the models in order to guarantee that the networks get an adequate amount of data to update and enhance their weights. Finding a harmonic balance between quick convergence and steady learning may be accomplished by using a learning rate of 0.01 that has been suitably adjusted. A cautious pace is maintained in order to avoid beyond the best solution, while at the same time ensuring that a suitable speed is maintained in order to gain improvements.

Results and Evaluation

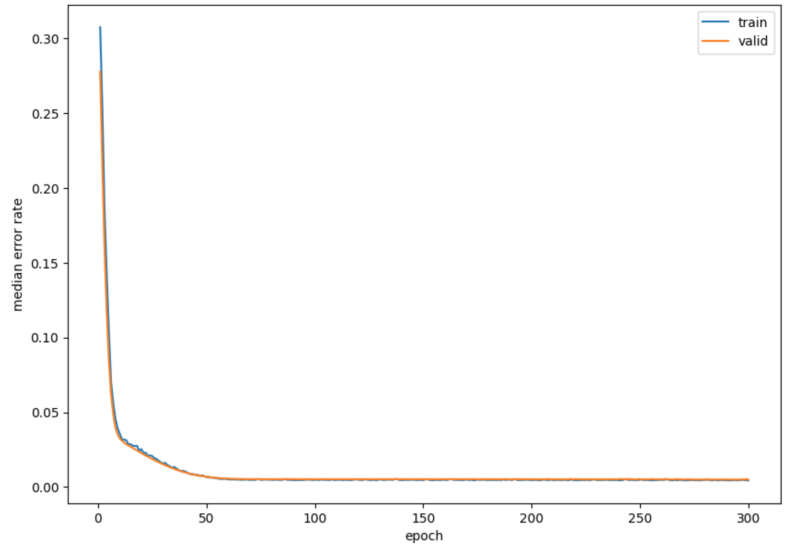
Linear Regression

For this model, we reached a validation error of 0.00497 on the 300th epoch.



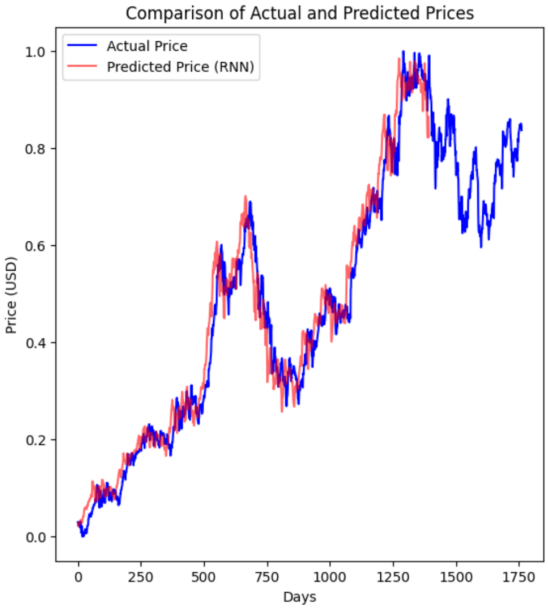
Multi-layer Perceptron Base Model

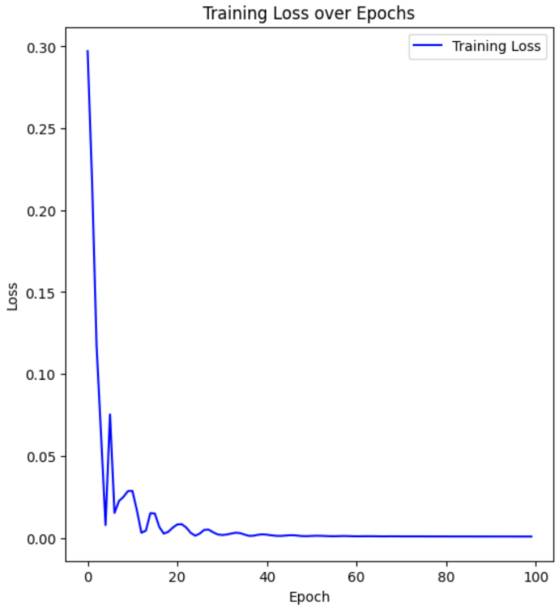
For this model, we reached a validation error of 0.00128 on the 171st epoch with hidden layers of [512, 256, 128].



RNN

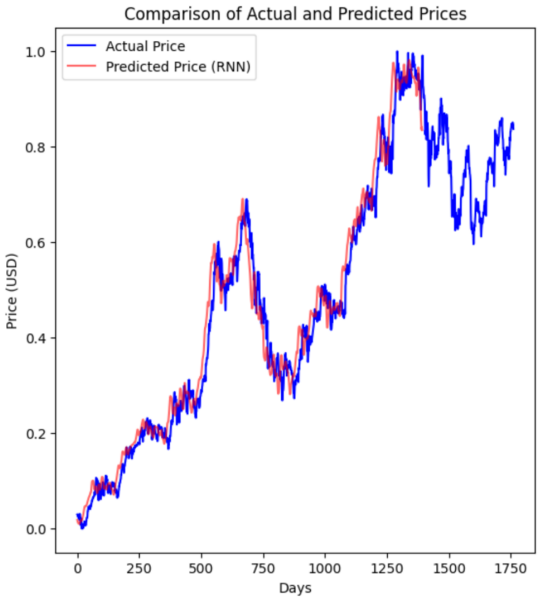
For this model, we reached an MSE of 0.00103.

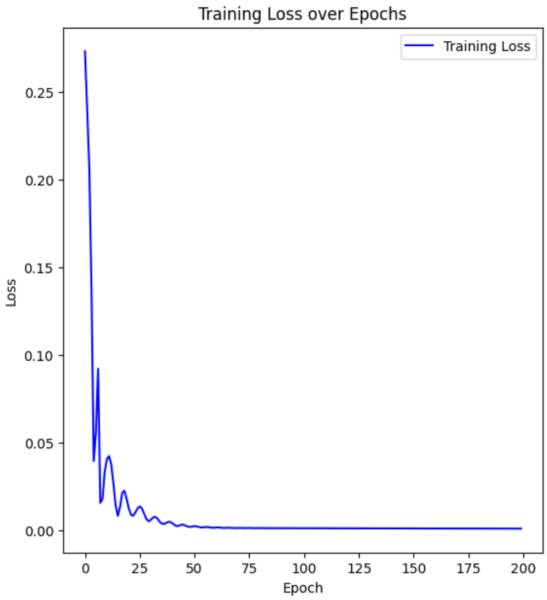




LSTM

For this model, we reached an MSE of 0.00216.





We used validation error as the evaluation value for the Linear Regression Model and the Multi-layer Perceptron Base Model. The validation error is calculated by taking the absolute value of the predicted price minus the actual price, divided by the actual price. We were unsurprised that the MLP has a much lower validation error than the linear model.
We used mean squared error as the evaluation value for the Recurrent Neural Network and Long Short-Term Memory. We were surprised that the RNN has a lower MSE than the LSTM, we think it could be because short-term trends more influence the stock price movements in the dataset. An RNN might capture these dynamics more effectively than an LSTM.

A potential business case could be developing a predictive tool for investment firms to optimize stock trading strategies based on AAPL's stock price movements.

Deployment

The deployment of results would first involve performing extensive validation using out-of-sample data to ensure the model's reliability and accuracy in real-world scenarios. Second, ensure an automated pipeline for real-time or periodic data ingestion, processing, and feeding into the model for up-to-date predictions. Third, develop an API for the model, allowing various applications or services within the organization to access its predictions. Fourth, regularly update the model with new data. Re-train or fine-tune to adapt to market changes or new data patterns. Last, collect user feedback to identify any issues or areas for improvement.

Some issues the firm should be aware of regarding deployment would be model choice and ethical considerations. For model choice, since the data that these models train on are data from

2010 to 2016, AAPL stock and the overall economic trend have changed a lot during the past years, and the models would need to be trained on more up-to-date data before the actual deployment. For ethical considerations, deploying powerful deep learning models in trading can significantly impact market dynamics, potentially leading to concerns about market manipulation or unfair advantages.

The first risk of the deployment plan is that the model may not perform as expected in real-world scenarios due to overfitting, underfitting, or failure to generalize to new data. We can implement validation techniques like k-fold cross-validation, regular testing against unseen data, and continuous performance monitoring. The second risk is that users' misinterpretation of the model's predictions leads to erroneous decisions. We can conduct comprehensive training for users. Develop clear guidelines for interpreting and acting on the model's predictions. The third risk is that there might be unintended ethical issues, such as biased predictions. We can conduct ethical reviews of models. Ensure transparency and fairness in model predictions.

Appendix

Contribution of each team member

Shawn Dai: Multi-layer Perceptron Base Model

Diane Guan: RNN Model

Taehui Kim: Data preparation and exploration

William Li: Linear Regression

Sarvesh Vishnu Muhundu Parthiban: LSTM

Members also work on the report and presentation slides together.