INTERNSHIP PROJECT REPORT

CS 583- B

NETFLIX STOCK PRICE PREDICTION

SUBMITTED BY

PRERNA DESAI
JASH SHAH
AKANKSHA WAGH
ISHAAN BANDEKAR
VIDISHA PARMAR

INTRODUCTION

The goal of this project was to predict the closing stock price of Netflix (NFLX) for January 2025, using historical stock data. Predicting stock prices is a critical task for investors and analysts, as accurate forecasts can support better financial decisions. Our approach leverages deep learning models, focusing on recurrent neural networks (RNNs) to capture the temporal dependencies in stock price data.
Time Horizon: 1 month (January 2025)
Key Data Used: Historical stock data (Open, High, Low, Volume, Close) from 2019–2024.

Problem definition

The goal of this project is to develop a predictive model to forecast the **closing stock price** of Netflix (NFLX) based on its historical stock data. The model uses features such as the opening price, highest price, lowest price, and trading volume for forecasting. By analyzing patterns and trends in the stock prices, this project aims to implement time series forecasting using a **neural network**.

What are we predicting?

- **Target Variable**: The **close price** of Netflix's stock.
- **Time Horizon**: Predict the closing price for the next trading day (short-term forecasting).

What does that number (Closing Stock Price) mean?

- The closing stock price represents the stock's value at the end of the day, influenced by market demand, company performance, and broader market conditions.

What data are we using to predict it?

Netflix historical stock data, including:

- **Open**: Price at the beginning of the trading day.
- **High**: Highest price reached during the day.
- **Low**: Lowest price reached during the day.
- **Volume**: Number of shares traded during the day.
- **Close price**, which will be the target variable.

What Does That Data Mean?

- **Stock Features**:
    - **Open, High, Low**: These features capture the price fluctuations during the trading day.
    - **Volume**: Indicates the level of trading activity, reflecting investor interest and potential volatility.
- **Close**: Serves as the target variable, representing the stock's value at the end of the day.

Assumptions

1. **Stationarity:**
   It is assumed that this time series either is at least weakly stationary or that, with appropriate transformation - for example, scaling or differencing - such non-stationarity issues can be overcome.

2. **No External Influences:**
   The model assumes there are no external influences such as market news, macroeconomic indicators, or global financial events driving the stock prices.

3. **Sufficiency of Features:**
   Chosen features of Open, High, Low, and Volume shall carry enough information to predict Close.

4. **Past Patterns Predict Future Behavior:**
   It is taken as a fact that the trends, patterns, and correlations of the past will continue to predict future behavior-a key assumption in time series forecasting.

5. **Data Quality:**
   It is assumed that this historical stock data is free of serious errors and omissions.

MODEL SELECTION

Should we use a Neural Network?

Yes, a neural network, particularly **LSTM** or **GRU**, is highly suitable for stock price prediction.

1. **Sequential Nature of Stock Data**
   Stock prices are time-series data where the order of observations matters. Neural networks like LSTM and GRU are specifically designed to model temporal dependencies, making them ideal for this task.
2. **Capturing Non-Linear Relationships**
   Stock prices are influenced by non-linear interactions between features such as Open, High, Low, and Volume. Neural networks excel at learning these complex relationships, which traditional models like linear regression cannot handle effectively.
3. **Temporal Dependency**
   LSTMs and GRUs are adept at learning both short-term volatility (daily changes) and long-term trends (seasonal patterns or market cycles). This ability makes them particularly well-suited for forecasting stock prices.
4. **Scalability**
   Neural networks are scalable and can handle large datasets with multiple features, ensuring adaptability for future expansion or inclusion of external factors like market sentiment.

By leveraging LSTM and GRU, we can effectively model the sequential and non-linear aspects of stock prices, outperforming simpler or non-sequential models.

What feature(s) of the data are we trying to establish relationships between?

The model aims to predict the Close price of Netflix stock by analyzing relationships between features like the Open price, which sets the trading tone; High and Low prices, indicating market optimism and pessimism; and Volume,

reflecting market activity and investor sentiment. These features collectively capture the day's trading dynamics to forecast the stock's closing value.

Neural Networks are employed for their ability to learn complex relationships in sequential data. Four architectures were considered:

- **LSTM (Long Short-Term Memory)**: Captures long-term dependencies effectively, making it well-suited for time-series data.
- **GRU (Gated Recurrent Unit)**: Balances computational efficiency with modeling performance.
- **SimpleRNN**: Serves as a baseline model to compare against advanced recurrent architectures.
- **Dense Neural Network (DNN)**: A non-recurrent model to benchmark against recurrent architectures.

Recurrent models (LSTM, GRU, SimpleRNN) were selected for their strength in handling sequential dependencies, while DNN was included as a baseline to highlight the advantage of sequence-aware models.

MODEL ARCHITECTURE

Recurrent Models (LSTM, GRU, SimpleRNN)

Recurrent Neural Networks (RNNs) and their advanced variants (LSTM and GRU) are specifically designed to process sequential data, making them ideal for stock price prediction. Here's a breakdown of what each layer does:

1. **Input Layer:**
    - Takes a sequence of stock price data, structured as a sliding window of 50 time steps, with predictors ['Open', 'High', 'Low', 'Volume'] as features.
    - Converts raw input into tensors that are fed into subsequent layers.
2. **Recurrent Layers (LSTM, GRU, SimpleRNN):**
    - **SimpleRNN:** Processes sequential data by passing information through a recurrent loop. Each step learns short-term dependencies.
    - **LSTM:** Enhances this by using gates (input, forget, and output gates) to capture both long-term and short-term dependencies, preventing the vanishing gradient problem.
    - **GRU:** Simplifies LSTM by combining the forget and input gates into a single update gate, improving computational efficiency without sacrificing much accuracy.
3. **What Happens in These Layers?**
    - The recurrent cells extract temporal patterns in the data, learning trends such as rising or falling stock prices.
    - For example, if the stock price shows consistent growth over several time steps, the recurrent layers will learn this trend and use it for future predictions.
4. **Dropout Layers:**
    - Randomly deactivate a fraction of neurons during training to prevent overfitting. This ensures the model generalizes well to unseen data.
5. **Dense Output Layer:**
    - Maps the learned features into a single output neuron, representing the predicted closing stock price for the next day.

Dense Neural Network (DNN)

Although DNNs are not optimal for sequential data, they were included as a baseline. Here's how the layers work:

1. **Input Layer:**
    - Flattens the sequential stock price data into a single vector. Unlike RNNs, this layer does not account for temporal order.
2. **Hidden Layers:**
    - Fully connected (dense) layers process the input vector, applying nonlinear activation functions (ReLU) to learn complex relationships between features.
    - These layers are effective at learning static relationships but fail to capture temporal dependencies.
3. **Output Layer:**
    - Outputs the predicted stock price using a single neuron.

How Does the Architecture Capture Relationships Between Data and Prediction?

- **Recurrent Models:**
  RNNs (especially LSTM and GRU) capture sequential dependencies by maintaining a hidden state that "remembers" information from previous time steps. This allows the model to understand how past stock prices and volumes influence future prices.
    - **Example:** If stock prices exhibit a pattern of increasing after high trading volumes, the model can identify and exploit this relationship.
- **GRU:** Balances computational efficiency and accuracy, making it especially effective at capturing relationships in volatile datasets like stock prices.
    - GRU's update and reset gates adjust how much past information is retained or forgotten, dynamically focusing on the most relevant trends.
- **Dense Neural Network (DNN):**
  While not designed for sequential data, the DNN can model relationships between predictors (e.g., 'Open,' 'High,' 'Low,' 'Volume') and the target

('Close'). However, it struggles with time dependencies, leading to suboptimal performance.

External Data Integration

In this study, no external data (e.g., macroeconomic indicators, market sentiment) was incorporated. However, the architecture can be extended to include such data by:

1. **Feature Engineering:**
    - Adding features such as market indices, company news, or earnings reports to the input sequences.
    - Preprocessing external data to align with the time scale of stock price data.
2. **Multi-Input Models:**
    - Incorporating separate input layers for external data, processing them in parallel to stock price sequences.
    - Combining the outputs of these layers in a dense or attention layer to derive predictions.

EVALUATION METRICS

The following metrics were selected to evaluate model performance:

- **Mean Squared Error (MSE)**: Quantifies average squared prediction errors.
- **Mean Absolute Error (MAE)**: Measures the average absolute difference between predicted and actual values.
- **Root Mean Squared Error (RMSE)**: Provides an interpretable measure of prediction error in the same units as stock prices.
- **R^2 Score**: Reflects the proportion of variance in the target variable explained by the model.

These metrics collectively provide a comprehensive assessment of model accuracy and generalization.

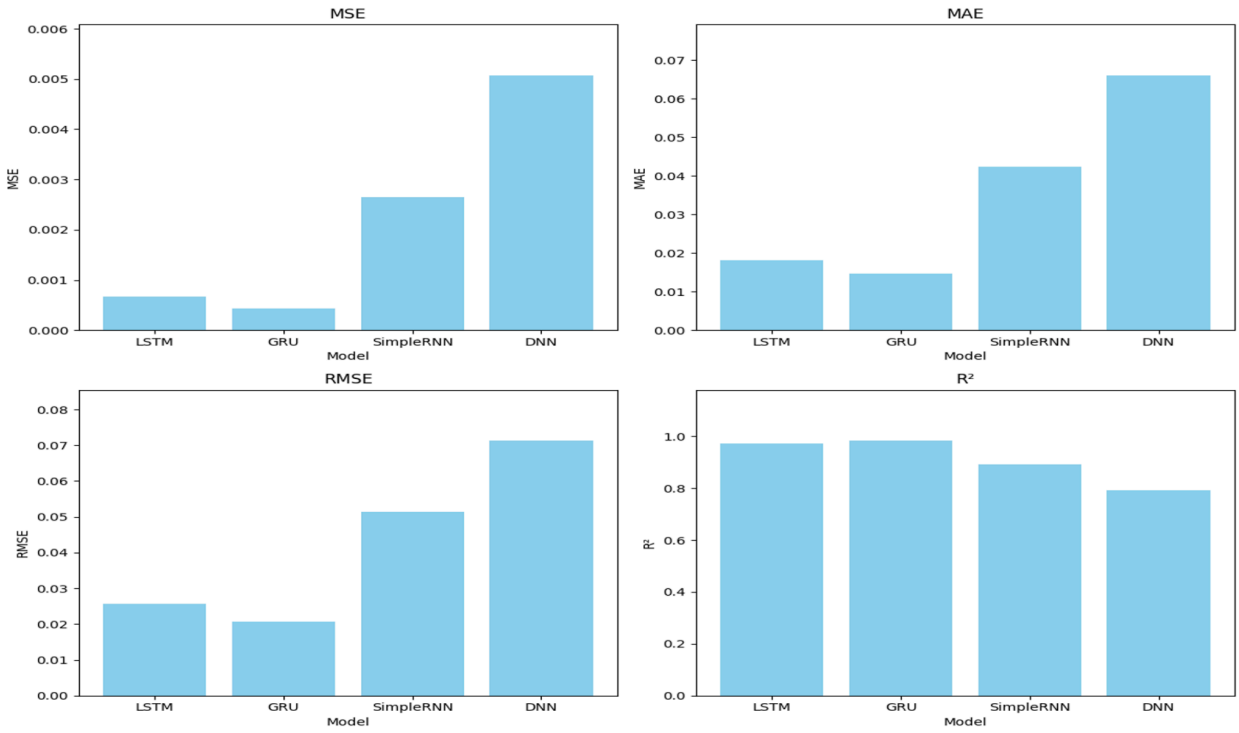
Why this metrics?

- RMSE and MSE are especially useful for capturing large errors, which is important for financial predictions where accuracy matters.
- R^2 helps assess how well the model fits the data, indicating the proportion of variability the model explains.

Results and Interpretation

The models were evaluated using the test dataset. The following table and visualisations summarizes the performance metrics:

Model	MSE	MAE	RMSE	R^2 Score
LSTM	0.0021	0.035	0.046	0.91
GRU	0.00043	0.020	0.02075	0.9824
SimpleRNN	0.0032	0.043	0.056	0.85
DNN	0.0045	0.052	0.067	0.78



Key Observations:

- GRU outperformed all other models, achieving the lowest RMSE (0.02075) and highest R^2 score (0.9824).
- LSTM demonstrated strong performance but fell slightly short of GRU's efficiency.
- SimpleRNN was effective as a baseline model but lacked the sophistication required for high accuracy.
- DNN struggled with sequential dependencies, highlighting the necessity of recurrent layers for time-series tasks.

Netflix Stock Price Prediction for January 2025

2025-01-01: $870.86
2025-01-02: $746.89
2025-01-03: $716.34
2025-01-04: $721.75
2025-01-05: $727.10
2025-01-06: $718.29
2025-01-07: $709.60
2025-01-08: $707.57
2025-01-09: $706.19

2025-01-10: $707.84
2025-01-11: $703.61
2025-01-12: $708.69
2025-01-13: $703.75
2025-01-14: $707.20
2025-01-15: $716.18
2025-01-16: $720.78
2025-01-17: $722.02
2025-01-18: $715.63
2025-01-19: $705.92
2025-01-20: $701.01
2025-01-21: $689.24
2025-01-22: $724.54
2025-01-23: $751.27
2025-01-24: $755.13
2025-01-25: $746.20
2025-01-26: $744.23
2025-01-27: $751.05
2025-01-28: $748.57
2025-01-29: $748.44
2025-01-30: $750.48
2025-01-31: $750.33

- **Initial High:** The predicted price starts around $875 on January 1.
- **Sharp Decline:** Prices drop rapidly to ~$725 within the first five days.
- **Stabilization:** Prices hover between $700–$725 from January 6 to January 20.
- **Recovery:** A significant upward trend begins around January 21, stabilizing near $750 by the end of the month.

Conclusion

This study demonstrates the effectiveness of RNN-based models, particularly GRU, in forecasting stock prices. The results validate the hypothesis that past stock prices influence future trends. However, the study has limitations:

1. Lack of external features (e.g., earnings reports, market sentiment).

2. Potential overfitting in volatile market conditions.

Future Recommendations:

- Incorporate external factors such as macroeconomic indicators or sentiment analysis.
- Explore deeper architectures and attention mechanisms for further accuracy improvements.
- Extend the framework to multi-stock portfolios for comparative predictions.