Stock Price Prediction Using Time Series Analysis and Neural Networks

1. Data Familiarization and Preparation

The project consists of two primary phases. In the first phase, we aim to familiarize ourselves with stock data and perform pre-processing to ensure its suitability as input for machine learning-based predictive models. This phase involves the following key steps:

- **Data Acquisition:** We obtained historical stock data relevant to our analysis.
- **Data Segmentation:** The dataset was filtered to focus on specific timeframes essential to our study.
- **Data Cleaning:** Erroneous or irrelevant data points were identified and removed to ensure data integrity.
- **Data Visualization:** We generated candlestick charts to analyze price movements and trends. These charts depict open, high, low, and close prices for each time interval, providing a comprehensive visual representation of market behavior.

This rigorous preprocessing ensures that the dataset is well-structured and reflective of real-world scenarios, thereby improving the effectiveness of our neural network models in stock price prediction.

1.1 Data Download and Initial Processing

We initiated the process by downloading stock data using the Pandas library, specifically focusing on stocks included in the S&P 500. The dataset initially contained 503 stocks across 8 columns. We then refined the data by including only stocks with available data from 2010 onwards, reducing the dataset to 290 stocks.

Subsequently, we acquired daily price data for these stocks via the Yahoo Finance library. The dataset includes:

- **Open:** The stock's opening price for the day.
- **High:** The highest trading price during the day.
- **Low:** The lowest trading price during the day.
- **Close:** The closing price of the stock.
- **Adjusted Close:** The closing price adjusted for corporate actions such as dividends.
- **Volume:** The number of shares traded.

The final dataset comprises 3,513 rows and 1,740 columns, covering data from January 4, 2010, to the present. It is worth noting that trading data is unavailable for weekends and the first three days of each year due to market closures.

1.2 Time Series Data Exploration and Cleaning

In time series data, missing values can arise due to various factors. To address this:

- Missing values at the start of the series were handled using backward filling, replacing null entries with the next available data point.
- Missing values in the middle or end were treated with forward filling, using the last recorded value.
- Stocks inactive during specific periods retained their last available prices.

Following the cleaning process, we removed a stock (B.BF) due to complete absence of data (all values were NaN). We verified data integrity by ensuring no missing values remained.

1.3 Visualization and Analysis

We conducted exploratory data analysis by:

- Plotting a histogram of the close price returns for the stock **ROK**, showcasing the frequency of price changes in percentage terms, useful for investment risk management. Most price variations fell within the -3% to +3% range, indicating relative stability.
- Generating a **candlestick chart** for ROK, covering daily price movements from January 4, 2010, to December 18, 2023. White candles represent price increases, while black candles denote price declines.

2. Neural Network Training Using New Data

In the second phase, we acquired a different dataset and replicated the preprocessing steps (excluding visualization). We refined and normalized data for each stock separately before modeling. We selected five stocks (**GOOGL, AMZN, AAPL, META, and MSFT**) with data from 2018 onwards to avoid missing values.

For **GOOGL**, we conducted additional analysis by training predictive models with both normalized and non-normalized data. Subsequently, we utilized the trained models to forecast stock price movements and examined their accuracy using **naïve forecasting**.

2.1 Time Series Cross-Validation

Traditional cross-validation is unsuitable for time series due to look-ahead bias. Instead, we implemented:

- **Rolling cross-validation:** Expanding training data iteratively.
- **Expanding window cross-validation:** Using progressively larger datasets for validation.

2.2 Model Input and Output Preparation

To reduce dimensionality, only the adjusted close price was used as an input feature. We adopted a **sliding window approach**, segmenting the time series into 20-day windows to predict the price for the 21st day. The window was then shifted forward by one day, repeating the process.

Data was normalized to a [0,1] range for optimal model performance. We also experimented with non-normalized data for comparative analysis.

2.3 Neural Network Models

We implemented the following neural network models:

- **LSTM (Long Short-Term Memory)**: Captures long-term dependencies and mitigates the vanishing gradient problem.
- **GRU (Gated Recurrent Unit)**: A simplified alternative to LSTM with two gates.
- **Bi-LSTM (Bidirectional LSTM)**: Processes data in both forward and reverse directions for enhanced accuracy.
- **MLP (Multilayer Perceptron)**: A feedforward network, less effective for time series.
- **CNN (Convolutional Neural Network)**: Extracts spatial features from time series data.
- **ConvLSTM (Convolutional LSTM)**: Combines CNN and LSTM for both spatial and temporal feature extraction.

Each model was trained on the **Adjusted close prices** of the five selected stocks.

2.3.1 Performance Analysis

- LSTM, GRU, Bi-LSTM, and CNN performed well with low Mean Squared Error (MSE) and Mean Absolute Error (MAE).
- MLP and ConvLSTM exhibited slightly weaker performance.
  For non-normalized data, LSTM-based models struggled to converge, while CNN and MLP performed comparatively better, highlighting the importance of normalization.
- ConvLSTM showed a declining loss over the first 20 epochs but failed to reach zero, indicating difficulty in capturing patterns. Increasing epochs might improve its performance.

- Overall, models trained on normalized data provided more accurate future price predictions, reinforcing the necessity of data normalization in stock forecasting..

2.4 Naïve Forecasting Benchmark

We implemented a **Naïve Forecast** model, where the next day's price is predicted as the current day's price. Despite its simplicity, it served as a useful baseline with surprisingly reasonable results.

3. Conclusion

This project explored multiple neural network models for stock price prediction. Key findings include:

- **LSTM, GRU, Bi-LSTM, and CNN** demonstrated strong predictive performance, especially with normalized data.
- **MLP and ConvLSTM** were less effective due to their architectures.
- **The Naïve Forecast model** provided a strong baseline, showing that even simple methods can yield meaningful results.

The study highlights the significance of **data preprocessing, model selection, and evaluation metrics** in time series forecasting. Future research could focus on incorporating additional predictive features, extending prediction horizons, and experimenting with hybrid architectures to enhance forecasting accuracy.

Section titles for the code in Google Colab are as following:

Importing Required Libraries

1- Data Acquisition and Preprocessing

1.1 Data Downloading

1.2 Data Cleaning and Preparation

1.3 Statistical Analysis: Histogram of Stock Returns

1.4 Candlestick Chart for ROK Open Prices

2- Stock Price Prediction Using Neural Networks

2.1 Data Processing for Model Training

2.2 Loading Main Tickers Data from CSV

2.3 Google Stock Prediction

2.3.1 Training on Normalized Data

- **2.3.1.1 Data Preprocessing**
- **2.3.1.2 LSTM Model Training**
- **2.3.1.3 GRU Model Training**
- **2.3.1.4 BiLSTM Model Training**
- **2.3.1.5 MLP Model Training**
- **2.3.1.6 ConvLSTM Model Training**
- **2.3.1.7 CNN Model Training**
- **2.3.1.8 Stock Price Prediction Using Trained Models**

2.3.2 Training on Non-Normalized Data

- **2.3.2.1 Data Preprocessing**
- **2.3.2.2 LSTM Model Training**
- **2.3.2.3 GRU Model Training**
- **2.3.2.4 BiLSTM Model Training**
- **2.3.2.5 MLP Model Training**

- **2.3.2.6 ConvLSTM Model Training**
- **2.3.2.7 CNN Model Training**
- **2.3.2.8 Stock Price Prediction Using Trained Models**

2.4 Amazon Stock Prediction

2.4.1 Training on Normalized Data

- **2.4.1.1 Data Preprocessing**
- **2.4.1.2 LSTM Model Training**
- **2.4.1.3 GRU Model Training**
- **2.4.1.4 BiLSTM Model Training**
- **2.4.1.5 MLP Model Training**
- **2.4.1.6 ConvLSTM Model Training**
- **2.4.1.7 CNN Model Training**
- **2.4.1.8 Stock Price Prediction Using Trained Models**

2.5 Meta Stock Prediction

2.5.1 Training on Normalized Data

- **2.5.1.1 Data Preprocessing**
- **2.5.1.2 LSTM Model Training**
- **2.5.1.3 GRU Model Training**
- **2.5.1.4 BiLSTM Model Training**
- **2.5.1.5 MLP Model Training**
- **2.5.1.6 ConvLSTM Model Training**
- **2.5.1.7 CNN Model Training**
- **2.5.1.8 Stock Price Prediction Using Trained Models**

2.6 Microsoft Stock Prediction

2.6.1 Training on Normalized Data

- **2.6.1.1 Data Preprocessing**
- **2.6.1.2 LSTM Model Training**
- **2.6.1.3 GRU Model Training**
- **2.6.1.4 BiLSTM Model Training**
- **2.6.1.5 MLP Model Training**
- **2.6.1.6 ConvLSTM Model Training**
- **2.6.1.7 CNN Model Training**
- **2.6.1.8 Stock Price Prediction Using Trained Models**

2.7 Apple Stock Prediction

2.7.1 Training on Normalized Data

- **2.7.1.1 Data Preprocessing**
- **2.7.1.2 LSTM Model Training**
- **2.7.1.3 GRU Model Training**
- **2.7.1.4 BiLSTM Model Training**
- **2.7.1.5 MLP Model Training**
- **2.7.1.6 ConvLSTM Model Training**
- **2.7.1.7 CNN Model Training**
- **2.7.1.8 Stock Price Prediction Using Trained Models**

2.8 Naïve Forecasting for Google Stock