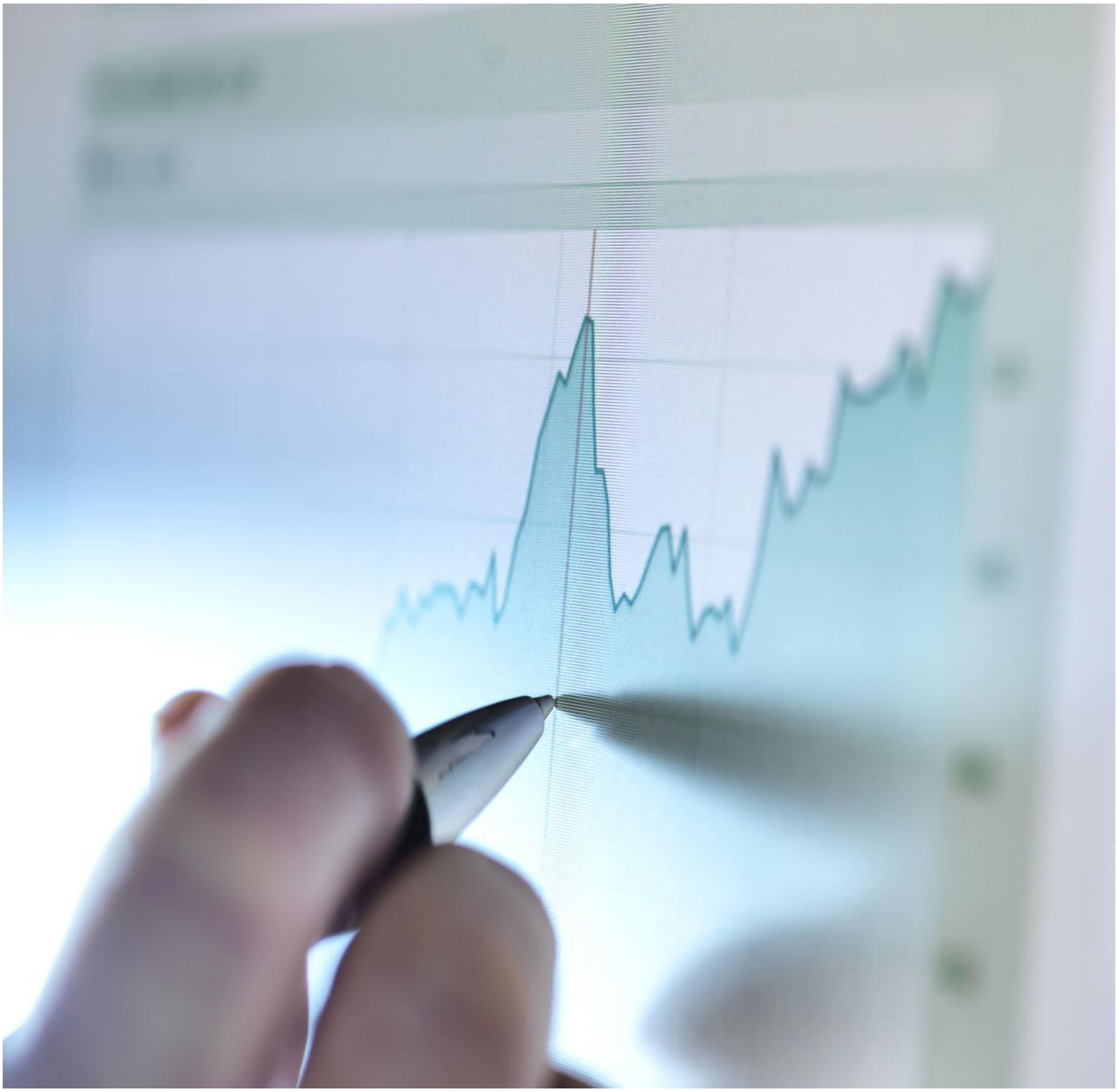

WiDS – UID #47

A GUIDE TO TENSORFLOW

TIME SERIES FORECASTING USING TENSORFLOW

Pratiek Sonare | 22B2440 | +91-9892598864

1. Introduction

1.1 Time Series Forecasting

Time series forecasting is a crucial aspect of data analysis and prediction, finding applications in various fields such as finance, economics, meteorology, and more. It involves the analysis of data points collected over time to uncover patterns, trends, and make predictions about future values. This technique is essential for decision-making processes, allowing businesses and researchers to anticipate future trends and plan accordingly.

1.2 Goal of the Project

The goal of this project is to leverage time series forecasting techniques, specifically Long Short-Term Memory (LSTM) neural networks and Recurrent Neural Networks (RNN), to predict the next stock closing prices. Stock price prediction is a challenging and significant task in the financial domain, with potential applications in investment decision-making and risk management.

2. Data Collection

The data for this project is collected using the Yahoo Finance API through the yfinance, pandas-datareader library. Yahoo Finance provides historical stock price data, including Open, High, Low, Close, and Volume. The chosen company for analysis is "AMZN" (Amazon), a prominent player in the technology and e-commerce sectors.

2.1 Time Range

The historical stock prices for the company "AMZN" are retrieved from January 1, 2012, to January 1, 2020, for training the LSTM model. The test data is collected from January 1, 2020, to the present. One can also choose to analyze the accuracy of the forecasting model by testing it with a much smaller data (for ex: from January 1, 2022 to the present) however, using a large test dataset seemed more appropriate.

3. Data Preprocessing

3.1 Min-Max Scaling

Data preprocessing is a crucial step to ensure uniformity in the input data. The closing prices of the stock are scaled using Min-Max scaling, bringing them within the range of 0 to 1. This normalization helps the model converge faster during training and ensures that no particular feature dominates the learning process.

3.2 Sequence Creation

To feed the LSTM model with sequential data, input sequences are created. Each input sequence consists of a window of historical closing prices, and the corresponding target is the subsequent closing price. This sequence creation establishes the temporal relationships within the data.

4. LSTM Model Architecture

4.1 Introduction to LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data. LSTMs are particularly well-suited for time series forecasting tasks where understanding past context is crucial for predicting future outcomes.

4.2 Model Architecture

The LSTM model architecture consists of three LSTM layers followed by a Dense output layer. The first LSTM layer is configured to return sequences, allowing it to provide a sequence of outputs to the subsequent layers. Dropout layers are added after each LSTM layer to enhance generalization and mitigate overfitting. One can get varying results on adjusting architecture structures and parameters, however, I have dedicated myself to use this basic architecture structure only.

4.3 Compilation and Training

The model is compiled using the Adam optimizer and mean squared error loss function. The compilation step prepares the model for training. The training process involves fitting the model to the training data for a specified number of epochs and a given batch size. This LSTM architecture is designed to capture temporal dependencies and patterns in the input sequences, enabling the model to make accurate predictions for future stock prices. The combination of multiple LSTM layers and dropout regularization enhances the model's ability to generalize well to unseen data.

5. Model Evaluation

The trained model is evaluated on the test data, and performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are calculated.

6. Results and Visualization

The trained LSTM model is employed to predict the next closing price of the stock. The prediction involves feeding the model with sequences of past closing prices, and the model outputs the predicted closing price for the next time step.

6.1 Interpretation of Results

The predicted closing price provides valuable information for investors and traders. By analyzing the predicted price, stakeholders can gain insights into potential future trends. If the predicted closing price is higher than the current one, it might suggest a potential upward trend, indicating a buying opportunity. Conversely, a predicted lower closing price might imply a potential downturn, prompting a cautious approach or even selling decisions.

6.2 Visualization

The results are visually represented through a plot that compares the actual closing prices with the predicted ones. Additionally, moving average curves are plotted to highlight trends and smooth out short-term fluctuations.

1. Using LSTM Architecture

With varying step-sizes (epoch), the batch-size was kept constant. For large amounts of data, the batch-size is also kept at larger values. However, on further analysis, it was found that varying batch-sizes did not bring significant improvement in the overall accuracy of the fit.

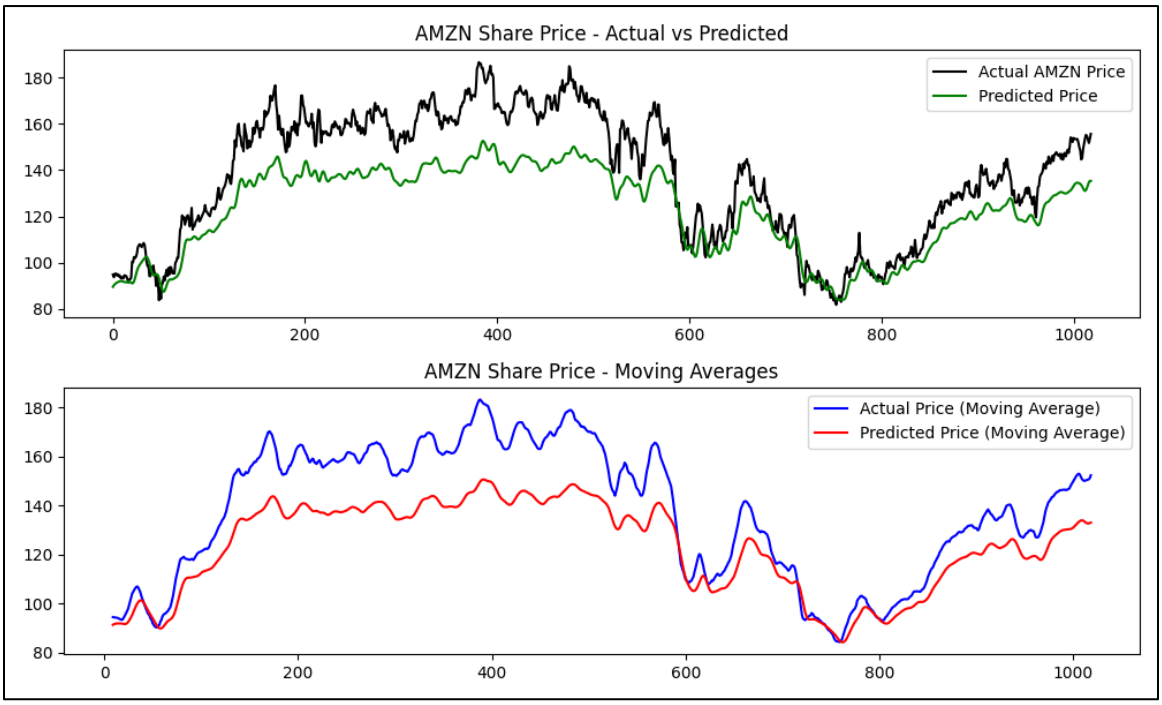

Fig 1.

Prediction: [[135.71634]]

Root Mean Squared Error (RMSE): 17.0774

Mean Absolute Error (MAE): 14.2078

Similarly, changing neural network parameters (neuron sizes, number of hidden layers) through trial and error caused Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to decrease eventually, to decrease to its minimum.

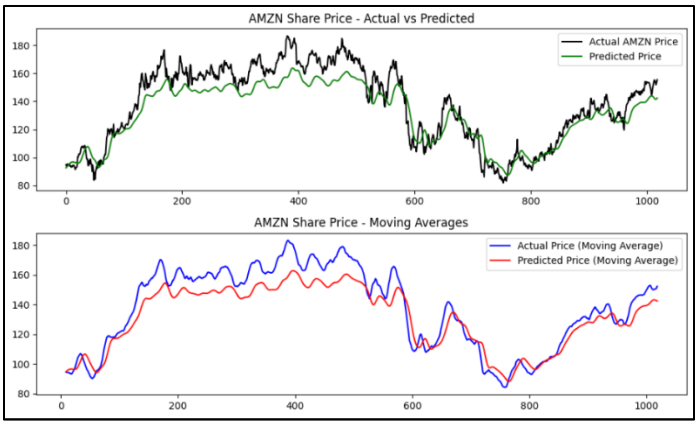

Fig 2.

Prediction: [[142.83893]]

Root Mean Squared Error (RMSE): 11.0565

Mean Absolute Error (MAE): 9.0709

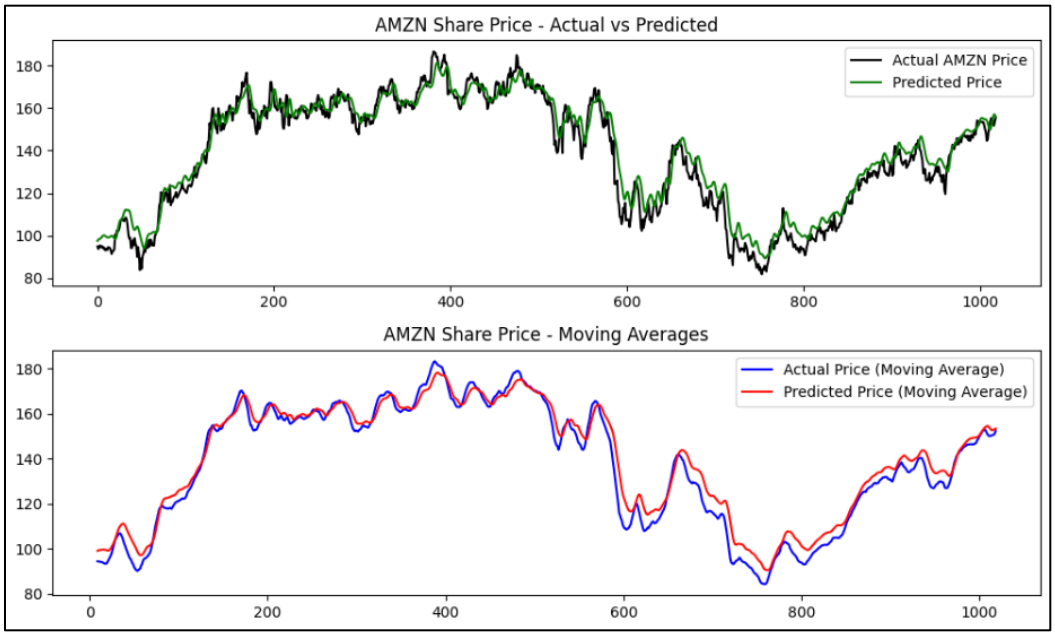

Fig 3.

Prediction: [[154.94463]]

Root Mean Squared Error (RMSE): 6.2422

Mean Absolute Error (MAE): 4.86032

*is the lowest ever error metrics received through trial and error