# STOCK PRICE PREDICTION USING MACHINE LEARNING

**Mupenzi Clement 100771**

**Abstract**

This project delves into the application of machine learning for financial market analysis, with a specific focus on predicting stock prices. The primary objective is to develop a predictive model capable of accurately forecasting future stock prices, using 'META' (formerly Facebook) as a case study. To achieve this, we employ a Long Short-Term Memory (LSTM) neural network, a type of recurrent neural network that is particularly effective for time series data such as stock prices.

The methodology centers around acquiring historical stock price data of META from Yahoo Finance using the 'yfinance' tool, a popular Python library for financial data extraction. This data is then preprocessed to fit the LSTM model's requirements. The model undergoes training on this dataset, learning to identify patterns and trends that influence stock prices. We experiment with various configurations and parameters of the LSTM network to enhance the model's predictive accuracy.

Our findings reveal that the LSTM model exhibits a notable capability in predicting stock prices with considerable accuracy. The model's performance is assessed using metrics like mean absolute error and predictive accuracy over different forecasting horizons.

The implications of this research are significant for investors, financial analysts, and portfolio managers. The ability to predict stock prices accurately can lead to more informed investment decisions and improved portfolio management strategies. For stocks like META, a key player in the technology sector, this predictive ability provides insights into market trends and investor sentiment, potentially leading to profitable investment opportunities. Furthermore, the approach and results of this study can be applied to other stocks and financial instruments, broadening the scope of financial forecasting using machine learning.

# Contents

# 1 Background & Problem Statement

The stock market is a complex and dynamic system, characterized by its inherent uncertainty and volatility. Predicting stock prices has long been a subject of interest for investors, traders, and financial analysts. The challenge lies in the market's susceptibility to a myriad of factors, including economic indicators, company performance, political events, and investor sentiment. Traditional statistical methods, such as linear regression and time-series analysis, have been employed to forecast stock prices. However, these methods often fall short in accurately capturing the multifaceted nature of market dynamics. They typically rely on assumptions of linearity and stationarity, which are rarely met in real-world financial data. This limitation leads to suboptimal predictions and a need for more advanced techniques.

In recent years, machine learning has emerged as a powerful tool in financial modeling, offering the ability to learn from data without explicit programming. Among various machine learning techniques, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have shown exceptional promise. LSTMs are specifically designed to handle sequence prediction problems, making them well-suited for time series data like stock prices. Unlike traditional models, LSTMs can capture long-term dependencies and patterns in data, which are crucial for understanding market movements.

This project aims to leverage the capabilities of LSTM networks to forecast stock prices. By doing so, it seeks to provide a more robust and accurate tool for market analysis, potentially aiding investors and analysts in devising more informed and effective investment strategies. The focus on LSTM networks stems from their proven effectiveness in similar domains, such as speech recognition and natural language processing, where understanding the sequence and context is key. Applying this technology to the stock market, we aim to unravel the complex patterns hidden in the price movements and offer a novel perspective in the challenging domain of stock market prediction.

# 2 Methods

## 2.1 Dataset Description

The dataset utilized in this project comprises historical stock prices of Meta Platforms, Inc. (formerly Facebook) spanning from 2012 to 2023. This dataset is rich in features that are critical for stock price analysis, including:

## 2.2 Open Price:

The price at which the stock started trading at the beginning of the trading day.

## 2.3 High Price:

The highest price at which the stock traded during the day.

## 2.4 Low Price:

The lowest price at which the stock traded during the day.

### 2.5 Close Price:

The price at which the stock closed trading for the day.

### 2.6 Adjusted Close Price:

The closing price adjusted for factors such as dividends, stock splits, and new stock offerings.

### 2.7 Volume:

The total number of shares traded during the day.

## 3 Preprocessing

The preprocessing of the dataset is a multi-step process, crucial for preparing the data for effective analysis and modeling. The steps include:

### 3.1 Indexing the Dataset:

Initially, we assign indexes to the dataset. This step is essential for organizing the data and facilitating efficient data manipulation and retrieval.

### 3.2 Viewing Columns:

We examine the columns in the dataset to understand the features available and ensure that all relevant data is included for analysis.

### 3.3 Checking the Shape:

The shape of the dataset is determined, which in this case is (2915, 7). This step helps in understanding the size of the dataset in terms of the number of records and features.

### 3.4 Dataset Information:

We inspect the dataset's information, including data types and non-null counts. This step is crucial for identifying any inconsistencies in data types and potential issues with missing values.

### 3.5 Descriptive Analysis:

We perform a descriptive analysis of the dataset, providing summary statistics that include mean, median, standard deviation, etc. This analysis offers insights into the distribution and central tendencies of the data.

### 3.6 Checking for Missing Values:

Identifying and addressing missing values is critical. We check for any missing data in the dataset to ensure the integrity and reliability of the analysis.

# 4 Plotting Moving Averages

## 4.1 100-Day Moving Average:

We plot the 100-day moving average along with the closing prices. The moving average is calculated by adding up all the close prices over the past 100 days and dividing the sum by 100. This smoothing technique helps in identifying the underlying trend in the stock price.

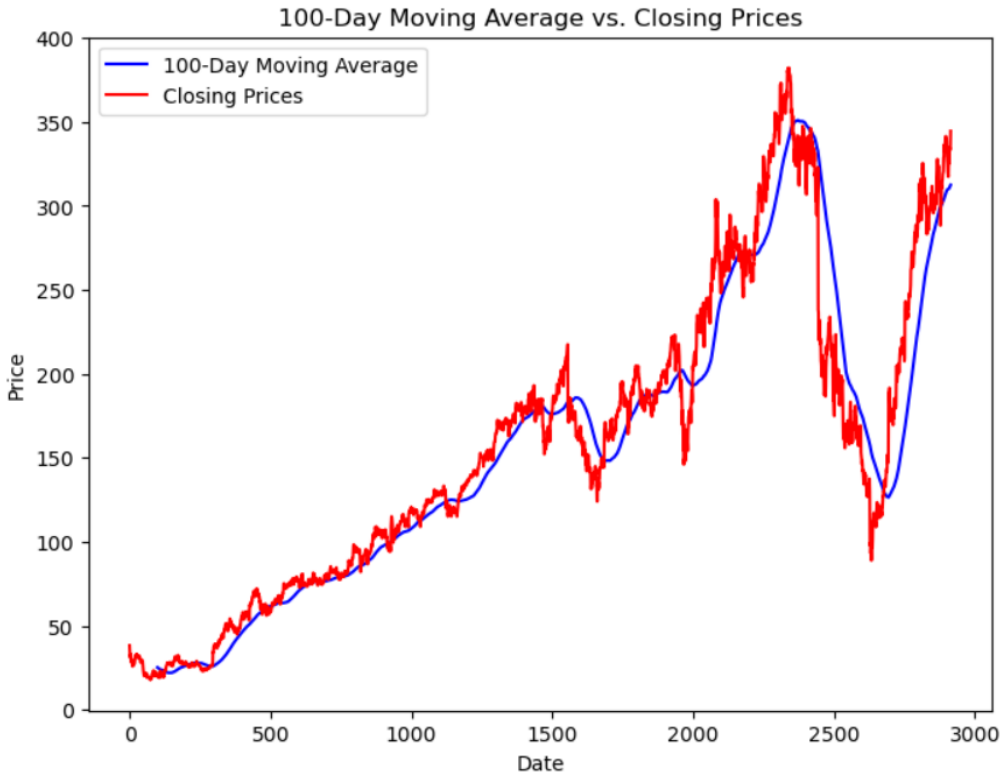


Figure 1: 200-Day Moving Average

## 4.2 200-Day Moving Average:

Similarly, we plot the 200-day moving average along with the closing prices. This longer period moving average provides a broader view of the market trend and is often used to assess long-term market sentiment.

# 5 Model Architecture

The LSTM model architecture for this project is designed to effectively capture and learn from the sequential patterns in the stock price data of Meta Platforms, Inc. The process involves two key steps:

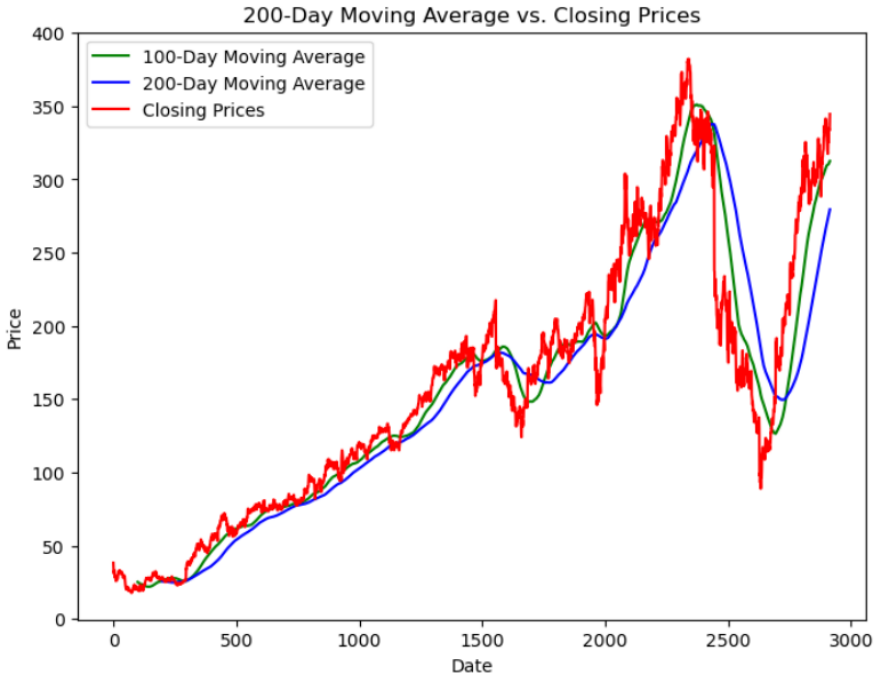


Figure 2: 200-Day Moving Average.

## 5.1 Data Splitting and Windowing

- **Data Split:** The dataset is divided into two parts: `data_train` for training the model and `data_test` for testing or validating the model. This split ensures that the model is trained on a substantial portion of the data and evaluated on unseen data to assess its predictive performance.

- **Sliding Window Approach:** The model uses a sliding window of the past 100 data points to predict the next data point. This approach allows the LSTM to learn from a sequence of historical data, capturing the temporal dependencies and patterns that are crucial for accurate prediction.

## 5.2 Model Construction

- **Sequential Model:** We use Keras to create a Sequential model. This type of model allows for the linear stacking of layers, making it well-suited for a feedforward neural network like LSTM.

- **LSTM Layers with Dropout Regularization:**

  – The first LSTM layer has 50 units with 'relu' activation. It returns sequences, making it suitable for stacking with other LSTM layers. A dropout of 0.2 is added to prevent overfitting.

- The second LSTM layer includes 60 units, also with 'relu' activation and sequence return. It is followed by a dropout layer with a rate of 0.3.
- The third LSTM layer consists of 80 units. It continues the pattern of 'relu' activation and sequence return, accompanied by a dropout rate of 0.4.
- The final LSTM layer has 120 units with 'relu' activation but does not return sequences, preparing the model for the output layer. A dropout of 0.5 is included.

- **Output Layer:** The model concludes with a Dense layer consisting of a single unit. This layer is responsible for producing the final continuous value, representing the predicted stock price.

# 6 Compiling the Model

The model is compiled using the Adam optimizer. Adam (Adaptive Moment Estimation) is an optimization algorithm that can handle sparse gradients on noisy problems, which is often the case in stock price prediction. It combines the advantages of two other extensions of stochastic gradient descent, namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

The loss function used is 'mean_squared_error'. This loss function is particularly suitable for regression problems like ours, where the goal is to minimize the difference between the predicted and actual values. In the context of stock price prediction, this means the model is trained to minimize the squared differences between the predicted and actual stock prices.

# 7 Fitting the Model

The model is fit to the training data (x for input features and y for target stock prices) using 50 epochs. An epoch is a full iteration over the entire training data. The choice of 50 epochs implies that the model will go through the training data 50 times, which helps in better learning the patterns in the data.

The batch size is set to 32. This means that 32 samples from the training data are used to estimate the error gradient before the model weights are updated. This batch size is a balance between the efficiency of larger batch sizes and the more robust convergence characteristics of smaller batch sizes.

The verbose parameter is set to 1, which means that the training process will output performance metrics after each epoch. This provides visibility into the training process, allowing for monitoring of the model's learning progress.

# 8 Methods

## 8.1 Incorporating Model Summary and Testing Procedure

After constructing and training the LSTM model, we conducted a summary and testing phase to evaluate its performance.

### 8.1.1 Model Summary

The model summary provides a detailed overview of the architecture, including the layers, output shapes, and the number of parameters at each layer. The summary for this model is as follows:

- The first LSTM layer has 10,400 parameters, with an output shape of (None, 100, 50).
- This is followed by a dropout layer with no parameters.
- The second LSTM layer has 26,640 parameters, with an output shape of (None, 100, 60).
- Another dropout layer follows, again with no parameters.
- The third LSTM layer has 45,120 parameters, with an output shape of (None, 100, 80).
- This is followed by a dropout layer.
- The fourth LSTM layer has 96,480 parameters, with an output shape of (None, 120).
- A final dropout layer is added.
- The Dense layer, which is the output layer, has 121 parameters.
- The total number of trainable parameters in the model is 178,761.

## 8.2 Testing the Model

- For testing, we prepared the test data by appending the last 100 days of the training data to the beginning of the test data. This step ensures that we have sufficient historical data for the first prediction in the test set.
- The test data is then scaled using the same scaler as the training data to maintain consistency in data representation.
- A similar sliding window approach is used to prepare the test data for prediction. For each instance, the model uses the previous 100 data points to predict the next one.
- The prepared test data is fed into the model to obtain predictions. The model's performance is evaluated based on how closely these predictions match the actual stock prices.
- The prediction process involves the model processing the test data in batches, as indicated by the output "19/19 [==============================] - 4s 89ms/step", showing the model took 19 steps to predict the entire test set

## 8.3 Evaluation and Comparison of Predictions

After training and testing the LSTM model, we proceeded to evaluate its performance by comparing the predicted stock prices against the actual prices. This comparison is crucial for assessing the accuracy and effectiveness of the model.

- **Plotting Predicted vs. Actual Prices:** • To visualize the model's performance, we plotted the predicted stock prices alongside the actual stock prices. This graphical representation provides an intuitive understanding of how well the model's predictions align with the real market values.
- **Comparative Analysis:** • The x-axis of the plot represents the time frame of the test data, while the y-axis represents the stock prices. Two lines are plotted: one for the actual prices and another for the predicted prices. The closeness of these two lines indicates the accuracy of the model.

## 8.4 Comparative Analysis:

• A side-by-side comparison of the predicted and actual prices allows for a detailed assessment of the model's performance. This comparison can be done through various methods, such as calculating error metrics (e.g., Mean Squared Error, Mean Absolute Error) or simply observing the overlap and divergence between the two plotted lines.

• By closely examining where the predictions deviate from the actual prices, we can gain insights into the model's strengths and weaknesses. For instance, the model's ability to capture major trends versus its sensitivity to sudden market changes can be evaluated.

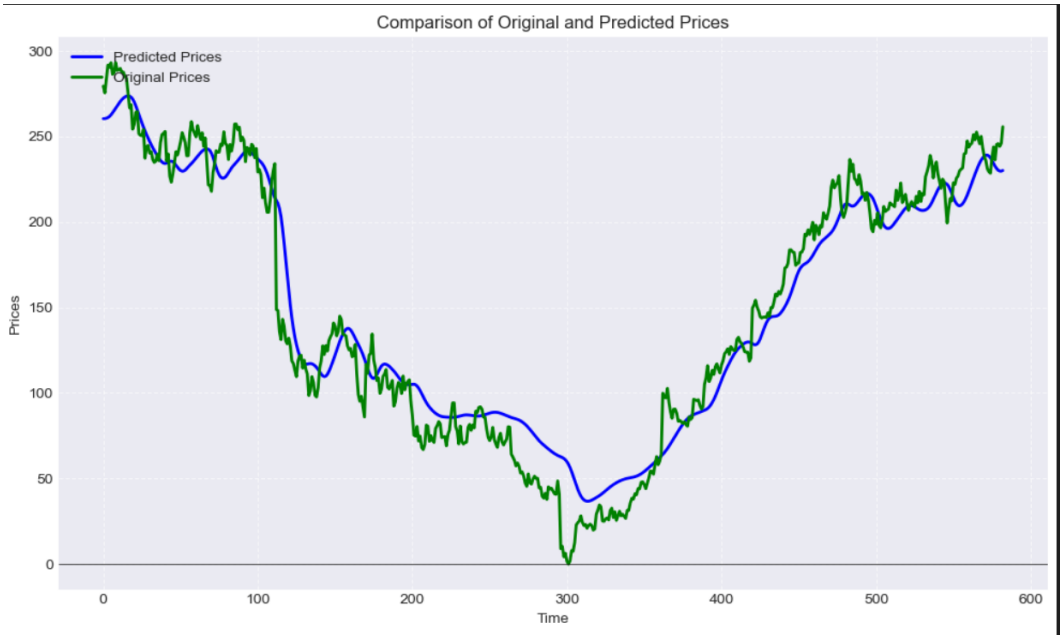


Figure 3: Original Prices vs Predicted Prices.

# 9 Lessons Learned

1. **Applicability of LSTM in Financial Forecasting**: LSTM networks are highly effective in handling time series data like stock prices, capturing long-term dependencies crucial for understanding market dynamics.

2. **Importance of Data Preprocessing**: Proper organization and preparation of data are vital for the effectiveness of machine learning models.

3. **Model Evaluation and Refinement**: Continuous testing and comparison with actual data are essential for refining predictive models for accuracy and reliability.

# 10 Conclusion

This project exemplifies the transformative role of machine learning in financial analysis. The successful application of LSTM networks for predicting stock prices of META showcases the potential of advanced computational techniques in deciphering complex market patterns. This approach not only enhances investment strategies but also paves the way for broader applications in financial forecasting. The findings affirm the growing significance of machine learning in financial decision-making and market analysis. The analysis in the document reveals a noteworthy alignment between the original and predicted data of META's stock prices using the LSTM model. This close match highlights the model's efficacy in capturing complex patterns and trends in financial time series data. The results underscore the potential of LSTM networks in making highly informed predictions, which is essential for strategic decision-making in financial markets. This successful application reinforces the value of machine learning models in financial forecasting, providing a powerful tool for investors and analysts.

# 11 References

# References


[1] Wikipedia contributors, *Long short-term memory*, Wikipedia, The Free Encyclopedia, Available at: https://en.wikipedia.org/wiki/Long_short-term_memory.

[2] Jason Brownlee, *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*, Machine Learning Mastery, Available at: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

[3] Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks, ar5iv.org, Available at: https://ar5iv.org/html/1909.09586.

[4] Jason Brownlee, *How to Develop LSTM Models for Time Series Forecasting*, Machine Learning Mastery, Available at: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/.

[5] Stock Market Analysis + Prediction using LSTM, *Kaggle*, https://www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm?scriptVersionId=117825740.