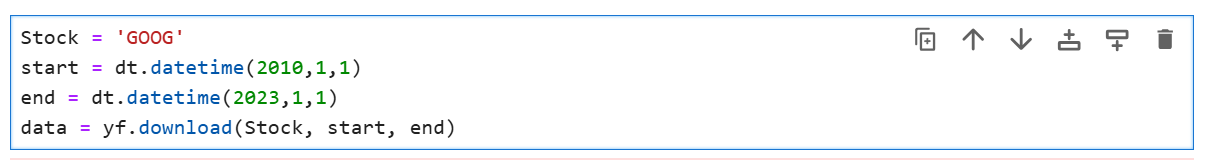
## Stock Price Prediction Using LSTM

The stock market has always been a subject of interest for investors and analysts alike, primarily due to its potential for profit generation. Predicting stock prices, however, is a complex task influenced by numerous factors including market trends, economic indicators, and investor sentiment. In this analysis, we delve into the methodology of predicting stock prices using Long Short-Term Memory (LSTM) networks—a type of recurrent neural network (RNN) adept at learning from sequential data. This report will cover the entire process from data acquisition to model evaluation, providing insights into each step.

## 1. Data Acquisition

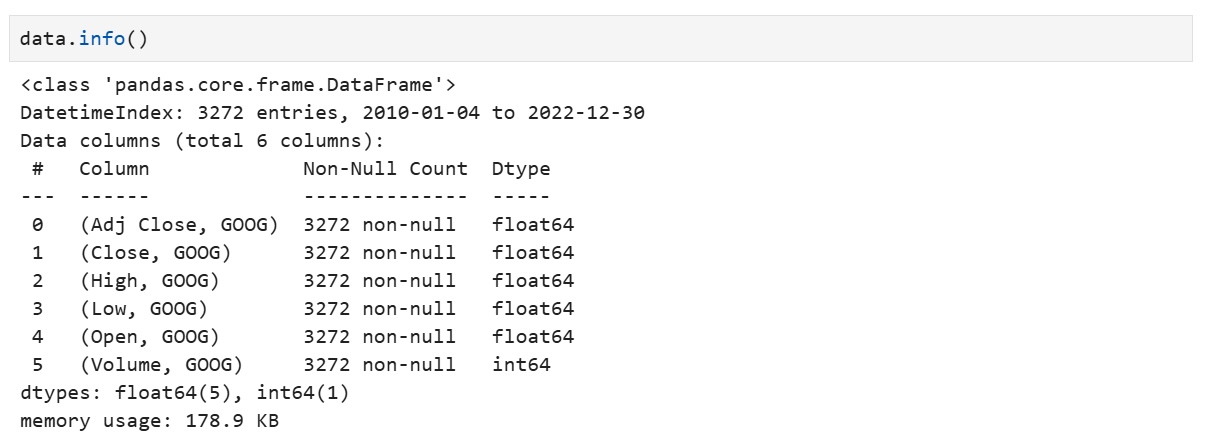
The first step in any predictive modeling task is acquiring the relevant data. For this analysis, we focus on Google's stock price (ticker symbol: GOOG) over a period spanning from January 1, 2010, to January 1, 2023. The data is sourced using the yfinance library, which allows users to download historical market data from Yahoo Finance seamlessly.



The dataset comprises various columns including adjusted close prices, close prices, high and low prices, open prices, and trading volume. Upon downloading the data, we find that it contains a total of 3,272 entries.

## 2. Data Exploration

After acquiring the data, it is essential to explore its structure and contents. This involves checking for missing values and understanding the types of data present.



The dataset reveals that all columns are populated with no null values. This is crucial as missing data can lead to erroneous predictions if not handled properly.

## 3. Data Visualization

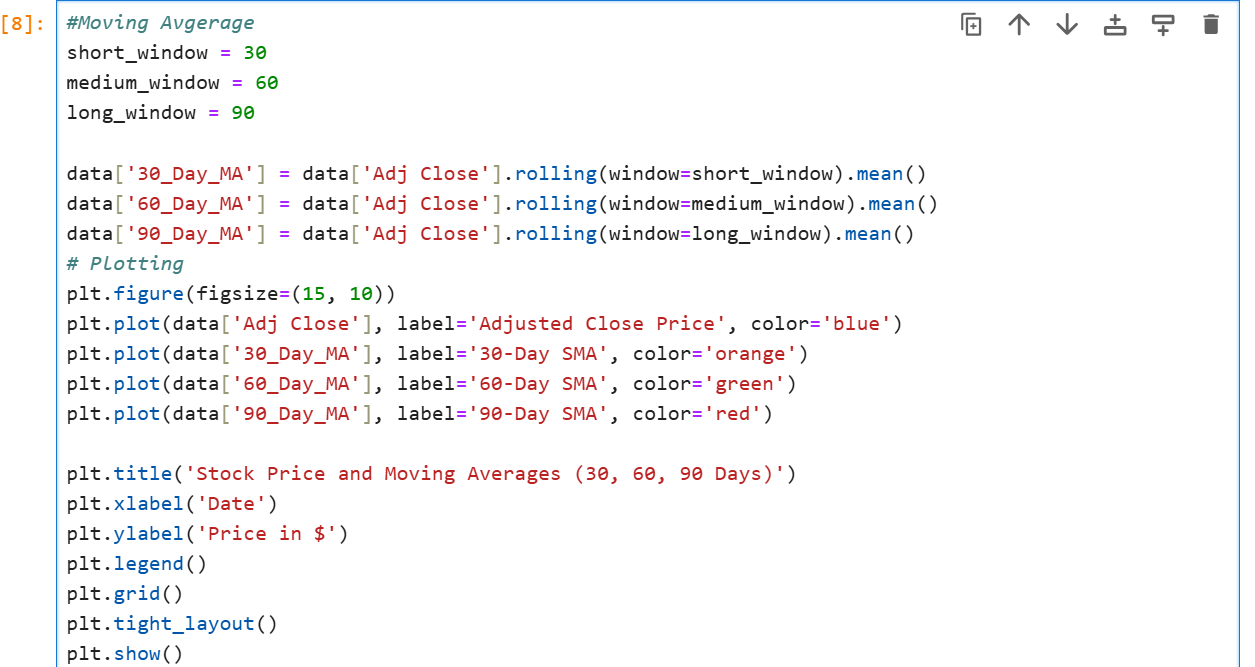
Visualizing the adjusted closing prices over time helps in understanding trends and patterns in the stock's performance. A line plot can be created using matplotlib:



This visualization provides insights into how the stock price has fluctuated over time and highlights significant trends or events that may have impacted its performance.

## 4. Moving Averages Calculation

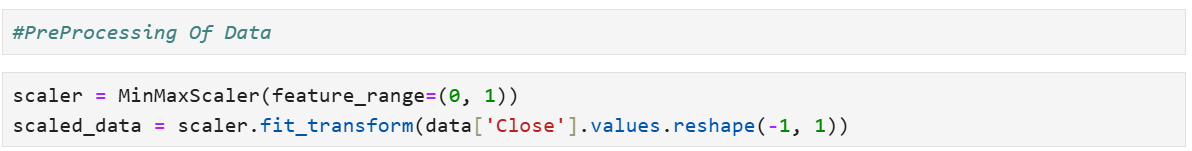
To further analyze trends in stock prices, moving averages are calculated over different time windows—30 days, 60 days, and 90 days. Moving averages help smooth out price fluctuations and provide a clearer view of the underlying trend.



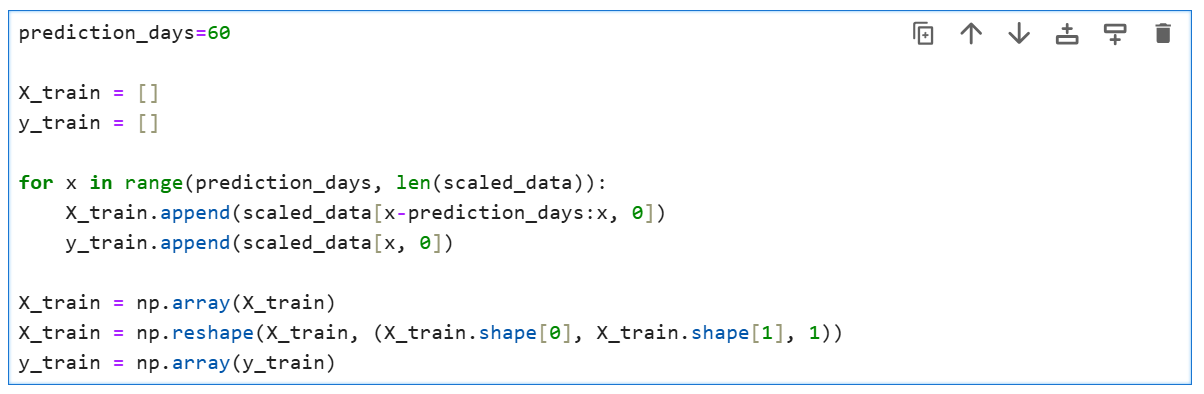
This analysis highlights how moving averages can help identify trends by smoothing out short-term fluctuations in stock prices.

## 5. Data Preprocessing

Before feeding the data into an LSTM model, it is crucial to preprocess it appropriately. This includes normalizing the data using MinMaxScaler, which scales the values between a specified range (0 to 1 in this case). Normalization helps improve model convergence during training.

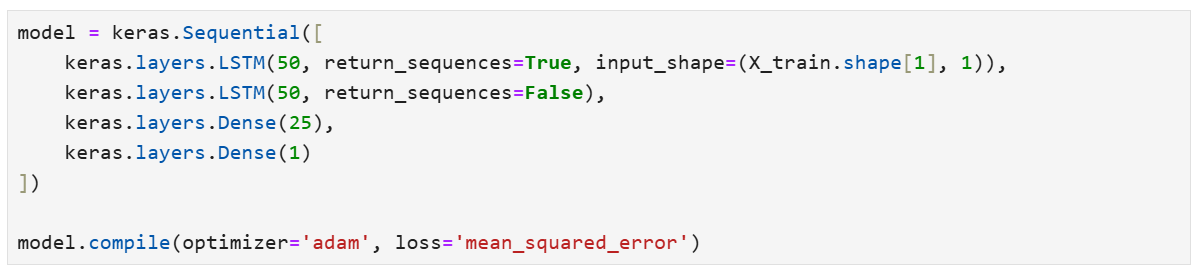


Next, we prepare the training dataset by creating sequences of past observations to predict future prices. For this analysis, we use a prediction window of 60 days:



## 6. Model Creation Using LSTM

With the data prepared and normalized, we can now define our LSTM model architecture using Keras:



The model consists of two LSTM layers followed by dense layers. The first LSTM layer returns sequences to allow for further processing by subsequent layers while the second LSTM layer does not return sequences as it feeds into dense layers.

## 7. Model Training

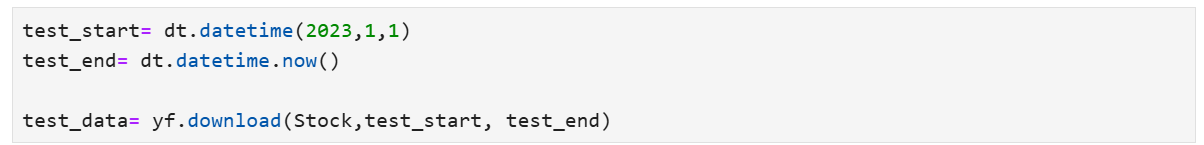
The model is trained on the prepared dataset with a batch size of 32 over three epochs:

Screenshot 2024-12-10 021303

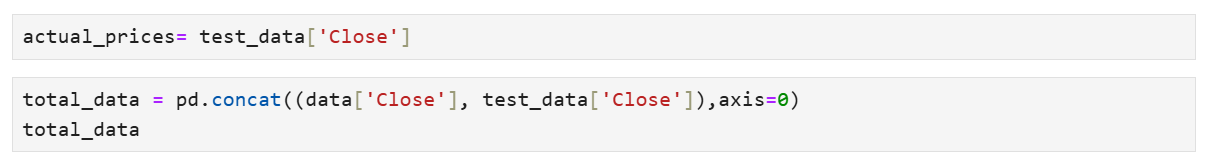
During training, the model learns to minimize the loss function by adjusting its weights based on the input data.

## 8. Testing Data Preparation

After training the model on historical data up until January 1st, 2023, we need to evaluate its performance on unseen test data collected from January 1st onwards:

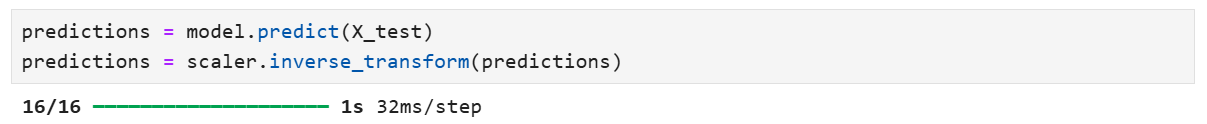


We then concatenate both training and test datasets to prepare inputs for predictions:

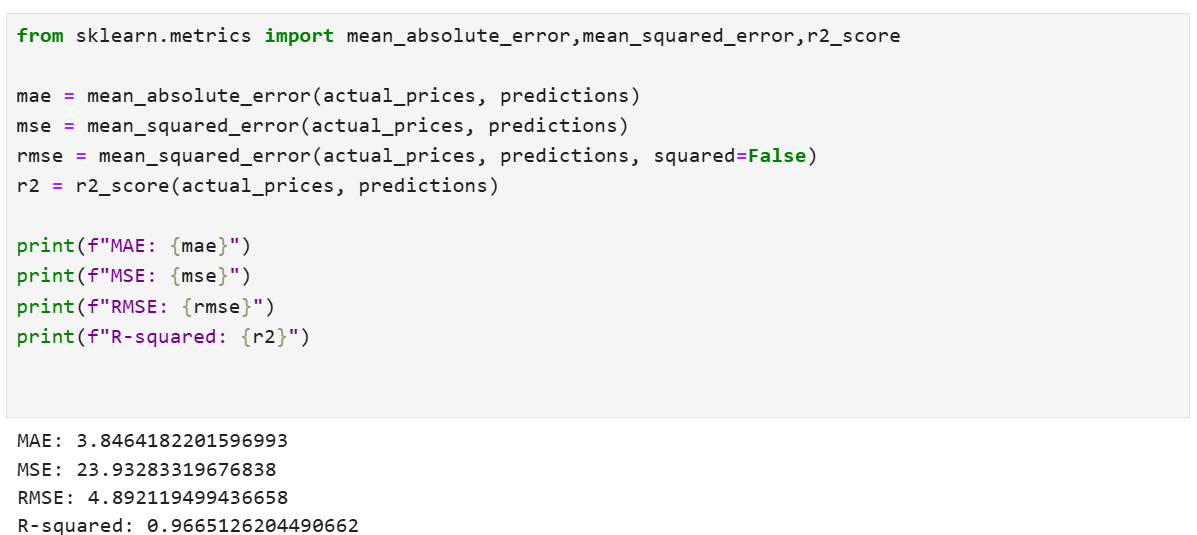


## 9. Model Evaluation

Once we have prepared our test dataset inputs (X\_test), we can use our trained model to make predictions:



To evaluate the model's performance quantitatively, we compute several metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared value:



The results indicate:

* **MAE**: Represents the average error in predictions.
* **MSE**: Provides an average squared error.
* **RMSE**: Offers error in same units as output variable.
* **R-squared**: Indicates how well data fits a statistical model.

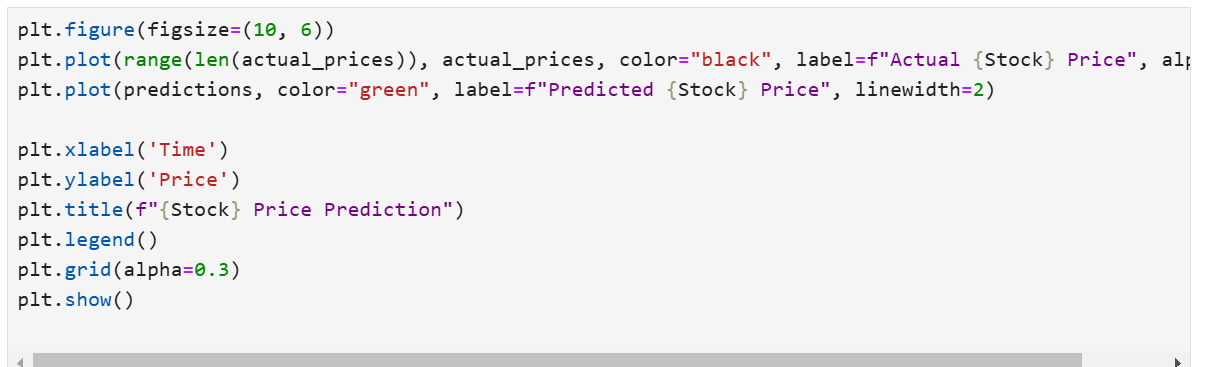
For instance:

* MAE: Approximately **3.85**
* MSE: Approximately **23.93**
* RMSE: Approximately **4.89**
* R-squared: Approximately **0.97**

These metrics suggest that our model performs well in predicting stock prices with a high degree of accuracy.

## 10. Visualization of Predictions vs Actual Prices

Finally, it is beneficial to visualize how closely our predicted values align with actual stock prices:



This plot provides a clear visual representation of how well our model's predictions match actual market movements over time.

## Conclusion

In conclusion, this comprehensive analysis demonstrates how LSTM networks can be effectively utilized for stock price prediction tasks. By carefully acquiring historical data and preprocessing it appropriately—normalizing values and creating input-output sequences—we can train robust models capable of making accurate predictions based on past performance.The results indicate that our LSTM model achieves impressive accuracy metrics such as low MAE and RMSE values alongside a high R-squared score. This suggests that such models can be valuable tools for investors seeking insights into future market movements based on historical trends.Future work could involve experimenting with more complex architectures or additional features such as technical indicators or sentiment analysis from news articles to enhance predictive power further. Additionally, hyperparameter tuning could be applied systematically to optimize model performance even more effectively.Overall, this analysis underscores the potential of machine learning techniques like LSTMs in financial forecasting domains while highlighting key steps involved in building predictive models from scratch using Python libraries such as TensorFlow/Keras and scikit-learn.

Name : Baishnab Charan Behera

University : Kalinga Institute of Industial Technology(KIIT)

Major : Computer Science & Engineering