**📈 Stock Market Analysis & Prediction Using LSTM**

**Introduction**

In this project, I explored the use of deep learning to predict stock prices, focusing specifically on Apple Inc. (AAPL). Stock price prediction is a challenging task due to the market's highly volatile and non-linear nature. For this, I used an LSTM (Long Short-Term Memory) model, which is particularly good at capturing patterns in sequential data like time series.

**Data Collection**

I pulled historical stock data from Yahoo Finance using the yfinance library. The dataset includes daily stock prices for Apple from January 1, 2012, to December 17, 2021. The features included were: Open, High, Low, Close, Adj Close, and Volume.

**Data Preprocessing**

I focused on the 'Close' price for prediction. The data was scaled using MinMaxScaler to normalize values between 0 and 1, which helps the model train more efficiently. I also created sequences of 60 days of past closing prices to predict the 61st day’s closing price.

**Model Architecture**

I built a sequential LSTM model using TensorFlow/Keras. Here's a breakdown:

* 2 LSTM layers with 50 units each.
* 1 Dense output layer with 1 unit.
* Mean Squared Error (MSE) as the loss function.
* Adam optimizer for training.

**Training**

The model was trained for 10 epochs on 80% of the data, with the remaining 20% used for validation/testing. The training loss decreased steadily, indicating that the model was learning.

**Predictions & Results**

After training, I tested the model on unseen data. The model’s predictions closely followed the actual stock price trends, although it struggled with sudden spikes and dips — a common limitation in time series forecasting.

**Visualization**

I plotted the actual vs. predicted stock prices, and the results showed that the model could reasonably forecast future stock prices based on historical data. While not perfect, the LSTM captured the general trend quite well.

**Conclusion**

This project showed that LSTMs can be a powerful tool for stock price prediction when used correctly. However, due to the inherently unpredictable nature of the stock market, no model can achieve perfect accuracy. Further improvements could include:

* Incorporating more features (like technical indicators or news sentiment).
* Trying different model architectures.
* Tuning hyperparameters for better performance.