# Stock Price Prediction Using AI

#### **Approach**

The primary goal of this project was to predict stock prices using a Long Short-Term Memory (LSTM) neural network. Below is a detailed summary of the approach, model selection rationale, evaluation metrics, and the obtained predictions.

1. Reviewing Existing Approaches:
   * I began by examining similar codebases and approaches taken by others for stock price prediction. This step was crucial for understanding the variety of methods available and their relative effectiveness.
2. Importing Dependencies:
   * All the necessary libraries and dependencies were imported at the start. This included common libraries such as NumPy, Pandas, Matplotlib, and TensorFlow/Keras for building the LSTM model.
3. Data Retrieval:
   * Historical stock data was retrieved using Yahoo Finance API, which is known for its reliability and ease of use. The yfinance library was utilized to access this API efficiently.
4. Loading and Preprocessing the Dataset:
   * The provided dataset was loaded and Cleaned. For cleaning I made sure to get the basics right
     + Removed Duplicated
     + Singular date format
     + Fill nulls
     + Filter out outliers
5. Feature Engineering:
   * I looked up papers and research and based on literature and articles about effective features for LSTM models in stock price prediction, I picked the following features and got their formulas with the help of google and chatGPT:
     + Percentage Change: Measures the rate of change in stock prices.
     + Lag: Previous stock prices used to predict future values.
     + Exponential Moving Average (EMA): A weighted moving average that gives more significance to recent prices.
     + Relative Strength Index (RSI): A momentum oscillator that measures the speed and change of price movements.
     + Daily Return and Cumulative Return: Metrics indicating daily price changes and overall performance over time.

#### **Model Selection**

1. Why LSTM?
   * I had narrowed down to 2 choices, LSTM and ARIMA both of which work well with time series data. However, seeing as how widely used LSTM was for stock price predictions, based on research comparing varieties models. And the fact that capability of learning long-term dependencies, making them highly suitable for time-series forecasting tasks such as stock price prediction. LSTMs can capture the temporal dependencies and trends in stock price data better than traditional machine learning models.
2. Model Architecture:
   * The LSTM model was designed with the following architecture:
     + Input Layer: Takes the engineered features.
     + LSTM Layers: Multiple LSTM layers to capture temporal dependencies.
     + Dense Layers: Fully connected layers to output the final prediction.
   * The model was compiled with an appropriate loss function and optimizer, specifically using Mean Squared Error (MSE) as the loss function and Adam optimizer for efficient training.

#### **Evaluation Metrics**

1. Mean Squared Error (MSE):
   * MSE is the average of the squared differences between the predicted and actual values. It penalizes larger errors more significantly, making it a suitable metric for regression tasks.
2. Root Mean Squared Error (RMSE):
   * RMSE is the square root of MSE, providing an error metric in the same units as the predicted values. It is more interpretable and gives a clear idea of the prediction accuracy in the original scale of stock prices.

#### Obtained Predictions

The LSTM model was trained on the historical stock data with the engineered features. After training, the model was evaluated on the test set using **MSE** and **RMSE** metrics. The following predictions were obtained:

* Test Set Performance:
  + MSE: 12.9
  + RMSE: 3.5

The code and the model runs completely fine, but the results are not what I was hoping for, there is still a lot of room for experimentation, finetuning and improvement. But I didn’t have enough time for that, as the deadline was close and the assignment was assigned during working days.