<u>Predicting Stock Prices Using Machine Learning</u>

Goal: *This document consists of some basic information about analyzing the stock market, as well as the steps I took to develop this software.*

The stock market is known for being volatile and dynamic. Accurate stock price prediction is extremely challenging because of multiple factors, such as politics, global economic conditions, unexpected events, a company's financial performance, and so on.

For this reason, I aim to make use of the Neptune API in Python, to predict stock prices using Machine Learning.

Stock Analysis:

1. Fundamental Analysis:
   - Evaluates a company's stock by examining its intrinsic value, including but not limited to tangible assets, financial statements, management effectiveness, strategic initiatives, and consumer behaviors.
   - Focusses more on long term and less on short term effects.
2. Technical Analysis:
   - Analyzes measurable data from stock market activities, such as stock prices, historical returns, and volume of historical trades, i.e., quantitative information that could identify trading signals and capture the movement patterns of the stock market.
   - Focusses more on short term, hence it is more heavily influenced by news.

For this model, I will be making use of technical analysis. And for now, will be making use solely of Simple and Exponential Moving Averages.

Dataset Analysis:

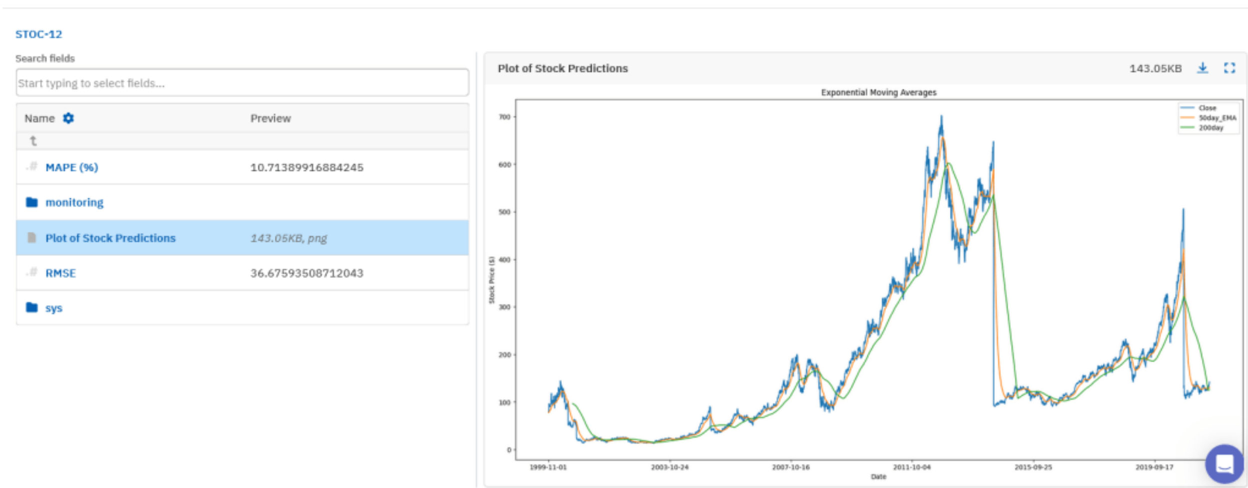
- I will be implementing the Alpha Vantage Stock API to load the data.
- The data being used will be Apple's stock information from the last 21 years (1999-11-01 to 2021-07-09)
- For model training, we'll use the oldest 80% of the data, and save the most recent 20% as the hold-out testing set.

Helper Functions:

- I will be considering two functions: RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error)
- $$MAPE = \frac{1}{N} * \sum_{t=1}^{N} \left| \frac{At - Ft}{At} \right|$$

- $$RMSE = \sqrt{\frac{1}{N} * \sum_{t=1}^{N}(At - Ft)^2}$$
- RMSE gives the differences between predicted and true values, whereas MAPE (%) measures this difference relative to the true values.
- I have also created another helper function that splits the stock prices data into training sequence X and the next output value Y.

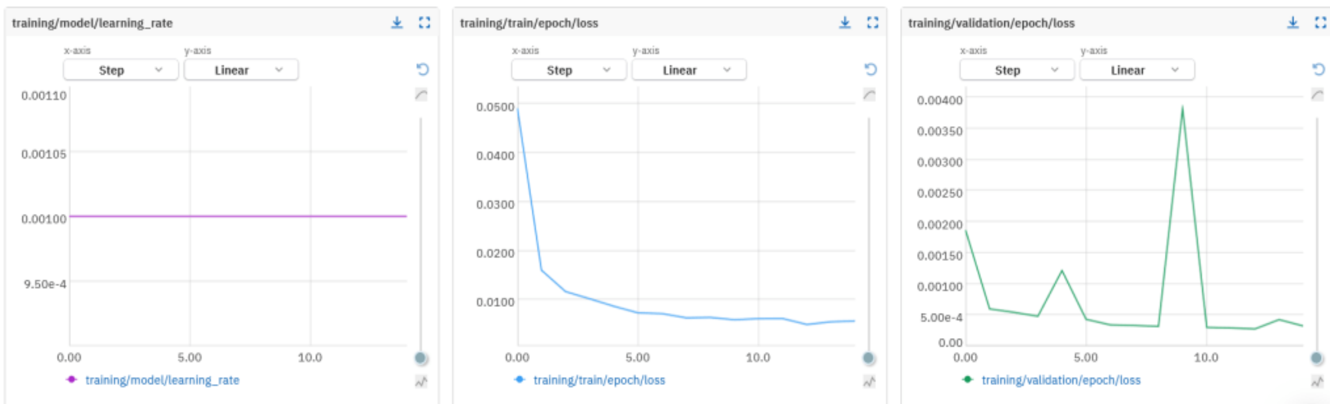
Examining the performance metrics tracked in Neptune, we have RMSE = 36.68, and MAPE = 10.71%, which is an improvement from SMA's 43.77 and 12.53% for RMSE and MAPE, respectively. The trend chart generated from this EMA model also implies that it outperforms the SMA:



Comparison of SMA and EMA side-by-side in Neptune:

For this project, I will be using LSTMs (Long Short-term Memory) for the time-series data. It can capture historical trend patterns and predict future values with high accuracy.

After setting up the Neptune and Keras integration, we get the following **training process** on Neptune:



The below trend chart shows a near-perfect overlay of the predicted and actual closing price for our testing set.