# Stock Market Modeling using Combined Gaussian Process and LSTM Models

Duncan Mazza    Shashank Swaminathan

## Introduction

The goal of our project is to create a stock market modeling API that would utilize multiple different types of machine learning models to learn patterns in stock prices and make predictions about future stock prices. With the goal of an API in mind, we have integrated automatic documentation building and our code base consists primarily of object-oriented code. To that end, we integrated Yahoo's API a class that serves as a common ground for the disparate models to acquire data.

## LSTM Recurrent Neural Network

The first of our machine learning models is a long-term short-term memory (LSTM) network, which is a specific type of recurrent neural networks. Networks classified as recurrent neural networks allow information to persist between passes of data through the network; this is similar to how humans use memory to aid their immediate understanding of the world. As the name suggests, LSTMs have the capability to encode information about previous states - both recent and not - to inform their output (Figure 1). LSTMs are designed for sequential data, making them a logical choice for our application.

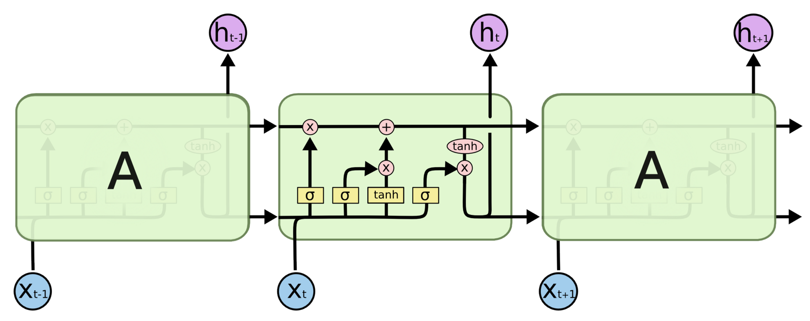


Figure 1: A standard visualization of a LSTM and the activations inside of its modules. $x_t$ is an element of a sequence, and $h_t$ (the hidden layer) is the output/prediction.

LSTMs can process sequences of arbitrary length with any pre-defined number of features, and output a sequence with a different and arbitrary number of features. To leverage these attributes, we pass in not only sequences containing the stock of the company we wish to predict, but also the stock prices of other companies. While stock prices are inherently unpredictable, we built our model on the notion that there there may exist learn-able relationships between companies' stocks.

A model trained for 100 epochs to predict Apple stock with the added features of Microsoft, Google, and Motorola stock produced the output shown in Figure 2. The results in are generated in training mode; in evaluation mode[a], the model predicts mostly homogeneous trends between input sequences. Further work is required to mitigate this issue and produce predictions that are more robust.

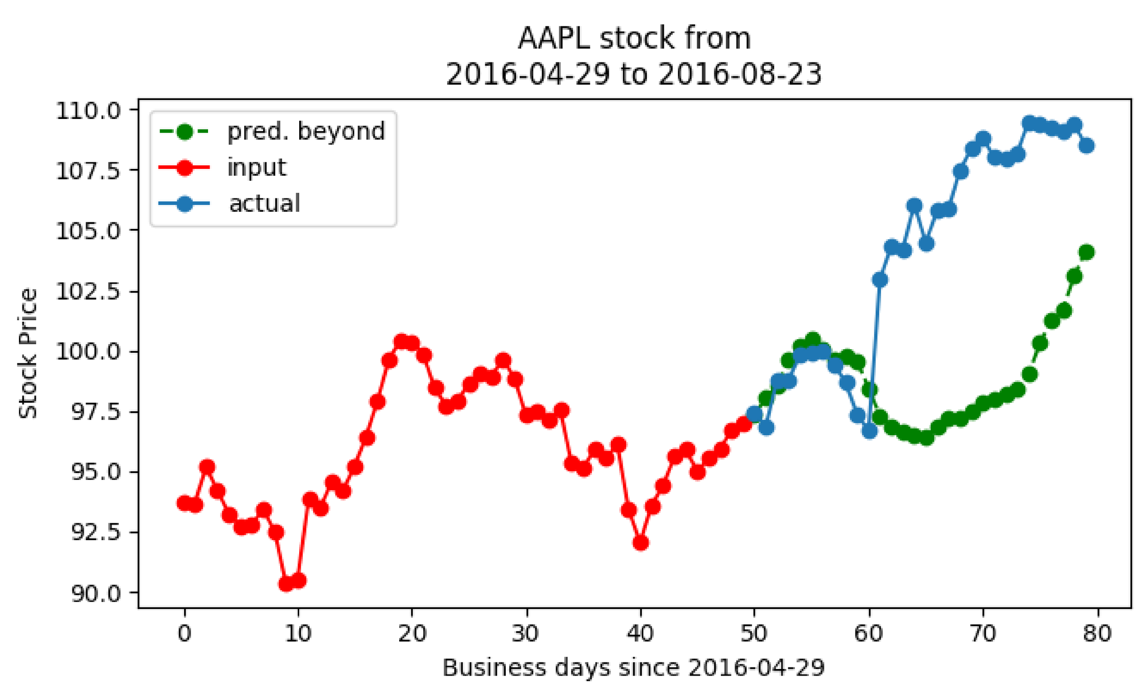


Figure 2: The model is trained on the percent changes of the company stock, (left), and reconstructs the stock prices. Green values are values predicted beyond the training data (red), meaning that the model has no knowledge of the actual stock price (blue).

[a]Evaluation mode prevents the dropout from occurring; a dropout probability of 0.3 was used.

## GPM

The second model, as seen in Figure 4, used is a Gaussian Process (GP) model, which uses a Bayesian regression approach. This method models the data by considering an infinite function space. It gives a probability to every function within this infinite space, corresponding to how likely the function is to model the data. Since representing an infinite function space is computationally intractable, we use Gaussian Processes to represent the distribution of functions over an infinite domain. We utilized 3 GPs in our model, each with a covariance function chosen to cover specific behavior we wanted from our model. The chosen covariance kernels are listed below:

1. Exponentiated Quadratic Kernel for long-term steady growth.
2. Rational Kernel for short to medium length fluctuations.
3. Periodic Kernel for medium length periodic behavior.
4. White Noise and Matern32 kernels as noise modelers.

Each of these kernels can be described in further detail by hyper-parameter distributions. These are plotted in Figure 1.

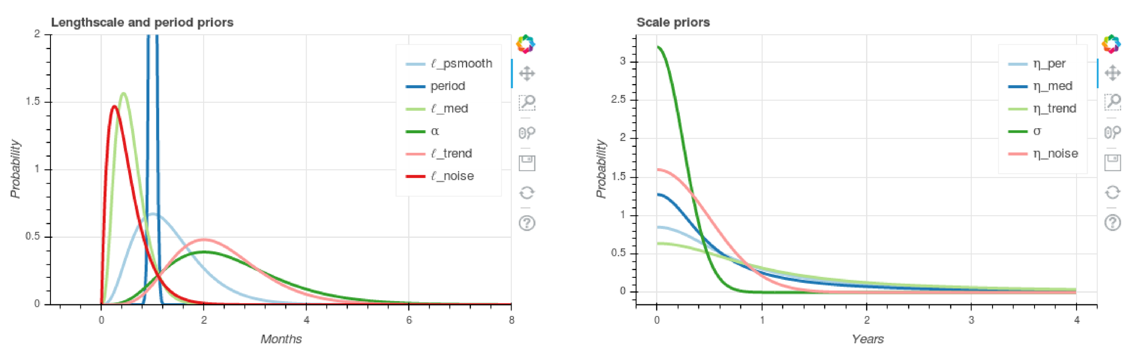

Figure 3: Plot of the model's hyper-parameters.

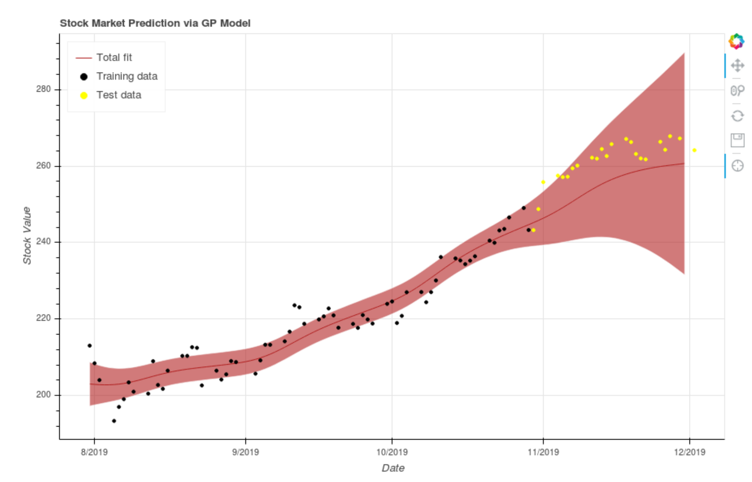

Figure 4: Result of the Gaussian Process Model.

## Model Combination

We used a simple weighted average approach for model combination. An example of the combined model is shown in Figure 5. This was all integrated into an interactive API shown in Figure 6.

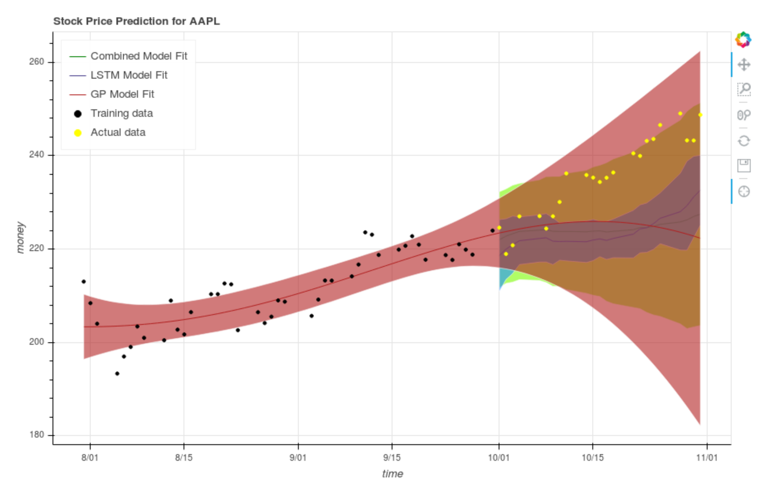

Figure 5: Output of the combined GP+LSTM model.

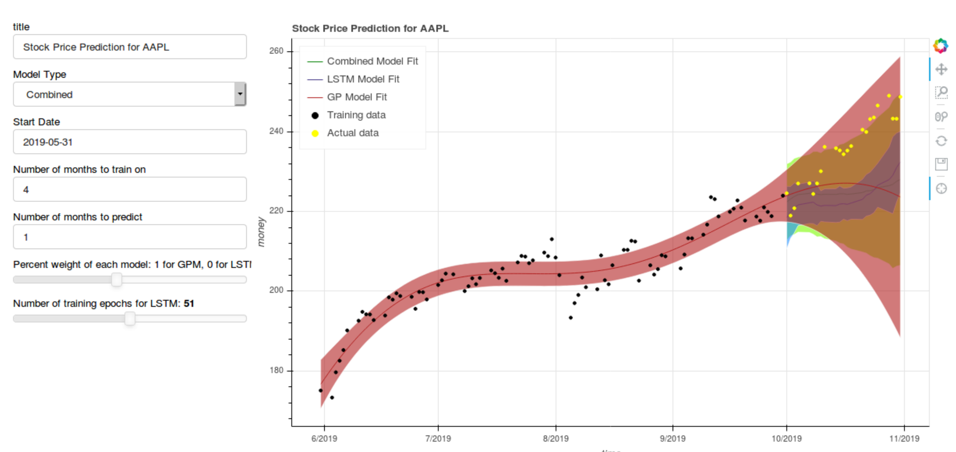

Figure 6: Screenshot of the interactive stock modeling API.