Statistical Machine Learning Final

Daniel J. Park

March 27, 2020

1 Introduction

For this final assignment, I analyze two different time series datasets with various tools for modeling series data: the Facebook Prophet, an LSTM Neural Network, and an ARIMA model.

2 COVID-19 Death Toll

COVID-19 is a respiratory disease caused by a novel coronavirus and is now considered a pandemic by the World Health Organization. I downloaded data on the cumulative number of deaths from COVID-19 globally from January 22, 2020 to March 23, 2020 (Figure 1). Because there are only 62 days' worth of observations, a machine learning algorithm such as an LSTM Neural Network would not be appropriate. With such few data points, the algorithm might perfectly memorize and overfit on the training data. It would generalize poorly to predict cumulative deaths at future time steps.

Instead, I decided to implement the Prophet model developed by Facebook's Core Data Science team. The Prophet uses an additive model of the form:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

where $g(t)$ models non-periodic changes in the value of the time series, $s(t)$ models seasonality (e.g. daily, weekly, yearly), and $h(t)$ captures the effects of holidays. We assume the errors ϵ_t follow a normal distribution.[1] According to

[1] https://peerj.com/preprints/3190v2/

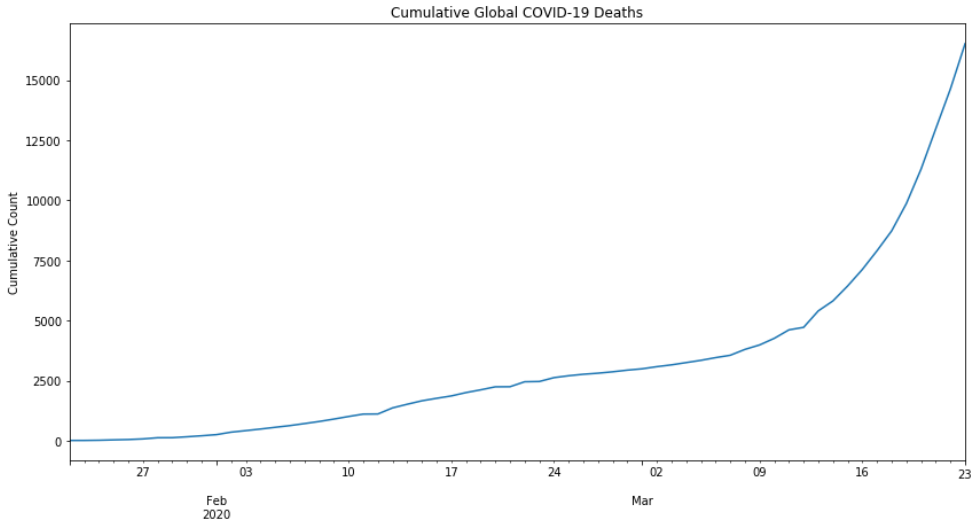

Figure 1: Global Cumulative Deaths from COVID-19

the developers, the Prophet "works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well."[2]

I trained a Prophet model using all 62 days worth of data, then made a forecast/prediction for the entire time range plus seven more days into the future. The results are shown in Figure 2.

The true observations seem to follow an exponential growth trend; however, the Prophet does not incorporate an exponential assumption into the model. Rather, the predictions look like a piecewise linear response function with two linear components: the first from January 22 to March 8 and the second from March 8 to March 30. In addition to the general linear trends, we can see that the model is including weekly seasonality, which is why there are local 'bumps' occurring every seven days, even though the observations don't show a weekly pattern.

Prophet performs quite well for dates up until Mar. 10 (except for trying to predict weekly seasonality where there is none), after which the extrapolated linear growth forecast will severely underestimate the true global cumulative deaths from COVID-19. In terms of accuracy metrics, the correlation between the actual and predicted deaths is $\rho = 0.979$ while the Mean Absolute Percentage Error is 1.518, which means that the model is making large prediction errors relative to the

[2] https://facebook.github.io/prophet/

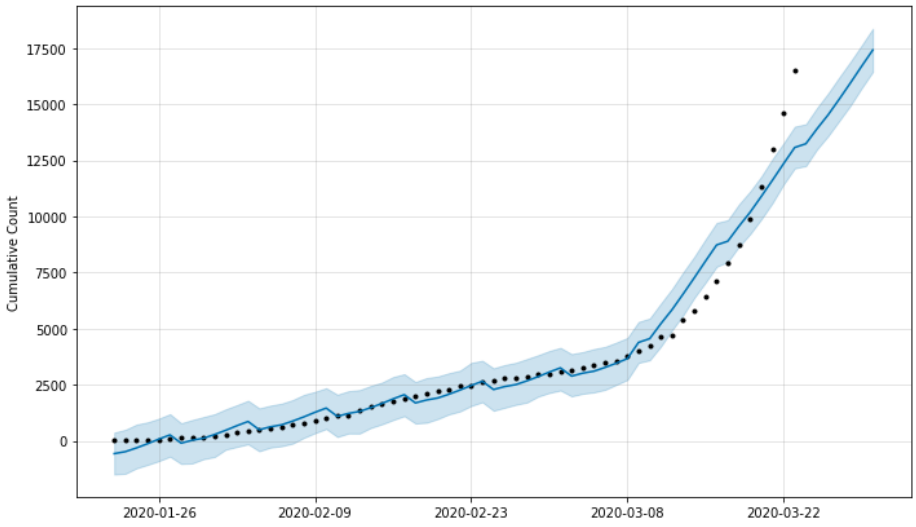

Figure 2: Actual and Forecasted Deaths from COVID-19

true observations. I do not recommend the Prophet model for predicting global deaths from COVID-19 since we do not have a long time series with multiple seasons' worth of data. Another model that incorporates an exponential growth assumption and possibly includes other parameters would be more appropriate.

3 Historical Soybean Prices

Soybeans are one of the world's most important foodstuffs and a key agricultural export for the United States. In the past year, soybean exports were much discussed in the trade negotiations between the U.S. and China. Given the importance of this commodity, I wanted to analyze historical soybean prices to predict future price movements. I downloaded a dataset containing the daily closing price (in USD per bushel) of soybeans for every trading day from December 5, 1968 to March 24, 2020.[3] I then used the following three methods for modelling time series for prediction.

[3] https://www.macrotrends.net/2531/soybean-prices-historical-chart-data

3.1 LSTM Neural Network

A Long-Short Term Memory Neural Network is a type of Recurrent Neural Network that keeps some information about training examples seen in previous time steps in a cell state and updates this cell state as it sees more examples. A key difference between an LSTM and a regular RNN is that the LSTM is able to learn long-term dependencies between inputs far apart in the series and not just between consecutive terms.

The data is shown in Figure 3, where the dotted lines mark the dates I used to split the dataset into training (prices up until December 31, 2008), validation (prices from January 1, 2008 to December 31, 2011), and test (prices after January 1, 2011) sets.

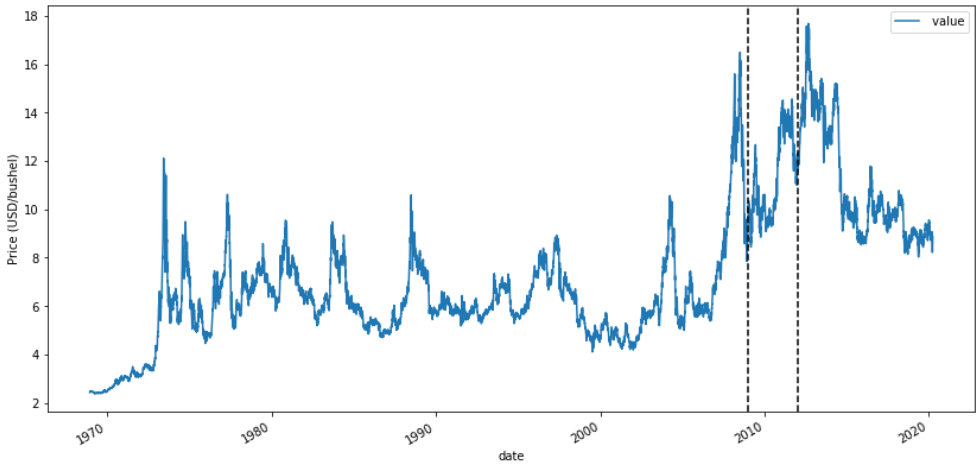

Figure 3: Historical Soybean Prices

After standardizing the prices, I had to transform the training data into the appropiate formate for an LSTM: a series of shorter sequences or sliding windows of 60 days length each. In my implementation, I trained an LSTM with 30 hidden units using the mean square error loss function and the Adam optimizer. The training and validation loss converge at Epoch 10, reaching 0.04 and 0.32 respectively (Figure 4.

With the implementation of an LSTM available here[4], we can set a `future` parameter that controls for how many time steps ahead the model predicts soybean

[4] https://romanorac.github.io/machine/learning/2019/09/27/time-series-prediction-with-lstm.html

price. For example, if we let $future = 5$, the model outputs at time t the expected price at time $t + 5$. In this case, I set $future = 0$ to see the model's prediction for each day on that day.

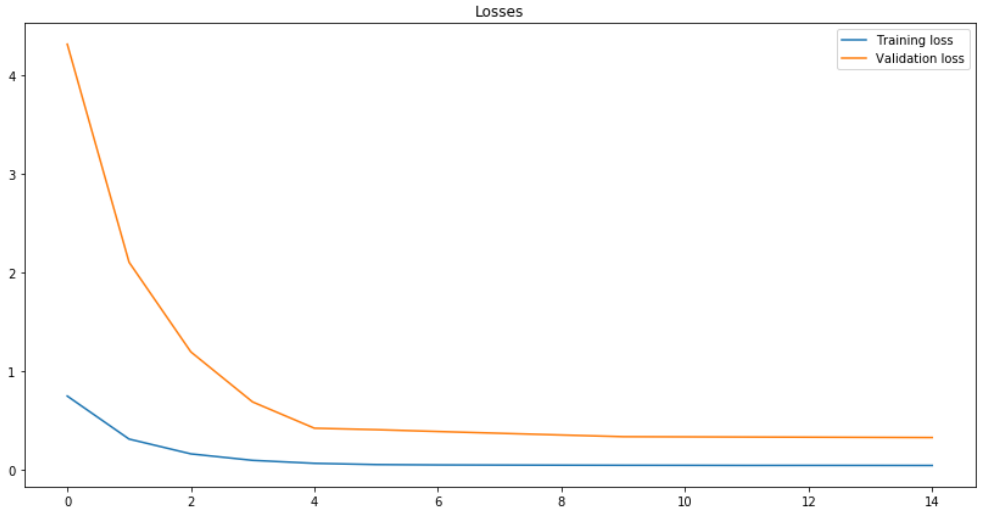

Figure 4: Standardized losses from training the LSTM

From Figure 5, we see that for the first 550 days in the test dataset, the model consistently underestimates the price and captures neither the average nor the volatility of the observations. After the 550th day, the predicted series more closely matches the actual, although the prediction continues to lag behind the true price movements.

3.2 Facebook Prophet

For the Prophet model, I concatenated the training and validation sets from the previous section and trained using the series of prices up until December 31, 2011. The Prophet captures the general trend in the training data up until about 2003, although it tends to underestimate peak prices (Figure 6). In red we see the true prices from the test dataset and the forecasted series in blue (with a 95% confidence interval region shaded). Not only does the true price fall outside of the confidence interval for most days after January 1, 2011, but the curve of the forecast trends upward while the actual prices are generally decreasing. For comparison, I note that the MAPE is 0.592 and the correlation is -0.803.

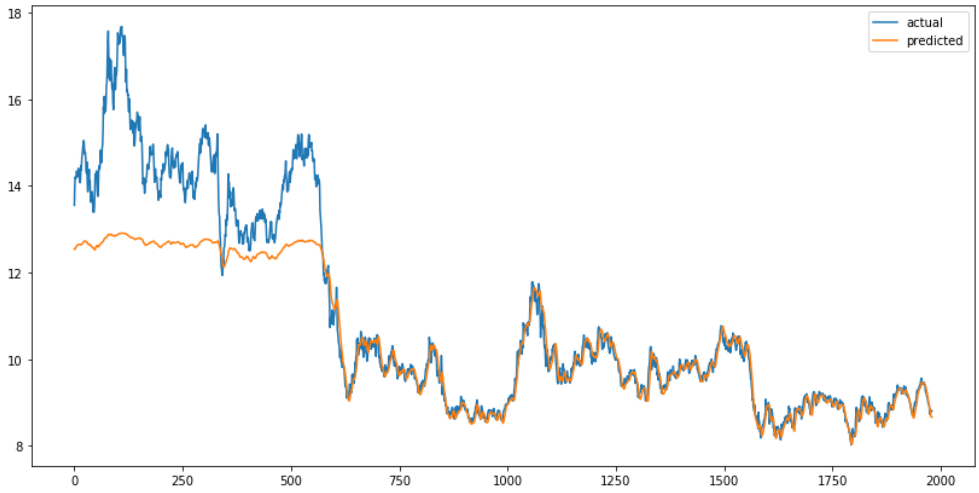

Figure 5: Actual vs. LSTM Predicted Soybean Prices for Test Data

3.3 ARIMA/SARIMAX Model

An Auto-Regressive Integrated Moving Average model assumes y_t from a stationary series is a linear combination of previous time steps and lagged forecast errors of the form:

$$y_t = \alpha + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \cdots + \phi_q \epsilon_{t-q} \tag{2}$$

An automated step-wise procedure found the best model according to the AIC criterion and the optimal parameters to be a SARIMAX($p = 2, = d = 0, q = 0$) model. All four terms in the model were significant with p-values close to zero. We can check how good the fit is with diagnostic plots (Figure 7). The residuals look like they have a mean of approximately zero. Other than that large spike, the residual plot displays uniform variance. The Normal QQ-plot shows some skew at the tail ends of the sample quantiles, so we should proceed on the error normality assumption with caution. Lastly, the Correlogram or ACF plot shows that the residuals are not autocorrelated, so the predictors already in the model are sufficient.

Figure 8 shows the forecast for soybean prices in purple with a 95% confidence interval shaded region. Compared to the Prophet forecast, the shape of the SARIMAX forecast follows the downward trend after January 1, 2011, and the confidence interval captures most of the actual prices. In terms of metrics, Ta-

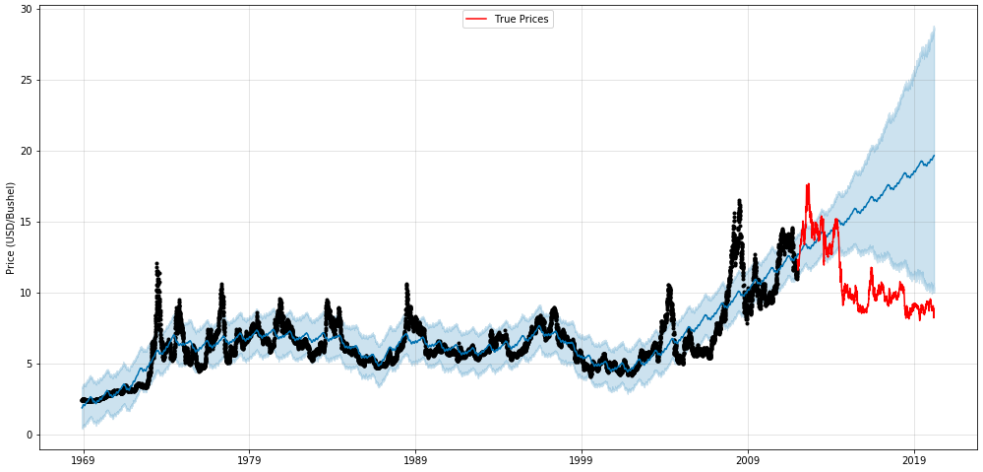

Figure 6: Forecasted Daily Closing Soybean Prices

ble 1 shows that SARIMAX represents a clear improvement toward greater test prediction accuracy. The MAPE and RMSE are more than 50% smaller, and the magnitude of the correlation coefficient suggests a closer fit.

	Prophet	SARIMAX
MAPE	0.592	0.238
RMSE	6.6135	3.027
Corr (ρ)	-0.803	0.852

Table 1: Accuracy Metrics

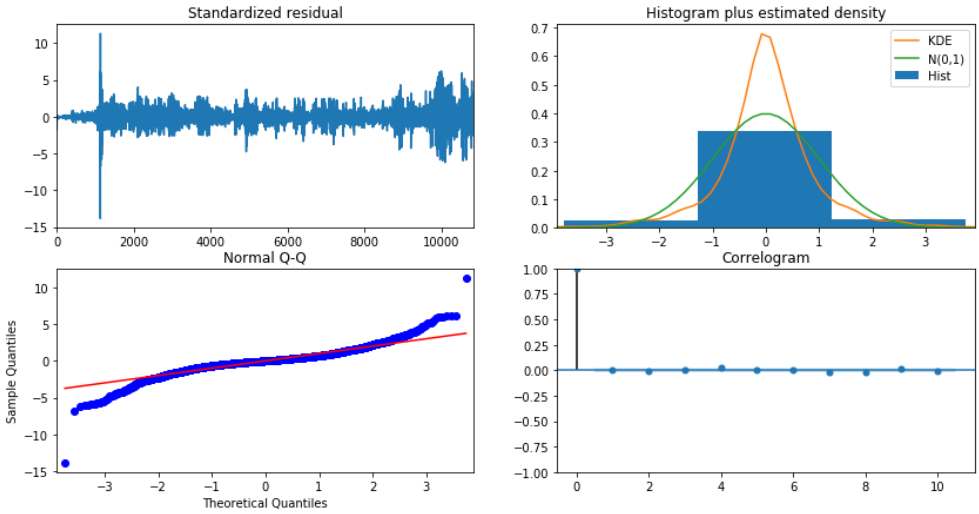

Figure 7: Diagnostic plots from the SARIMAX model

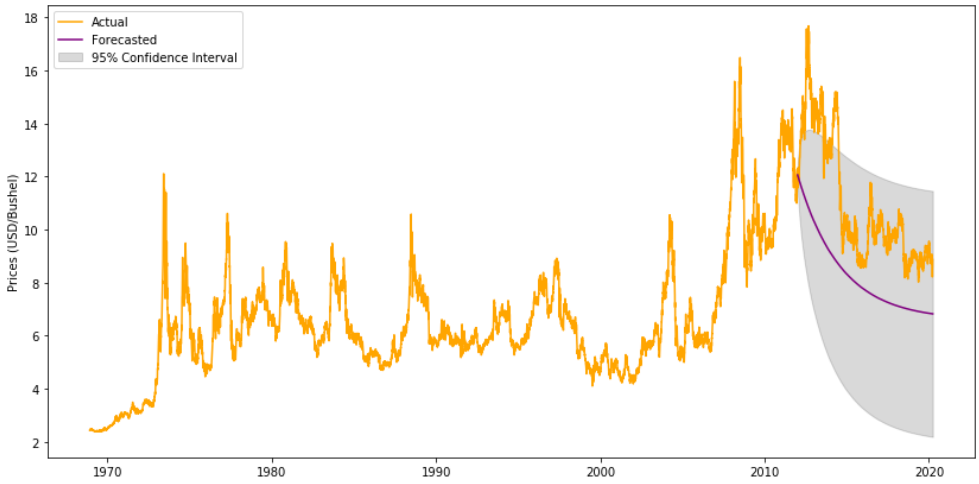

Figure 8: Actual vs. SARIMAX Predicted Soybean Prices