Forecasting Take Home Lisa VanderVoort

November 2020

Analysis of the Data

This is an analysis of regular motor gasoline retail prices data from January 1992 to January 2018. This is a dataset containing monthly average gasoline prices. There were no null values in the data and there were 26 full years of data and one datapoint from 2018. In order to provide an unbiased analysis and model, I chose to hold out data from 2016-2018 as my test data. Therefore, the data for these years is not included in my analysis, so as not to influence the building of the model.

To start the analysis, I examined the trend in gas price over time which can be seen in Figure 1. There were a number of interesting observations. First, the trend is mostly increasing, as you can see the price of gas has risen over time. However, in recent years, it appears the trend is on the decline. Additionally, there's also increased variance over time which can be seen in the more dramatic spikes in the later part of the data. Second, there are some outliers with dramatically high and low prices, particularly as time goes on. For example, there is a huge spike and subsequent drop in 2008/2009 that likely connects to the global financial crisis.

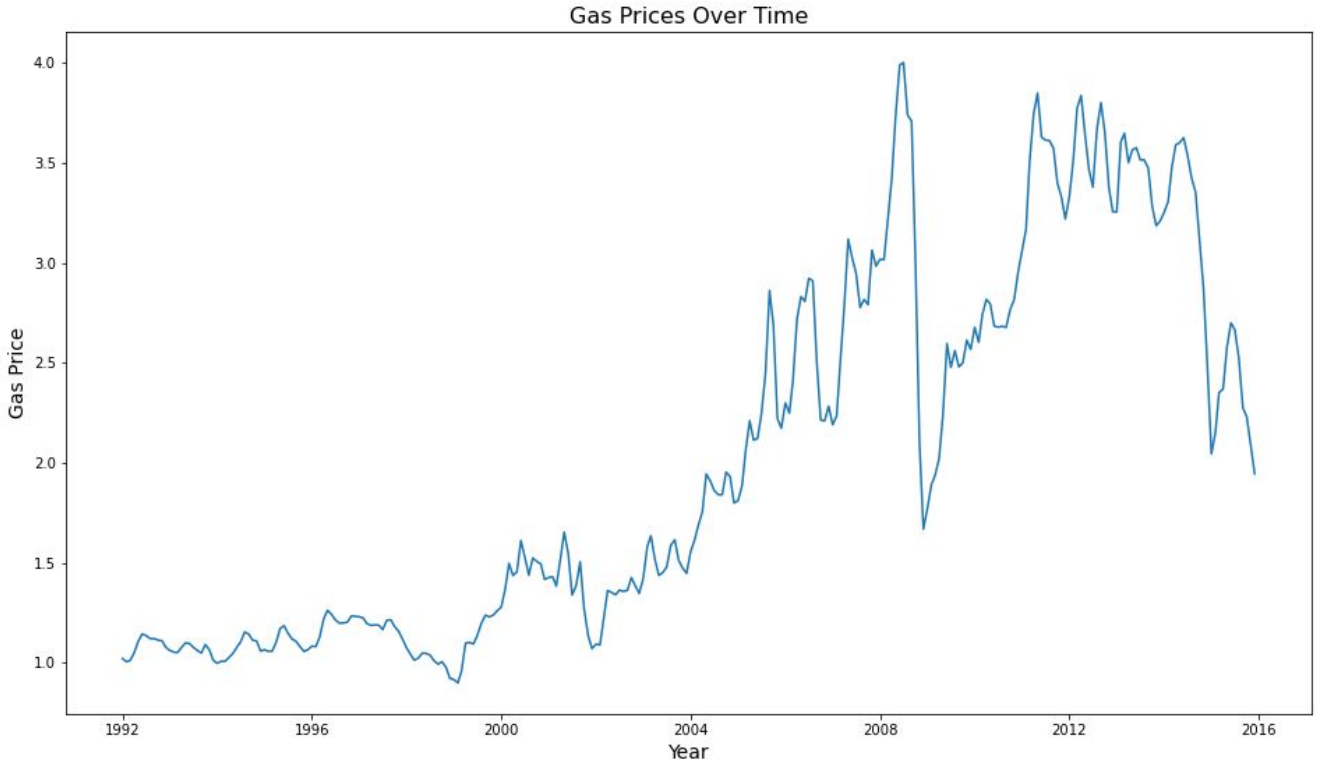

Figure 1: Gas prices over time for train data (January 1992-December 2016)

Next, I examined the seasonality of the data. In order to do this, I looked at how gas price changes by month across all years which can be seen in Figure 2 below. There is a lot going on in this graph and there isn't a super clear pattern across the years. However, it does appear that spikes in the prices tend to happen between May and September and dips tend to happen in December and January.

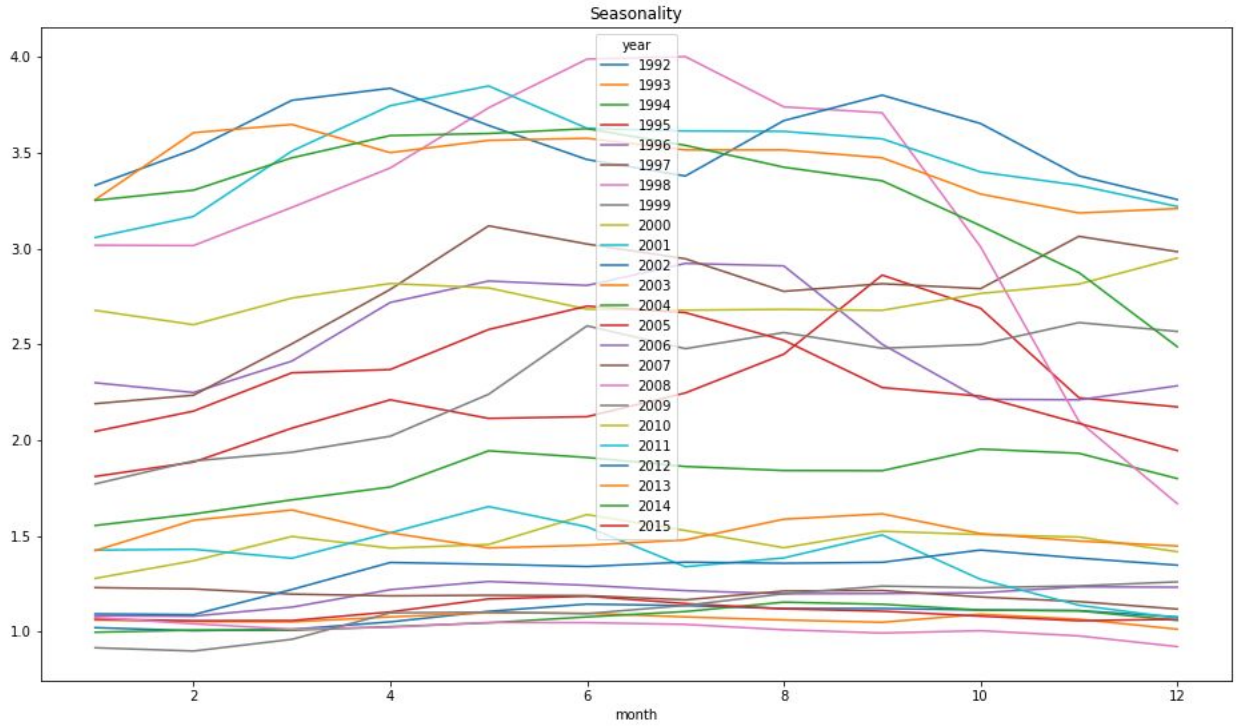

Figure 2: Gas prices as a function of month across all years in train data (January 1992-December 2016)

Next, I wanted to take a look at how the prices changed as a result of the financial crisis. After a bit of research, the Great Recession took place from December 2007 to June 2009. Figure 3, which is shown below, is the same as Figure 1, but with the addition of the start and end dates of the Great Recession. This information will become important in the modeling section below, as I don't want the model to learn these trends to repeat over time.

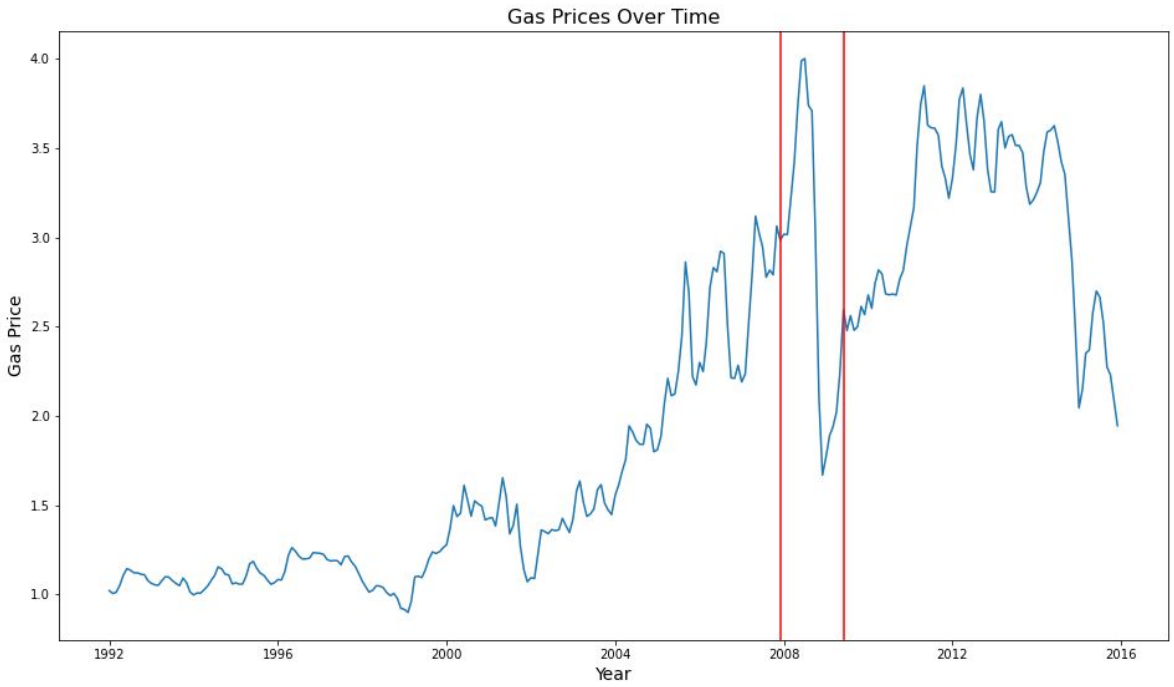

Figure 3: Start and end of the Great Recession shown above in red

Modeling Approach

Going into modeling, there were a few takeaways from the analysis that I wanted to pay careful attention to. First, the model is mostly increasing over time and beginning to decrease towards the end, so I need to be mindful of the trend component plot in Prophet. Second, the model shows increased variance over time, so I wanted to pay attention to the hyperparameters I can tune that will increase flexibility of the model. Third, there was a dramatic increase and decrease in prices during the financial crisis that I don't want the model to learn as a yearly trend. Therefore, I will need to deal with this separately. Finally, this data is monthly average data and I will need to adjust my predictions in Prophet to ensure that I am forecasting one monthly value, instead of daily values (the default in the model).

My approach to modeling is to always start simple, make observations and notes about the model, record error metrics, and then to iterate changing one aspect. In order to organize my process and document my thinking, I created the following table to show the iterations I went through with the model and the decisions I made as a result.

I built each model to forecast monthly predictions 2 years into the future. Additionally, I utilized the cross validation metrics built into Prophet to obtain error metrics about my training dataset, using an initial training period of 730 days (2 years) in the first cutoff, then making predictions every 180 days on a horizon of 365 days.

Figure 4: Table detailing the iterative modeling process

Iteration	Hyperparameter Tuning (with selected value)	Error Metrics	Observations/Notes
Baseline	Use all defaults built into Prophet	MAE: 0.337 RMSE: 0.482	Not a great model. Error metrics aren't too bad. However, in examining the forecast, the overall trend is only increasing, but the model struggles to capture the change in seasonality as the prices become more volatile. The model overfits in the earlier years and underfits in the later years. In examining the component graphs, the yearly seasonality is too wavy which means that the default has too many degrees of freedom.
1	Turn off yearly_seasonality. Add custom with lower fourier order (period=365.25, fourier_order=3, prior_scale=20, mode='additive').	MAE: 0.336 RMSE: 0.483	While there was virtually no change in error metrics, this iteration's yearly component graph was better aligned to the trend seen in most years and was smoother. The model is still doing a really poor job with the volatility in the data. I will keep this seasonality in the models going forward.

2	changepoint_prior_scale=0.1	MAE: 0.340 RMSE: 0.482	Prophet recommends changepoint_prior_scale as one of the most impactful features to tune. I decided to increase this parameter to begin with because the trend needs to be more flexible to account for the volatility from 2005-2013. After iterating through a few values, I found 0.1 to be the best value. While it marginally increased the MAE and marginally dropped the RMSE, I thought it was important to increase this value to allow the forecast to change more at the changepoints. There are some significant change points later in more recent years and I can see this better modeled in the trend component graph as it shows a decline in trend in recent years (which is what I see in the data). I will keep this seasonality in the models going forward.
3	changepoint_range=0.85	MAE: 0.340 RMSE: 0.487	I then decided to tune the changepoint_range for the model. Since I wanted the model to be able to adapt better to the volatile increases and decreases in price, I decided that increasing this value, which increases the proportion of history the trend is allowed to change, would help the model better learn the data. By increasing this value just slightly, the forecast graph was better able to see and adapt to the decline in prices in later years. While this did increase the MAE and RMSE marginally, I believe this is a valuable tradeoff in getting a more accurate prediction. I will keep this seasonality in the models going forward.
4	Additional regressor (details in Jupyter notebook)	MAE: 0.357 RMSE: 0.520	At this point, I wanted to add in an additional regressor to account for some of the rapid increases and decreases that are seen in 2008/2009. I created an additional column with a 1 indicating if a month took place during the Great Recession (December 2007 to June 2009) and 0 if it did not. Using this, again my error metrics went up a bit, but the graphs of the forecast improved which I believe will help prevent overfitting on the test set.

The forecast of the baseline model can be seen in Figure 5 below. While the model unfits a small amount in the very beginning of the data, the model struggles to adapt to the volatile increases and decreases as time goes on. Additionally, the model shows a completely increasing trend, while the later data points increase a decrease in price.

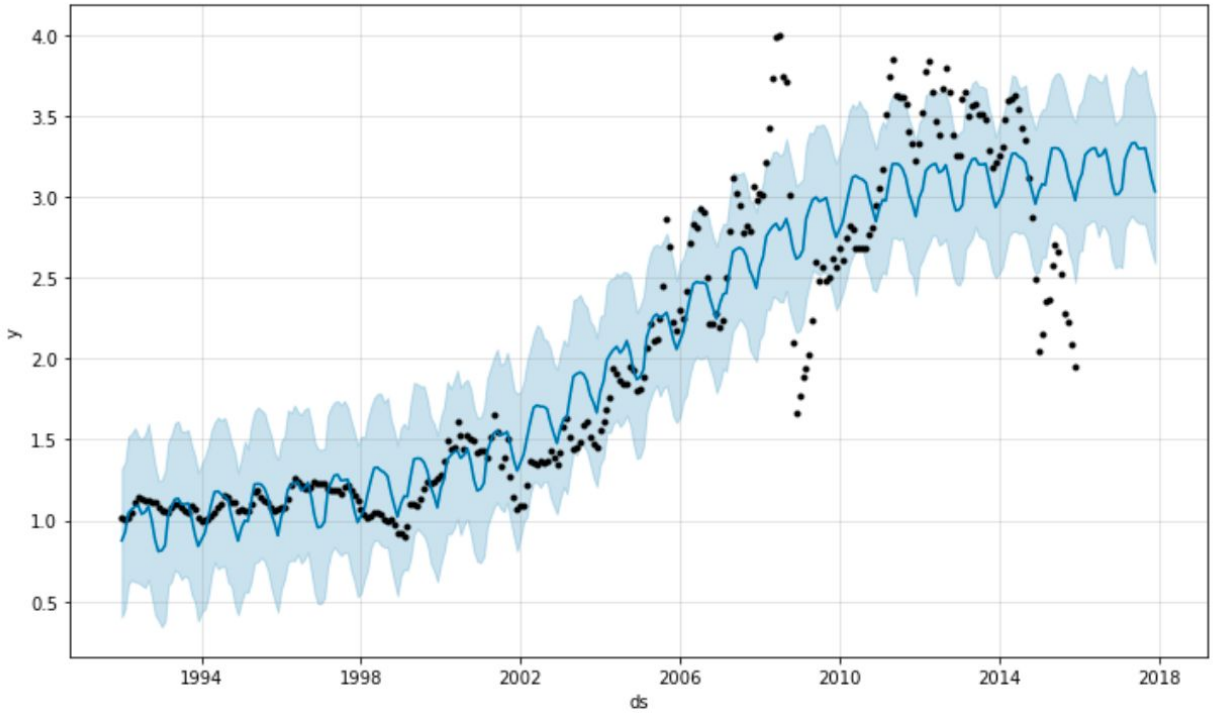

Figure 5: Forecast of baseline model

Based on this information, I decided to go with a model that includes the hyperparameters tuned in Figure 3. This includes adding custom yearly seasonality, increasing the changepoint_prior_scale and changepoint_range, and adding an additional regressor. In the final model, the forecast is better able to adapt and change with the rapid increase and decreases. The model also shows a decline in the trend in later years. The small increase in the error metrics was justified by the improved forecast which I believe will lead to better results on the test dataset.

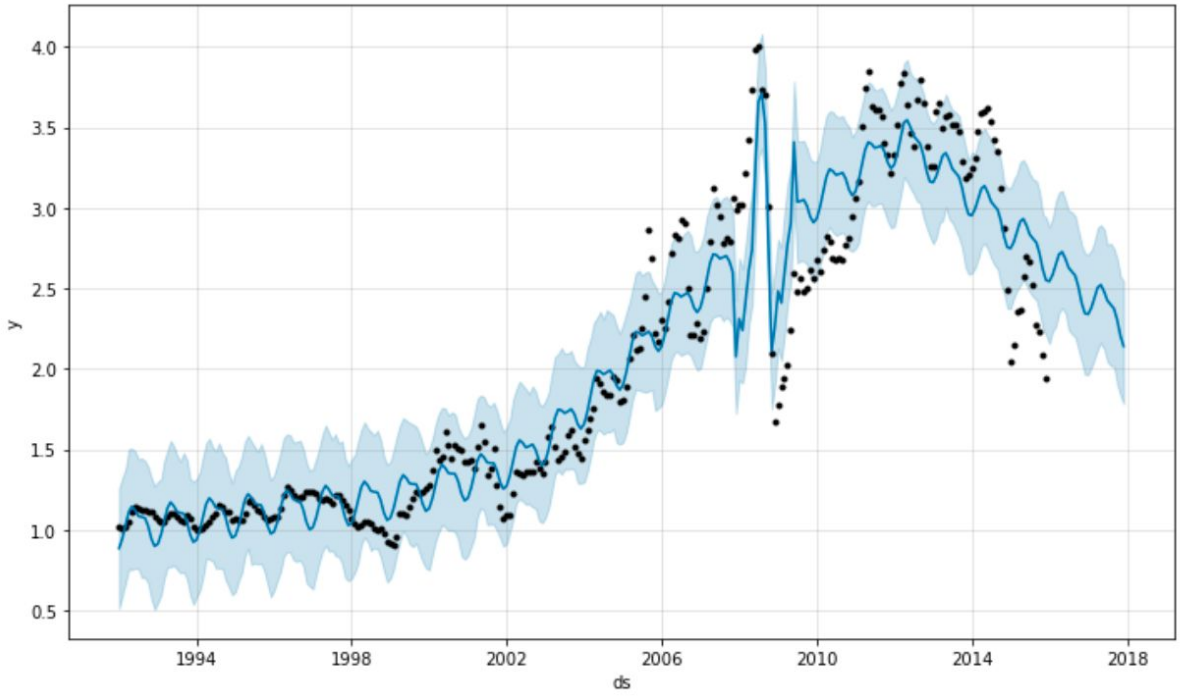

Figure 6: Forecast of final model on training data

When I evaluated this modeling on my training dataset, I obtained an average MAE of 0.357 and RMSE of 0.519 across the forecasted horizon. Evaluating my model on my test dataset, I obtained an average MAE of 0.353 and RMSE of 0.417.

Based on the model's performance, it's possible that my model is underfit. I would recommend continuing to iterate on the model and adding additional complexity. One area of future improvement of this model would be to add in additional regressors, such as information about the stock market, that might better explain the volatility in gas prices. Additionally, the model might be improved by decreasing the forecasting period since gas prices are easier to predict in the short-term rather than the long-term.

Running Analysis and Modeling Code

In order to run the code for the analysis in Jupyter notebook, follow these steps:

1. Build the image:
    a. bash driver.sh build
2. Start Jupter notebook
    a. bash driver.sh jupyter
3. Go to http://localhost:8888/notebooks/notebooks/forecasting_initial_analysis_and_modeling_lisa_vandervoort.ipynb
4. Stop the container when done
    a. bash driver.sh stop

In order to run the code for the modeling Python file, follow these steps:

1. Build the image:
    a. bash driver.sh build
2. Run the scripts/python_model.py
    a. bash driver.sh python-modeling