A Survey of Methodology for Multisite Gas Price Prediction

Haylee Ham

1. Introduction

2. Literature

In an effort to predict gas prices across a large geographic area, it is important to first note the forces at work in the setting of local gas prices. First, the spatial component of how gas prices are set in relation to prices at nearby gas stations should be considered. Various studies have addressed the question of the spatial relationship of gasoline prices. Robert Haining's 1984 paper entitled "Testing a Spatial Interacting-Markets Hypothesis" analyzed gas prices in Sheffield, England. He discussed the differences in price patterns in urban vs non-urban areas. He found that gas price in urban areas (with customers who can collect relative information on gas prices and travel to other retailers at almost no cost to themselves) set costs that are more correlated to neighboring gas stations than in non-urban areas, where markets do not overlap to such a great degree. Haining also found that during time of price cuts (a signal of competition) there is a significant amount of spatial clustering on the lowest end of gas prices. Prices also increased as distance off major roadways increased. On the other hand, Haining found that in times of rising prices, localized competition became weaker and interaction effects were not linked at any price level. This suggests that spatial patterns are more influenced by the relative location of gas stations and other inter-market interactions when prices are falling. This factors may also have an influence on price variance.

A study conducted by Andrew Eckert and Douglas S. West focused their study on price cycles of gasoline. They were specifically interested in attempting to identify collusion and predatory behavior in setting gas prices in Vancouver, Canada. Their study found that large increases in price do not happen when prices fall too close to the wholesale price, but rather on days of the week when demand is lowest. These increases begin in Vancouver and then move east. However, price decreases happen more quickly in the east, where the lowest concentration of population exists. This is consistent with previous findings of Edgeworth cycles, where price decreases begin begin in local areas where most competitors are clustered and then move across the entire market.

Eckert and West also seek to identify which phenomenons would be incorrectly measured or lost entirely depending on the type of data used to study these price cycles. They found that most studies use data where gas prices have been averaged to a weekly level and this granularity is not fine enough to capture the effect that underlying mechanism behind large jumps in in prices, which is that these price increases occur on days with low demand. This finding is especially important since past studies have attributed too much effect to the proximity of the price to wholesale costs. In fact, price restorations have occurred almost always on Tuesdays and Wednesdays, an effect that suggests that demand factors are driving volatility. They also confirmed that spatial patterns exist in the decision of gas prices and that local markets are highly influential, eschewing the theory that commuters smooth the spatial patterns by spreading their demand beyond local markets. This study was unable to reject the hypothesis that price reductions in a cycle are initiated in markets with dense spatial clustering of ARCO/Tempo stations.

A study entitled, “Do Gasoline Prices Respond Asymmetrically to Crude Oil Price Changes?” found a stronger relationship between increases in crude oil and gasoline prices as compared to to decreases in crude oil and gasoline prices. Thus, there is an asymmetric response between crude oil prices and prices of gasoline. Pinkse, et. al. also conducted a study using Vancouver data and focused on the possible existence of spatial clustering of contract-types as opposed to a random spatial configuration of gasoline stations and types.

Thus, with this literature concerning the identified patterns of the setting of gas prices, it can be noted that granularity of data must be considered, that is to say that the data should not be averaged out to larger time periods and should be collected as frequently and consistently as possible. It should also be noted that information about the day of the week, as well information about the brand of gas station, the proximity and density of other gas stations and their respective prices, and the wholesale cost of oil on each day of observation must be considered in order to fit a model that can attempt to parse these patterns and make accurate predictions.

3. Data

The data set that I will use for this study contain the average daily posted price for gasoline across 13,687 gas stations in Germany. This represents about 95% of all gas stations in the country. The data has been collected for the time period of May 16, 2014 to December 11, 2015. The data set includes the static variables of brand of gas station, brand of gasoline, distance to autobahn, whether the operator of the gas station is independent, the latitude and longitude of the station, a dichotomous variable indicating if the gas station lies directly on the autobahn. Also collected every day of the study is the price of E5 gasoline, the price of refined gasoline out of Rotterdam, the price of West Texas Instrument (WTI) crude oil, the price of Brent crude oil, and the day of the week. In total there are 8,270,468 observations in this data set. Each observation is a daily price point for the above mentioned gasoline for each station on each day in the data set, along with the corresponding market and geographic variables for each day and station.

While there are a total of 13,687 gas stations that are observed on at least one day in the study, only 12,374 of the stations have been observed on all 575 days. In order to keep the task of prediction balanced for each station, only the 12,374 stations that have an observation for each day for May 16, 2014 to December 11, 2015 have been retained for this paper. In total, this means that there are 7,115,050 observations to be used for the purpose of training and testing models to predict the price of gasoline at a specific station.

In order to add to the geographic understanding of the model, reverse geocoding using the latitude and longitude of each gas station has been performed and a variable has been added to the dataset that indicates which of the sixteen German states each gas station resides in.

The locations of the 12,374 stations that were used for this study can be seen in figure 1 below:

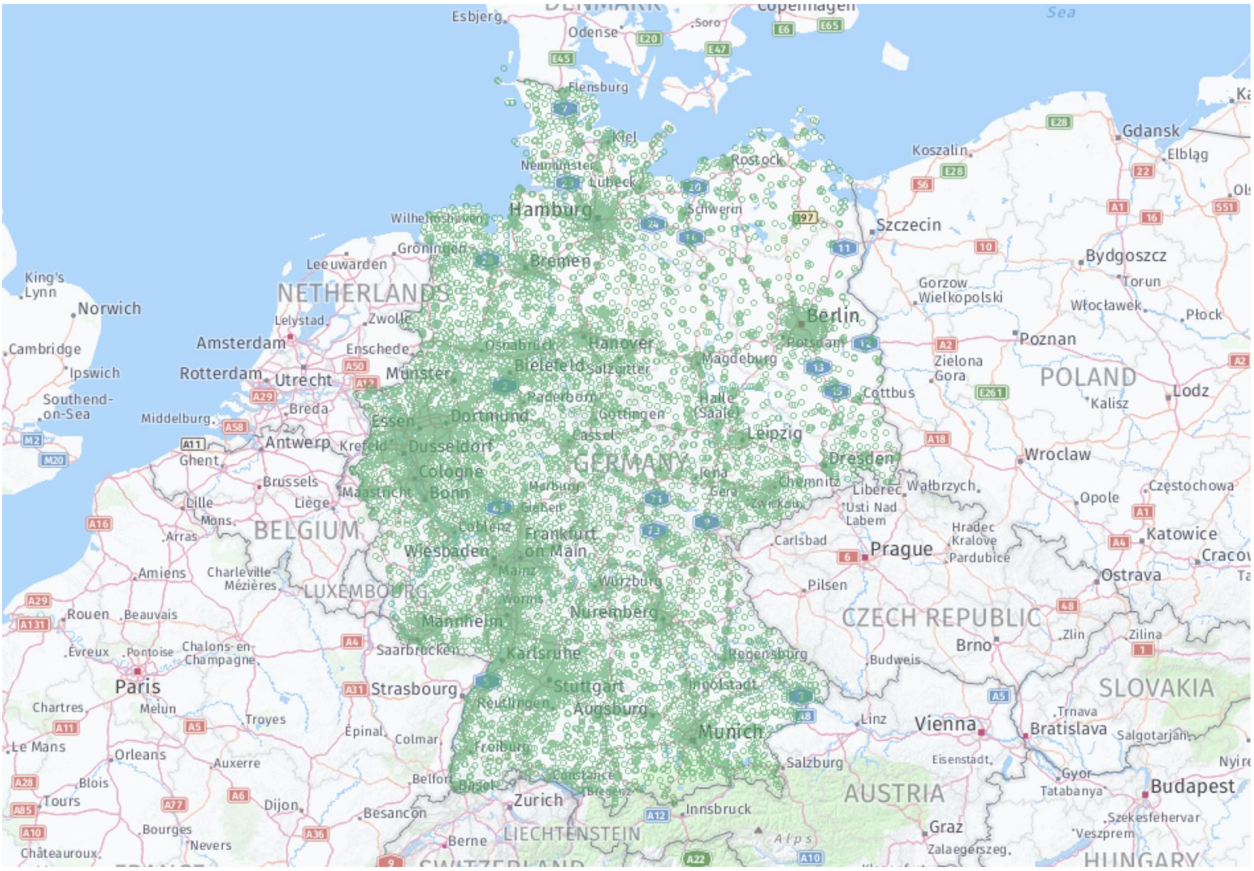

Figure 1: Locations of 12,374 gas stations used in this study based on latitude and longitude.

Table 1 below shows the descriptive statistics for select variables.

	Observations	Mean	Standard deviation	Minimum	Maximum
E5 gasoline price (€)	7,115,050	1.45364	0.1044921	0.8995	2.498042
Distance to the autobahn (m)	7,115,050	6850.198	7997.5	0.1899418	64109.73
Whether station on autobahn	7,115,050	0.03159851	0.1749287	0	1
Whether station brand is Aral	7,115,050	0.1677791	0.3736701	0	1
Whether station brand is Esso	7,115,050	0.07541521	0.2640602	0	1
Whether station brand is Jet	7,115,050	0.04323118	0.2033771	0	1
Whether station brand is Shell	7,115,050	0.1307829	0.3371628	0	1
Whether station brand is Total	7,115,050	0.02763396	0.1639217	0	1
Rotterdam price ($)	7,115,050	0.4941067	0.0999969	0.321701	0.6813258
Brent crude price ($)	7,115,050	70.62451	22.64122	39.28281	114.6798
WTI crude price ($)	7,115,050	65.40877	21.88225	37.26793	106.68

Table 1: Descriptive statistics for variables in the dataset.

The variable that indicates the day of the week of the observation can take the values 1-7. A value of 1 means that the observation was collected on a Monday. Each incremental value is successive throughout the days of the week.

The variable that indicates which state the gas station resides in has no missing values and has been altered from a categorical to a numerical variable. The state variable can range in value from 1 to 16.

This dataset deviates from a common multivariate time series by being multi-site. Indeed, there are 12,374 sites in the dataset. A variable was created that identifies each station using a unique integer so that each observation could be fed into the models sequentially according to date while allowing the models to identify the stations as separate entities.

A commonly necessary step in time series data analysis is that of removing the time trend and seasonality. A time trend in time series data has been formally defined as “an intrinsically determined monotonic function within a certain temporal span” (Wu, 2007). Neither such a time trend not seasonality trend were found in this time series. Given that the data only spans roughly 1.5 years, time trends that may have been apparent over greater temporal space were not found and no differencing of the data in order to enforce stationarity was performed. The gasoline price over the entire data span of the time series for a randomly chosen station is below in figure 2.

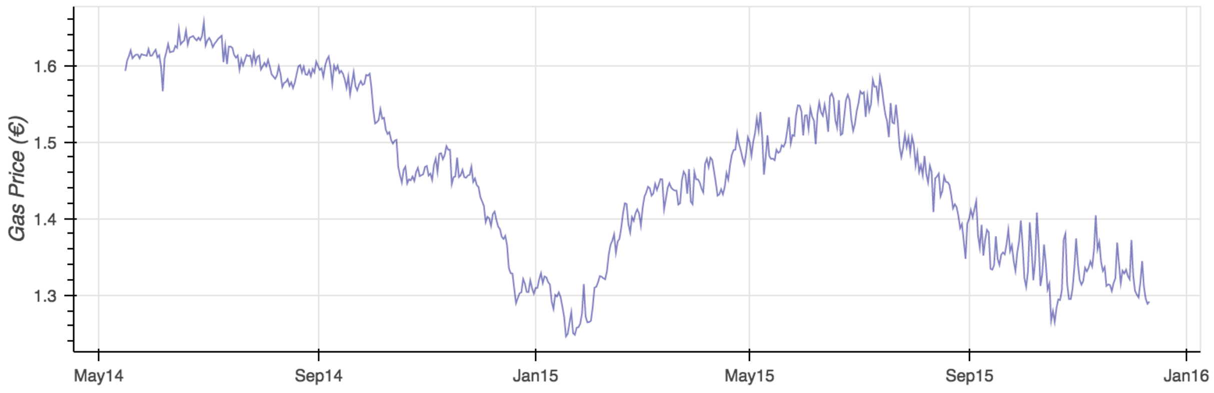

Figure 2: The gasoline price from May 16, 2014 to Dec 11, 2015 at station 2001.

Input map of price variation and means.

4. Methods

As new models for time series forecasting become popular, researchers have more options to consider when selecting which model to pursue. Increasingly, papers whose goal, like this one, is to compare the performance of various models are cropping up. It is especially enlightening to compare traditional econometric models with more modern machine learning approaches to time series forecasting.

For this reason, this paper examines the prediction accuracy of both simple and complex, traditional and modern models. The models considered become increasingly more computationally expensive in regards to both time and memory. First, a simple multiple linear regression was considered. Second, a vector autoregression model, well-suited for multivariate time series prediction, was fit and evaluated. Third, a random forest, less restrictive in its assumptions, was examined. Fourth, a long short term memory neural network, the neural network most often used for time series prediction, was considered.

4.1 Multiple linear regression

While using a multiple linear regression model is a very simplistic approach of modeling this prediction problem, the linear nature of the model may fit well for this problem given the linear relationship between the gas price and the refined gasoline price out of Rotterdam. Figure 3 below shows the linear relationship between gas price in time t and rotterdam gas price in time t at a randomly selected station.

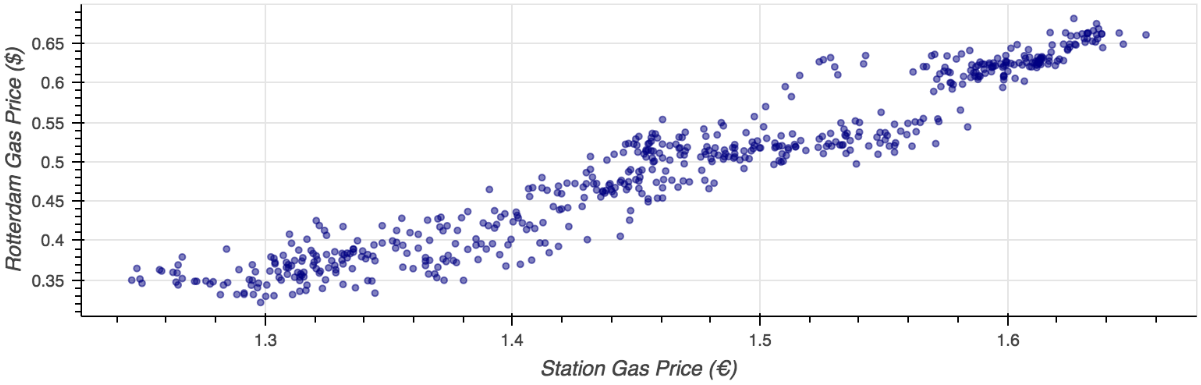

Figure 3: The scatter plot above shows the strongly linear relationship between the gas price at station 201 and the refined gas price out of Rotterdam, both in time t.

The model for the multiple linear regression model is as follows:

$$gas\_price_t = \beta_0 + \beta_1 weekday + \beta_2 dautobahn + \beta_3 autobahn + \beta_4 aral + \beta_5 esso + \beta_6 jet + \beta_7 shell + \beta_8 total + \beta_9 rotterdam + \beta_{10} brent + \beta_{11} wti + \beta_{12} state + \beta_{13} longitude + \beta_{14} latitude + \beta_{15} station + \beta_{16} num\_days + \varepsilon_t$$

Formula 1

Formula 1 also reveals the data structure for this model. The dependent variable is the gas price at the station in time t and the input variables are all of the explanatory exogenous variables appropriate for the station and time period.

Before training the multivariate linear model, each variable in the data is normalized. This is done using the method of subtracting by the variable's mean and dividing by its standard deviation. The resulting variables will each have a mean of zero and a variance of 1.

4.2 Random forest

A model that is not held back by the strong linear assumptions of the multivariate linear regression is considered next. Random forest regression is the result of multiple decision trees. As described by Kane in a paper comparing the effectiveness of the ARIMA and random forest methods for time series forecasting: "Decision trees recursively partition data in the regression space until the amount of variation in the subspace is small. A predictor for the subspace can then be created simply by taking the average value of the dependent data corresponding to the independent data in the subspace...Predictions for new data are obtained by finding the predictor corresponding the partition where the new input variable resides" (Kane, 2014).

Each decision tree uses only a subset of features and data available in order to capture as much of the structure of the data as possible by viewing various subsets of it. Individual trees are quite noisy but have low bias when grown appropriately deep. The noise of the individual trees is quieted when all of the trees are averaged as part of the random forest method (Hastie, 2001).

After the random forest has been trained, values for the features at future points in time are fed into the forest and each tree votes on the value that it has classified as the prediction for the output variable (Kane, 2014).

As with the multiple regression model, the data is fed into the random forest after each variable has been normalized.

4.3 Long short term memory neural network

The final method to be considered for this particular multisite time series forecasting is a long short term memory neural network (LSTM). An LSTM is a specialized version of a recurrent neural network (RNN). In a very well known blog post, Christopher Olah explains that recurrent neural networks solve the problem that traditional neural networks face of not being able to make decisions in context. That is to say that traditional feedforward networks are not informed by the output they generate from previous observations. In a time-series problem, the ability to understand that the output from the previous steps should be considered in the following estimation of the output variable is very important. RNNs solve this problem by having loops. This allows the information from previous estimations to persist. Olah uses the following illustration, figure 4, to explain RNNs:

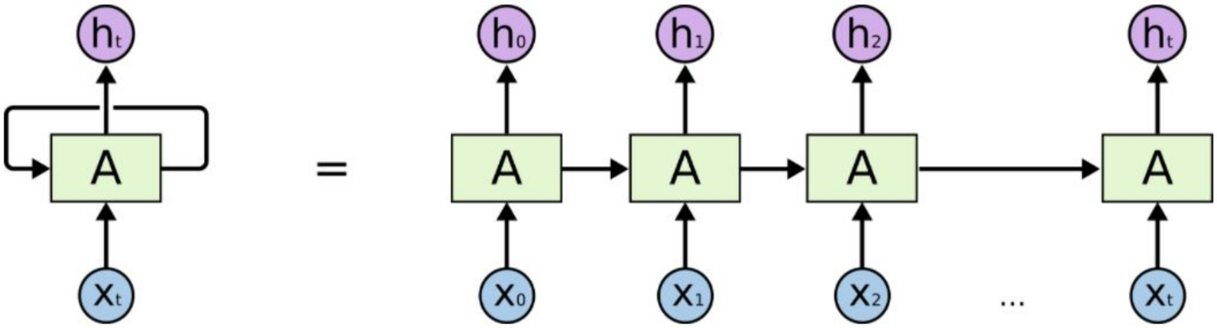

Figure 4: An unrolled recurrent neural network. “In the above diagram, a chunk of neural network, A, looks at some input xt and outputs a value ht. A loop allows information to be passed from one step of the network to the next” (Olah, 2015).

The problem that persists even with an RNN is that it is possible that important information for forecasting the current observation occurred many observations previously. When the gap between the current observation and the needed information from previous observations grows sufficiently large, the RNN is incapable of collecting that information. LSTM networks solve this problem by being able to learn “long-term dependencies” (Olah, 2015). LSTM networks use multiple sigmoid layers to learn these dependencies. Information in the cell state is updated, kept, or forgotten by the network based on the sigmoid and tanh layers. These added layers allow the LSTM to make decisions about the relevancy of the new information that is passing through and information that is needed for future prediction can persist much longer than it does in a traditional RNN (Olah, 2015).

For the purposes of this study, the data has been formatted into a supervised learning problem to be fed into the LSTM network in the same way that it was organized for both the random forest and the multiple linear regression models. The target variable is the gasoline price at a particular station at time t and the features are the station and day specific variables for time t.

5. Data Preparation

The multivariate linear model, random forest, and LSTM neural network all rely on the same inputs for the prediction as were used to train. Three of these inputs, WTI, Brent crude oil, and Rotterdam gasoline prices, are variables whose future values cannot be known when predictions are made. For this reason, these three variables have been treated separately as univariate time series. Simple linear models have been trained using the values of the initial 545 days in the time series in order to predict the values of the final 30 days in each time series. When predictions were made for the final 30 days of gas prices at each of the 12,374 stations, the predicted values (rather than the observed values for these “future” days) were fed into each model as the values for the variables WTI, Brent crude oil, and Rotterdam gasoline price.

6. Results

6.1 Persistence model

In measuring the overall prediction quality of the models, it is important to begin with a baseline of error for the predictions. It is customary to use the most naive model possible in order to create these baseline error measures. A persistence model is one such extremely naive approach for a time series prediction problem. With a persistence model, the previously observed values for the target variable will persist through time and are treated as the predicted values in the future (Brownlee, February 2017).

In this application, the values that are being predicted are the gas prices at each gas station for the final month of the data set, November 12, 2015 to December 11, 2015. By using the observed gas prices in the final month of the training portion of the dataset, the entirety of which runs from May 11, 2014 to November 11, 2015, I employ a persistence model which naively predicts that the prices from the last month of the training set will be the prices for the month I set out to predict.

With these predictions for the gas price at each gas station for the month of November 12, 2015 to December 11, 2015, I calculate the root mean squared error (RMSE) of these predictions compared to the actual observed prices for that month. The result is a RMSE of 0.034. Since the RMSE has the same units as the quantity being estimated, the persistence model can naively predict gas prices one month in the future that are on average deviant from the true values by 3.4 cents. This will be used as our baseline when evaluating the results of the models created for this study since it is a very naive and computationally inexpensive process by which to make predictions.

6.2 Multivariate linear model

The linear model was trained using the data from the first 545 days of the dataset. With its simple formulation and computationally inexpensive training method, the multivariate linear model requires the least amount of training time and computation power out of the three non-baseline models. This aspect makes the linear model an attractive option for this multi-site prediction problem, where training deeper, more complex models can be very temporally and computationally expensive. However, the predictions from this model were not accurate enough to make this simple solution viable. The gas prices predicted by the multivariate linear model for the final 30 days had a RMSE of 0.035, a higher RMSE than the one produced by the baseline persistence model.

While the computational ease of training and testing the multivariate linear model is advantageous, in its simplest form the multivariate linear model is not accurate enough to beat out the baseline persistence model. (It should be noted that the persistence model is computationally less expensive than even the linear model, since it only requires shifting values in the dataframe and no actual model is trained.)

The accuracy of the predictions output by the linear model varies across the 30 days. Figure 5 below shows the RMSE for each of the 30 days from the predicted period.

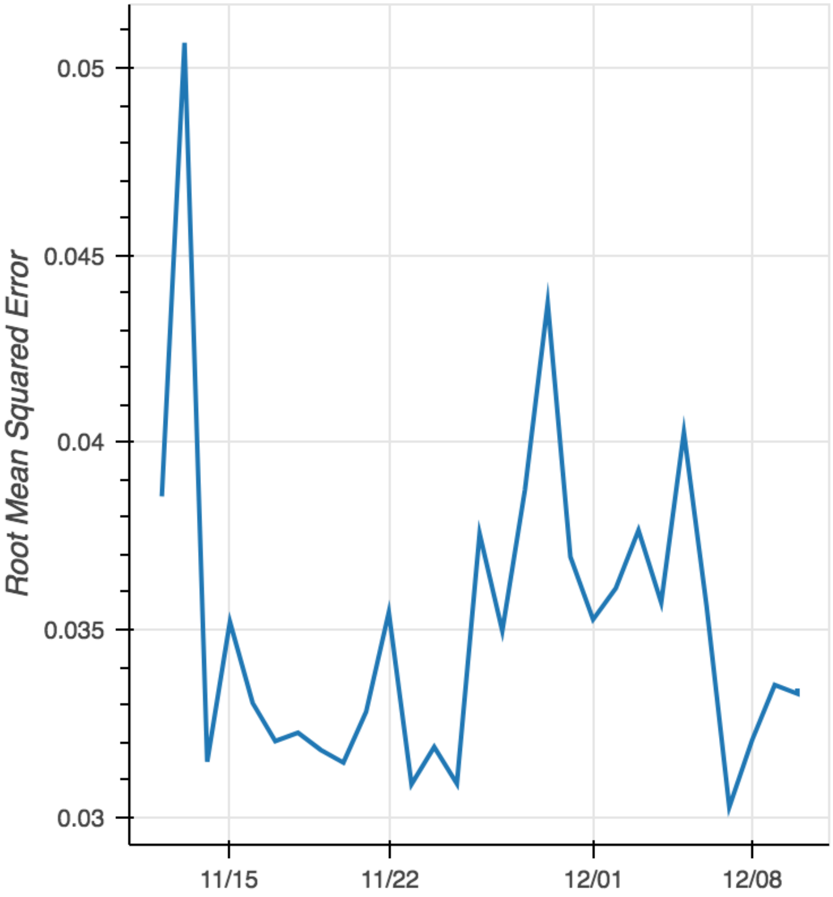

Figure 5: RMSE for each of the 30 days when gas prices were predicted by the multivariate linear model. The average RMSE for the entire 30 day period was 0.035.

The peak of the graph on the second day of prediction mirrors the sudden increase in gas price seen on that day that would have been difficult for the multivariate linear model to predict. Figure 6 shows the observed gasoline prices at a randomly selected station for the month in which the predictions were made.

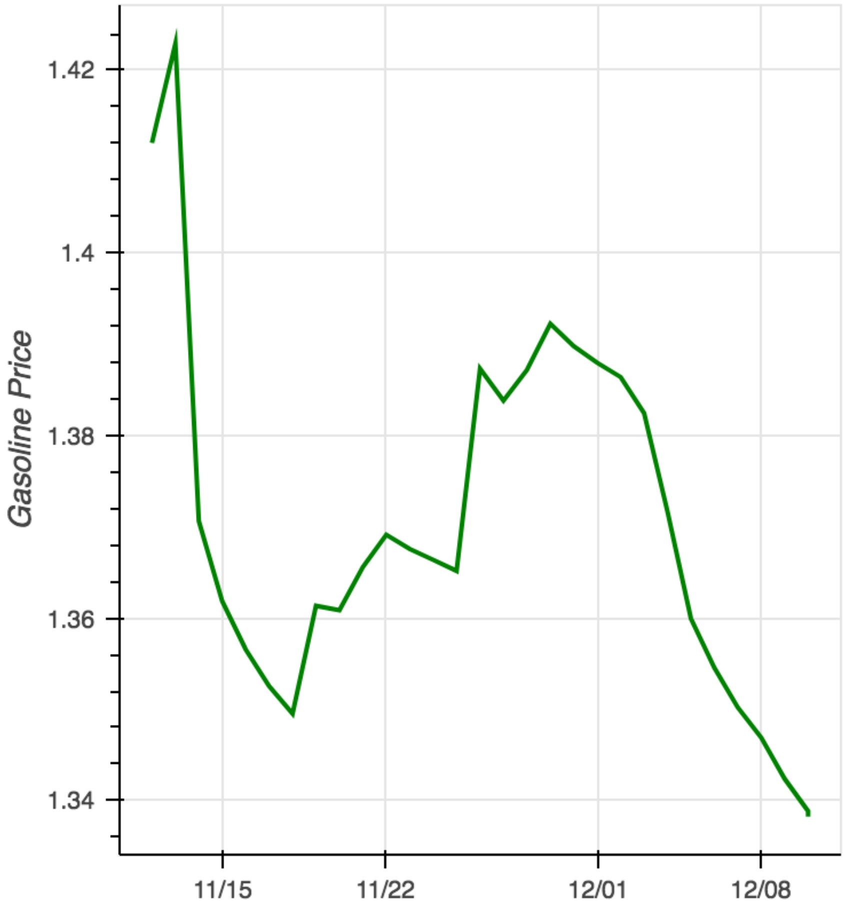

Figure 6: The observed gasoline prices at station 201 from November 12, 2015 to December 11, 2015.

6.3 Random forest

The random forest model was trained on the same initial 545 days of data that the linear model was trained on. The same predictions for the non-static WTI, Brent crude oil, and Rotterdam gasoline prices were used in place of the observed values during prediction. However, the random forest model has tuning hyperparameters that go beyond the simplicity of the linear model and allow the model to be better specified for this exercise in prediction. The hyperparameters for the random forest model that I tuned are the number of estimators (or trees) in the forest, the maximum depth to which each tree is allowed to grow, the number of features used for each tree, the minimum number of samples required to split a node, and the minimum number of samples required at each leaf node of a tree (Koehrsen, 2018).

I performed a random grid search across different values of each of these parameters. Most notably, I trained models with between 10 and 150 estimators and with maximum depths of between 2 and 50. With each added estimator and increase in maximum depth, the model takes longer to train and uses more memory. However, the improved accuracy of the model begins to taper and stops improving once maximum depth has reached 25.

The best RMSE was achieved by a model trained with 100 estimators, a maximum depth of 25, all features used by each estimator, a minimum number of samples required to split a node of 2, and a minimum number of samples required at each leaf node of 1. The RMSE for this model was 0.025.

The RMSE for the random forest model is a one cent improvement in accuracy over the multivariate linear model and a 0.09 cents improvement over the baseline.

Figure 7 shows how the random forest model performed each day of the 30 days used for prediction.

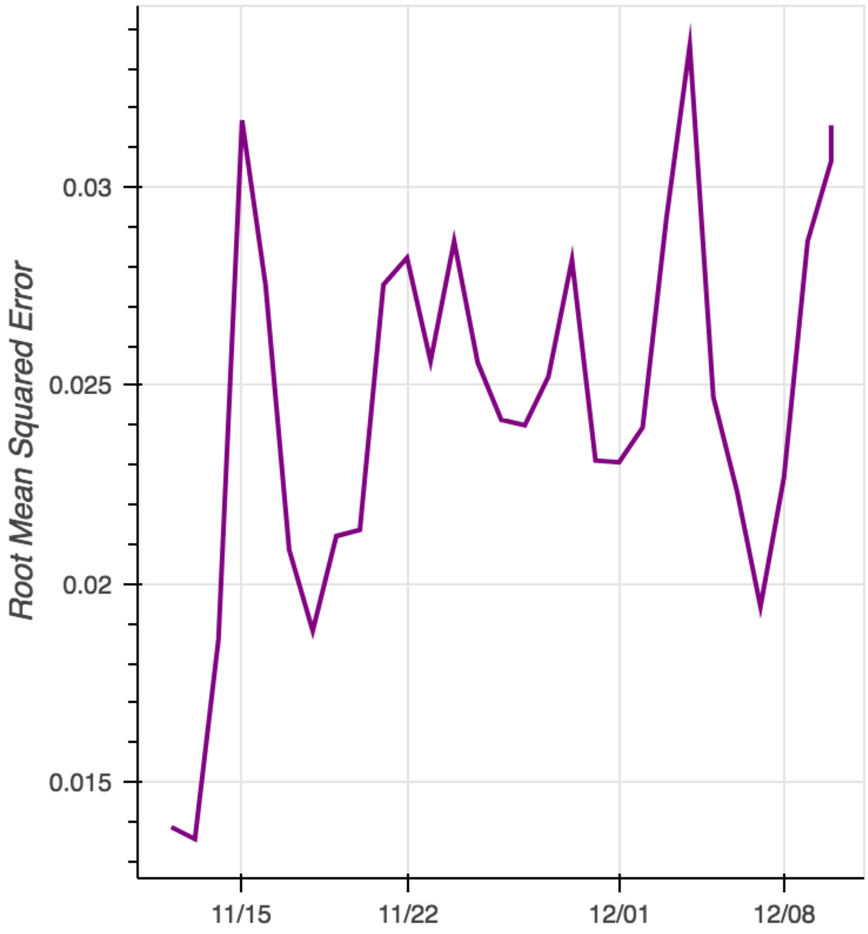

Figure 7: This figure shows the RMSE between the predicted and observed gas prices for each day of the 30 day prediction interval that resulted from the tuned random forest model.

6.4 LSTM neural network

The neural network model that was created for this study is the originally developed LSTM comprised of one hidden LSTM layer and a feedforward output layer (Brownlee, July 2017). The model was trained using the same input data of the initial 545 days that were used to train the linear and random forest models.

The LSTM was optimized using the Adam algorithm. Adam is an alternative to classical stochastic gradient descent. It differs by setting and updating, using the second moments of the gradients, a separate learning rate for each parameter in the model, rather than the single learning rate used in stochastic gradient descent (Kingma, 2015). I trained the LSTM on a series of values for the hyperparameters number of epochs and batch size. Models were trained on a number of epochs ranging from 1 to 500 and batch sizes ranging from 500 to 7200.

Training the LSTM model requires significant time; in fact, training the 500 epoch model required more time than any other LSTM or random forest model trained for this study. The model trained using 500 epochs and a batch size of 1000 took nearly 10 hours to train using high memory CPUs on Google Cloud Compute Engine. However, the LSTM model trained with 500 epochs required less memory than training the random forest with the most estimators. In order to improve the time intensive aspect of training the LSTM model and, more vitally, to control for overfitting, early stopping can be utilized. Early stopping is a technique wherein the neural network can choose to stop training before the stipulated number of epochs have passed if the test error begins to increase. When test error increases while training error decreases or remains stagnant, the neural network is overfitting the training data and will not perform well on out-of-sample data. Early stopping attempts to detect overfitting and stop training prematurely in order to avoid it.

While tuning the LSTM model, I employed the early stopping technique for combinations of batch size and number of epochs where it was apparent that overfitting was occurring.

Several combinations of hyperparameters yielded the same lowest RMSE. The combination that is the least computationally expensive was chosen. With the number of epochs set to 20 and batch size set to 1000 with early stopping incorporated into the training, the RMSE of the LSTM model is 0.030. The early stopping parameter chose to stop training after the 20th epoch. The relationship between train and test loss is shown in figure 8 below.

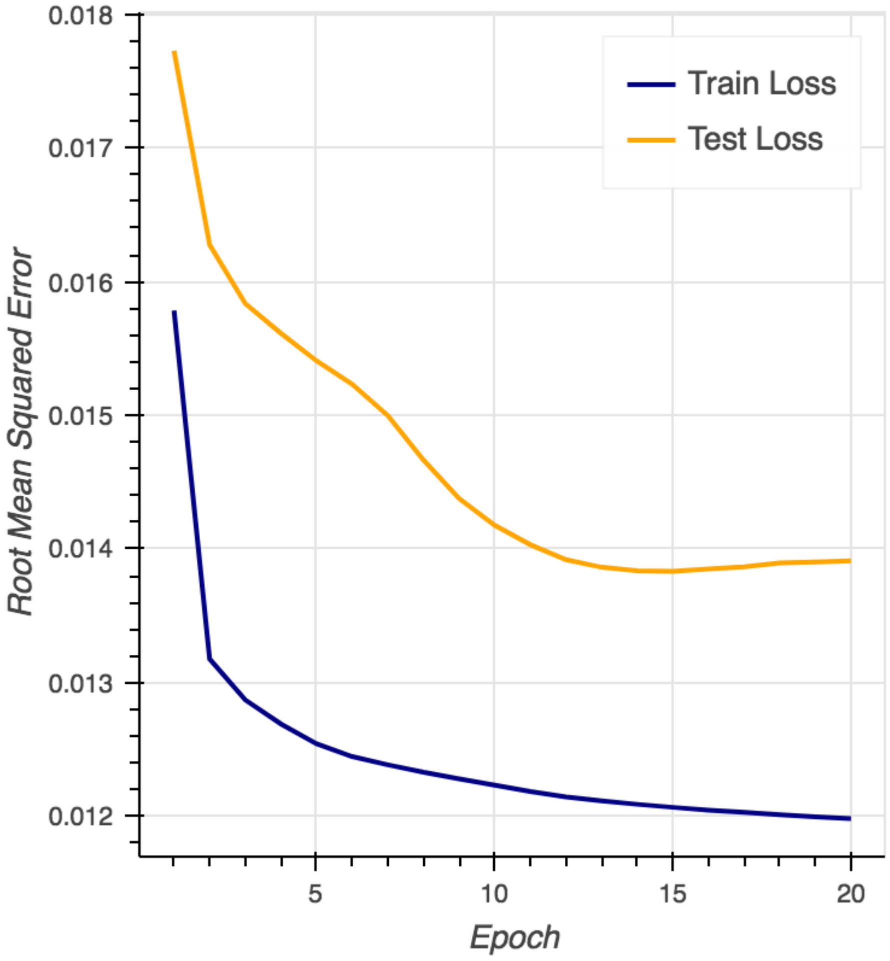

Figure 8: The test loss and train loss are graphed at each epoch as the LSTM model is trained. The technique of early stopping caused training to stop after epoch 20 since the test loss was beginning to increase.

Figure 9 shows the RMSE from the LSTM model for each day of the 30 days in the prediction interval. The graph is highly variable, much like the respective graph for the linear model, however there is a smaller variance in the values of RMSE across the time period resulting from the LSTM model.

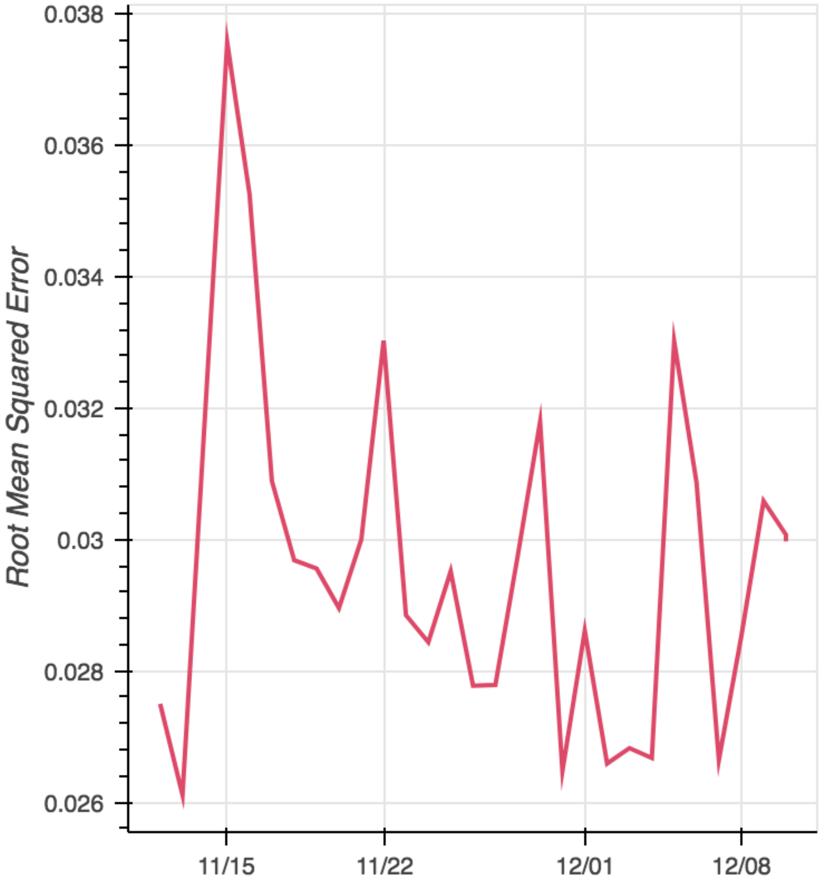

Figure 9: The RMSE is calculated for each day of the 30 day prediction interval that resulted from the tuned LSTM model.

7. Conclusion

An extension to this work would be to tune a model of stacked LSTM neural networks.

References

Borenstein, Severin, A. Colin Cameron, and Richard Gilbert. “Do Gasoline Prices Respond Asymmetrically to Crude Oil Price Changes?” *The Quarterly Journal of Economics* Volume 112, 1 (1997). 305–339. https://doi.org/10.1162/003355397555118.

Brownlee, Jason. “How to Make Baseline Predictions for Time Series Forecasting with Python.” Machine Learning Mastery, 21 Feb. 2017, machinelearningmastery.com/persistence-time-series-forecasting-with-python/.

Brownlee, Jason. “Stacked Long Short-Term Memory Networks.” Machine Learning Mastery, 19 July 2017, machinelearningmastery.com/stacked-long-short-term-memory-networks/.

Eckert, Andrew and Douglas S. West. "Retail Gasoline Price Cycles across Spatially Dispersed Gasoline Stations." *The Journal of Law and Economics* 47, 1 (2004). 245-273.

Haining, Robert. "Testing a Spatial Interacting-Markets Hypothesis." *The Review of Economics and Statistics* 66, 4 (1984). 576-83. doi:10.2307/1935981.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. New York, NY, USA: Springer New York Inc., 2001.

Kane, MJ, Price N, Scotch M, and Rabinowitz P. “Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks.” *BMC Bioinformatics* 15, 1 (2014). doi:10.1186/1471-2105-15-276.

Kingma, Diederik P. and Jimmy Lei Ba. “Adam: A Method for Stochastic Optimization”. *ICLR* (2015). https://arxiv.org/abs/1412.6980.

Koehrsen, William. “Hyperparameter Tuning the Random Forest in Python – Towards Data Science.” Towards Data Science, Towards Data Science, 10 Jan. 2018, towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.

Olah, Christopher. “Understanding LSTM Networks.” Understanding LSTM Networks -- Colah's Blog, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

Pinkse, J. and Margaret E. Slade. “Contracting in space: An application of spatial statistics to discrete-choice models”. *Journal of Econometrics* 85, 1 (1998). 125-154. https://doi.org/10.1016/S0304-4076(97)00097-3.

Pinkse, J. and Margaret E. Slade and C. Brett. “Spatial Price Competition: A Semiparametric Approach.” *Econometrica* 70 (2002). 1111–1153. doi:10.1111/1468-0262.00320.

Wu, Zhaohua, Norden E. Huang, Steven R. Long and Chung-Kang Peng. “On the trend, detrending, and variability of nonlinear and nonstationary time series.” *PNAS* 104, 38 (2007). 14889-14894. https://doi.org/10.1073/pnas.0701020104.