US ELECTRICITY PRICE AND VOLATILITY ANALYSIS

COWEN SUSTAINABLE INVESTMENTS

ANALYTICS IN PRACTICE

IEOR 4524

MEMBERS

Akshay Kumar	Ankit Yadav	Beichen Liu
ak4271@columbia.edu	ay2436@columbia.edu	bl2713@columbia.edu

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Scope review

The electricity market (unlike the regular stock market, where you often find use cases of price prediction) is set up as an Intra-Day and Day-Ahead market. The dynamics which guide the prices in both these markets are different and hence have to be dealt with separately.

At the same time, prices and their corresponding underlying factors that drive these values may be different for different regions and customers. As a result, it's imperative to define the scope of the problem into tangible deliverables.

The team, post discussions with Cowen has agreed to target the following as the first set of deliverables -

1) **Market of Interest** -

We'll be focusing on the Day-Ahead electricity price market, instead of the Intra-Day and Ancillary Services market.

2) **Geography of Interest** -

Given California's commitment to shift to green energy, this state will be our primary geography of interest.

3) **Price of Interest** -

Electricity prices differ from the retail to the wholesale market. Cowen has prioritized impact on direct consumers over impact on suppliers and distributors. Hence, we'll be focusing on the retail market.

4) **Exogenous Variables** -

External or exogenous variables are going to be extremely significant in our hybrid model. Since variables like temperature and transmission costs play a part in defining the seasonality of data, we'll be looking forward to incorporating them into our model. Variables like state regulations (which also drive price behaviors) are being ignored for the time being.

Data Collection

Historical Price

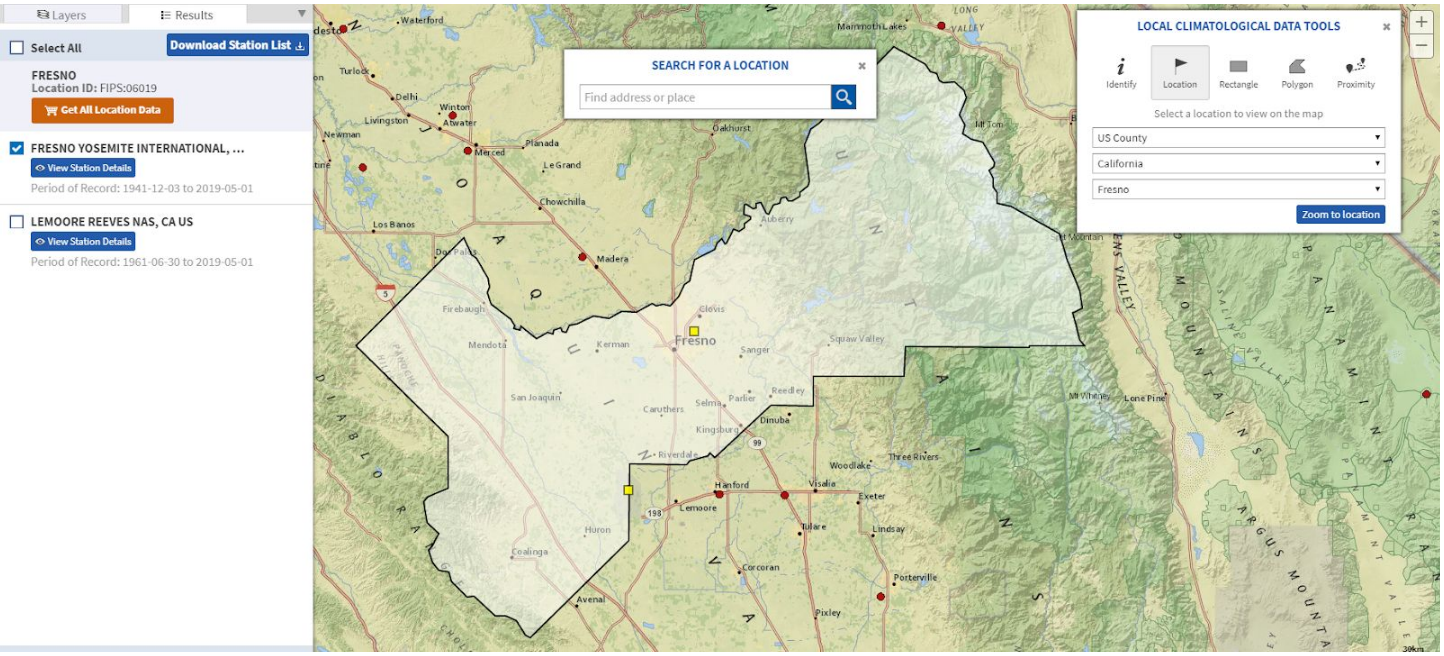
The team was advised to use publicly available data from various ISO websites. This involved using web scrapers wherever possible (for example with NYISO and CAISO) or using data that was manually downloaded. The following table depicts information that was gathered from multiple sources -

No.	ISO Name	Data Covered	Duration
1	NYISO	Node level	2016-2019
2	ISO- NE	Zone level	2011-2019
3	Midcontinent Independent System Operator	Node level	2011-2018
4	AESO	Node level	2008-2018
5	Electric Reliability Council of Texas (ERCOT)	Zone level	2016-2019
6	Independent Electricity System Operator (IESO)	Zone level	2011-2019

The data analysis and modeling was conducting on hourly price data from CAISO that was shared by the project sponsor.

Exogenous Variables

Although the initial discussions involved leveraging Earth Institute and University of California Berkeley for gathering sensor data on temperature, the team was advised to use climate data from https://gis.ncdc.noaa.gov/maps/ncei/lcd. A snapshot of how the data is visualized on this tool for Fresno in California is depicted below -



FRESNO
Location ID: FIPS:06019

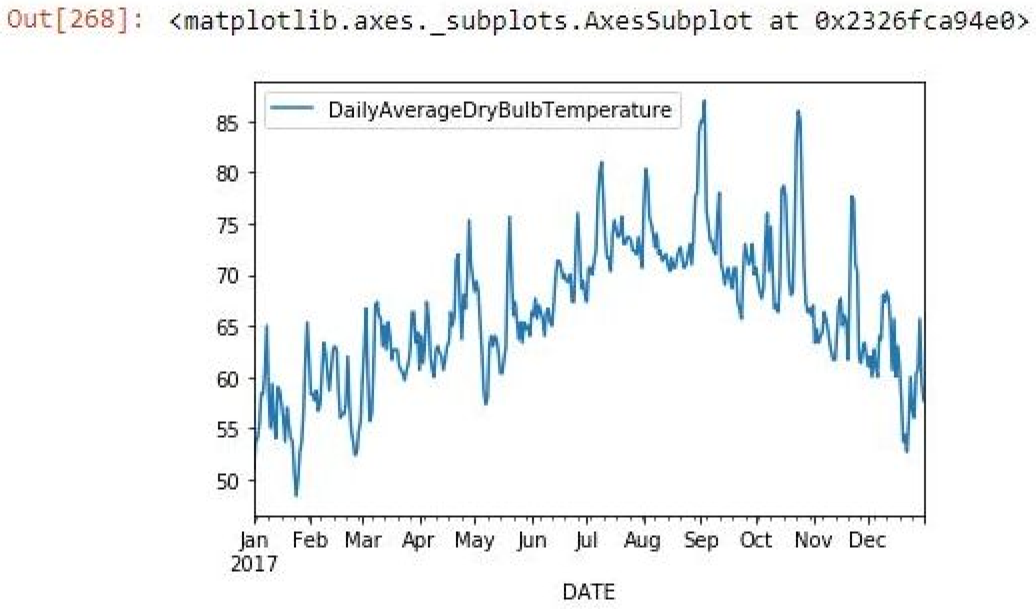
☑ FRESNO YOSEMITE INTERNATIONAL, ...
Period of Record: 1941-12-03 to 2019-05-01

□ LEMOORE REEVES NAS, CA US
Period of Record: 1961-06-30 to 2019-05-01

LOCAL CLIMATOLOGICAL DATA TOOLS
Select a location to view on the map
US County: US County
California
Fresno

SEARCH FOR A LOCATION
Find address or place

The tools shares climate data distributed over monthly, hourly, daily ranges covering information of temperature, pressure, humidity, wind speed, snow fall, heavy fog, dew, altimeter settings etc. The exported file had about 124 unique columns, but for the purpose of our modeling exercise, we primarily focused on the temperature data. An example distribution of temperature for the year 2017 for Los Angeles appears as follows -



```
Out[268]: <matplotlib.axes._subplots.AxesSubplot at 0x2326fca94e0>
```

Data Preprocessing and Visualization

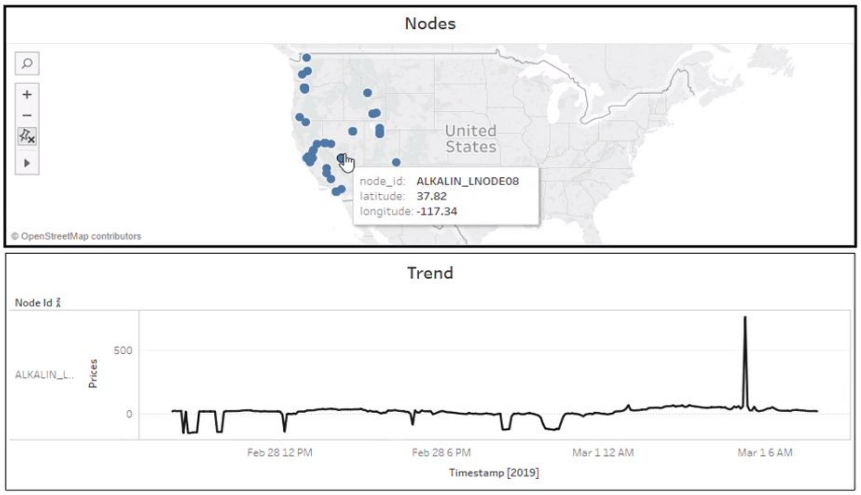
The raw data is hourly-price in each node. In order to make efficient models, we take the maximum price and minimum price in a day and find their difference – spread price—for model building.

To begin with, we try building models for each node from the data in CAISO. However, given that it takes 2 minutes for a model to run for a node, the total time it'll take to run for all of the nodes (i.e. 5792) would be a lot. Following a discussion with the project sponsor, we subset our data and focus on major cities like Los Angeles, San Francisco, San Diego, San Jose, Fresno, Sacramento, Seattle. For every city, we calculate the average price of all nodes for a given day and use the information in our model.

Data visualization

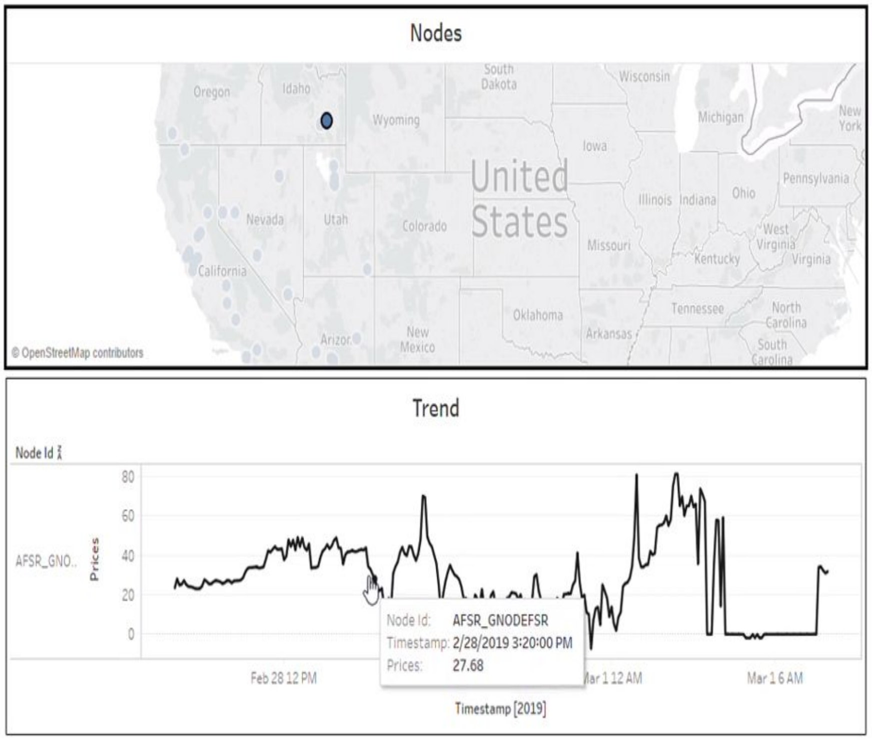
A sample Tableau dashboard that shows prices of 6000 nodes is shown below. The tableau shows the distributions and trends of electricity prices across the US (data was gathered from CAISO).

Figure 1 Tableau Example



node_id: ALKALIN_LNODE08
latitude: 37.82
longitude: -117.34

Figure 2 Tableau Example

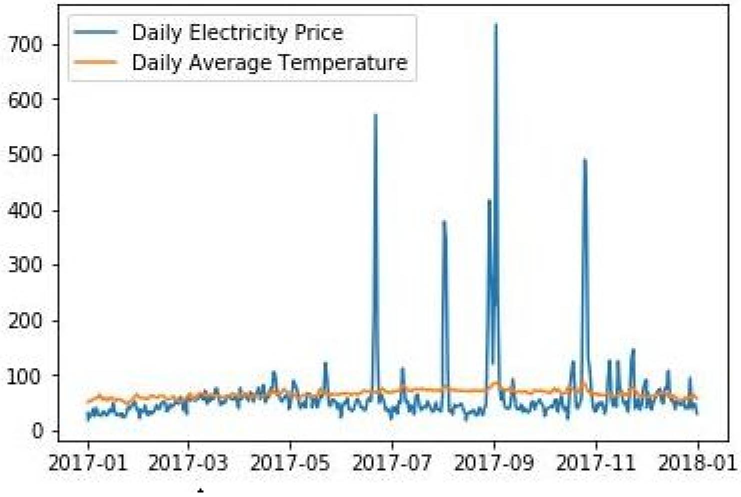


Node Id: AFSR_GNODEFSR
Timestamp: 2/28/2019 3:20:00 PM
Prices: 27.68

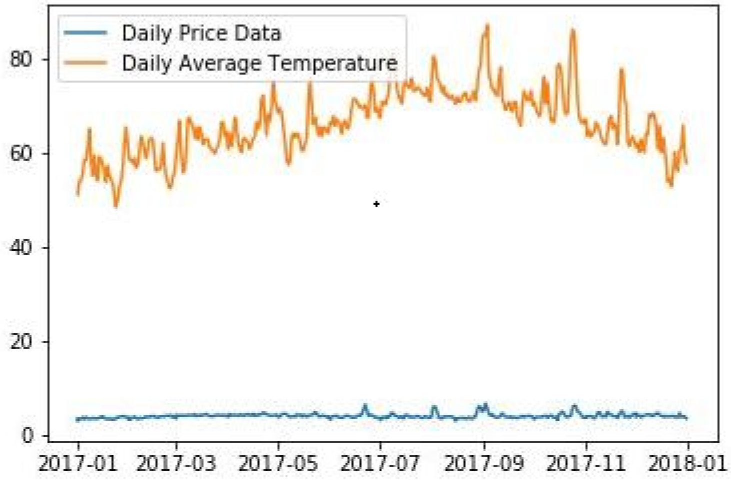
Sample graphs for daily temperature and daily spreads for a few cities we analyzed -

Los Angeles

Daily Price Spread vs Average Temperature

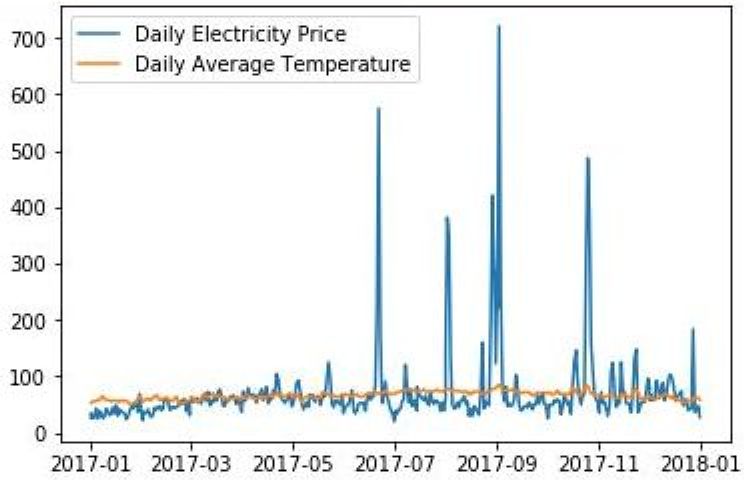


Daily Log Price Spread vs Average Temperature

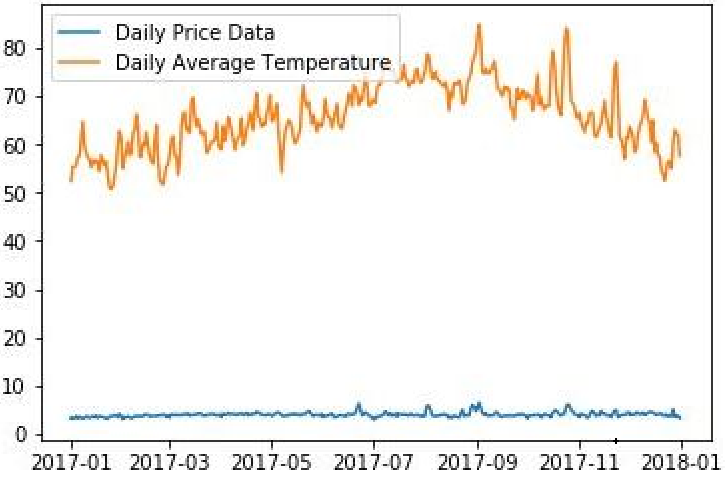


Similarly for San Diego,

Daily Price Spread vs Average Temperature



Daily Log Price Spread vs Average Temperature



Algorithm Modeling

As a part of the project, we tried working with four models- ARMA, ARIMA, ARIMAX, LSTM. We concluded that LSTM and ARIMA X gave the best results. The following gives a brief introduction of the models and our application.

ARMA & ARIMA

The ARIMA model is a generalization of an ARMA model. Both of the ARMA and ARIMA models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data shows evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

The AR (auto-regressive) part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The MA (moving-average) part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p,d,q)(P,D,Q)m, where m refers to the number of periods in each season, and the uppercase P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model. The model can be generalize as following:

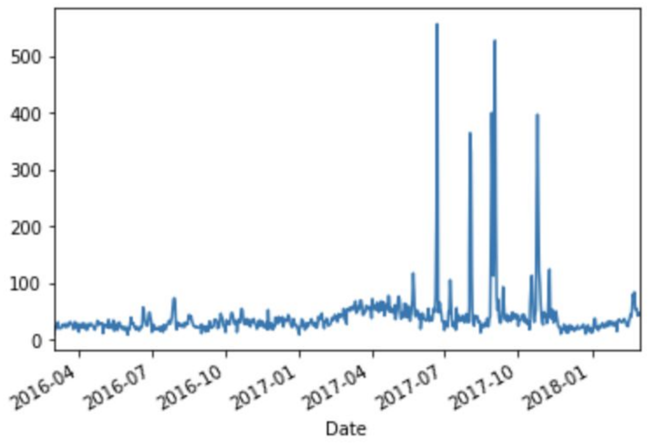
$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t = \delta + \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t$$

Model Evaluation

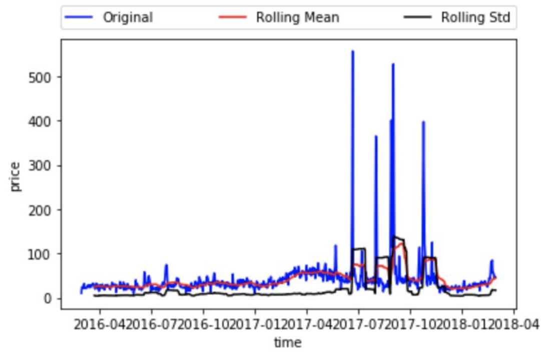
Node data

To begin with, we take a node – “106THSO_LNODED1” – as an example to see the trend of price. We plot the observed price, rolling mean & rolling standard error of price and log-price. Using ACF and PACF, we found the best parameters for our ARMA/ARIMA model.

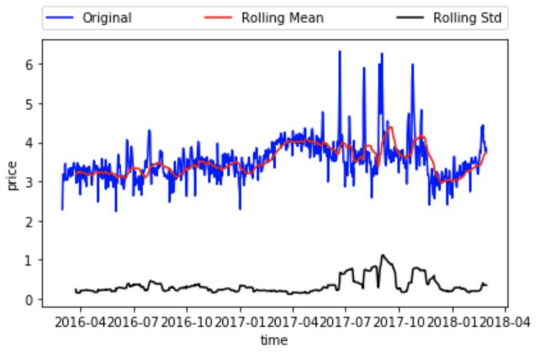
Node Observed Price



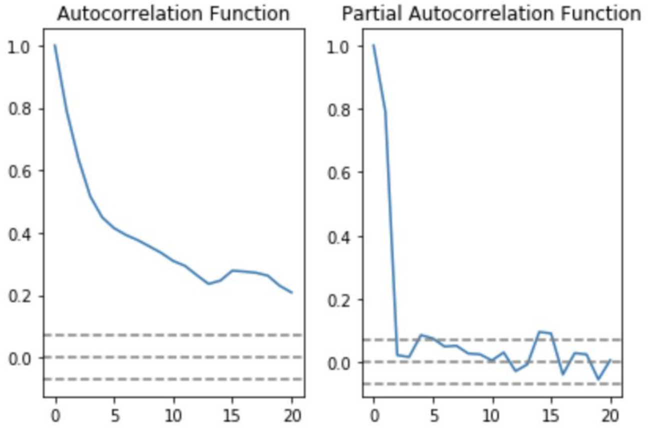
Node Price Rolling Mean & Std



Node Log-price Rolling Mean & Std

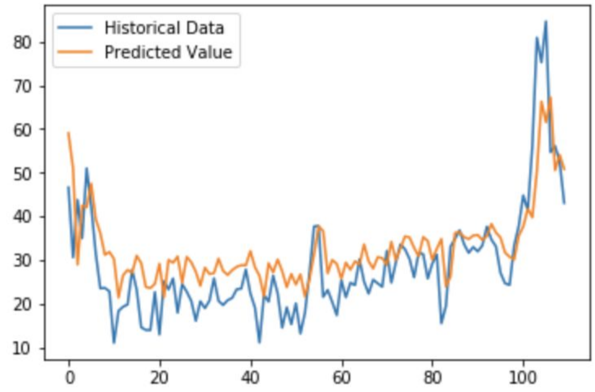


ACF & PACF



According to the graph above, we fitted the node price into ARMA and ARIMA with p and q equal to (5,2) and got a mean-square-error of 76.90737623833544 for ARIMA (with d = 0).

Prediction Evaluation



We observed that this prediction performed fairly well. Adding d greater than 0(degree of differencing) to the ARMA model seemed like a bad decision, since the MSE just increased. Using auto-arima, we tried several combination of model parameters and got the best mse as 7463.705446933585. As a result, we concluded a simpler ARIMA model with degree of differencing equal to 0 (which behaves like ARMA) rather than an ARIMA model with a higher degree of differencing may be a better choice for our dataset.

Scaling to City level data

ARIMAX

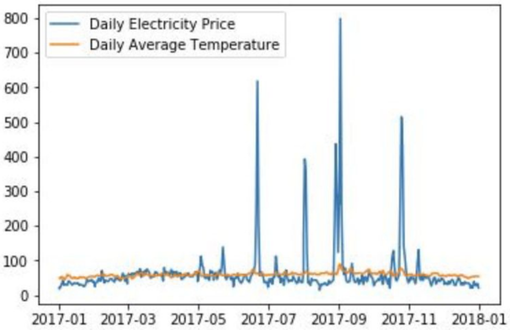
Based on the performance of node data, we tried to scale our model for the price spreads of cities. We took the average of the nodes' prices in a specific city as the price of the city. We had already gathered the temperature data for each city as discussed in the aforementioned sections.

We now incorporate the temperature data as an exogenous variable into our ARIMA model and make predictions using a modified ARIMA or ARIMA X model. City specific results are shared below.

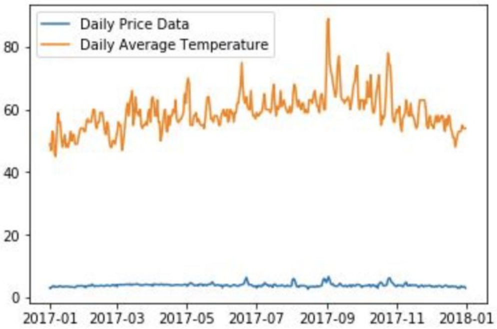
San Francisco

The parameters to run this model on this city are ARIMA (1,0,2). We add the temperature variable to the ARIMA model to generate predictions using ARIMAX. The electricity price spreads and temperature data are shown as below.

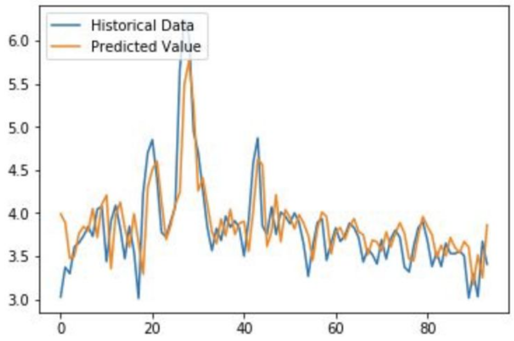
Electricity price and temperature



Log-Electricity price and temperature



Prediction evaluation



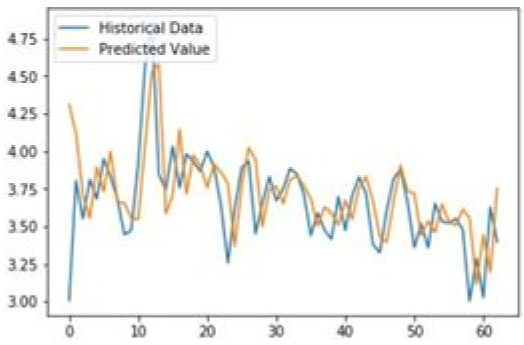
The mse of the model above is 0.1284001645, indicating that temperature does help explain variability in the data. Also, the log price spread helps make smoother predictions instead of raw price spreads which is highly volatile.

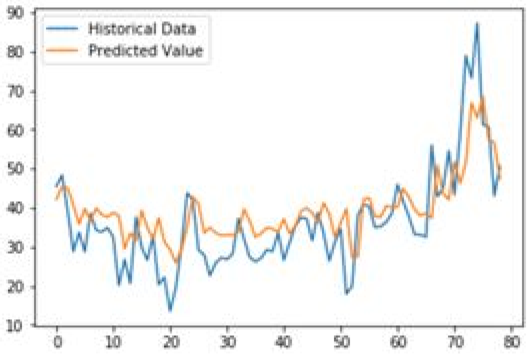
Similar observations for other cities are shared below -

San Jose

ARIMAX Prediction

ARIMA Prediction

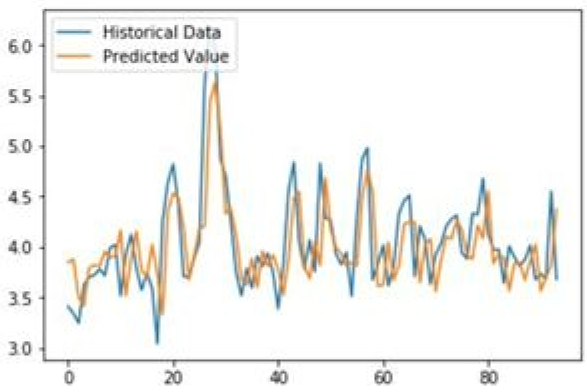


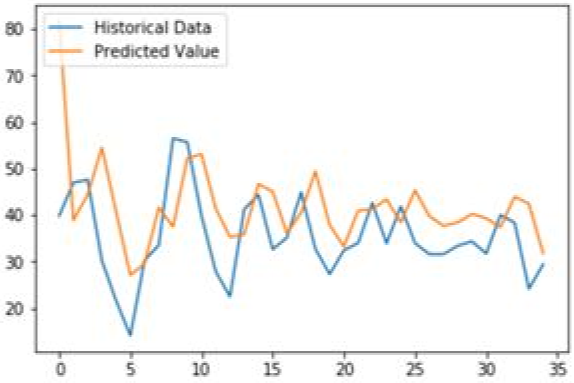


Los Angeles

ARIMAX Prediction

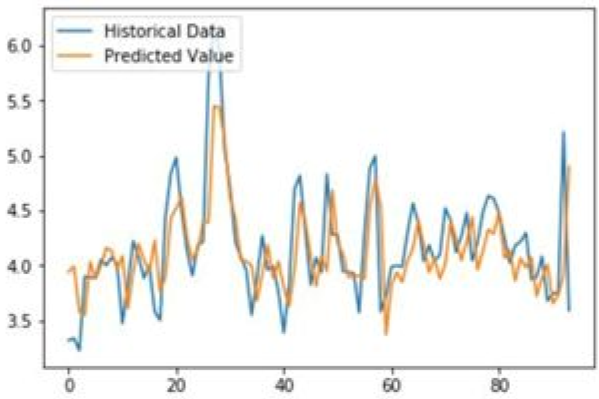
ARIMA Prediction



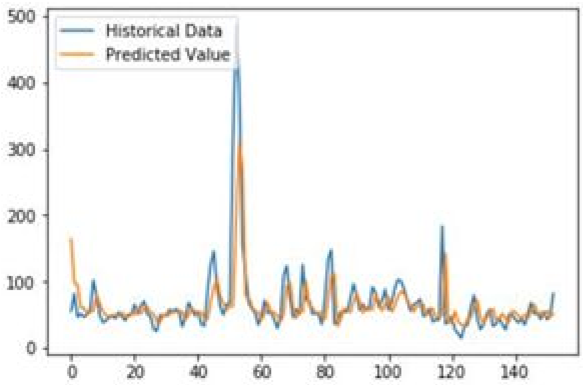


San Diego

ARIMAX Prediction



ARIMA Prediction



LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network, (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can).

There are several architectures of LSTM units. A common architecture is composed of a cell (the memory part of the LSTM unit) and three "regulators", usually called gates, of the flow of information inside the LSTM unit: an input gate, an output gate and a forget gate. Some variations of the LSTM unit do not have one or more of these gates or maybe have other gates. For example, gated recurrent units (GRUs) do not have an output gate.

Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The activation function of the LSTM gates is often the logistic function.

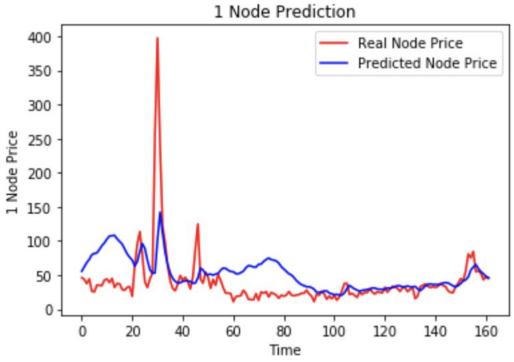
There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections, which need to be learned during training, determine how the gates operate.

Model & Evaluation

Node data

First, we tried the LSTM model for one node – “106THSO_LNODED1” – as an example to see the trend of price. The prediction result is as following, with loss equals to 0.0047.

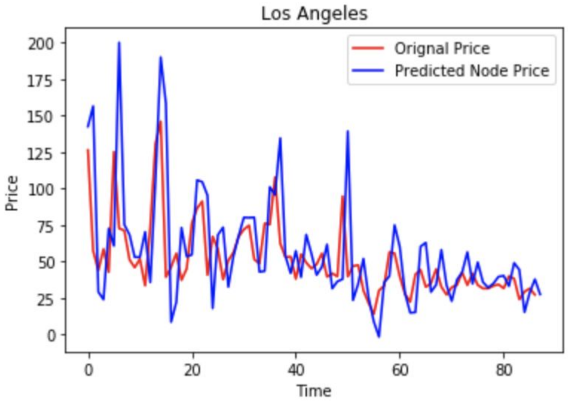
Prediction evaluation



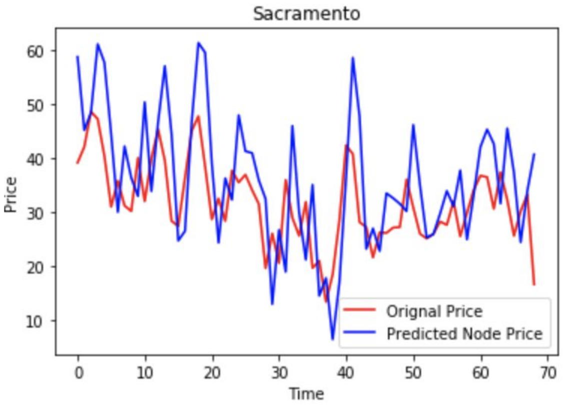
City data

We built the LSTM models for six cities and the result are shown below. The loss of Los Angeles is 0.0027, San Francisco is 0.0035, San Jose is 0.0034, San Diego is 0.0033, Sacramento is 0.0019, Fresno is 0.0038.

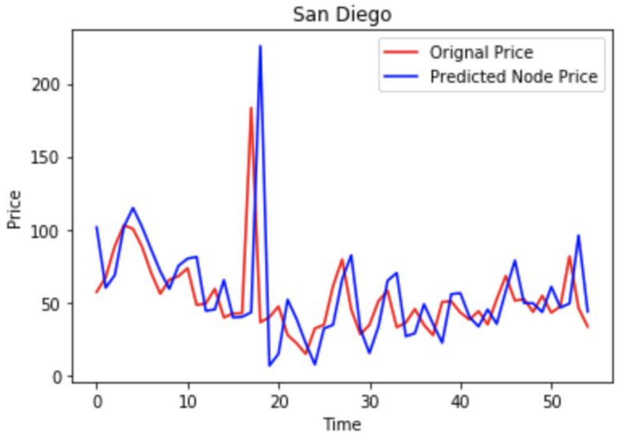
LSTM of Los Angeles



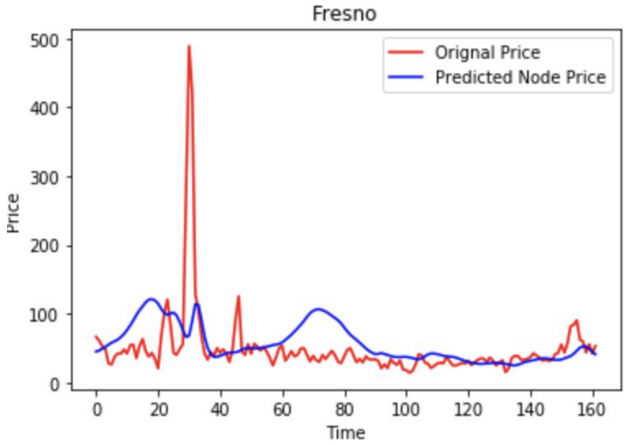
LSTM of Sacramento



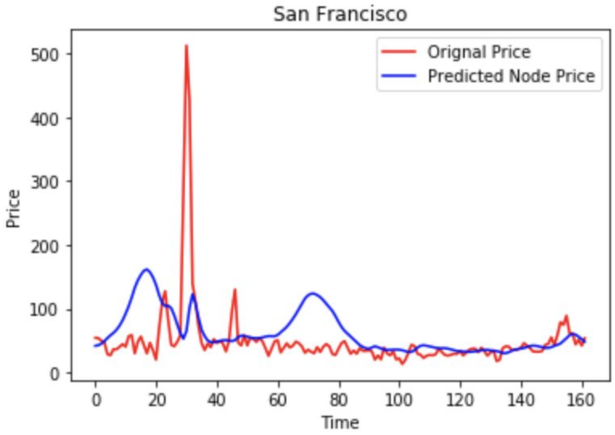
LSTM of Sacramento



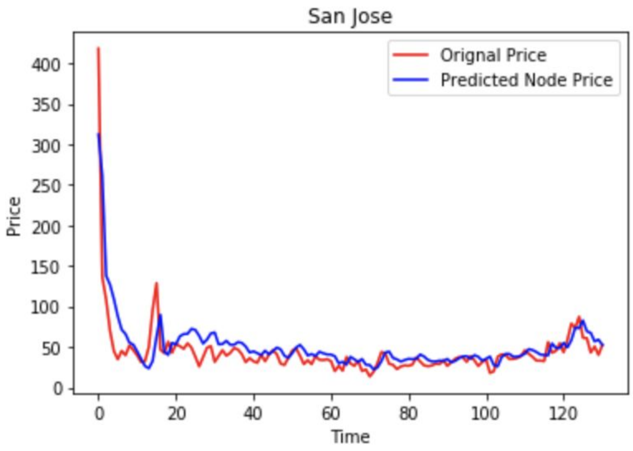
LSTM of Fresno



LSTM of San Francisco



LSTM of San Jose



Conclusion

We’ve observed that our models, specifically ARIMAX and LSTM perform fairly well. Our predictions were close to the actual values and the MSEs/Loss values for each models are listed below -

City	LSTM Loss	ARIMAX MSE	ARIMA MSE
San Francisco	0.0035	0.1284	80.4130
San Jose	0.0034	0.0905	79.8928
Los Angeles	0.0027	0.1569	154.0351
San Diego	0.0033	0.1566	1727.5343
Sacramento	0.0019	0.121	45.7675
Fresno	0.0038	0.1161	75.7081

We can conclude that the LSTM performs better in terms of predicting future electricity price spreads compared to ARIMA. ARIMAX however, leveraging the temperature information per city helps explain variability in the data and is definitely an important factor for prediction. We hypothesize that a multivariate LSTM that leverages exogenous variables like temperature will definitely have a higher predictive power.

Future Work

The following are a few aspects that we can focus on for the future -

- At the moment, we’ve only focused on temperature as the exogenous variable for our modeling. We can extend our analysis to include additional variables like fuel cost, transmission cost etc. This should help explain additional variability in the data.
- Our model only focuses on California. We can scale our algorithm to predict prices all over the United States (subject to availability of pricing data and exogenous variables).
- Exploring price volatility and behavior in the retail and wholesale markets, recognizing the difference between the two which should help give us some insights into how both the markets behave.
- Incorporating data on charging stations to identify how price movements may impact traffic at various charging stations.