Capstone Project 3 – Final Report

Brad Painting
Mentor: Dhiraj Khanna
Springboard 2021

Contents

Introduction

Electricity Market

Demand for electricity is in constant flux. However, the electric grid does not have the ability to store energy, and the supply of power must modulate to meet variations in demand from residential, commercial, and industrial consumers. Too much power requires power plants to disconnect from the grid. More seriously, a shortage of power can result in blackouts that disrupt critical needs.

Many electric markets in the U.S. are deregulated, which means that the public utility manages transmission only, and independent power producers decide whether it is economical to produce power. The decision of how much power to sell is commonly determined by daily actions where bids are placed for the sale of electricity in 1-hour increments for one day in advance[1].

Predicting Demand

There are complex dynamics affecting total electricity demand: randomness in the daily habits of individuals, unpredictable elements of commercial and industrial processes, and uncertainty in weather forecasts all make precise forecasts difficult. Renewable energy is also comprising an increasing percentage of total capacity, which varies with fluctuations in wind speed, solar irradiance, and rainfall. Additionally, utilities can directly influence demand through active load-shedding measures, such as direct load control (DLC) devices in air conditioners and water heaters that restrict their full capacity during peak periods[2].

Balancing Authorities

The free-market auction for electricity production joined with the uncertainties in demand creates a crucial need for short-term electricity forecasts. The task of matching supply and demand is the responsibility of various "balancing authorities" managing different regions throughout the U.S [3].

The Energy Information Administration (EIA) provides hourly data on power generation by source (e.g., nuclear, coal) and total demand reported by balancing authorities. Additionally, they provide the day-ahead demand forecasts which balancing authorities use to manage and maintain a reliable supply of electricity.

Method

This project uses the demand data from EIA to fit to a univariate time-series model and test 24-hour forecasts against actual demand during those periods. An exogenous variable for temperature was then added in attempt to improve accuracy. Models explored include ARIMA / ARIMAX models as well as Facebook Prophet.

The purpose of the model is to forecast demand data only 24 hours in advance. Often in machine learning problems, 20 to 30 percent of the available observations are set aside as a test set and the model is trained on the remaining rows. However, this method does not work well for this problem. The data has over 24,000 1-hour rows and the forecast is for only 24 hours, so it would be only a sliver of the test set. A single train-test split would fail to measure how the model performs on different days of the week and during different seasons of the year, because forecast horizons beyond 24-hours are not relevant to the problem presented by day-ahead auctions for sale of electricity.

I accounted for this variation by splitting the data into 20 different training sets of varying lengths and testing the 24-hour forecast against the actual data in each case.

Exploratory Data Analysis

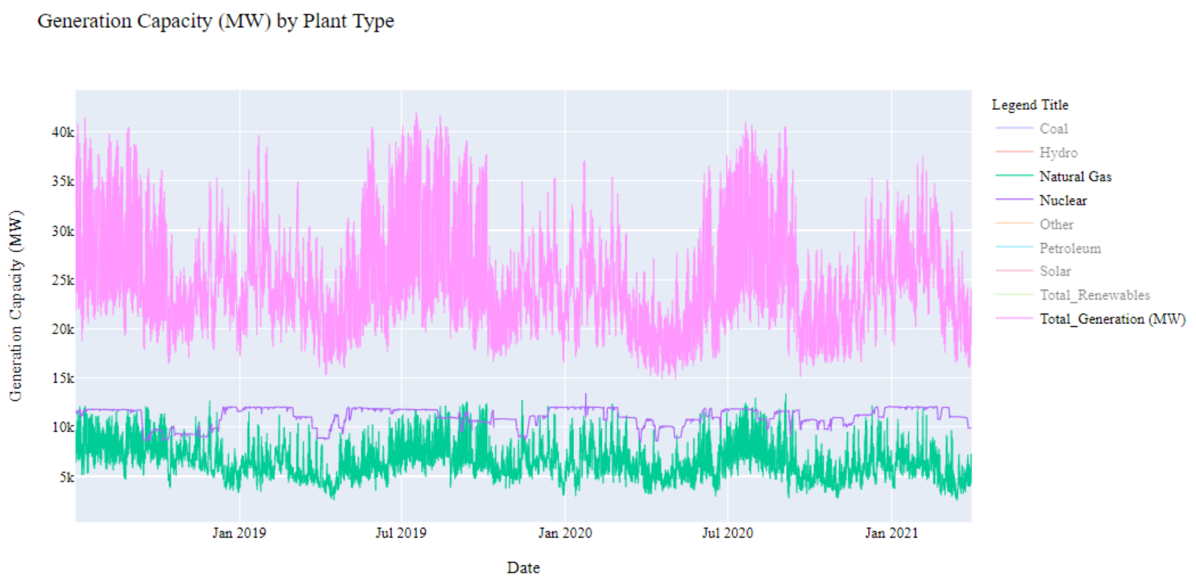
Overview

Total power generation is similar to total power demand, but there are minor differences resulting from interchange between balancing authorities; when there is a shortage or surplus of power, it can sometimes be bought or sold to different regions. I performed EDA on both the supply and demand side to get a general overview of the problem and because I was initially unaware of the interchange data creating discrepancies between supply and demand.

Generation Data

I visualized the generation data with the expectation that patterns in power supply would help demonstrate patterns in demand. Different types of power plants have varying abilities to modulate the amount of electricity generated; nuclear is primarily used for steady baseloads, while natural gas plants are able to ramp up generation to meet peak demand periods. The difference in patterns is shown in Figure 1, created with Plotly.

Figure 1: Power generation time series



Correlations between power sources are shown in Figure 2. I dropped Petroleum from the correlation matrix because it had negligible contribution. Coal and natural gas behave most similarly to each other, while solar and nuclear are mostly uncorrelated with other power sources. However, hydro, solar, and other power sources make up only a small portion of total generation, seen in Figure 3.

Figure 2: Power generation correlation matrix

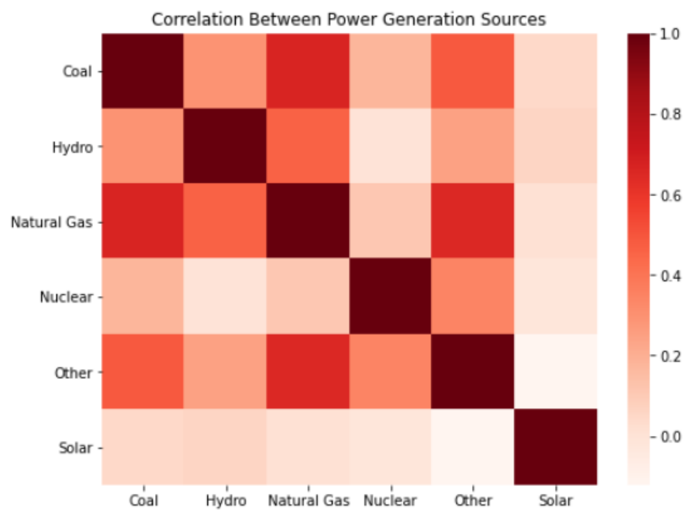
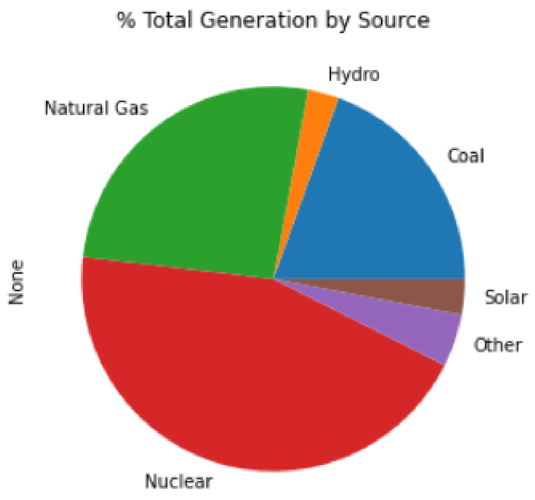


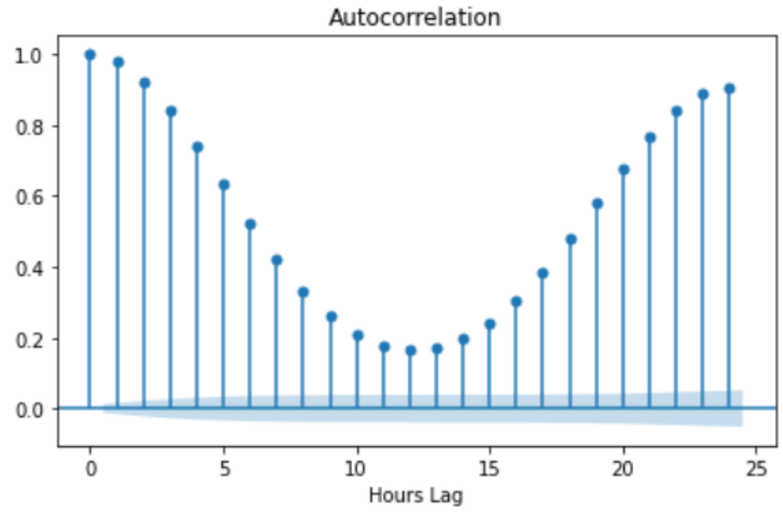
Figure 3: Percent Contribution to Total Power Generation



Demand Data

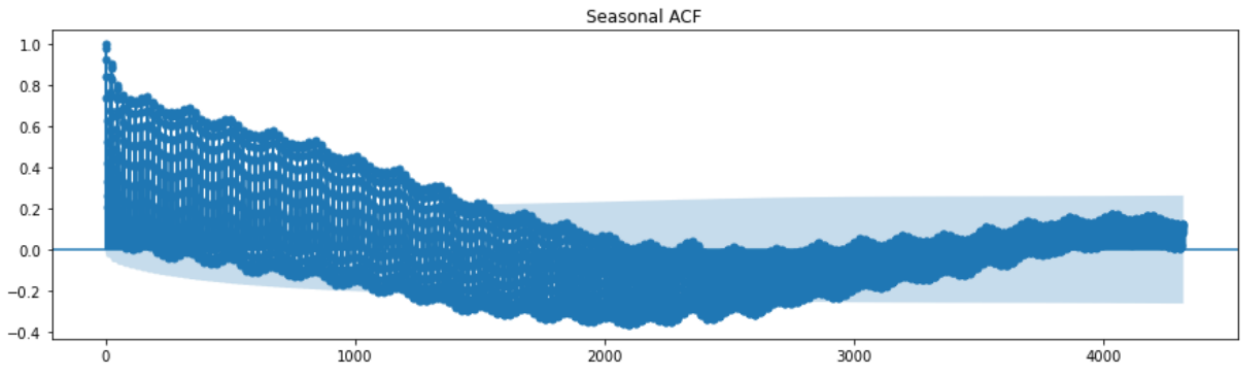
I began investigating the predictive power of ARIMA models on the demand time series by using the statsmodels package to plot autocorrelation and partial autocorrelation.

Autoregressive component – Daily



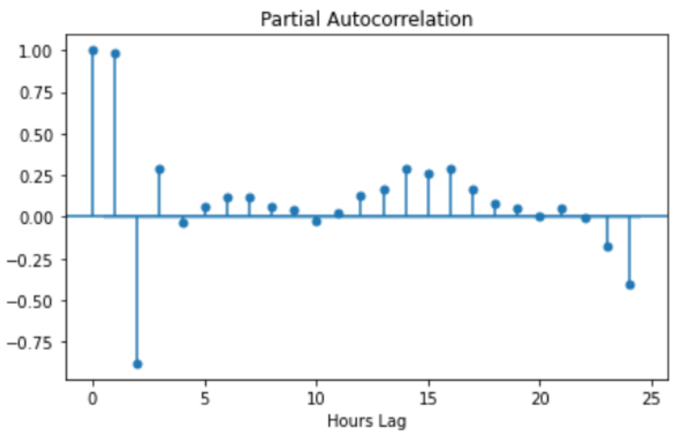
I first inspected over a 24-hour period. The ACF plot is in line with expectations. The power consumption during any hour is highly correlated with the previous hour, which would be explained by the inertia of aggregate daily activities across a population. The power consumption 24 hours away (during the same hour of the next day) is also highly correlated, because daily patterns tend to repeat. The lowest correlation is halfway through the day. This is unsurprising, because power consumption at noon should be significantly different than power consumption at midnight.

Autoregression – Seasonal



To see seasonal effects of autocorrelation, I plotted ACF over 180 days. The long-term ACF plot shows that the series becomes less correlated with the lagged version of itself as the lag approaches around 2000 hours. With 8760 hours per year, this is approximately a quarter of the way through a year, or the time between seasons.

Moving Average



The moving average component of electricity demand is described by a partial autocorrelation function (PACF). This shows that an unexpected increase in electricity demand in the previous hour is predictive of continued increase two hours into the future. The third hour is inversely correlated with the 3-hour lag, and then beyond that, the moving average weakens as a predictor. I am not sure what mechanism is responsible for this, but may be caused by the length and profile of peak demand periods such as when people get home from work.

Data Collection and Wrangling

Generation and Demand Data

I obtained electric power generation and demand data from EIA's Hourly Electric Grid Monitor API. The data portal was in Beta testing when I began this project but moved into production on April 30th 2021. The generation data is provided by production source (such as coal, natural gas, or solar). The forecasted demand and actual demand are single-feature datasets representing the aggregate for the region.

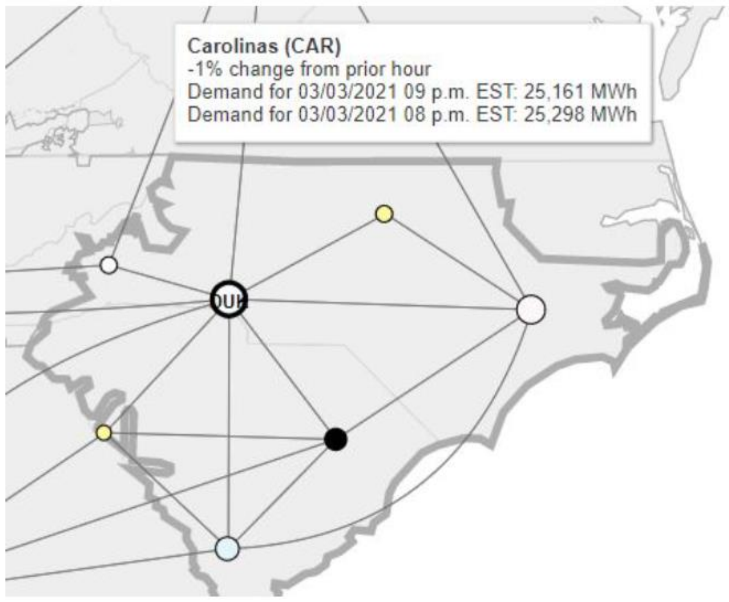
Weather Data

Weather – specifically temperature – is expected to affect demand because of its influence on electric heating and cooling. The demand data is hourly, so I needed a data source with at least hourly resolution on historical temperatures. I used the NCEI (National Centers for Environmental Information) API to pull the data.

Balancing authorities do not know the exact temperatures 24 hours in advance and need to rely on local forecasts. I could not find a source a source for historical 24-hour forecasts, so I incorporated a random error into the historical measurements based on the average error in 1-day ahead forecasts described by Segarra et al. [4].

The region handled by the Carolinas balancing authority is shown in **Error! Reference source not found.**. It is difficult to choose a single representative weather forecast because the region covers two states, and forecasts are for specific locations. I choose to use the average of forecasts for major cities in approximately the center of each state (Raleigh, NC and Columbia, SC).

Figure 4: Carolinas Balancing Authority Regin


Carolinas (CAR)
-1% change from prior hour
Demand for 03/03/2021 09 p.m. EST: 25,161 MWh
Demand for 03/03/2021 08 p.m. EST: 25,298 MWh
DUK

NCEI data included many variables including both wet bulb and dry bulb temperature. I selected dry bulb temperature instead of wet bulb (which incorporates the effect of humidity). A hybrid of dry bulb and wet bulb could possibly work better since air conditioning loads increase with higher humidity, but this is a potential topic for further research.

The main data cleaning required on weather data was dealing with non-numeric data containing an “s” and missing values represented by asterisks. The “s” indicates “suspected value” according to the documentation, so I replaced all occurrences of “s” with a blank string and then converted the column to the numeric data type. I addressed the missing values by replacing the asterisks with np.NaN and then backfilling with the pandas df.fillna() method.

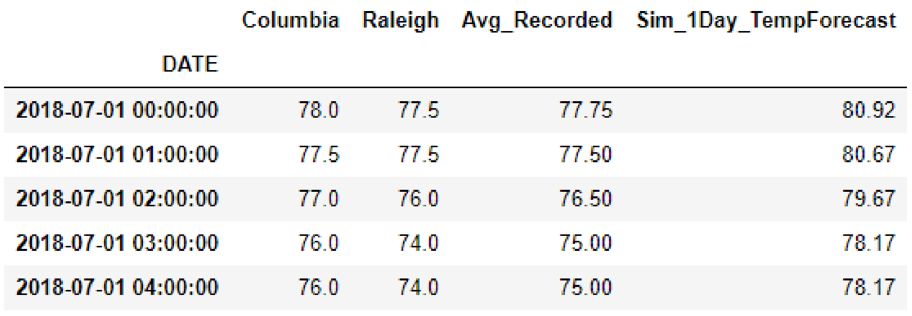
Figure 5: Non-numerical values in weather data

```
2020-03-09 13:56:00 72s
2020-12-28 13:56:00 63s
2021-03-31 17:56:00 *
2021-05-03 15:56:00 *
```

Adjusting for temperature forecast accuracy

The average mean-absolute error of the day-ahead temperature forecast is 1.39 degrees C, as assessed by Segarra et al (2019). Because the NCEI temperature measurements are at irregular intervals, the first step was to resample to 24-hour periods using the pandas resample(‘1D’) method. I converted the Celsius unit to Fahrenheit to match NCEI data and then used np.random.normal(), setting the scale parameter to the mean absolute error of their findings. This generated an array of temperature forecast errors.

These errors were again resampled to 1-hour intervals to line up with the EIA data. I forward filled a few missing values resulting from discrepancies in timeframes of the 1-day and 1-hour resamplings. I then added these errors to the average measured temperature between the two cities to create a simulated day-ahead forecast column.

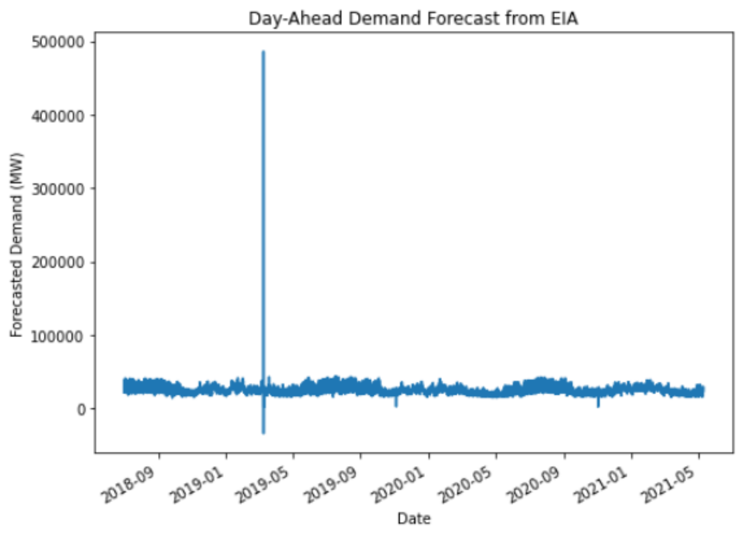


DATE	Columbia	Raleigh	Avg_Recorded	Sim_1Day_TempForecast
2018-07-01 00:00:00	78.0	77.5	77.75	80.92
2018-07-01 01:00:00	77.5	77.5	77.50	80.67
2018-07-01 02:00:00	77.0	76.0	76.50	79.67
2018-07-01 03:00:00	76.0	74.0	75.00	78.17
2018-07-01 04:00:00	76.0	74.0	75.00	78.17

Benchmark Forecasts

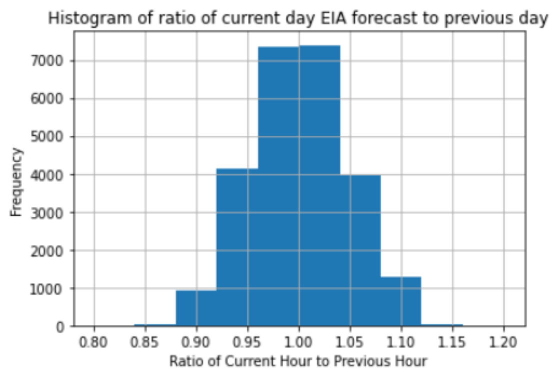
The balancing authority 24-hour forecasts collected by EIA serves as a benchmark for assessing the results of the machine learning models. It is collected through a government survey form (EIA-930) that requests data in CSV or XML format (the balancing authority can also elect to automate their data sharing). Figure 6 shows it contains some data errors visible as outliers.

Figure 6: EIA Demand Forecast Data



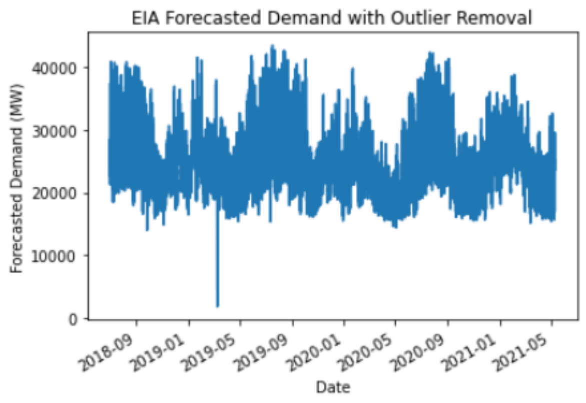
To filter out probable errors, I looked at the distribution of changes in forecasted demand over 1-hour intervals to detect any large spikes. Figure 7 shows that it is rare for the forecasted demand to be less than 0.85 of the previous hour or more than 1.15 of the previous hour. For any values that fell outside this range, I replaced them with the rolling median of the past 24 hours.

Figure 7: Histogram of change in forecasted demand from previous hour



The error-filtered time series is shown in Figure 8. There is still a significant anomaly in April; however, it is less severe and using a tighter window for filtering errors may eliminate good data. Additionally, the machine learning model is not trained on this data and it is only for comparison.

Figure 8: EIA Demand Forecasts after Outlier Removal



Modeling and Testing

SARIMAX Model

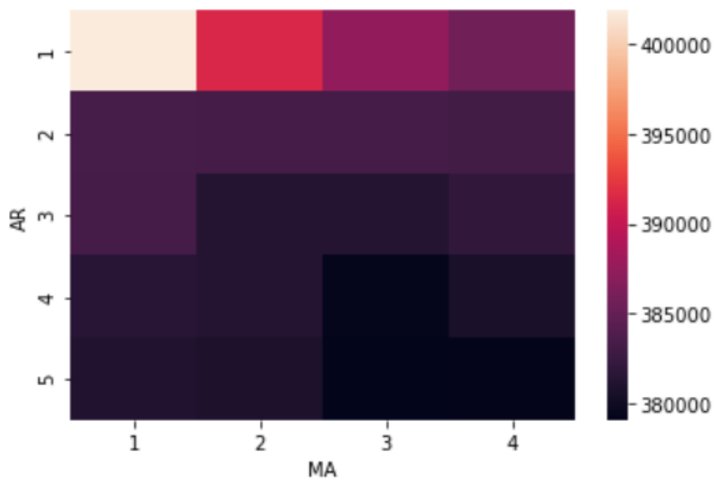
Parameter Tuning

Because of the size of the data set and number of tests required to validate the model, tuning the parameters is time consuming. The SARIMAX model contains the argument order=(p,d,q), where p = number of AR lags to include, d = integration order of the process (to make data stationary), and q = number of MA lags to include. From the EDA, the time series showed both AR and MA components, so these parameters must be specified.

I used the arma_order_select_ic method in the statsmodels package to autofit AR and MA parameters. Including too many lags can cause overfitting, and too few can lose precision. I chose the Bayesian Information Criterion (‘bic’) argument for the output reported, in which the number of parameters that result in the lowest BIC is generally best.

Figure 9 shows a heatmap of the different BIC values for different combinations of AR parameters. It appears that an AR of 4 and MA of 3 may be optimal. These values seem reasonable based on the ACF and PACF graphs, which showed that autocorrelation coefficient was above 0.8 for up to 4 lags, and that the PACF coefficient was high (in positive or negative direction) for up to 3 lags, with a sharp drop-off thereafter.

Figure 9: Heatmap of BIC values for ARMA parameters



Stationarity

The integrated "I" parameter of the ARIMA model accounts for differencing required to make the data stationary. I inspected stationary using the statsmodels KPSS ("Kwiatkowski-Phillips-Schmidt-Shin") test. The null hypothesis is that the data is stationary. I used a p-value of 0.05 as the threshold for rejecting the null hypothesis. Without differencing, KPSS returned p = 0.01, meaning the data is probably nonstationary. After a single order of differencing, KPSS returned p = 0.1, which means the null hypothesis cannot be rejected for the differenced time series and the integrated parameter should use d=1.

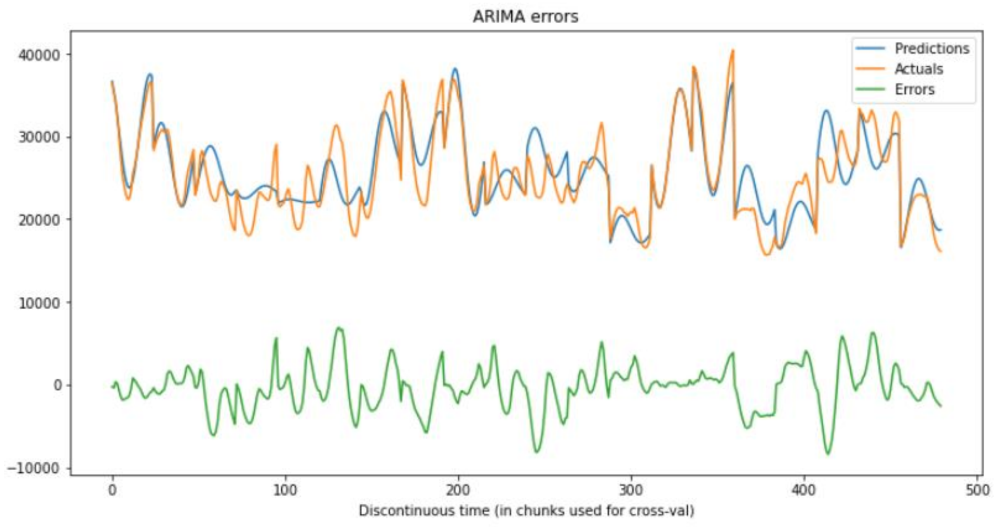
hypothesis cannot be rejected for the differenced time series hypothesis cannot be rejected for the differenced time series hypothesis cannot be rejected for the differenced time series this is a test on whether

Model Results

ARIMA Model

The (4,1,3) order ARIMA model was tested by making 20 separate day-ahead forecasts and comparing to actual demand data, with the results displayed in Figure 10.

Figure 10: ARIMA forecast compare to actual demand

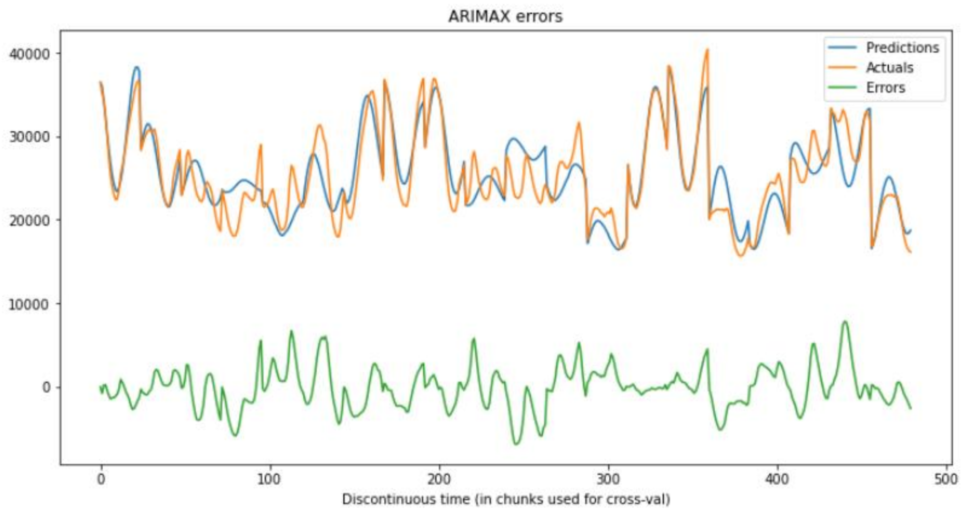


MAPE = 0.08363699443588107

ARIMAX (ARIMA with Exogenous Variable)

The addition of temperature data provided a slight improvement over the base ARIMA model. An additional step could be to look at incorporating a seasonal order argument into the SARIMAX model. However, this could be redundant because the exogenous temperature variable is highly correlated with season.

Figure 11: ARIMAX forecast compared to actual demand

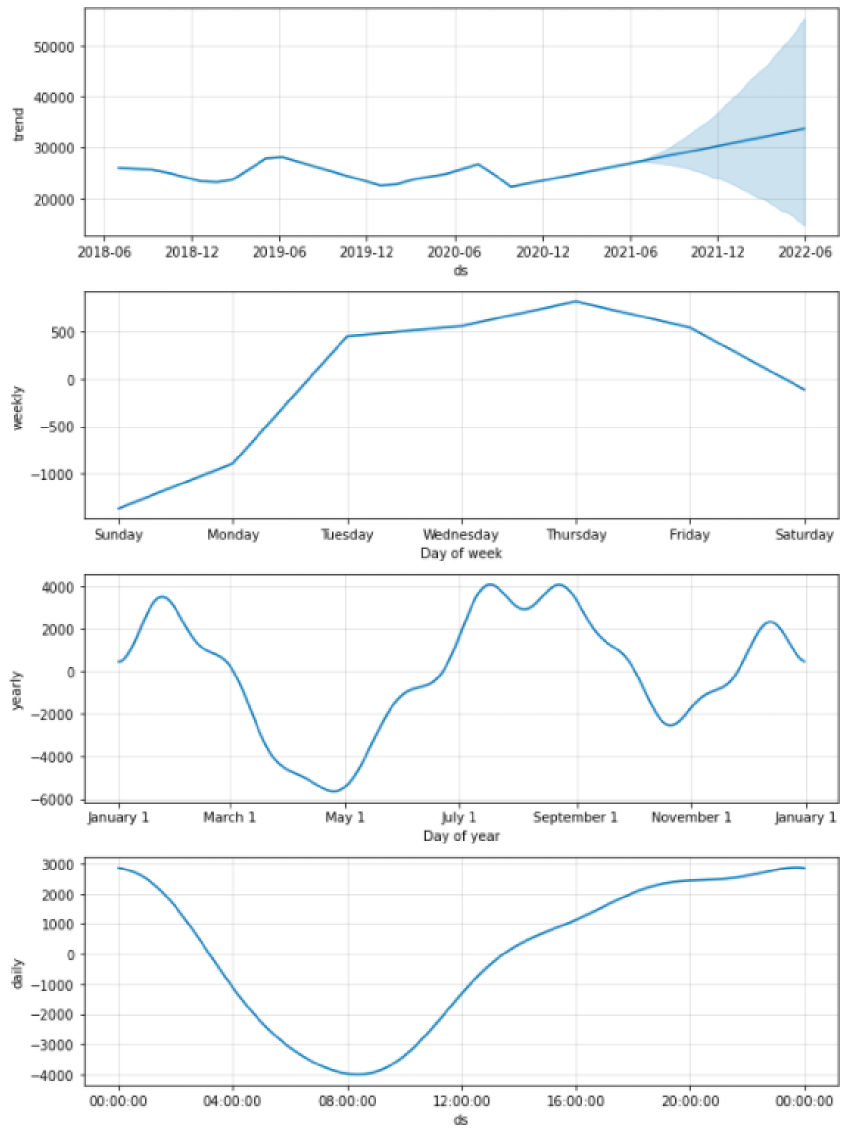


MAPE = 0.07813559302193628

Facebook Prophet Model

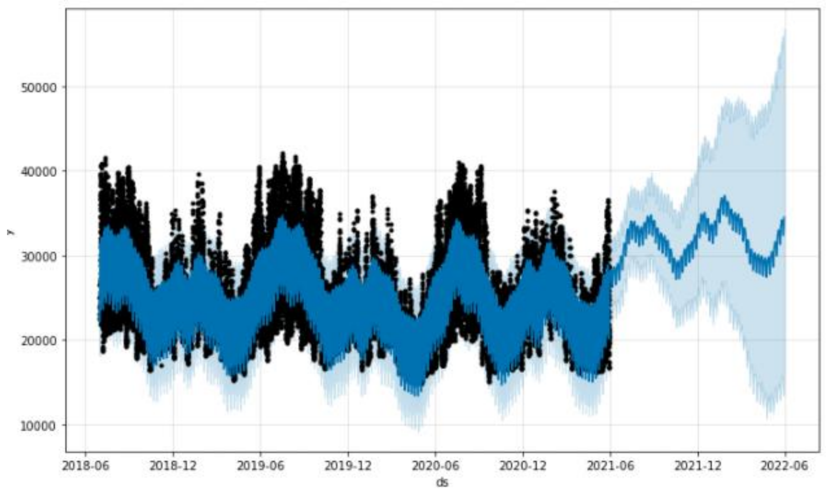
I looked at using the Facebook Prophet Model as an alternative to ARIMA. Contrary to ARIMA, Prophet uses an additive regressive model in which the time series is broken into trends of different periodicity which are then added together, such as daily, weekly, yearly, and long-term trend. One advantage over ARIMA is that Prophet can support holiday effects, which are likely relevant to electricity consumption. The additive components of the Prophet model for the electricity data are shown in Figure 12.

Figure 12: Facebook Prophet Model Additive Components



Unfortunately, I was unable to easily fit the additive model to the training data. The initial fit using the default parameters is shown in Figure 13.

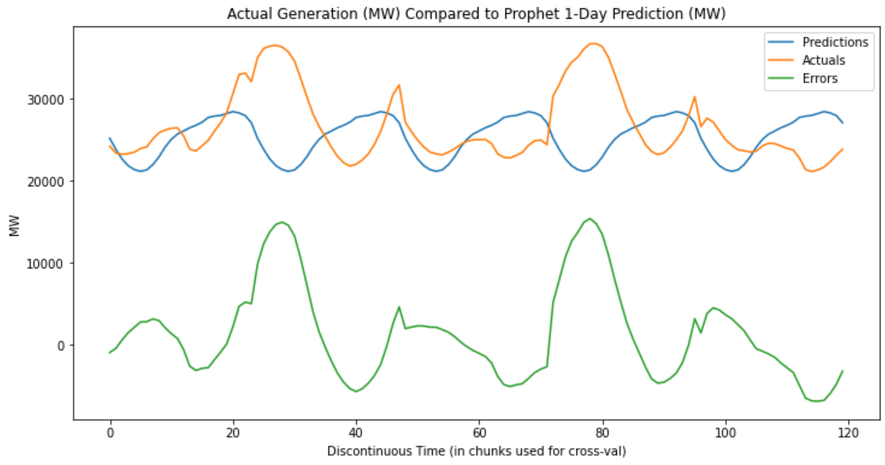
Figure 13: Facebook Prophet Model Fitting



An additional constraint is that the computation time for fitting and forecasting was much longer for the Prophet model. In attempt to make a fair comparison to the ARIMA / ARIMAX models, I added the temperature data to the model as an additional regressor and ran 5 individual day-ahead electricity forecasts (I used only 5 for Prophet versus 20 for ARIMAX due to the computation time).

The results of the forecasts, shown in Figure 14, were not encouraging with a MAPE of 15.4%, so I stuck with the ARIMAX model.

Figure 14: Prophet Model Forecast Errors



Model Issues

I encountered two types of errors when using the statsmodels SARIMAX model. The first was an "LU Decomposition Error" that aborted the loop during the forecasting. The second type did not produce an explicit Python error, but generated extreme negative values in the forecasts that were at least an order of magnitude off from any historical demand data.

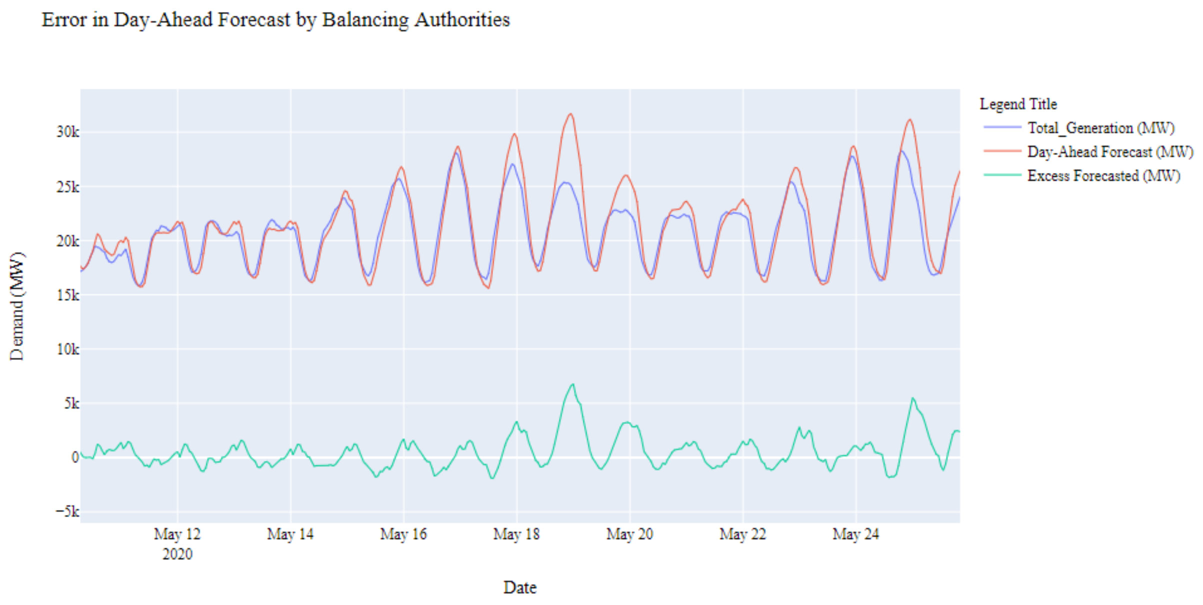
I did not track the random seed number when the negative-value error occurred and am unable to replicate it. However, I was able to repeat the "LU decomposition error" by setting the random seed to 38. The error is discussed in the statsmodels Github as a potential issue with the package[5]. I circumvented the error by adding an "enforce_stationarity=False" argument to the instantiation of the SARIMAX model, as described in Github and suggested by my mentor. The model ran smoothly with the added argument and the SARIMAX models continued to work well when I changed the random seed to 47.

I also experimented with the "ARIMA" model in the statsmodels module to compare to the more flexible "SARIMAX" from statsmodels to make sure the latter package was not adding any disfunction to the model fitting and forecasting. The models behaved identically when I input the same parameters to each package and used the same random seed.

Conclusions

The ultimate goal of this project was to determine if a machine learning model could approach or exceed the accuracy of electricity forecasts collected by balancing authorities. As described under "Benchmark Forecasts", I prepared this data for comparison to the actual demand data. A typical ~2-week period is shown below.

Figure 15: Balancing Authority Forecast vs Actual Demand



The overall MAPE for the EIA forecasts is 5.09% versus a MAPE of 7.81% using the ARIMAX model. However, as described in the data cleaning, I left some anomalous values in the EIA forecast dataset due to difficulty of cleaning and uncertainty about where the line is between data errors and bad forecasts. The existence of anomalies and/or erroneous data in the EIA dataset can also be an argument for supplementing the survey of balancing authorities with machine learning.

Another qualification is that I am not familiar with the exact process for day-ahead electricity auctions. My models assume that there is no gap between the historical data and the next 24 hours; the actual demand during the current hour can be used towards forecasting data in the next hour and the 23 hours thereafter. In reality there is likely some lag in waiting for recent power data to become available before it can be used for forecasting.

Overall, the ARIMAX model is a promising supplement to the current forecasting process. Machine learning may help identify data errors from balancing authorities and can possibly augment other forecasting methods. The models in this project could possibly be made more accurate by narrowing the electricity forecasts and weather data to more localized regions, as well as spending more time tuning parameters.

Sources

[1] Edwards, D., 2009. *Energy Trading and Investing; Trading, Risk Management, and Structuring Deals in Energy Markets*. 1st ed. McGraw-Hill Education, p.8.

[2] Demand Load Management: https://www.power-grid.com/smart-grid/new-wave-of-direct-load-control-update-on-dlc-systems-technology/

[3] Roles of EIA: https://www.eia.gov/beta/electricity/gridmonitor/about

[4] Accuracy of Weather Forecasts: https://doi.org/10.3390/en12071309).

[5] SARIMAX Error: https://github.com/statsmodels/statsmodels/issues/5459

[6] Source of Data:
https://www.eia.gov/electricity/gridmonitor/dashboard/electric_overview/US48/US48