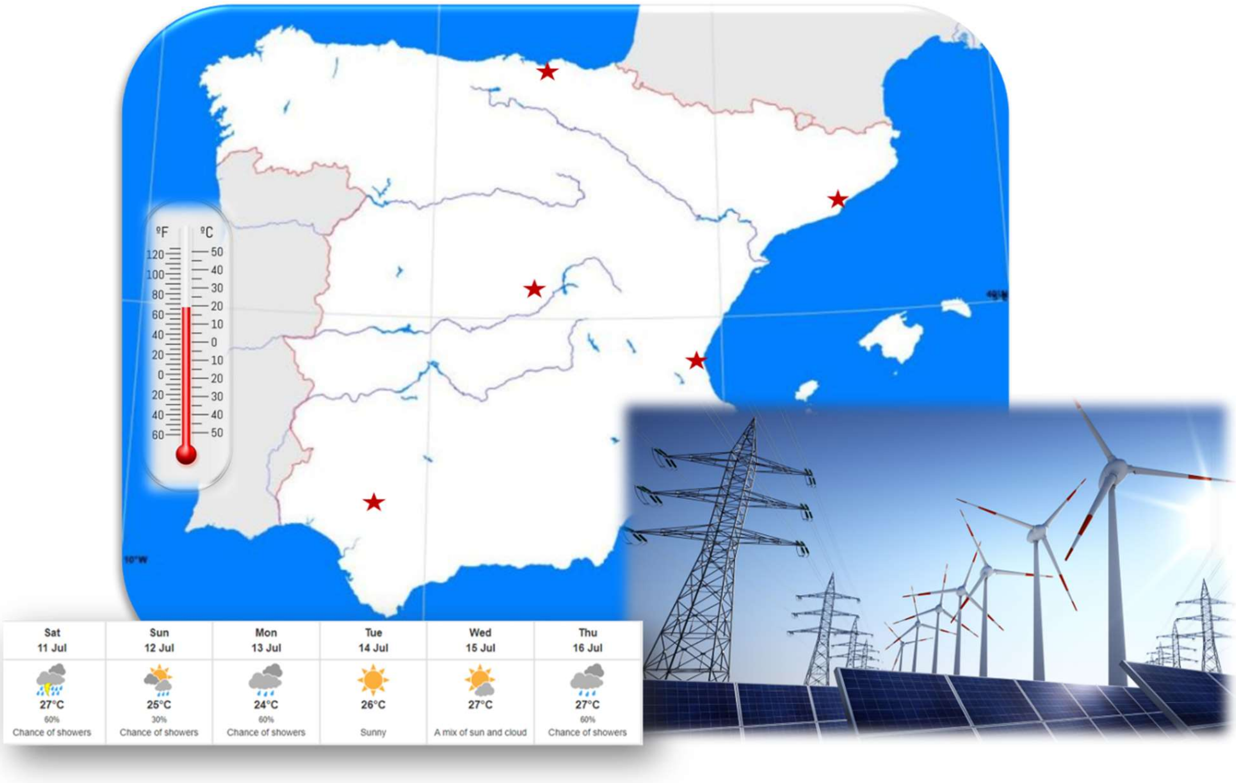
Elizabeth Hardwick, Capstone Project Final Report:

Electricity Demand Forecasting



Synopsis

I attempted to forecast hourly electricity demand approximately seven months in advance using four total years of hourly electricity demand data for all of Spain and weather data for five major cities in Spain. Using FB Prophet I was able to forecast demand with almost 93% accuracy on average for the seven month forecast window.

Problem

Accurate power generation forecasts, both short term and long term, are critical to a green energy future. Accurate forecasts enable renewable generators to correctly price their energy and maximize contribution from renewable sources (while minimizing carbon sources), and are important to operators for system stability and planning. By being able to forecast well in advance how much electricity demand there will be in a given period of time, the system operator can determine how much of that demand can be met by solar, wind, and other renewable sources and how much and when extra fossil generation will be needed to fill any gaps between the base generation amounts and peak demands.

I tried numerous models attempting to forecast hourly demand approximately seven months in advance using four total years of hourly electricity demand data for all of Spain and weather data for five major cities in Spain. Models tested include simple linear regression, Random Forest, KNN, SARIMAX, FB Prophet, and XGBoost.

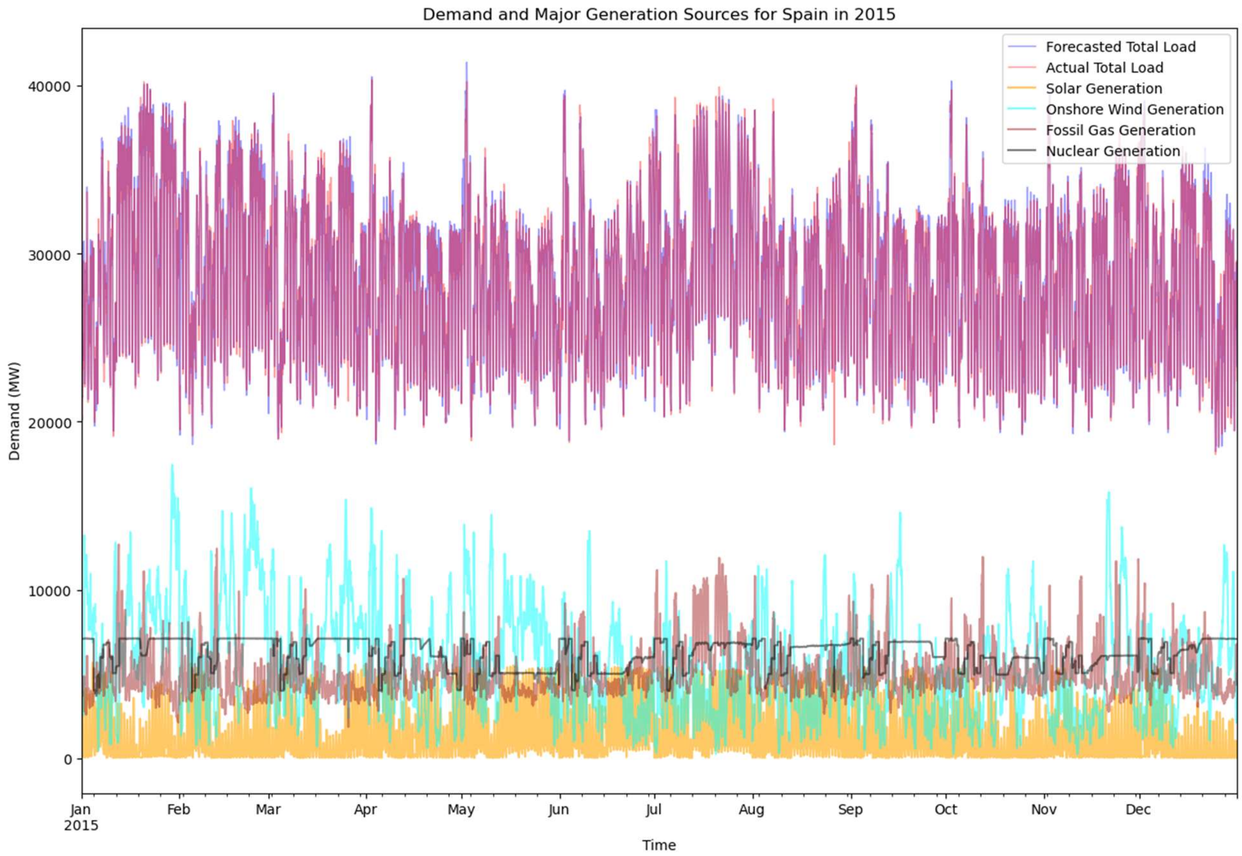
Data Wrangling

The datasets I used for the forecast were obtained from Kaggle (sourced from Open Weather, ENTSOE transparency platform, and Red Electrica). The energy dataset contained 35064 hourly records with electricity demand, predicted demand, price, predicted price, as well as generation totals by method (solar, fossil gas, fossil coal, wind, etc.) and the weather dataset contained hourly records with temperature, wind speed, humidity, rainfall, snowfall, and other weather descriptor variables for each of five geographically diverse cities in Spain.

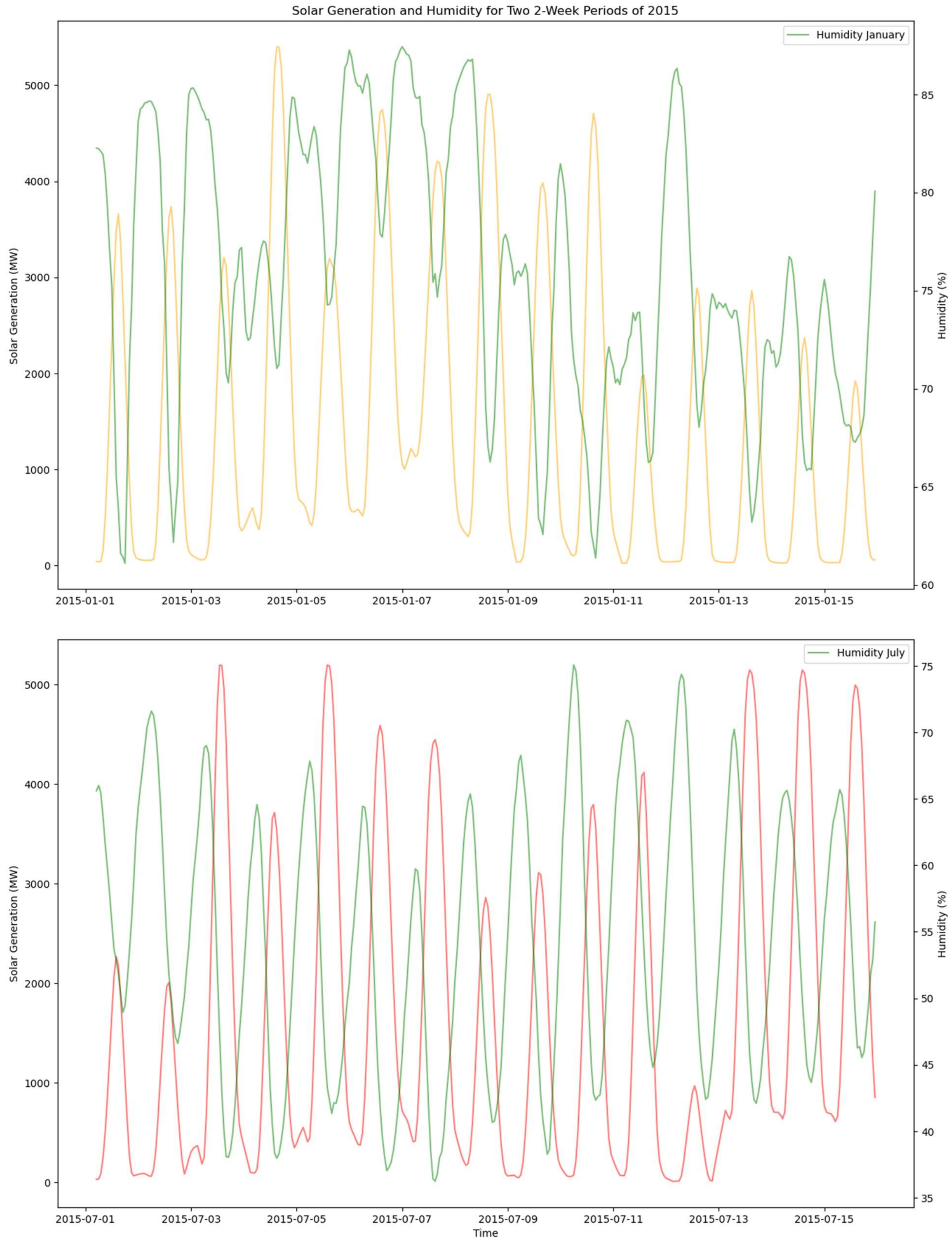
I found and deleted duplicate data, interpolated some missing values in the energy data (preserving continuous hourly data was important), deleted some entire columns that contained lots of missing data, averaged the weather data for the five cities to have only one of each weather variable per record, indexed everything by time, and then merged the weather and energy datasets into one.

Exploratory Data Analysis

One of the most enlightening visuals I created during EDA plotted a time series of one year of demand and also showed some of the major generation sources (solar, wind, fossil gas, and nuclear). This plot clearly illustrated the weekly seasonality with demand being higher during the week and lower on the weekend, as well as the winter/spring/summer/fall seasonality where demand is highest in winter and summer when more electricity is needed for heating and cooling, respectively.

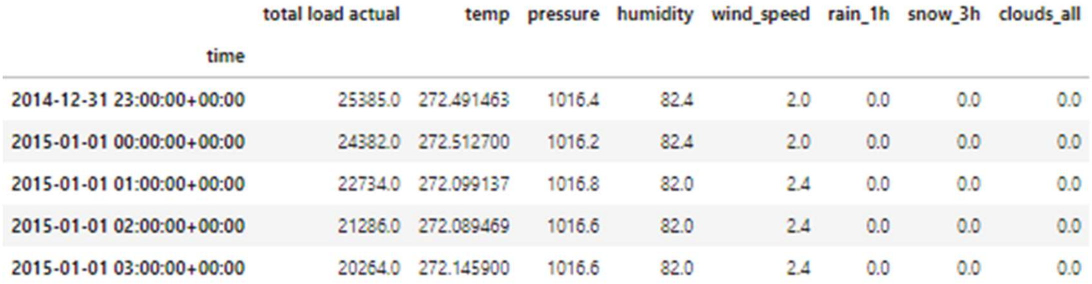


The EDA process also helped with some outlier detection such as erroneously high atmospheric pressures, which I imputed with the median, and some records with zero fossil and nuclear generation which were also likely erroneous. I also learned that humidity is an excellent predictor of solar generation. On a daily basis, humidity is high at night when there is no solar generation and low during the day when solar generation is occurring. In the absence of a solar radiation measurement, humidity is an excellent predictor, at least in Spain's climate.


Solar Generation and Humidity for Two 2-Week Periods of 2015

Feature Engineering

The base dataset with which I initiated modeling included time-indexed: total load (demand (MWH), temperature (K), pressure (mb), humidity (%), wind speed (km/hr), rain, snow, and cloud percents.



time	total load actual	temp	pressure	humidity	wind_speed	rain_1h	snow_3h	clouds_all
2014-12-31 23:00:00+00:00	25385.0	272.491463	1016.4	82.4	2.0	0.0	0.0	0.0
2015-01-01 00:00:00+00:00	24382.0	272.512700	1016.2	82.4	2.0	0.0	0.0	0.0
2015-01-01 01:00:00+00:00	22734.0	272.099137	1016.8	82.0	2.4	0.0	0.0	0.0
2015-01-01 02:00:00+00:00	21286.0	272.089469	1016.6	82.0	2.4	0.0	0.0	0.0
2015-01-01 03:00:00+00:00	20264.0	272.145900	1016.6	82.0	2.4	0.0	0.0	0.0

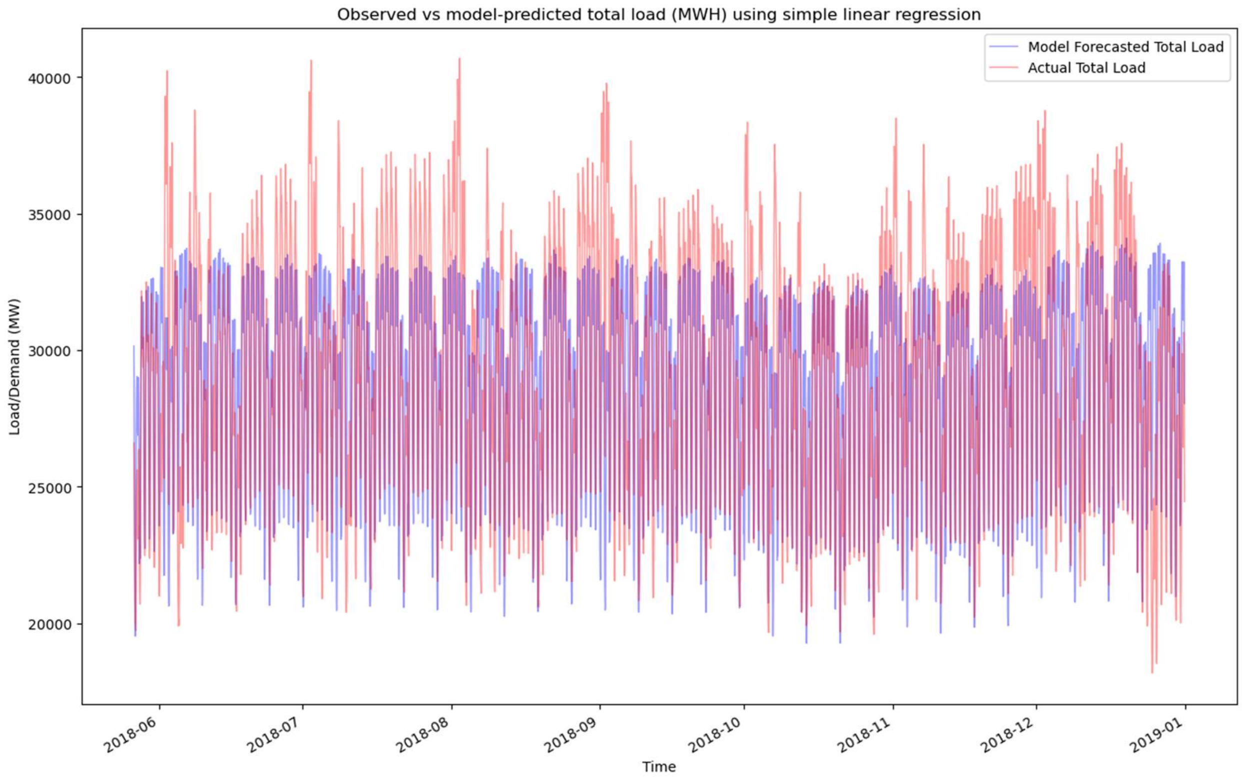
However, there were some useful time-based features I created for regression-based models. I used one-hot encoding to create binary categorical variables for the hour of the time (1-24), the day of the week, work/non-work day, and season (winter/spring/summer/fall).

I also created Fourier series terms for the multiple seasonal periods to use in the SARIMAX model to try to help it handle the multiple seasonalities, since it can only handle one seasonality. I used the FourierFeatures class of the sktime library to do this.

Modeling

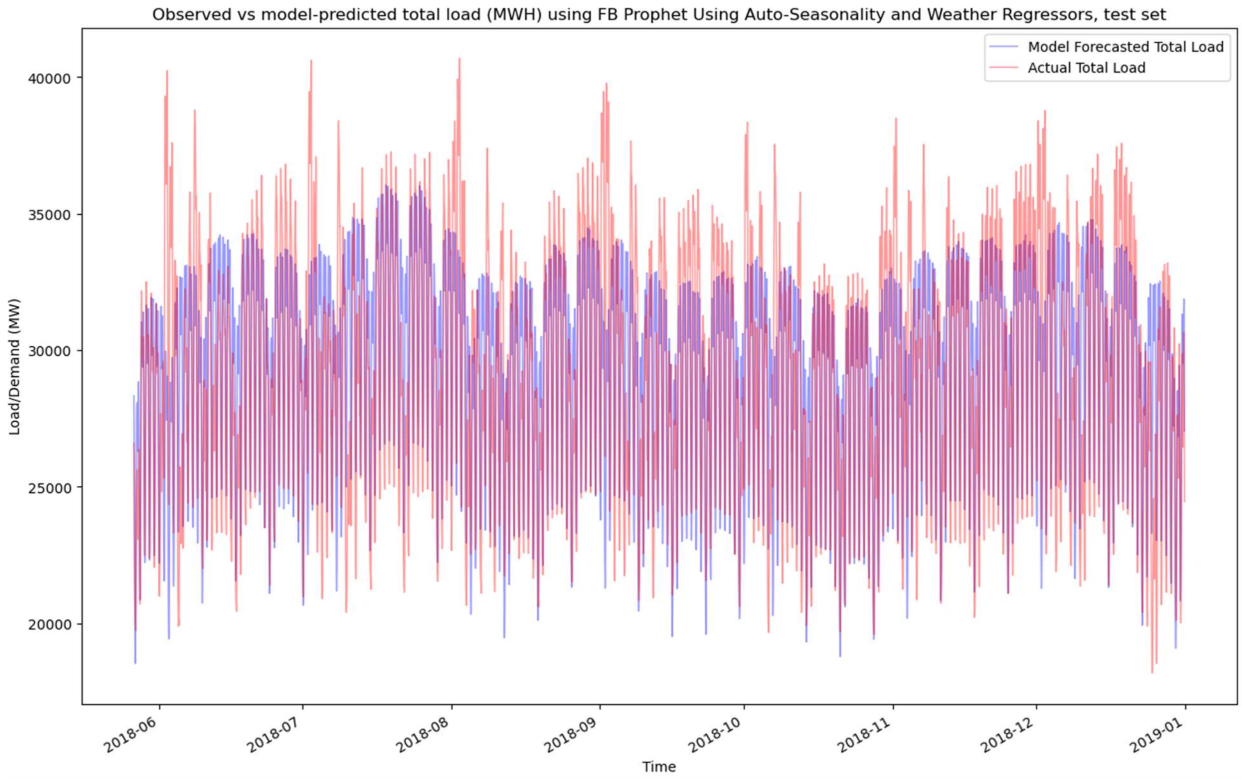
I first tried out a simple linear regression and a baseline year-over-year (just repeating the demand from the same time of the previous year). Neither of these performed very well but they provided a good baseline to compare other models to. I tried the linear regression on the larger dataset with all the engineered features and also a reduced dataset that only included work day vs non-work day and season features. The set with more features performed better.

I then tried Random Forest and KNN models, with both tuned using 5-fold Random Search cross validation. Random Forest performed worse than linear regression and KNN about the same as linear regression.



Next, I dove into classical time series forecasting using statsmodels SARIMAX to see how it would fair. SARIMAX is only capable of handling a single seasonality, and does better when that seasonality is short with respect to the data frequency, so I had the model handle the daily seasonality. But since the data also have strong weekly and yearly seasonality, this model is only useful for short term forecasting. I had hoped that the Fourier transform features for the other two seasonalities fed into the model as exogenous variables would help but it doesn't seem like they did.

Prophet is a model developed by Facebook for time series forecasting that handles multiple seasonalities well and works best with time series having strong seasonal effects as long as several seasons of training data are available. It can also take in exogenous variables like the weather data in this case. Since Prophet is made for time series, I omitted the time-based engineered features and used only the basic weather variables as input. Prophet performed quite well on both train and test sets (~2700 MWH for RMSE and 7.2% error on average).



Finally, I also tested XGBoost, which is a gradient boosted trees-based model. XGBoost is good at finding patterns in data (seasonal times series data being very patterned), but cannot really extrapolate. However, since this dataset has essentially no trend, extrapolating is not necessary and it performed quite well. I tuned the hyperparameters with 5-fold RandomSearchCV and it predicted the seven month forecast window almost on par with Prophet with an average RMSE of about 2800 MWH and just over 7% error on average. Keep in mind that although XGBoost performed as well as Prophet on this dataset, if the energy demand was increasing or decreasing on any significant scale, we'd definitely want to use Prophet rather than XGBoost, as it is able to extrapolate.

Conclusions

Take Home:

Prophet and XGBoost are both good models for long term energy demand forecasting. They both capture the general pattern of the three seasonalities quite well but tend to underpredict peak demands, at least with this dataset. They would be good models to use for general long-term planning of the right energy generation mix, unless there is an overall trend to the data, in which case Prophet should be used. These models both have quick processing times that are about two orders of magnitude faster than SARIMAX. For forecasts of several hours to a few days ahead, SARIMAX does a good job as long as there are no anomalous events that cause demand to spike or drop.

Model Name	Test or Train	Forecast Window	RMSE (MWH)	MAE (MWH)	MAPE
Prophet	Train		2666	2003	7.15
	Test	7-month	2750	2069	7.23
	Test	6 days	2386	1655	5.05
XGBoost	Train		2536	1871	6.69
	Test	7-month	2820	2077	7.11
SARIMA	Train		366	245	0.95
	Test	6 days	2006	1720	6.51
Simple Linear Regression (full feature space)	Train		2861	2203	764
	Test	7-month	2850	2175	7.55
KNN	Test	7-month	3004	2188	7.44
Baseline Year-over-Year		7-month	3101	2249	na

Reflections:

As an example of such an anomalous event as mentioned above, when I plotted the forecast for the first week of the test set for SARIMAX and Prophet, both models grossly underpredict on the 7th day, which is a Saturday. Typically, the weekends have a lower demand than weekdays, so I did some digging and found out that Pedro Sánchez was sworn in as Spain’s new prime minister on Saturday June 2, 2018. Perhaps the whole country was at home or the bar (indoors at least) watching the swearing in on television and power demand was much higher than normal for a Saturday. This goes to show how a single event can throw off a model's prediction. It would be difficult for almost any model to handle something like this without a human manually telling the model to treat this day differently in some way.

Further Investigation:

Due to time constraints, there were only so many model tweaks I could compare, but there are a few worth noting that I have not yet investigated. Since Prophet and XGBoost ended up being the most promising models, it would be worth feeding both different versions of training datasets with more or fewer engineered features to see how that changes performance.

Other avenues with this data that would be interesting to investigate would be to predict the contribution of wind-generated energy given the wind speed and energy price given wind generation, solar generation, and total demand.

Energy forecasting is an interesting and important topic in machine learning and although I’ve not even scratched the surface with this project, I’m keen to dive deeper and learn more in the future!