

Problem Statement

The Energy Information Administration (EIA) has a plethora of hourly electricity demand data for different regions in the lower 48 states. Energy companies rely on accurate forecasts to predict upcoming demand. The goal of this project is to train a machine learning model on weather/electricity data to generate a more nuanced energy-demand forecast. The client for this type of project would be any energy company looking to better understand how the demand for their electricity varies hour to hour. An improved forecast helps energy companies take next steps to fully capitalize on and perhaps better brace for large spikes in electricity demand. Additionally an improved predictive model will shed insight on the features that most heavily affect electricity demand.

Data Cleaning and Wrangling

My project consists of primarily two data sets: (1) electricity data from the EIA, i.e., electricity demand, and (2) hourly, local climatological data (LCD) [downloaded from the National Oceanic and Atmospheric Administration \(NOAA\) website](#). To limit the scope of the project, I retrieved both data sets from the city of Los Angeles and the city of Seattle (specifically, Los Angeles Department of Water and Power data from EIA and the Los Angeles Downtown LCD data and Seattle City and Light from EIA and Seattle Tacoma International Airport LCD data). Electricity data were retrieved using EIA's API and then unpacked into a pandas `DataFrame`. Since LCD data are not available via NOAA's API, I manually downloaded LCD data from the NOAA website as a CSV file which I imported to a pandas `DataFrame`. Electricity data contain hourly entries from July 2015 to present. To align the data correctly, I downloaded the LCD data from July 2015 to September 2018 and later cut the electricity data to fit the date range.

The electricity data required some cleaning--there were very few null values in the set, however there were a very small number of outliers where data was either entered in as zero or a large negative number. Removing outliers cut only $\sim .01\%$ of the data. LCD data on the other hand required more cleansing. Throughout the cleaning of the LCD data, I frequently referred to the [LCD documentation](#) for guidance on how to treat the features. Firstly, all features (except daily cooling/heating degrees) that did not record hourly entries, i.e., monthly and daily records, were dropped. I kept hourly heating and cooling degrees because they are very relevant to energy consumption. I used `bfill` to fill the the hourly entries of a specific day with that day's heating/cooling degrees. Additionally I added hourly heating and cooling degrees to get higher granularity on a metric which is relevant to energy consumption. The hourly heating/cooling degree days are the number of degrees below/above 65 degrees Fahrenheit (negative values are set to zero) and represents the amounting heating/cooling necessary for that temperature. I dropped a number of hourly columns with excessive null entries compared to the size of the data set and those with low predictive power. I dropped hourly pressure tendency and change because we of excess null values and because measures of pressure were already present, hourly wind direction and gust speed for the same reasons, and hourly present weather type because of excess null values and the dispersity of possible values, indicating low predictive power.

One of the main challenges in cleaning the LCD data was that there were in some cases multiple entries for the same hour. I wanted to have just one entry per hour such that I could eventually align LCD data with the hourly entries in the electricity data. I wrote a function that goes through the weather data and for each instance where there are multiple hourly entries, I keep the hourly entry closest to the hour and drop the remaining rows. For instance, if there are entries at 11:05, 11:36, and 11:56, I kept the entry corresponding to 11:05 and dropped the

other rows. I performed the cleaning this way because the data closest to the hour is most representative for that hour. I then renamed the date such that the minute column is set to 00 to once again align with the electricity data. Either way, the values for multiple, per-hour entries are very similar, so the choice of which entry to keep doesn't make a huge difference.

The hourly sky condition feature required additional cleaning. The hourly sky condition feature contains information about each cloud layer (up to three) using six different categories to describe the cloud coverage. As stated in the documentation, the full state of the sky can be best determined by the category of the last (i.e., highest) layer. With this in mind, I replaced the sky condition feature with the condition of the upper most layer to make the feature consistent. This feature also included some information about the cloud layer height which I excluded because these data were only present for a small number of entries. Finally, there were a number of features of mixed float and string types. After looking at their unique values, I saw that some values were string and floats arbitrarily. I simply converted strings to floats to make the features consistent.

Next we deal with null values and outliers. To determine whether to fill null values with median of `ffill`, I plotted histograms, violin plots, and printed stats for each feature. These plots can be found [here](#) (for LA) and [here](#) (for Seattle). Features where the median was close to the mean suggests few outliers and that the median is an appropriate statistic to fill null entries. Conversely, null values in features with outliers present were filled using `ffill`, assuming that previous values have some predictive value for subsequent values since we are dealing with time-stream data. For outliers, because there was not a single feature where the presence of outliers significantly shifted the median from the mean and because the data is such high granularity, outliers were kept without further cleaning. Lastly, we cut both data sets to fit the same date range as mentioned above, and combine the electricity and weather data into a

single `DataFrame` on date index. Tables of summary statistics for both Seattle and LA can be seen in **Figure 1**.

To obtain more training data, I performed the same analysis described above with weather and electricity data from Seattle. With both data sets (LA and Seattle), we are left with ~54K data points from July 2015 to September 2018 to perform statistical analysis and predictive modeling. The code for this process can be found [here](#).

	mean	median	std
demand	3317.094400	3191.00	785.756576
hourlyrelativehumidity	62.482905	66.00	20.829641
hourlydewpointtempf	51.144188	54.00	11.886757
hourlydrybulbtempf	66.454703	66.00	9.638729
hourlywetbulbtempf	58.249122	59.00	7.749118
hourlyheatingdegrees	7.371355	6.00	6.932305
dailycoolingdegreedays	4.866420	2.00	5.595377
dailyheatingdegreedays	1.971355	0.00	3.554354
hourlywindspeed	1.623656	0.00	2.415391
hourlyvisibility	9.220403	10.00	1.796501
hourlycoolingdegrees	0.651968	0.00	1.617127
hourlyaltimetersetting	29.967976	29.96	0.110782
hourlystationpressure	29.772348	29.76	0.108847
hourlysealevelpressure	29.964036	29.95	0.106616
hourlyprecip	0.000552	0.00	0.007481

	mean	median	std
demand	1110.918797	1108.00	202.371059
hourlyrelativehumidity	71.809753	75.00	17.561429
hourlydrybulbtempf	54.625480	54.00	12.272807
dailyheatingdegreedays	11.074687	10.00	9.714834
hourlywetbulbtempf	49.385309	50.00	9.047850
hourlyheatingdegrees	15.641228	15.00	8.983962
hourlydewpointtempf	44.533922	46.00	8.816830
hourlywindspeed	7.910760	7.00	4.465327
dailycoolingdegreedays	1.271269	0.00	3.071634
hourlyvisibility	9.466663	10.00	1.814851
hourlycoolingdegrees	0.040656	0.00	0.330297
hourlystationpressure	29.564245	29.58	0.206405
hourlyaltimetersetting	30.031201	30.04	0.191500
hourlysealevelpressure	30.057155	30.06	0.189585
hourlyprecip	0.002953	0.00	0.013927

Figure 1. Summary statistics for all the features in the data set for both LA (top) and Seattle (bottom). As discussed, since the median is not too far away from the mean for almost

all of these features, filling null values with either the median of `ffill` is an appropriate procedure.

Exploratory Data Analysis

Here we perform some initial exploratory data analysis (EDA) for both the Seattle and Los Angeles weather/electricity data sets. One of the first questions we ask is what is the relationship between the independent variables (the weather features) and the dependent variable (electricity demand measured in MWh). Pandas has a convenient method (`.corr()`) for finding the Pearson correlation coefficients between all the different variables. **Figure 2** shows the sorted Pearson correlation coefficients for both Seattle and Los Angeles. The results from the Pearson correlation coefficients between Seattle and Los Angeles are interesting. Namely that cooling degree days (CDD) are a significant, positive correlation and heating degree days (HDD) are a significant, negative correlation for LA, whereas the exact opposite is true for Seattle. **Figure 3** shows how the electricity demand varies in time for both LA and Seattle and sheds light on why we observe this behavior. Seattle's electricity demand spikes in the winter months, indicating that the majority of power is used for heating, whereas LA's electricity demand spikes in the summer months, indicating that the majority of power is used for cooling. This is consistent with the computed Pearson correlation coefficients, but a more thorough multivariate regression would be informative.

DEMAND CORRELATIONS (PEARSON) FOR LA		DEMAND CORRELATIONS (PEARSON) FOR SEATTLE	
dailycoolingdegreedays	0.572275	hourlyheatingdegrees	0.582695
hourlydewpointtempf	0.383226	dailyheatingdegreedays	0.574442
hourlyrelativehumidity	0.365116	hourlyrelativehumidity	0.269787
hourlywetbulbtempf	0.302208	hourlyprecip	0.061526
hourlycoolingdegrees	0.192912	hourlywindspeed	0.030342
hourlydrybulbtempf	0.044278	hourlysealevelpressure	0.005554
hourlyvisibility	0.009670	hourlyaltimetersetting	-0.027498
hourlyprecip	-0.022645	hourlystationpressure	-0.034864
dailyheatingdegreedays	-0.221475	hourlyvisibility	-0.068125
hourlywindspeed	-0.232047	hourlycoolingdegrees	-0.079621
hourlystationpressure	-0.265702	dailycoolingdegreedays	-0.115990
hourlysealevelpressure	-0.266134	hourlydewpointtempf	-0.519638
hourlyaltimetersetting	-0.267439	hourlydrybulbtempf	-0.552038
hourlyheatingdegrees	-0.290998	hourlywetbulbtempf	-0.580742

Figure 2. Pearson correlation coefficients for both LA (left) and Seattle (right). As discussed, CDD are positive correlations for LA, while they are negative in Seattle, and vice versa for HDD.

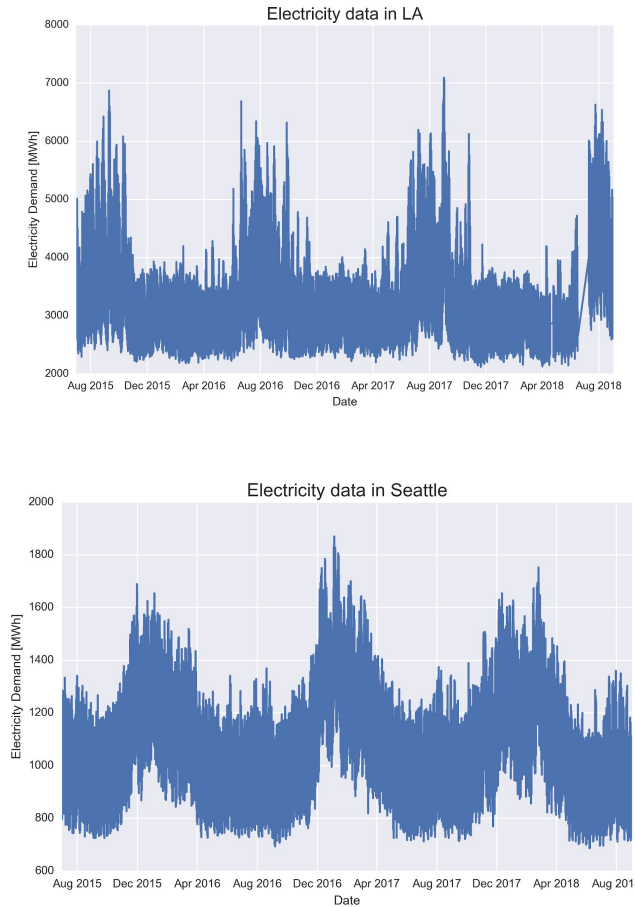


Figure 3. Electricity demand versus time for both LA (top) and Seattle (bottom). Demand peaks in the winter in Seattle, whereas it peaks in the summer in LA.

Next, we are interested in the significance of the CDD and HDD correlations since they are the strongest. Using bootstrapping statistics with ten thousand simulations, we compute p-values of exactly zero for all HDD and CDD features, indicating strong confidence ($p < 10^{-4}$) that the correlations are significant.

Additionally, **Figure 4** shows computed r^2 values for both the Seattle and Los Angeles data sets. Here we can see the relative power of each of the features for both data sets. Once again, HDD is a significant feature for Seattle while CDD is a significant feature for LA.

DEMAND CORRELATIONS (r ²) FOR LA			DEMAND CORRELATIONS (r ²) FOR SEATTLE		
	col	r ²		col	r ²
9	dailycoolingdegreedays	0.327499	1	hourlyheatingdegrees	0.339533
7	hourlydewpointtempf	0.146862	5	hourlywetbulbtempf	0.337261
13	hourlyrelativehumidity	0.133310	10	dailyheatingdegreedays	0.329984
5	hourlywetbulbtempf	0.091330	4	hourlydrybulbtempf	0.304746
1	hourlyheatingdegrees	0.084680	7	hourlydewpointtempf	0.270023
0	hourlyaltimetersetting	0.071524	13	hourlyrelativehumidity	0.072785
3	hourlysealevelpressure	0.070827	9	dailycoolingdegreedays	0.013454
12	hourlystationpressure	0.070597	6	hourlycoolingdegrees	0.006340
11	hourlywindspeed	0.053846	2	hourlyvisibility	0.004641
10	dailyheatingdegreedays	0.049051	8	hourlyprecip	0.003785
6	hourlycoolingdegrees	0.037215	12	hourlystationpressure	0.001215
4	hourlydrybulbtempf	0.001961	11	hourlywindspeed	0.000921
8	hourlyprecip	0.000513	0	hourlyaltimetersetting	0.000756
2	hourlyvisibility	0.000094	3	hourlysealevelpressure	0.000031

Figure 4. r² values for both LA (left) and Seattle (right).

The code that produced the above plots can be found [here](#).