Identifying Relationships and Simulating Pricing Across Weather and Energy Market Nodes

Alexander Ruff, Mohammad Meysami, Jie Wang

Sponsored by Ascend Analytics

February 12, 2019 - May 16, 2019

1 Abstract

Deterministic models can bias results with their limited pathways into the future. The changing dynamics of the energy market, due to the rising penetration of renewable sources, has led to inconsistencies with these deterministic computations. Inaccurate forecasting leads to challenges in risk assessment. And thus a new, more capable way of accurately predicting system behavior is desired. We aim to identify key relationships in market and weather data that inform the load and production of energy. And further, apply machine learning to reproduce and simulate the price of energy in the California market.

2 Introduction

Ascend Analytics focuses on consulting and creating custom analytic solutions for their clients in the wholesale electricity market. Ascend offers software solutions that integrate both physical and financial aspects to provide important insight into operations, portfolio analytics and planning. This project was inspired by the need for a more accurate model to predict the price of energy in the real-time market (RTM). Ascend has seen success analyzing day-ahead market data (DAM) and hourly level data for the RTM. (See [CAISO, 2019b] for a description of the two market processes of interest). However, these tools fail to adequately predict the higher-

Alexander Ruff: Data manipulation and price simulation.
Mohammad Meysami: Preliminary analysis and regression models.
Jie Wang: Time series analysis and background research.

resolution (5 - 15 minute) data sets of the RTM. The scope of our research will focus on successful simulation of the DAM to confirm their findings. We reserve predictions for higher resolution data for future study.

Beyond this information, we are not aware of any other past work or published articles that have been performed by others. According to Ascend, "Literature indicates that the most successful models other than neural networks have Boolean hypotheses." Our research will help to investigate this claim and provide further context towards identifying the relationships in the data.

The approach for this study is broken in to three phases which will be performed for the DAM.

Phase I involves choosing, parsing and merging data sets to create two separate sets over neighboring time periods selecting the first as training data and the second as test data. For instance, if creating monthly sets, December 2018 would be training data and January 2019 would be test data.

Phase II is to perform analysis on the training set to identify relationships and compare regression algorithms to choose the model that best fits all other variables (ie. Weather, renewable energy forecast, system load, etc.), to the dependent variable price of energy (measured in dollars per megawatt-hour or $/MWh).

Phase III is to then model and simulate the price of energy for the test set and compare the predicted price to the actual price. We can repeat this for other models and compare the results.

3 Methods

3.1 Data Parsing

Data can be described broadly by two categories: Weather and Market. In the data harvesting and parsing phase, the goal is to pick relevant data sets from the online portals outlined above. The data are not necessarily organized in their current state so we will take care when cleaning and merging the sets. The methods included in the Pandas library for Python will be implemented to manipulate the data.

The California Independent System Operator (CAISO) provides open access to energy market data for the state of California. This will be the source of our market data. Online portal at [CAISO, 2019a].

The National Climatic Data Center (NCDC) provides access to weather data, and from this portal, we will pull the appropriate weather station data for the area surrounding our market nodes of interest. Online portal at [NOAA, 2019].

After pairing specific weather stations and market nodes together by location, we can reduce now combined sets further by keeping only variables of interest and eliminating others.

3.2 Time Series Analysis

Time series analysis attempts to find meaning in the data by looking at trends in time. Such trends could reveal previously unseen relationships. In general, observations in neighboring time steps should be related to each other more than those far apart.

3.2.1 Autocorrelation

"Autocorrelation is the correlation of a signal with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations as a function of the time lag between them" [Pandey,]. Identifying repeating patterns within a selected time period may help to inform future models.

3.2.2 Window Functions

For the windowing analysis, we will plot the windows and looks for the window function which could provide the most optimal curve, which could inform us of the ideal model, transformation and so on.

(a) Rolling Window

Rolling windows overlap and “roll” along at the same frequency as the data. By default, all data points within a window are equally weighted in the aggregation.

(b) Hamming Window

Customary cosine-sum windows have the form,

$$w(n) = a_0 - (1 - a_0) cos\left(\frac{2\pi n}{N}\right) \qquad 0 \le n \le N$$

A special case of the Hann function, the Hamming window is recommended for smoothing the truncated autocovariance function in the time domain. The Hamming window is defined specifically as:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1$$

3.3 Regression Models

The data exploration phase incorporates first formal analysis using regression followed by intuition and repeated implementation to improve models based on the results.

We choose regression instead of classification as our predicted values for price are numerical and not categorical. This analysis will help us build intuition on the relationships between exploratory variables and show us how correlated they are to one another. Further these models will allow us to evaluate the significance of each predictor. In general, having more predictors leads to increased model complexity. If a sufficiently small amount of accuracy is lost in the predicted price of a model with reduced complexity (fewer input variables), than this reduced model is more favorable as implementing it will be computationally less expensive. With the correlations and significance of the predictors identified, we will be able to create the most efficient model. We will generate these models using methods in the Sklearn library for Python.

3.3.1 Generalized Linear Methods

In mathematical notation, if $\hat{y}$ is the predicted value. Then,

$$\hat{y}(\beta, x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p$$

where x_i is the ith predictor and $\hat{\beta}_i$ is the ith stimated coefficient for $i = 1, ..., p$. Note that β_0 corresponds to the value of the intercept.

(a) Ordinary Least Squares (OLS)

In order to minimize residual sum of squares (RSS) of observed responses, a linear model estimates coefficients $\beta = (\beta_0, ..., \beta_p)$. It attempts to solve:

$$\min_{\beta} ||X\beta - y||_2{}^2$$

Minimizing the above yields coefficients of the form,

$$\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} X^T y$$

When one of the predictors X_i is linearly dependent to another predictor X_j, $X^T X$ is not inevitable so that there is not one unique solution for $\hat{\beta}_{\text{OLS}}$. The issue gets even worse when $X^T X$ is inevitable but highly correlated and the columns of the design matrix X have an "approximate" linear dependence. Here, the design matrix $X^T X$ becomes close to singular and as a result, the least-squares estimate becomes highly sensitive to random errors in the observed response, producing a large variance. One way to overcome this issue is using penalized regression to fix $X^T X$ singularity by changing it to $(X^T X + \lambda\beta)$ where λ is the tuning parameter [Pedregosa et al., 2011].

Mohammad Meysami: Section 3.3.1, (a), (b), (c), and (d)

(b) Backward Selection

There are many techniques for model selection. In this project we use backward selection in order to find the significant predictors that have more impact on price. In backward selection, we fit the OLS model including all the predictors and in each step we remove the predictor with the highest P-value. We keep removing insignificant predictors until we have only predictors at 5% significance. [Faraway, 2016].

(c) Ridge

Ridge regression is similar to OLS but the coefficients are estimated differently. In ridge regression the $\hat{\beta}_{\text{Ridge}}$ is estimated by minimizing,

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

where $\lambda > 0$ is a tuning parameter and different λ values provide different predictor coefficient estimates $\hat{\beta}$. Here,

$$\hat{\beta}_{\text{Ridge}} = (X^T X + \lambda \mathbf{I})^{-1} X^T y$$

where $\mathbf{I}$ is an $n \times n$ identity matrix.

Clearly, when $\lambda = 0$, $\hat{\beta}_{\text{OLS}} = \hat{\beta}_{\text{Ridge}}$ and as $\lambda \to \infty$, $\hat{\beta}_{\text{Ridge}} \approx 0$ [James et al., 2014]. Sklearn provides methods for ridge regression with built in cross-validation for finding the optimal value of λ.

(d) Lasso

Lasso regression is very similar to ridge except it uses l_1-norm for the shrinkage penalty term. In order to estimate the lasso coefficients, $\hat{\beta}_{\text{Lasso}}$, we minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

We prefer less complex models. Eliminating insignificant regressors reduces the variables on which the model depends. This allows for easier model interpretation. Again, Sklearn provides built in cross-validation for the Lasso method in order to determine the optimal λ value.

3.3.2 Ensemble Methods

A single decision tree results in a model prediction of an output variable that can be either categorical or numerical. It learns through basic decision rules it builds from input features. Ensemble methods combine predictions of many decision trees to improve robustness.

Ensemble methods either work by averaging or boosting.

Averaging methods build each part of the ensemble independently simply averaging all of the outcomes together to return the best result.

Boosting methods, build sequentially to achieve the desired result. Decision trees are useful for boosting since they can work with a variety of data types and can ultimately simulate much more complex models.

(a) Random Forest Regression (RFR)

RFR is an averaging method. "In random forests, each decision tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model" [Bobkov et al., 2016].

(b) Gradient Boosted Decision Trees (GBRT)

GBRT is a boosting method. It considers models of the form:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$$

where h_m are functions called weak learners (weaker trees) in the context of boosting. Gradient Tree Boosting uses decision trees of fixed size as weak learners.

GBRT builds the additive model by:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

where the new trees attempt minimization of the loss function L, given the previous ensemble F_{m-1}:

$$h_m = \arg\min_h \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x_i))$$

The initial model F_0 is problem specific. GBRT attempts minimization numerically by steepest descent.

The negative gradient of the current model F_{m-1} can be calculated for any loss function by:

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_F L(y_i, F_{m-1}(x_i))$$

Here step length γ_m is chosen with line search:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}).$$

[Pedregosa et al., 2011]

3.4 Price Simulation

After fitting the training set to a model, predictions for price will be made using the input variables from the test set. These predictions are then compared to the known actual price from the test set and evaluated for accuracy. We consider a number of scoring methods for quantifying the quality of predictions. Comparing these scores will allow us to select an optimal model for simulation of both markets. Sklearn provides methods for determining these scoring metrics.

(a) Mean Squared Error (MSE)

MSE corresponds to the average of the squared difference between actual and predicted responses. Better models minimize this as much as possible over n_{samples}. The formula is given by:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2$$

(b) The Coefficient of Determination (R^2)

The R^2 score measures model fit. It explains the variation of the model. Higher scores are closer to 1.0. Because models can be arbitrarily worse, it is not impossible to see negative scores.

The formula is given by:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i$.

4 Results

4.1 Variables of Interest

Variables of interest are chosen from the raw data sets based on factors such as relevance, absence of N/A values, and collection times.

Name	Original Set	Type	Units
Sky Condition	Weather	Categorical	BKN, CLR, FEW, OVC, SCT
Wind Speed	Weather	Numerical	mph
Wind Direction	Weather	Numerical	Degrees
Temperature	Weather	Numerical	Degrees Fahrenheit
Humidity	Weather	Numerical	g/kg
Station Pressure	Weather	Numerical	Pa
Sunrise	Weather	Numerical	hh:mm
Sunset	Weather	Numerical	hh:mm
Price	Energy	Numerical	$/MWh
Load	Energy	Numerical	MWh
Solar Actual	Energy	Numerical	MWh
Wind Actual	Energy	Numerical	MWh
Solar Forecast	Energy	Numerical	MWh
Wind Forecast	Energy	Numerical	MWh
Hour	Both	Numerical	0-23
Time	Both	Numerical	yyyy-mm-dd hh:mm:ss

Table 1: The chosen variables of study for the DAM. These remain after eliminating extraneous variables.

4.2 Data Visualization

These figures provide basic visualization of the variable relationships and build intuition for our analysis.

Mohammad Meysami: Figures 1-5

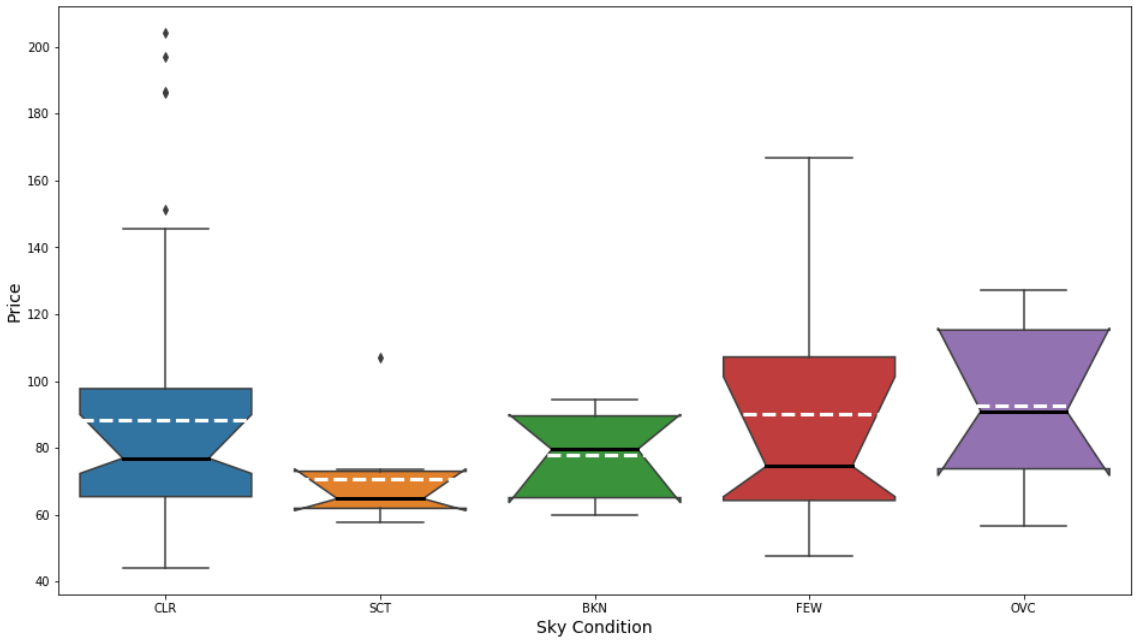

Figure 1: Mean (*white dashed line*) and median (*black solid line*) of the price of energy for each Sky Condition. Notches show the 95% confidence interval for the median. December 1-8, 2018.

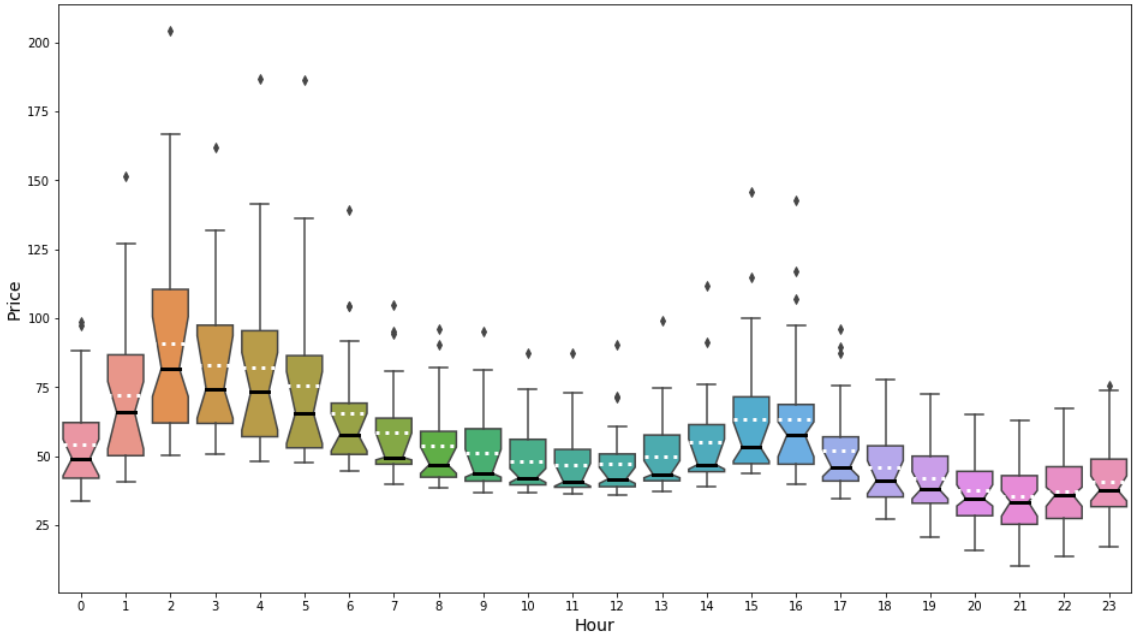

Figure 2: Mean (*white dashed line*) and median (*black solid line*) of the price of energy for each hour. Notches show the 95% confidence interval for the median. December 1-8, 2018.

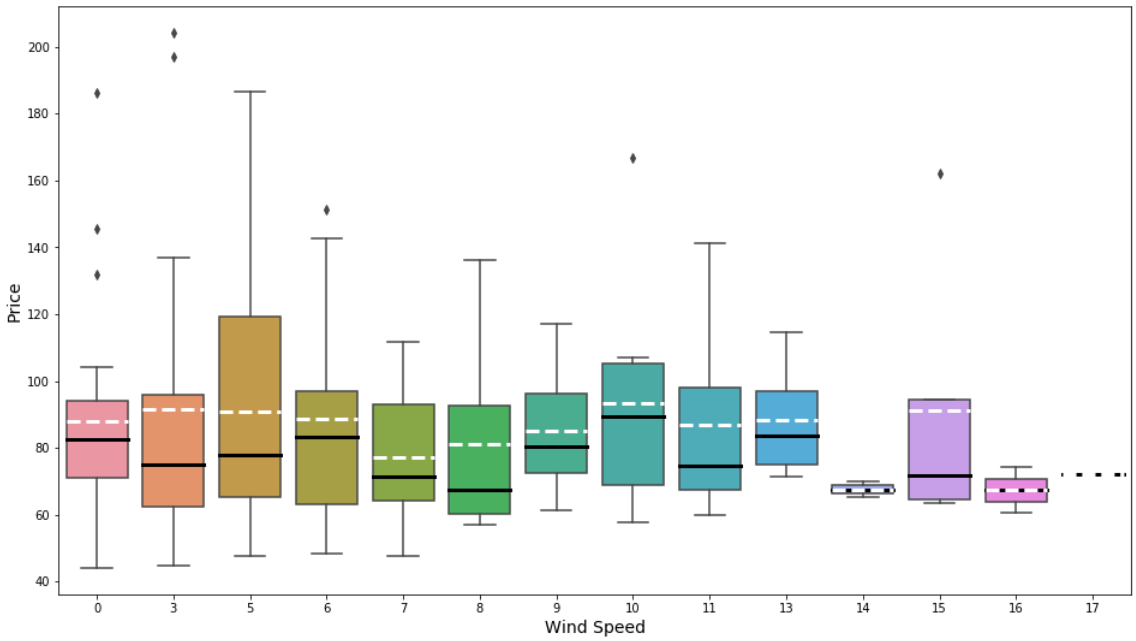

Figure 3: Mean (*white dashed line*) and median (*black solid line*) of the price of energy for each wind speeds from 0-17 mph. December 1-8, 2018.

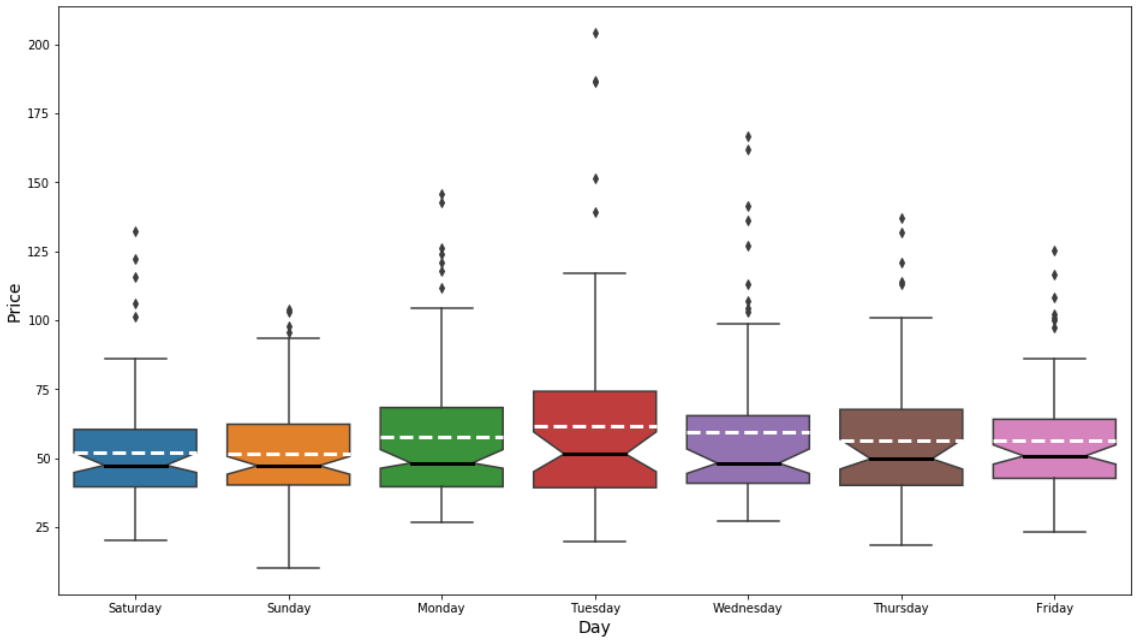

Figure 4: Mean (*white dashed line*) and median (*black solid line*) price of energy by day of the week. Notches show the 95% confidence interval for the median. December 1-8, 2018.

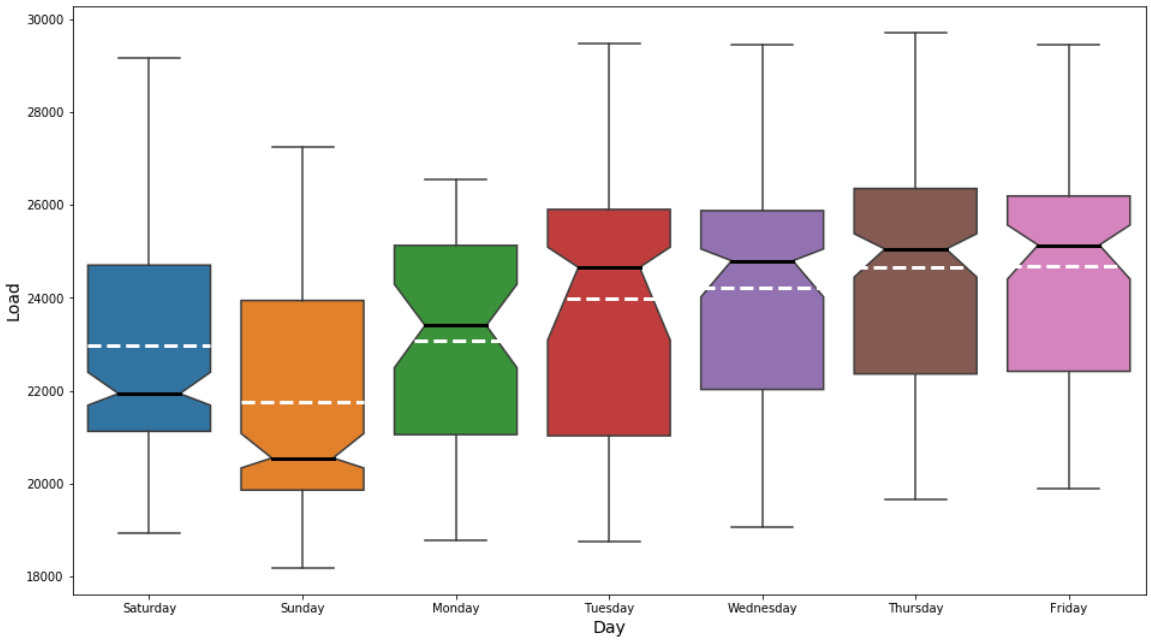

Figure 5: Mean (*white dashed line*) and median (*black solid line*) energy load by day of the week. Notches show the 95% confidence interval for the median. December 1-8, 2018.

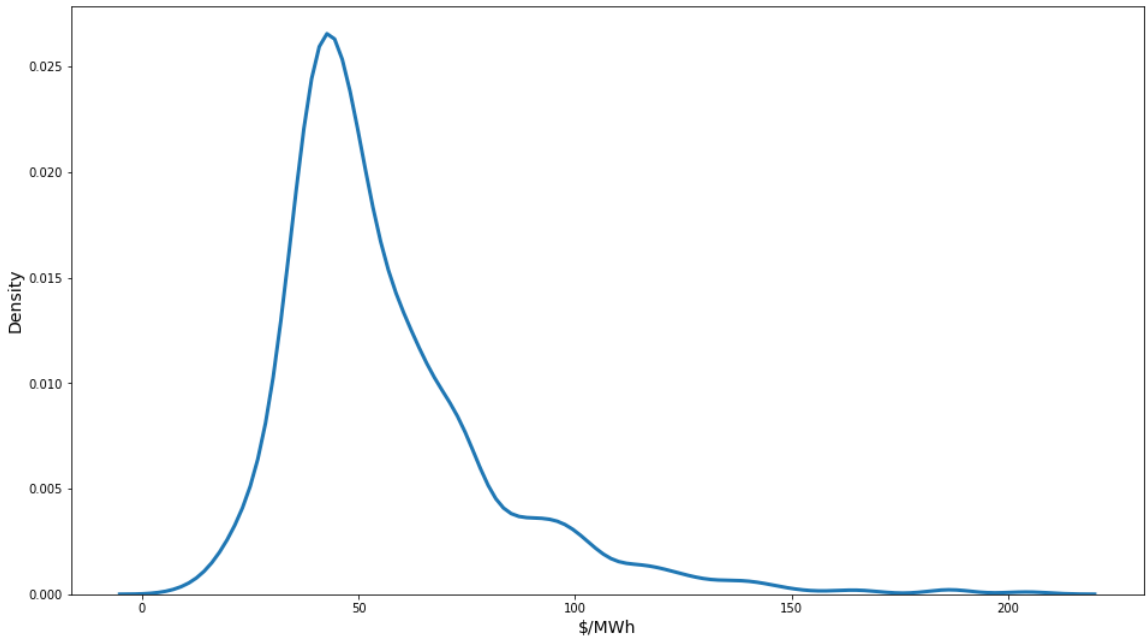

Figure 6: Kernel density estimate of price distribution for the full month of December data set.

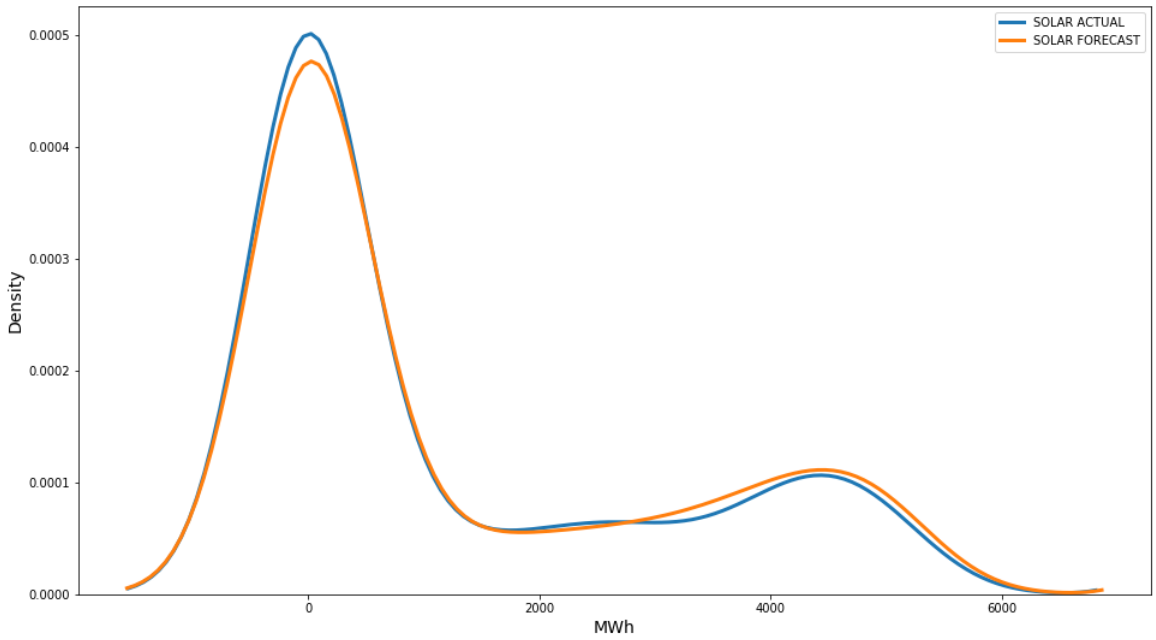

Figure 7: Kernel density estimate of solar energy distribution for the full month of December data set.

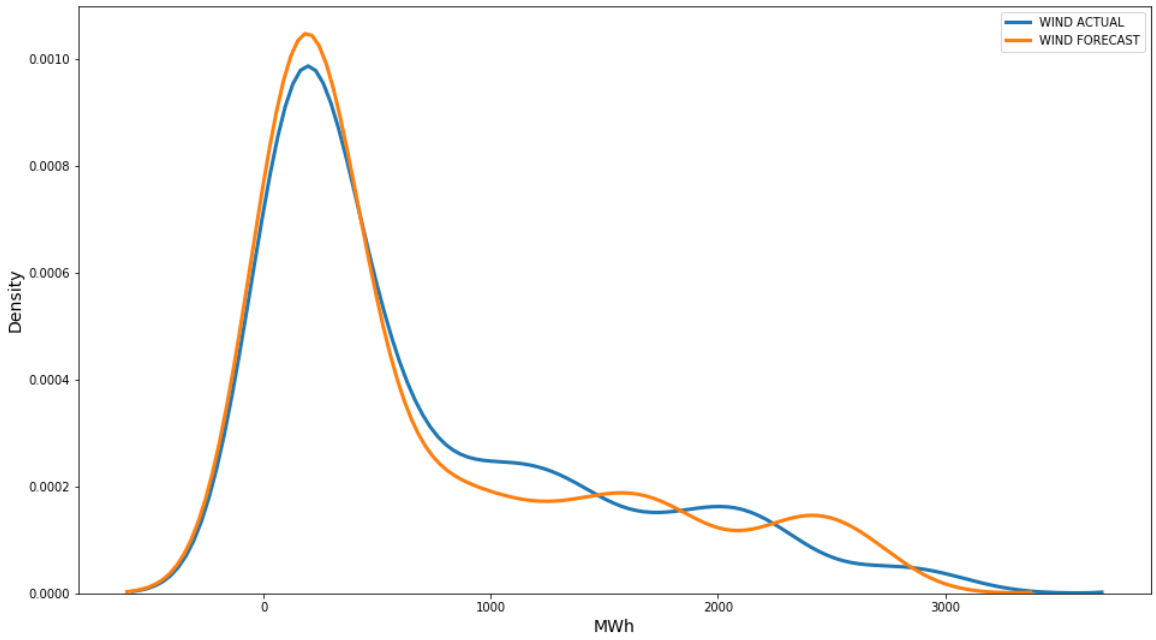

Figure 8: Kernel density estimate of wind energy distribution for the full month of December data set.

4.3 Time Series Analysis

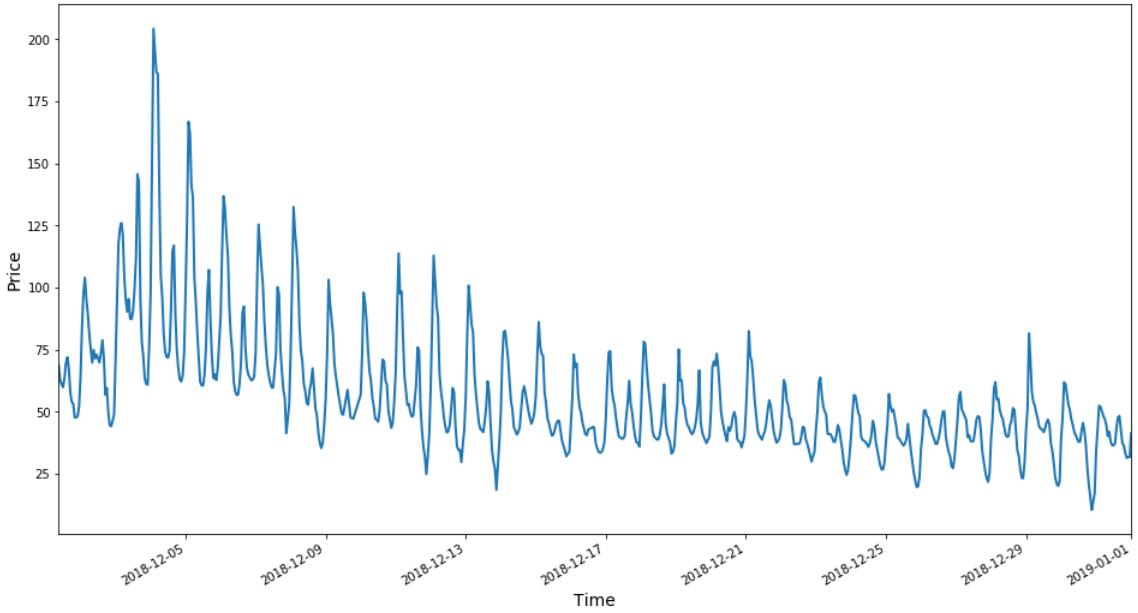

Figure 9: Price vs. time for the full month of December data set.

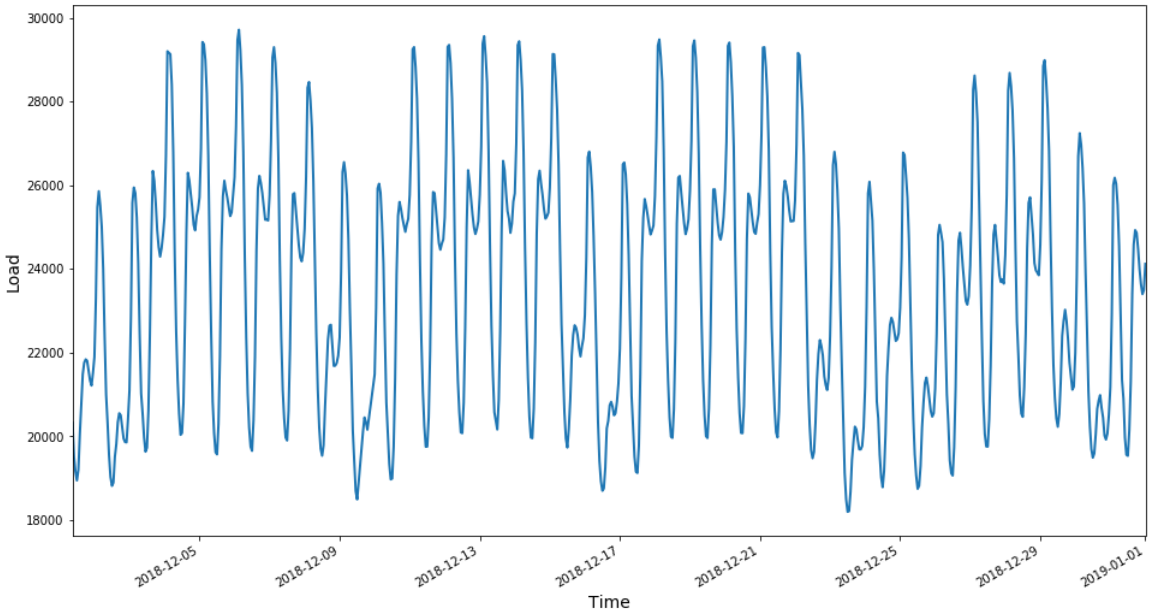

Figure 10: Load vs. time for the full month of December data set.

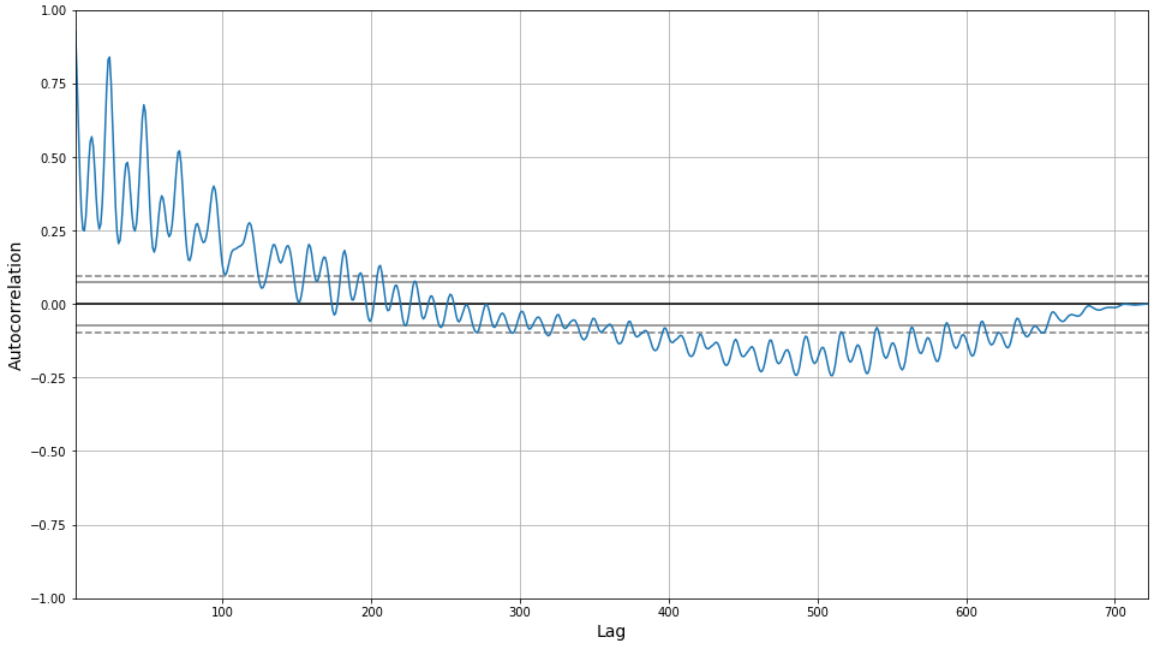

Figure 11: Autocorrelation plot of price for the full month of December.

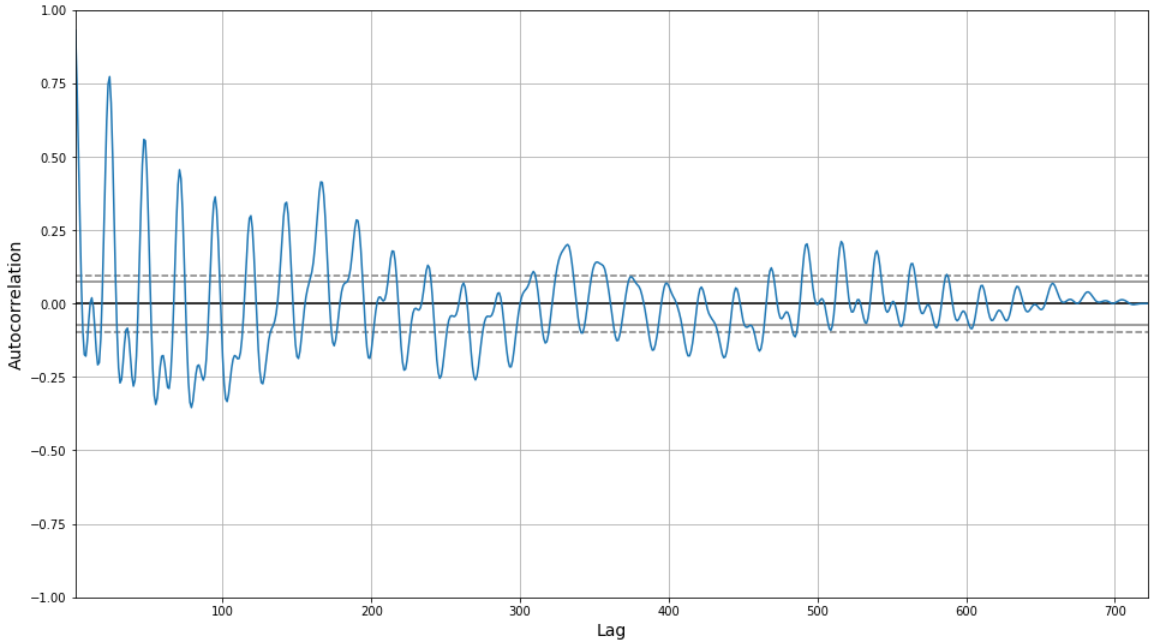

Figure 12: Autocorrelation plot of load for the full month of December.

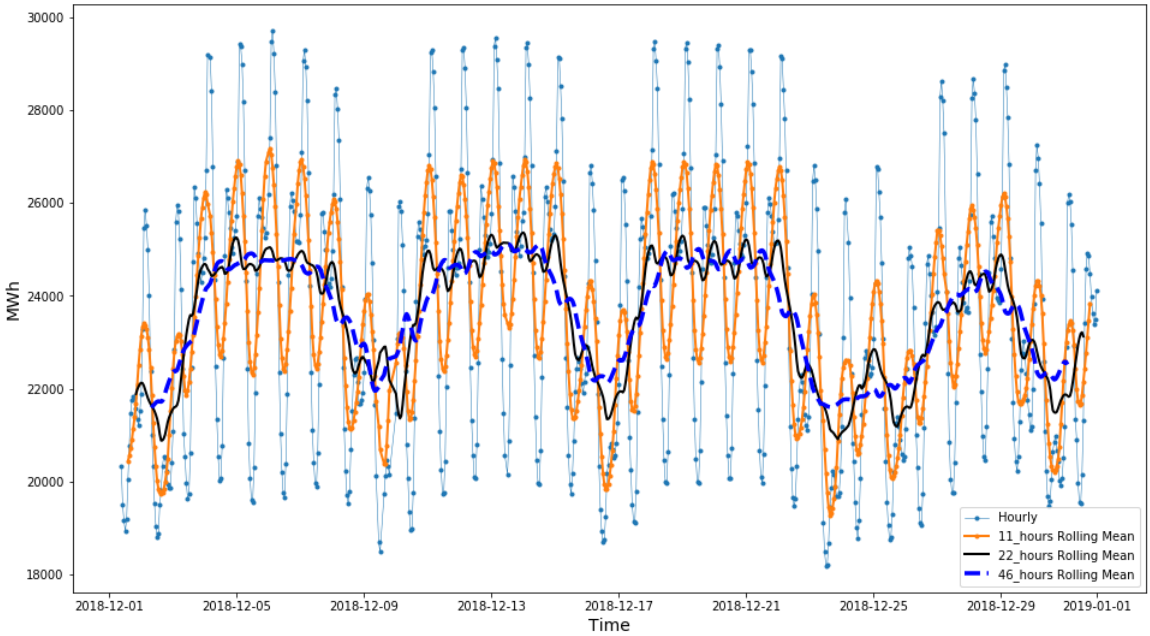

Figure 13: Rolling window plot of load for 11, 22 and 46 hours.

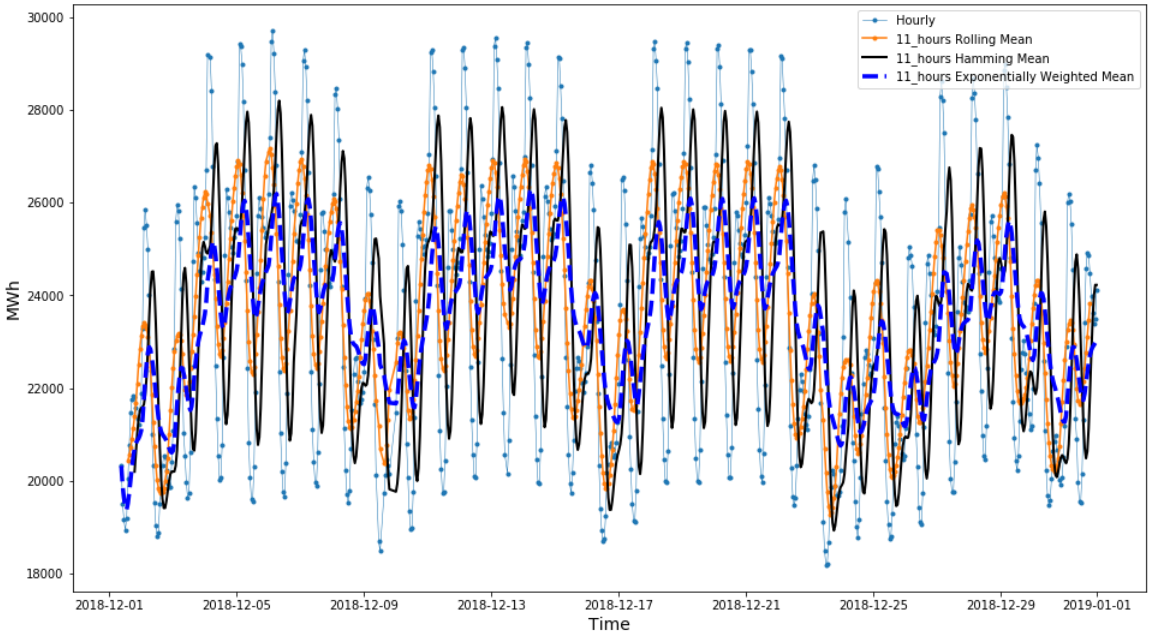

Figure 14: 11 hour window plot of load with rolling, hamming, and exponentially weighted means.

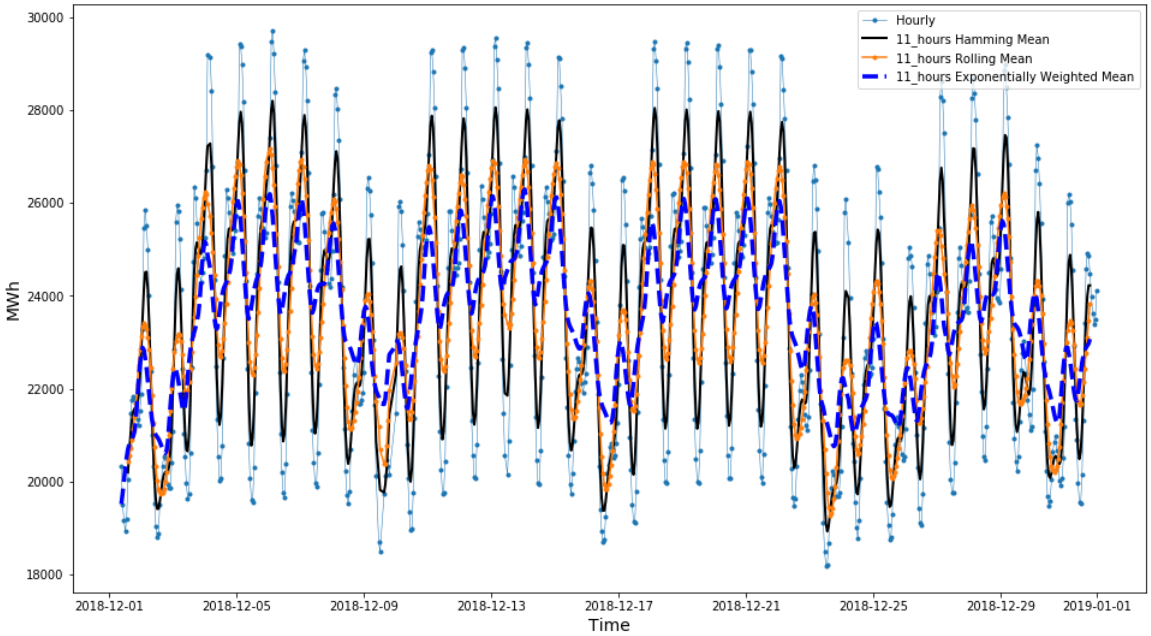

Figure 15: Shifted 11 hour window plot of load with rolling, hamming, and exponentially weighted means.

4.4 Model Selection

4.4.1 OLS

Variable	Coefficient	Std. Error	t-test	P-Value	2.5%	97.5%
Intercept	3788.9644	3583.632	1.057	0.293	-3312.954	1.09e+04
SKY CONDITION[CLR]	-0.0096	7.691	-0.001	0.999	-15.252	15.233
SKY CONDITION[FEW]	4.4463	6.169	0.721	0.473	-7.778	16.671
SKY CONDITION[OVC]	8.0645	7.296	1.105	0.271	-6.395	22.524
SKY CONDITION[SCT]	4.4046	7.769	0.567	0.572	-10.992	19.802
VISIBILITY	-2.0098	4.039	-0.498	0.620	-10.015	5.995
DEW POINT TEMP	0.4726	0.499	0.948	0.345	-0.516	1.461
RELATIVE HUMIDITY	-0.3489	0.154	-2.270	0.025	-0.654	-0.044
WIND SPEED	-0.1480	0.562	-0.263	0.793	-1.262	0.966
STATION PRESSURE	-252.6292	308.723	-0.818	0.415	-864.445	359.187
SEA LEVEL PRESSURE	301.0003	303.789	0.991	0.324	-301.038	903.039
SUNRISE	-8.1342	3.865	-2.105	0.038	-15.793	-0.475
HOUR	-1.3282	0.410	-3.236	0.002	-2.141	-0.515
LOAD	0.0079	0.001	11.015	0.000	0.006	0.009
SOLAR FORECAST	-0.0049	0.008	-0.633	0.528	-0.020	0.010
WIND FORECAST	-0.0036	0.004	-0.916	0.362	-0.011	0.004
SOLAR ACTUAL	0.0012	0.007	0.168	0.867	-0.013	0.015
WIND ACTUAL	-0.0064	0.005	-1.270	0.207	-0.016	0.004

Table 2: Calculated coefficients and P-Values using OLS on the December 1-8, 2018 data set. 2.5% and 97.5% refer to the respective percentiles.

Mohammad Meysami: OLS

At a significance level of at least 5%, four predictors RELATIVE HUMIDITY, HOUR, LOAD, and SUNRISE are statistically significant based on this data. This corresponds to the model given by:

$$\begin{aligned}\text{Price} = 3788 &- 0.3489 \times \text{RELATIVE HUMIDITY} \\ &- 8.1342 \times \text{SUNRISE} \\ &- 1.3282 \times \text{HOUR} \\ &+ 0.0079 \times \text{LOAD}\end{aligned}$$

4.4.2 Backward Method

All significant predictors found by backward selection are shown below.

Variable	**Coefficient**	**Std. Error**	**t-test**	**P-Value**	**2.5%**	**97.5%**
Intercept	2710.3056	2721.633	0.996	0.321	-2678.799	8099.410
DEW POINT TEMP	0.7698	0.401	1.920	0.057	-0.024	1.564
RELATIVE HUMIDITY	-0.3763	0.116	-3.241	0.002	-0.606	-0.146
SEA LEVEL PRESSURE	56.1496	29.923	1.876	0.063	-3.101	115.400
SUNRISE	-6.8557	3.040	-2.255	0.026	-12.875	-0.836
HOUR	-1.3182	0.391	-3.369	0.001	-2.093	-0.543
LOAD	0.0082	0.001	13.006	0.000	0.007	0.009
SOLAR FORECAST	-0.0036	0.001	-2.480	0.015	-0.006	-0.001
WIND ACTUAL	-0.0099	0.003	-3.194	0.002	-0.016	-0.004

Table 3: Calculated coefficients and P-Values using backward selection on the December 1-8, 2018 data set. Reduced to only show signif. variables. 2.5% and 97.5% refer to the respective percentiles.

Using backward selection, all the predictors in the above summary are significant at a level of 5%. So the desired model fit is given more accurately by:

$$\begin{aligned}\text{Price} = 2710.3056 &+ 0.7698 \times \text{DEW POINT TEMP} \\ &- 0.3763 \times \text{RELATIVE HUMIDITY} \\ &+ 56.1496 \times \text{SEA LEVEL PRESSURE} \\ &- 6.8557 \times \text{SUNRISE} \\ &- 1.3182 \times \text{HOUR} \\ &+ 0.0082 \times \text{LOAD} \\ &- 0.0036 \times \text{SOLAR FORECAST} \\ &- 0.0099 \times \text{WIND ACTUAL}\end{aligned}$$

Mohammad Meysami: Backward Method

4.4.3 Ridge

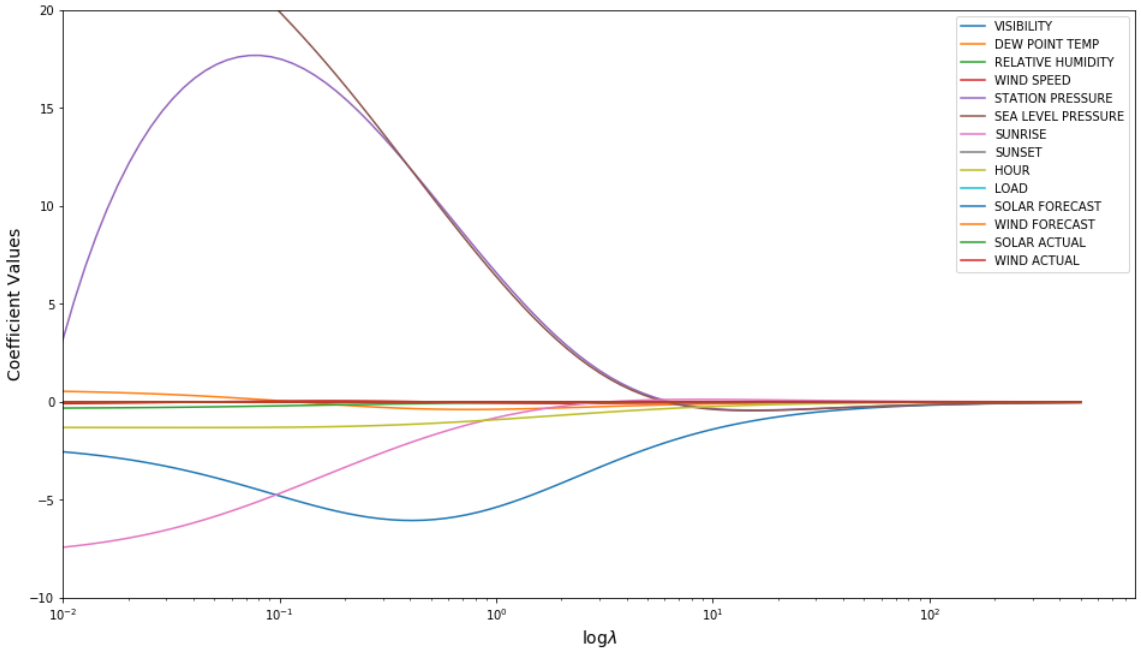

Figure 16: The change in coefficient values for different values of λ. All predictors are included.

In this case the optimal value for tuning parameter is $\lambda = 0.072$. As λ changes, the weight of the coefficients in the ridge model also change. With the optimal value of λ, MSE ($\sqrt{\text{MSE}}$) is 325.247 (18.034). The following table gives the corresponding coefficients.

Variable	Ridge Coefficient
VISIBILITY	-4.345507
DEW POINT TEMP	0.159722
RELATIVE HUMIDITY	-0.224212
WIND SPEED	0.031958
STATION PRESSURE	17.676609
SEA LEVEL PRESSURE	21.479997
SUNRISE	-5.248015
SUNSET	0.000000
HOUR	-1.304344
LOAD	0.006887
SOLAR FORECAST	-0.001767
WIND FORECAST	-0.004240
SOLAR ACTUAL	-0.001508
WIND ACTUAL	-0.004495

Table 4: Calculated coefficient estimates for Ridge Regression.

4.4.4 Lasso

Again using cross-validation, we attempt to choose an optimal value for λ using lasso regression.

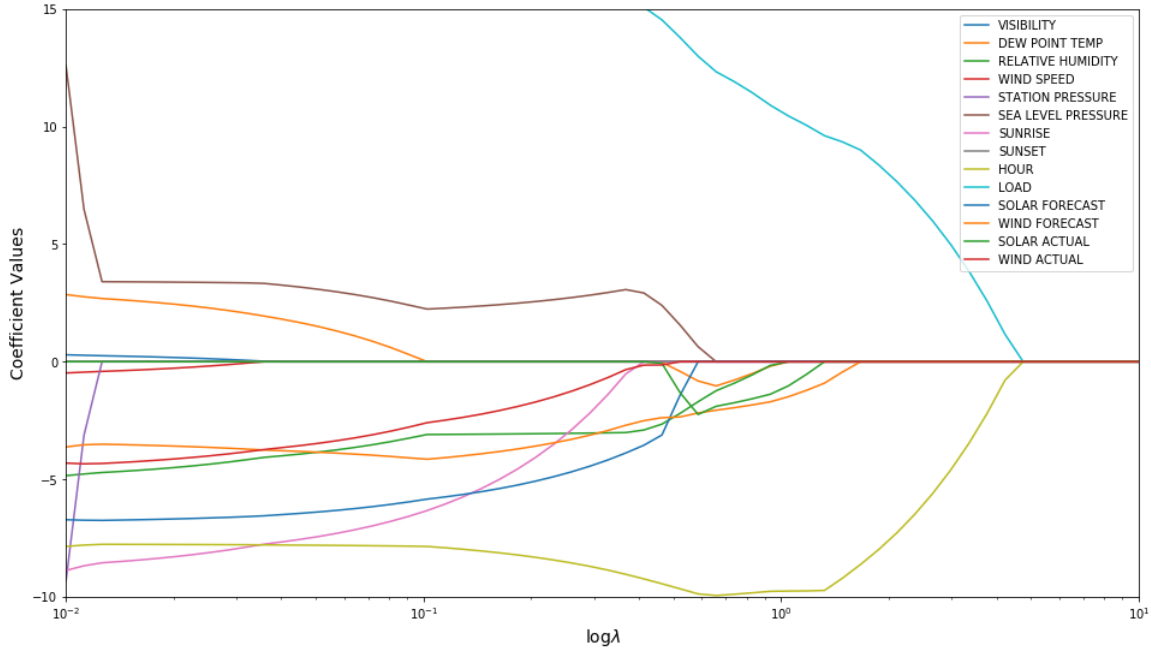

Figure 17: The change in coefficient values for different values of λ.

In this case, the optimal $\lambda = 0.074$ and MSE ($\sqrt{\text{MSE}}$) associated with it is equal to 326.311 (18.064).

The following table gives the corresponding coefficients for the lasso method. This model has reduced complexity and includes only 8 predictors.

Variable	lasso Coefficient
VISIBILITY	-0.000000
DEW POINT TEMP	0.000000
RELATIVE HUMIDITY	-0.175064
WIND SPEED	0.000000
STATION PRESSURE	0.000000
SEA LEVEL PRESSURE	21.686815
SUNRISE	-4.268319
SUNSET	0.000000
HOUR	-1.201886
LOAD	0.006044
SOLAR FORECAST	-0.003366
WIND FORECAST	-0.003577
SOLAR ACTUAL	-0.000000
WIND ACTUAL	-0.002748

Table 5: Calculated coefficient estimates for Lasso Regression.

4.4.5 GBRT

The GBRT model can return the importance of each predictor used in the model. Knowing this will allow us to build a model with fewer predictors and reduce complexity.

Feature	Importance
Sunrise	45.23%
Load	34.35%
Hour	6.65%
Solar Actual	5.92%
Wind Forecast	2.28%
Wind Actual	1.46%
Solar Forecast	1.46%
Sunset	1.34%
Air Pressure	0.66%
Temperature	0.31%
Humidity	0.18%
Wind Speed	0.12%

Table 6: Importance of each feature in GBRT model from the month of December data set.

4.5 Price Simulation

After training the various models, predictors from the test set are applied to generate a predicted value for price. This prediction is then compared to the actual given price from the test set.

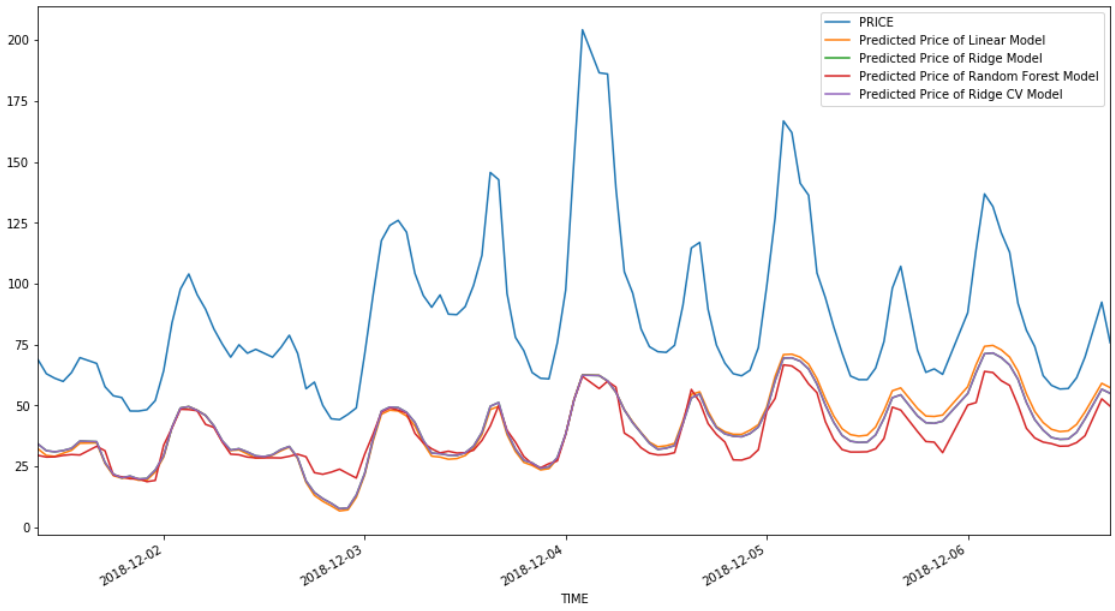

Figure 18: Predicted price of test data by various models compared to actual price from test data, blue. Training set: December 1-8, 2017. Test set: December 1-8, 2018

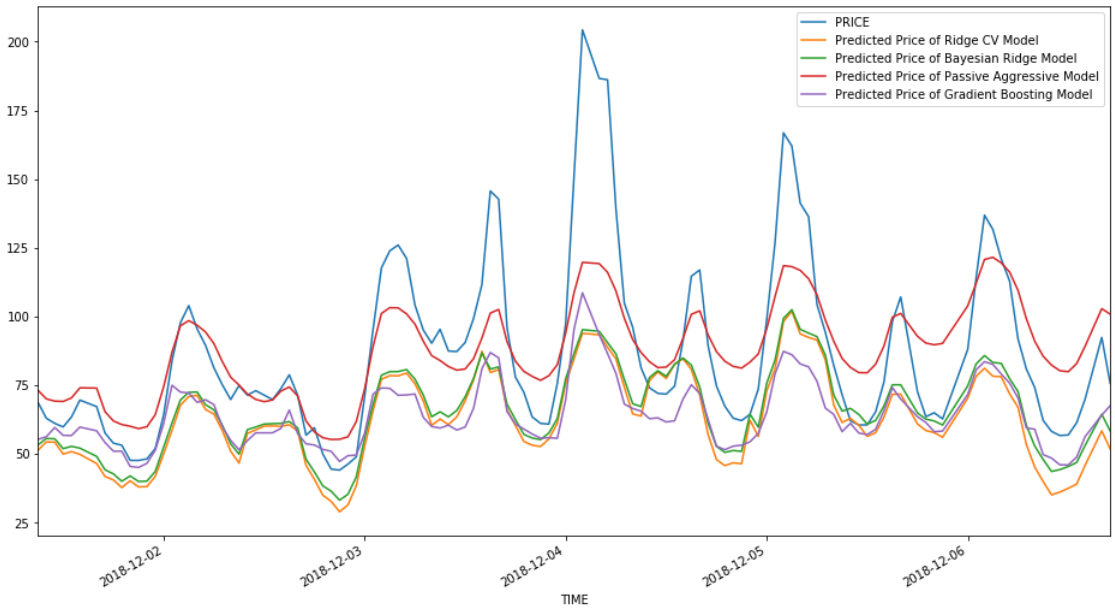

Figure 19: Predicted price of test data by various models compared to actual price from test data, blue. Training set: November 23-30, 2018. Test set: December 1-8, 2018

We see much higher accuracy when the training and test sets are closer together in time.

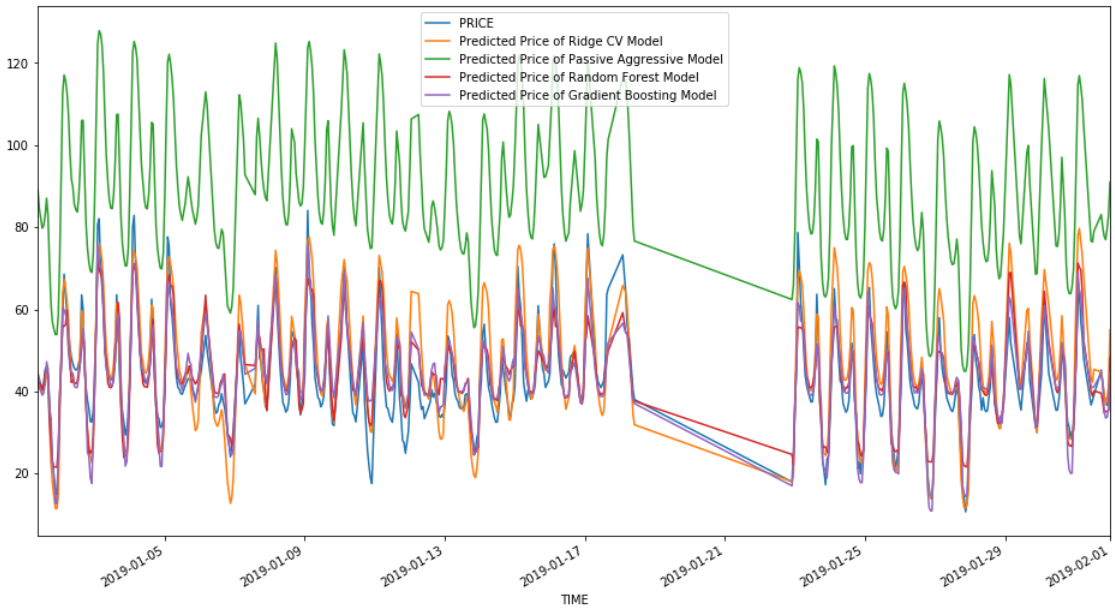

Figure 20: Predicted price of test data by various models compared to actual price from test data, blue. Training set: December 1-31, 2018. Test set: January 1-31, 2019

The results for the month long simulations are far more accurate than the one week models. These are scored for comparison.

Model	MSE	$\sqrt{MSE}$	R^2
Ridge	61.49	7.84	0.60
Passive Aggressive	2172.16	46.60	-13.11
Random Forest	32.05	5.66	0.79
Gradient Boosting	26.09	5.10	0.83

Table 7: Model scoring metrics. Training set: December 1-31, 2018. Test set: January 1-31, 2019

4.5.1 Reducing Complexity

GBRT gives the smallest comparative $\sqrt{MSE}$. Using the calculated feature importance of the GBRT model (Table 6) we generate a model with reduced complexity by eliminating predictors that are less than 1% significant.

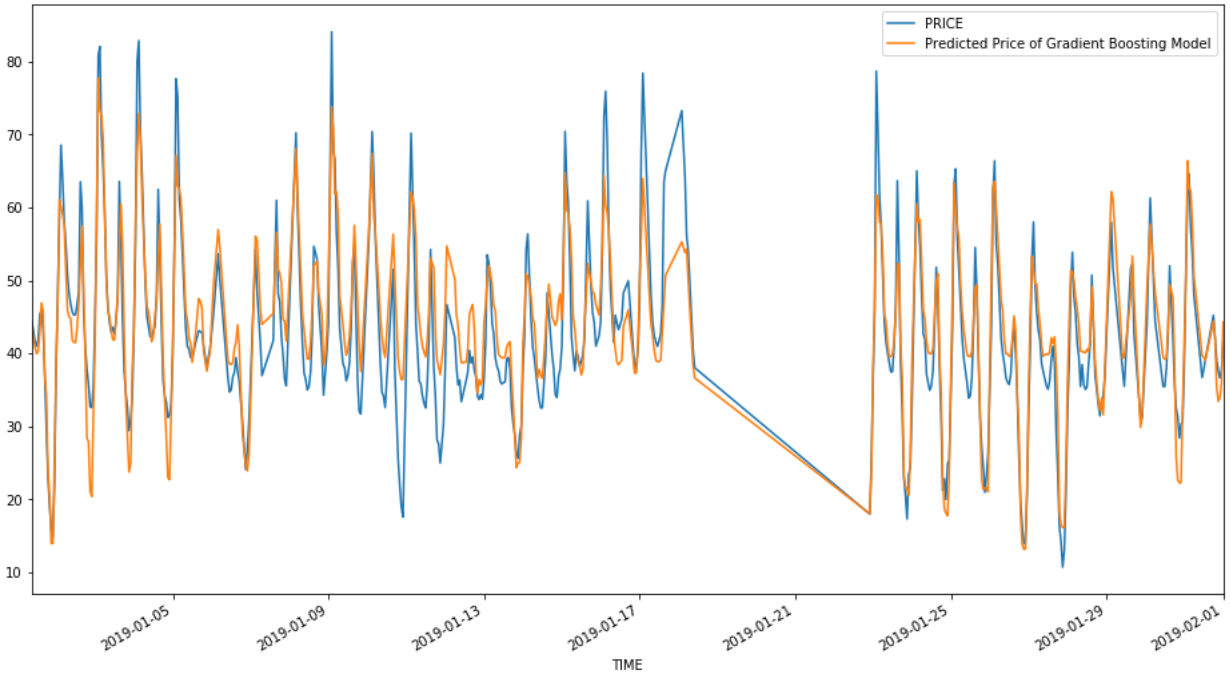

Figure 21: Predicted price of test data by simplified GBRT model, orange, compared to actual price from test data, blue. Training set: December 1-31, 2018. Test set: January 1-31, 2019

GBRT Predictors	MSE	$\sqrt{MSE}$	R^2
12	26.09	5.10	0.83
8	27.36	5.23	0.82

Table 8: Model scoring metrics for full and reduced GBRT models. Training set: December 1-31, 2018. Test set: January 1-31, 2019

5 Discussion

5.1 Preliminary Analysis

Figure 1 shows us that price is not heavily influenced by the current sky condition. The five categories are grouped fairly close together. Since sky condition is categorical and insignificant by OLS (Table 2) we eliminate this predictor. This reduction allows the use of regression algorithms in Sklearn.

From our research, Figure 2 represents a relationship that is referred to as the "Duck Curve" of energy markets (See [Wikipedia contributors, 2019]). The mean price by hour suggests that prices are highest in the morning, gradually decreasing till afternoon, spiking up slightly, then decreasing again. It is postulated that this is due to the increasing availability of solar energy as the day progresses, with a spike in demand when most consumers arrive home after work. Feature importance, Table 6, confirms the impact of solar energy on price.

Figure 3 suggests that the price is not affected greatly by Wind Speed. Our intuition is that rather than being driven by Wind Speed alone, the presence of a constant force of wind actually drives generation of energy, which effects price. Future analysis could benefit from data showing wind consistency.

Interestingly, Figures 4 and 5 reveal unexpected results. Load by day of the week tells us there is a clear difference in median load on the weekends versus weekdays. The confidence intervals on Saturday and Sunday do not overlap the weekday intervals which indicates this difference. This difference is not evident in price by weekday. While there is some variation from weekends to weekdays in price, it is not as significant as the difference in load.

The kernel density estimates for price and renewables, Figures 6, 7 and 8, are useful for trying to determine the distributions of these variables. Renewable distributions show similar plots for forecast and actual which means forecast distributions adequately model actual energy generation. These could help inform future models.

5.2 Time Series Analysis

Figure 10 shows the periodicity of load over the course of December. Observe the repetitive sequence of peaks and troughs. This and Figure 5 suggest that load is significantly higher on weekdays than on weekends. For the same set, we do not observe similar periodicity for price as shown in Figure 9.

In the autocorrelation plots, Figures 11 and 12, price initially shows some correlation to itself, but fades after

about 8 days. In contrast, load seems to be highly correlated with itself for a longer period of time. This tells us that exploring load for trends in time may lead to opportunities for better forecasting.

In Figure 13, we observe rolling mean over several time steps attempting to capture the correct periodic window. The 11 hour step seems to capture the peaks and troughs best. In Figure 14 we use the 11 hour time step and attempt to use different window functions to capture the periodicity. The hamming window gives the best results, but is slightly out of phase. By shifting the window, Figure 15, all functions capture the phase better, and thus our shifted window hamming mean is optimal for forecasting load.

5.3 Model Selection

OLS, Table 2, allows us to build a Linear Regression model to predict pricing in the DAM. The code considers all variables, and determines which should be used for the model. Significant predictors are determined by low P-Values, but OLS should be performed again with the larger data sets for more reliable results. We cannot trust the OLS model because we remove all the insignificant regressors instantly without taking into account the effect of removing each regressor. To eliminate this issue, we use backward selection which eliminates regressors one by one to find the optimal model.

In each iteration of the Backward Method, Table 3, we eliminate the predictor with the highest p-value. As we can see, since the matrix $X^T X$ is highly correlated, it results in multicollinearity. This prompts us to try other penalized regression techniques that are more robust for handling multicollinearity; such as ridge and lasso regression.

For ridge regression, since the number of predictors is fixed, it does not allow for the reduction of model complexity. All the predictors are kept even though some of the their coefficients are close to zero (Figure 16). However, in lasso regression when λ increases, Figure 17, some predictor coefficients become zero. So, unlike with ridge, we could use lasso to reduce model complexity through better selection. In compression, lasso's MSE is slightly higher than ridge's. However, the lasso model is less complex, Table 5, since it has only eight predictors. We observe a trade off between reduction of MSE and complexity. While reducing the number of predictors to 8 increases MSE, it may be that the increase is tolerable in favor of a simpler model.

5.4 Simulation Comparison

Figure 18 suggests that using training and test sets that are far apart in time results in poor simulation results. And after adjusting the training set to the week before the test set, Figure 19 shows an improvement

in the accuracy of the predictions. While improved, we still cannot rely on these models as the MSE is fairly large. Importantly, the shape of the actual price seems to be well captured by these simulations.

Figure 20 shows our first acceptable simulations. These arise from models trained on the entire month of December data set. It seems that more training data is the key to better test predictions. Moving forward, we will use larger sets for analysis based on these results. The scoring metrics in Table 7 show the $\sqrt{\text{MSE}}$ is smallest for GBRT.

After choosing fewer predictors for GBRT, Table 6, we again simulate prices for the test set, Figure 21. The resulting scoring metrics shown in Table 8 represent the trade off between reducing complexity and increasing MSE. This confirms the tolerable loss of accuracy for the simpler model. It stands that our current best model is GBRT with 8 predictors.

5.5 Model Interpretation

Gradient boosting models are difficult to visualize. Single trees are easily interpreted as they can be returned from an algorithm. GBRT models, however, are made up of many trees and visualizing this isn't possible. There are ways to infer relationships without direct inspection though, generally by analyzing feature importance. As in Table 6, feature importance is our best interpretation for GBRT models.

6 Conclusion

Over the course of this study, we have successfully simulated prices for the DAM by finding models that adequately fit weather and energy market data. Future analysis will take the discoveries made here and apply them to higher resolution sets in order to simulate prices in the RTM. We suspect that our exploration will help reveal similar relationships at this level and will serve as a good starting point for other interested parties. Dealing with over-fitting, and identifying the trade off between variance and bias are learning goals that could optimize further research. We believe the information we have learned from the DAM will help Ascend confirm their findings and that the steps taken will outline future research into this market.

AJ and Mohammad: Conclusion

References

[Bobkov et al., 2016] Bobkov, V., Bobkova, A., Porshnev, S., and Zuzin, V. (2016). The application of ensemble learning for delineation of the left ventricle on echocardiographic records. In *2016 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, pages 1–5. IEEE.

[CAISO, 2019a] CAISO (2005-2019a). Energy market database. http://oasis.caiso.com/mrioasis/logon.do.

[CAISO, 2019b] CAISO (2019b). Energy market processes. http://www.caiso.com/market/Pages/MarketProcesses.aspx.

[Faraway, 2016] Faraway, J. J. (2016). *Linear models with R*. Chapman and Hall/CRC.

[James et al., 2014] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). An introduction to statistical learning. 112.

[NOAA, 2019] NOAA (2019). Weather database. https://www.ncdc.noaa.gov/cdo-web/datatools.

[Pandey,] Pandey. Econometrics. http://ddceutkal.ac.in/Syllabus/MA_Economics/ECO-3.2-ECONOMETRICS.pdf.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Wikipedia contributors, 2019] Wikipedia contributors (2019). Duck curve — Wikipedia, the free encyclopedia. [Online; accessed 18-May-2019].

7 Appendix: Required Files

Please see compressed file (ascend_code_and_data.zip) for data sets in .csv format and code notebooks in .ipynb format.

7.1 Data Sets

1. Merged_Data.csv
2. Merge_Data2.csv
3. PURI_DECEMBER18.csv
4. PURI_JANUARY19.csv
5. PURISIMA_COMBINED_2017-12-01.csv
6. PURISIMA_COMBINED_2018-04-01.csv
7. PURISIMA_COMBINED_2018-11-23.csv
8. PURISIMA_COMBINED_2018-12-01.csv

7.2 Code Notebooks

1. prelim_analysis.ipynb
2. time_series_and_visualization.ipynb
3. ridge_lasso_tuning.ipynb
4. price_simulation.ipynb

Mohammad Meysami: prelim-analysis.ipynb and ridge-lasso-tuning.ipynb