Hypothesis Testing - Expected Value*

Daniel Maryanovsky[1] and Premanand Kumar[2]

Abstract— **The hypothesis here is that given the historic value of power consumption and the external weather data, it is possible to predict the power consumption on any given day in the future with reasonable accuracy. We are going to do predictive analytic using Machine learning to find the expected value of power for the day. This is important because we aim to create an automation system which can control the devices inside a connected building. The AI will compare its performance as against the expected value for the given day. This will also be useful for strategic discharge of battery system aimed to reduce the peak power cost of a building.**

I. Introduction

The power consumption for any given day depends on the day it is with respect to week, month and year, the growth of the company relative to its first year and the outside temperature. The aim is to extract features from the data available and setup the machine learning problem to predict the power consumption within 5 % of the MAE of the historic power consumption. This will help us to have a measurable metric for the proposed reinforcement learning to compare its performance against the expected value for the task of optimization. There are also other factors like change in pattern because of billing structure, adding new machines to the system which will change the total landscape of the power consumption and could make even the historic data obsolete. We will tackle these problems later in the paper.

II. General Flow

The general flow is as discussed in Figure 1.

A. Data

The data contains the dataframe of power consumption in 15/30 min interval with time stamps and a separate dataframe containing information about the weather on a hourly basis. The weather data is taken from dark-sky website using api calls through python.

B. Feature Extraction

Currently the features used now in the machine learning algorithm are,

1) day of the week : One hot encoded or phase encoded
2) day of the month : One hot encoded or phase encoded
3) month of the year : One hot encoded or phase encoded
4) average temperature of a day
5) maximum temperature of a day
6) minimum temperature of a day

*This work was supported by Locbit, Inc.
[1]Daniel Maryanovsky is Chief Data Scientist at Locbit
[2]Premanand Kumar is a Data Scientist at Locbit
[3]Surya Krishna is a Data Science Intern at Locbit

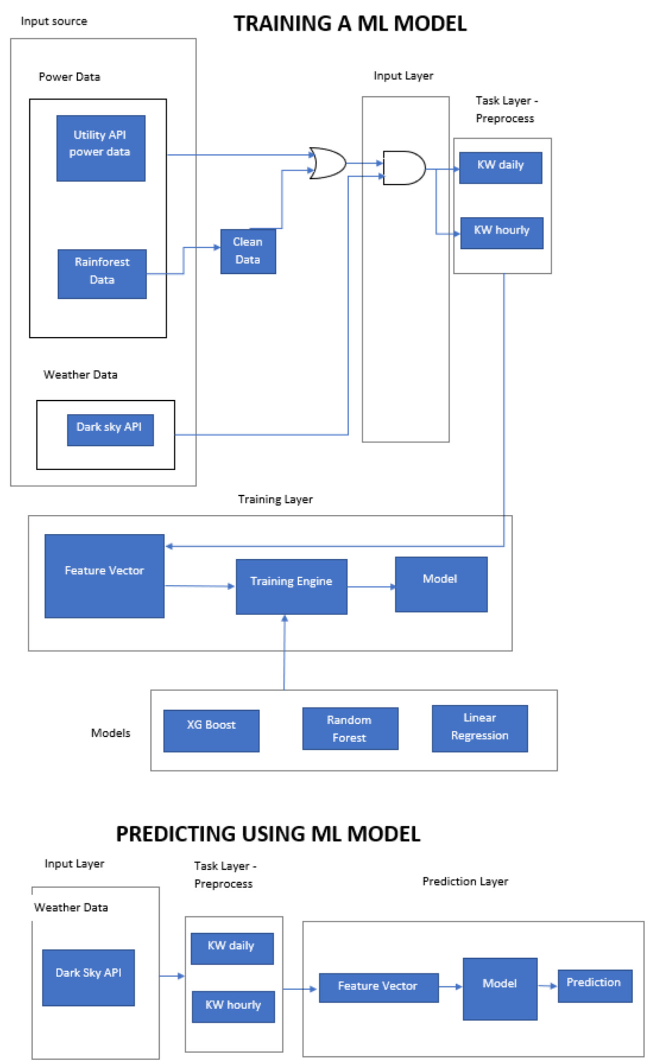

Fig. 1: General Flow.

7) average humidity of a day

The first three features are extracted from the timestamp while the remaining were taken from the temperature data from dark sky. The sum of all the interval kWh reading for any given day is our target value.

C. Data Splitting

The shuffled data is first split into 80-20 train and hold outset. The 80 percent train is further split into 80-20 train and validation set. The hyper parameter tuning will be done using train and validation set.

D. ML algorithms

We have the following Machine learning algorithms to choose from so far,

- Linear regression : The basic of ML algorithms which gives weight to each features and tries to figure out the final target value.
- Random Forest Regression : Since most of the features are one hot encoded, we chose Random Forest which can potentially mimic decision making as one of the features.
- XGBoost: This is a better variation of Random Forest and as expected produced similar results.
- Neural network: A simple two layered neural network for regression.

Apart from these we are also working on the following models to provide more weighting to the latest data as compared to the historical data.

- LSTM architecture to dynamically adjust the forecast based on recent values.
- CNN-LSTM architecture to capture both the local trend caused by business cycle and the overall climate correction.

III. Results on different sites

A. Zion - Kearny Mesa

This was by far the best, ideal site for our task. The unwavering cyclic nature of the business [fig 2] and the apparent uniformity even in the chaotic signals of daily power produced some astonishing results in predicting the power usage.

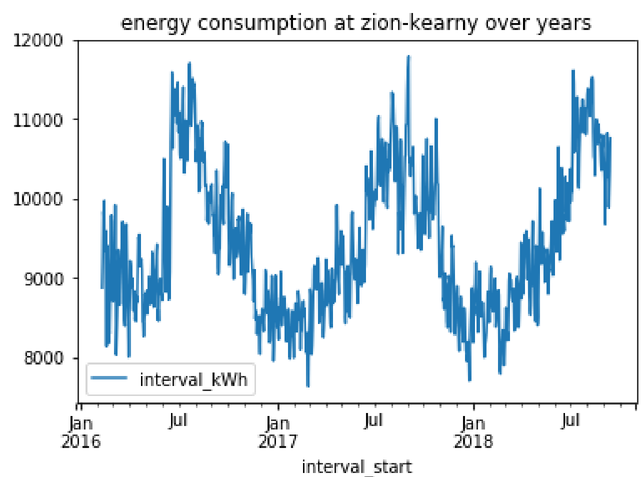

Fig. 2: Zion -Overall power consumption over the years

1) Training Phase

The training was done on utility-api data from 2016-02-10 to 2018-09-10. 5 fold cross validation was done on the training set and the validation set was used to identify the generality of the proposed solution. The results are as in table 1.

TABLE I: Training results on Zion - data from 2016-02-10 to 2018-09-10

Algorithm	kind	Accuracy*	r_score	MAE
Regression	Training	97.701	0.893	215.37
Regression	Validation	97.78	0.889	210.25
Random Forest	Training	97.119	0.805	269.94
Random Forest	Validation	96.68	0.713	314.53
XGBoost	Training	97.47	0.845	236.74
XGBoost	Validation	97.12	0.778	272.33

We can see that the simplest regression model has shown to be the best in all metrics due to data being relatively cyclic. So, the regression model was deployed into production.

2) Prediction

The prediction on production is as in figure 3.

Fig. 3: Zion - Prediction

Not only that prediction error was low (except on thanksgiving day), we were able to predict the general trend with greater deal of accuracy making the implementation very successful.

B. Stellar Care

Stellar care is a different business case as compared to Zion. Apart from heavy reliance on the weather, occupancy will also be key factor in determining the power consumption on any given day. However, hoping that the occupancy will remain constant/cyclic over the historic and future time, we went ahead with the model. The data we had was from 2016-12-30 to 2018-12-03 [figure 4]. We can see that there is no definite trend going on with the data and so the prediction task is going to suffer with the lack of additional information.

1) Training Phase

The training was done on utility-api data from 2016-02-10 to 2018-09-10. 5 fold cross validation was done on the training set and the validation set was used to identify the generality of the proposed solution. The results are as in table 1.

Though XGBoost produced the best result as comparable to zion, the lack of cycle is still a concern. Also the gap between the r_score of training and validation is alarming, indicating the lack of generality.

2) Prediction

The prediction for stellar care after it was employed is as in Figure 5. The error for the first two days were worryingly high. The model under predicted the power by 16-20 percent.

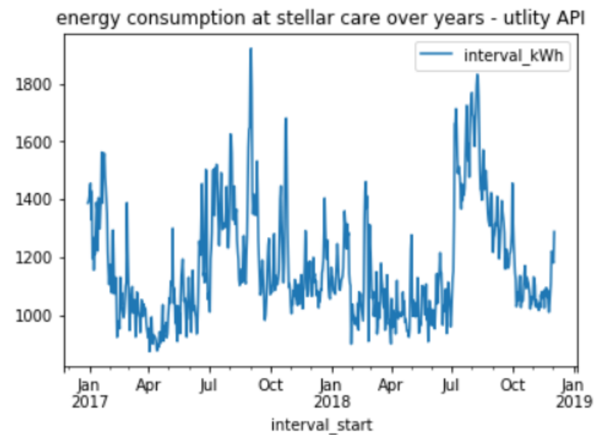

Fig. 4: Stellar Care General Power Consumption over years

TABLE II: Training results on Stellar Care - data from 2016-02-10 to 2018-09-10

Algorithm	kind	Accuracy*	r_score	MAE
Regression	Training	92.16	0.435	93.363
Regression	Validation	91.62	0.319	NA
Random Forest	Training	94.058	0.633	71.88
Random Forest	Validation	94.992	0.734	NA
XGBoost	Training	97.47	0.952	41.586
XGBoost	Validation	96.012	0.844	NA

A look at their power consumption did reveal an unusually high activity during the afternoon [table 2].

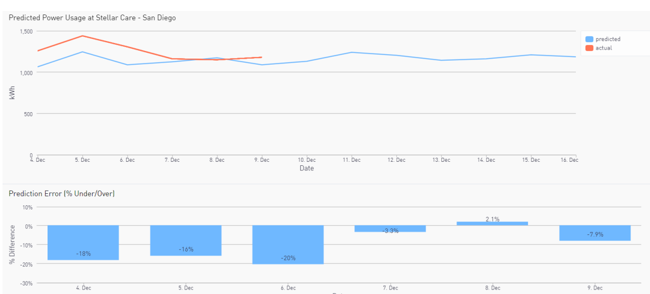

Fig. 5: Stellar Care Prediction

TABLE III: Stellar Care - Prediction

Date	Predicted	Actual	Error %	Figure
12/04/2018	1063.8	1255.9	-18	6
12/05/2018	1245	1441.8	-16	7
12/06/2018	1087.6	1307.9	-20	8
12/07/2018	1122.3	1158.9	-3.3	9
12/08/2018	1171.4	1146.4	2.1	10
12/09/2018	1090.5	1176.8	-7.9	11
12/10/2018	1132.9	1304.2	-15	12

From the table 2 and the respective figures, it is clear that stellar care has three activities

1) Active morning
2) sleep afternoon
3) wake night

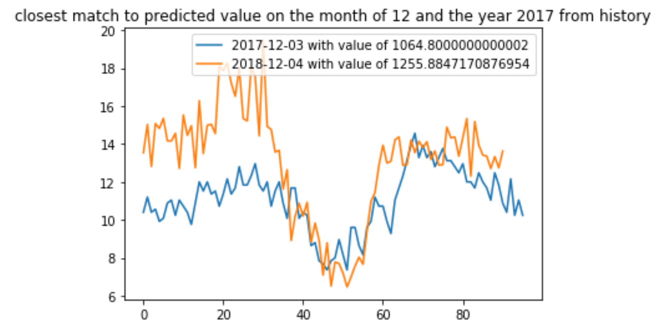

Fig. 6: Stellar Care - Day to Day Comparison

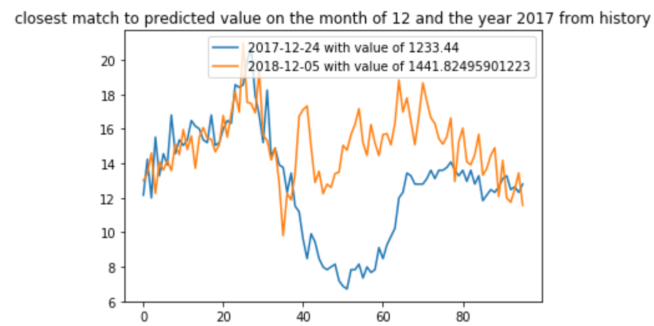

Fig. 7: Stellar Care - Day to Day Comparison

The predictions whenever they are off finds clear displacement in one of the three zones. Understanding why it happens will lead to,

1) Better prediction
2) Better optimization

IV. Sweet Water Gas

This is the third business case as compared to the other 2 we have seen so far. The general power consumption is as figure 13. Thought the general trend seems to be increasing, the data seems cyclic unlike stellar care and so, the model should work well.

1) *Training*

The results of the training is as in table 4. Unlike Zion, though we had good accuracy, we had poor r_scores. We went ahead with XGBoost which had reasonable r_score and a good accuracy.

TABLE IV: Training results on Sweet Water Gas

Algorithm	kind	Accuracy*	r_score	MAE
Regression	Training	93.31	0.530	21.62
Regression	Validation	91.74	-0.022	26.52
Random Forest	Training	93.35	0.412	21.48
Random Forest	Validation	94.992	-0.09397	24.76
XGBoost	Training	97.99	0.964	6.47
XGBoost	Validation	95.82	0.499	13.43

2) *Prediction*

The prediction for sweet water gas is as in figure 6. The prediction seems to deviate from the actual value for the dates between 3rd December to 11th December as in table

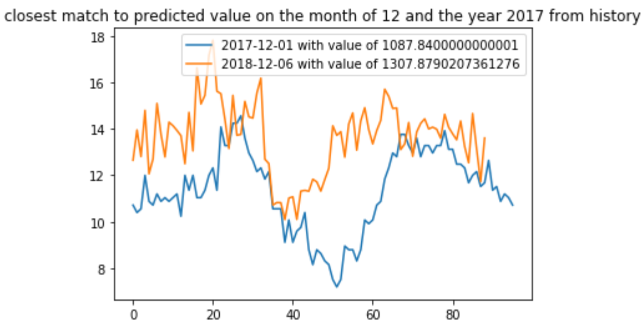

Fig. 8: Stellar Care - Day to Day Comparison

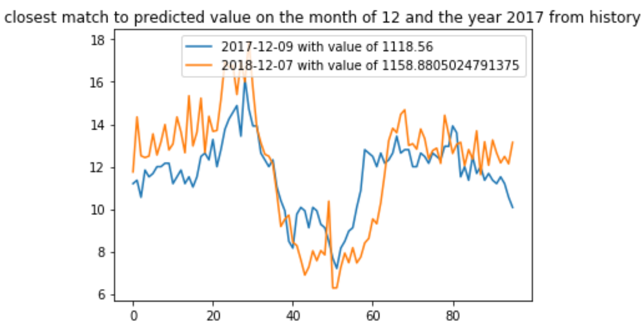

Fig. 9: Stellar Care - Day to Day Comparison

5. The values recorded as historic high for december[Figure 24]. So lets analyze the difference in pattern after that date. It is clear that over the time in a single day, the power consumption increase from midnight to midnight and lately by comparison the site has developed localized peaks during early morning as compared to the history and also the troughs are not as low as it used to be. Though we do not have immediate reasoning for those happenings, we are confident that it could be explained by some behaviour that could lead to better prediction/optimization.

TABLE V: Sweet water gas - Prediction

Date	Predicted	Actual	Error %	Figure
12/03/2018	299.77	281.06	-6.7	15
12/04/2018	315.53	290.55	-8.6	16
12/05/2018	310.3	286.26	-8.4	17
12/06/2018	318.52	294.96	-8	18
12/07/2018	322.34	290.88	-11	19
12/08/2018	316.02	284.89	-11	20
12/09/2018	304.33	282.71	-7.6	21
12/10/2018	309.97	288.92	-7.3	22
12/11/2018	315.52	287.09	-9.9	23

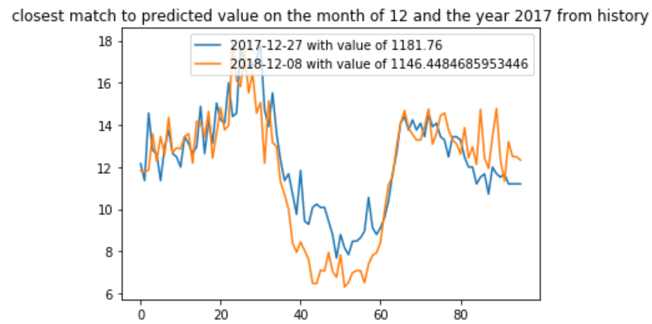

Fig. 10: Stellar Care - Day to Day Comparison

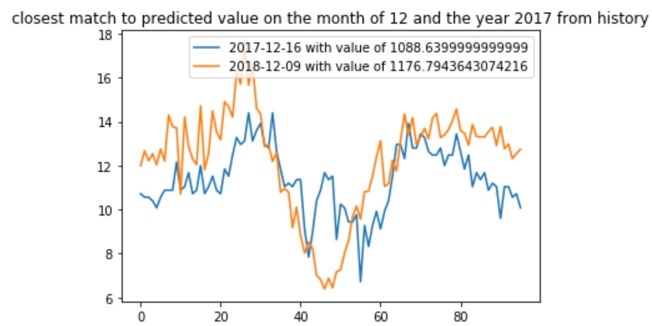

Fig. 11: Stellar Care - Day to Day Comparison

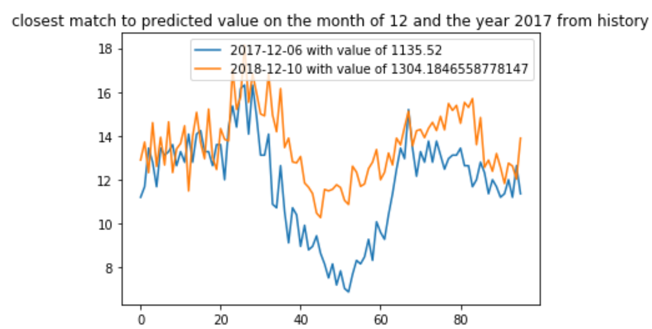

Fig. 12: Stellar Care - Day to Day Comparison

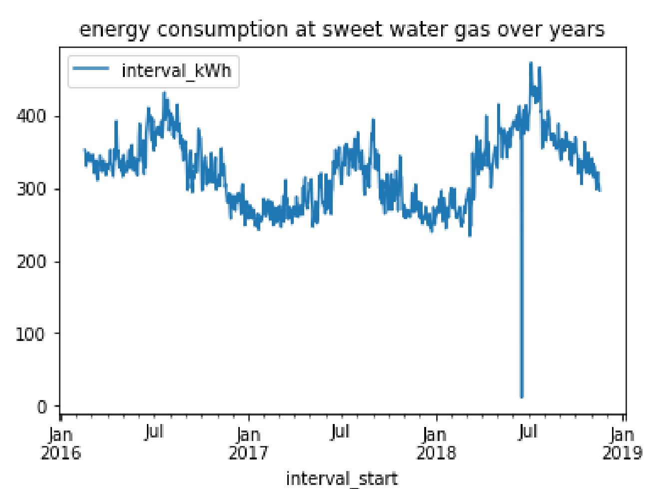

Fig. 13: Sweet Water General Power Consumption over years

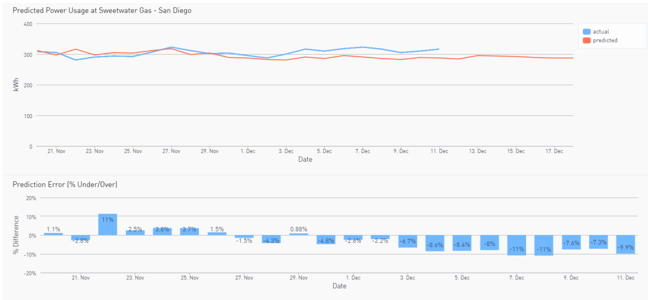

Fig. 14: Sweet Water Prediction

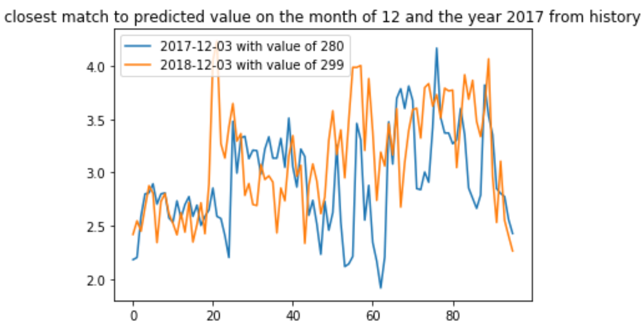

Fig. 15: Sweet Water - Day to Day Comparison

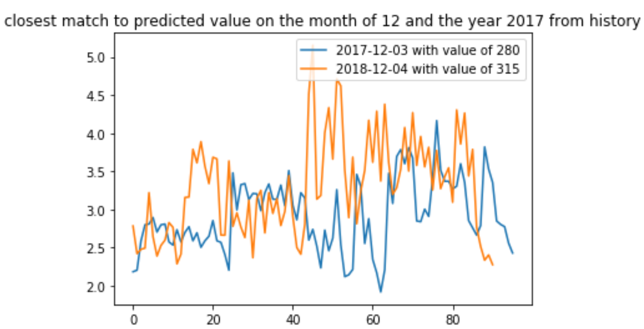

Fig. 16: Sweet Water - Day to Day Comparison

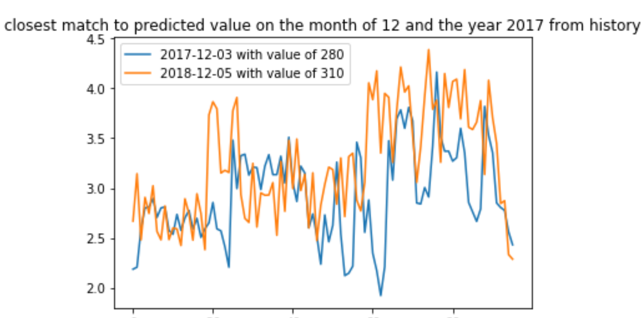

Fig. 17: Sweet Water - Day to Day Comparison

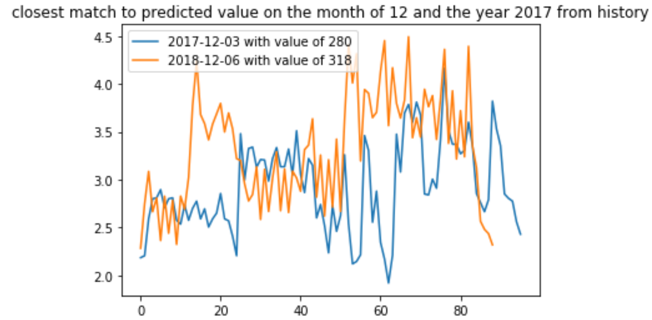

Fig. 18: Sweet Water - Day to Day Comparison

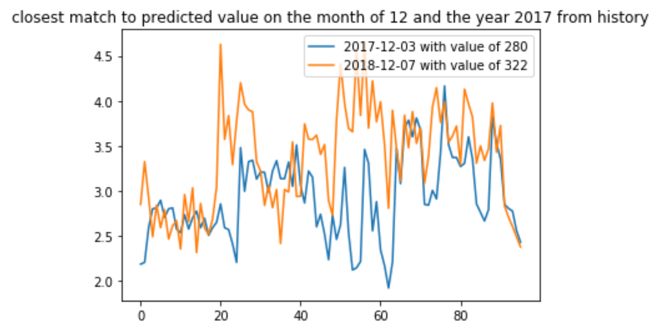

Fig. 19: Sweet Water - Day to Day Comparison

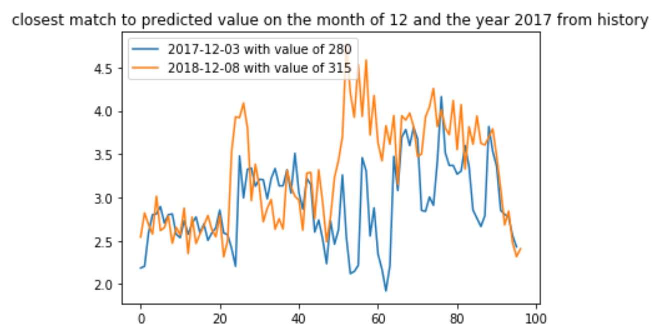

Fig. 20: Sweet Water - Day to Day Comparison

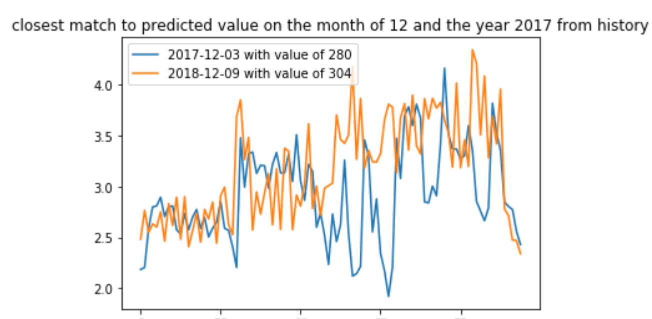

Fig. 21: Sweet Water - Day to Day Comparison

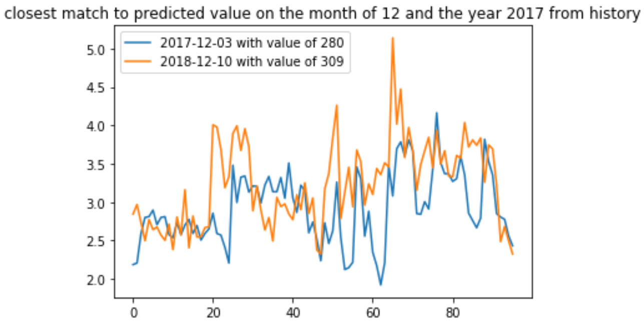

Fig. 22: Sweet Water - Day to Day Comparison

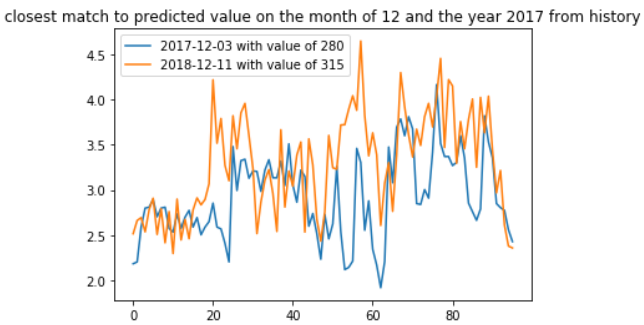

Fig. 23: Sweet Water - Day to Day Comparison

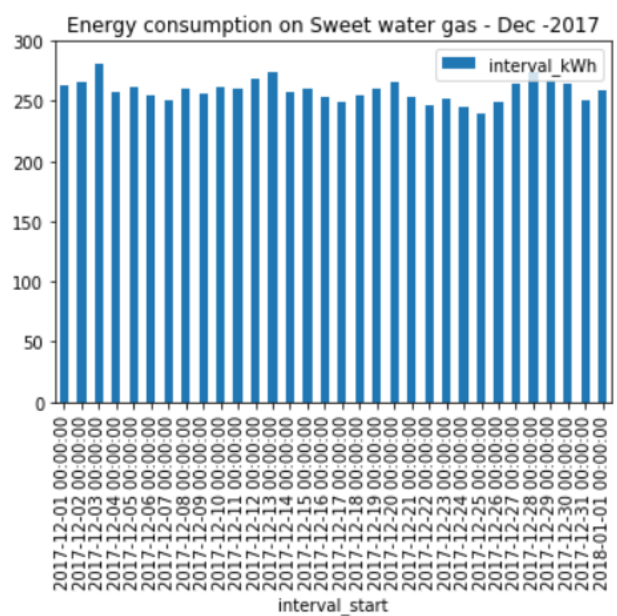

Fig. 24: Sweet Water - Day to Day Comparison