Energy Consumption Forecasting

Rastegar, Arvin
arvin.rastegar@studenti.unipd.it

September 21, 2022

1 Introduction

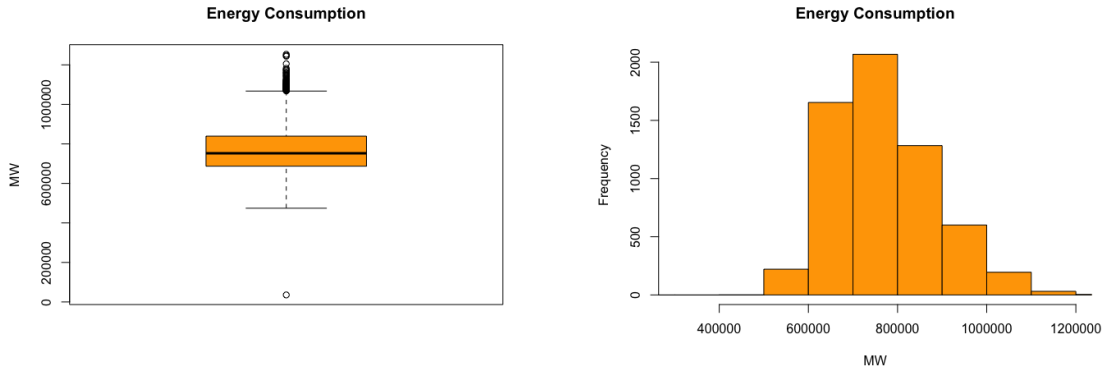
In this project, we will walk through time series forecasting using XGBoost. The data we will be using is hourly energy consumption. When we want to estimate energy consumption, we usually have factors in mind, for example, seasons, different times of day, and geographical location. We aim to forecast the correct energy required for each hour and determine important variables in this respect.

1.1 Dataset

The data presented in this challenge are the hourly power consumption data which comes from PJM's website and are in megawatts (MW). The regions have changed over the years, so data may only appear for certain dates per region.

Datetime	**PJME_MW**	**DATE**
2002-01-01	714857	2002-01-01 01:00:00
2002-01-02	822277	2002-01-02 01:00:00
2002-01-03	828285	2002-01-03 01:00:00
2002-01-04	809171	2002-01-04 01:00:00
2002-01-05	729723	2002-01-05 01:00:00

We can see the dataset aggregated on days in the above table for future reference.



(a) Box Plot for Energy

(b) Histogram for Energy

Figure 1: Hourly Energy Consumption per day

The data is from 2002-01-01 until 2018-08-03. The minimum energy consumption per day is 35486 MW, and the maximum is 1253516 MW usage per day. By looking at the histogram, we can see that the data is right-skewed since we use a lot more energy on some days.

1.2 Exploring

First of all, we plot the hourly data to observe the data throughout time as shown in the figure 2.

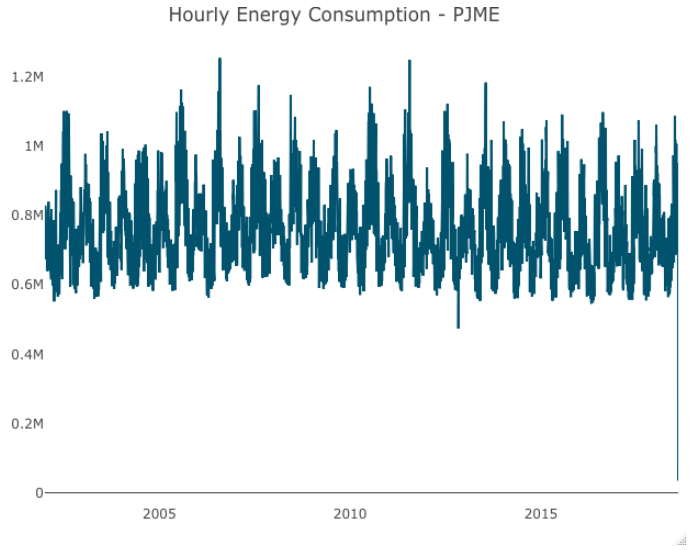


Figure 2: Energy Consumption plot.

Then we decompose by the additive model as shown in the figure 3 which has this formula:

$$Y_t = T_t + S_t + e_t$$

Y_t is the observed values, T_t is the trend, S_t is the seasonality and e_t is the random values in the decomposition.

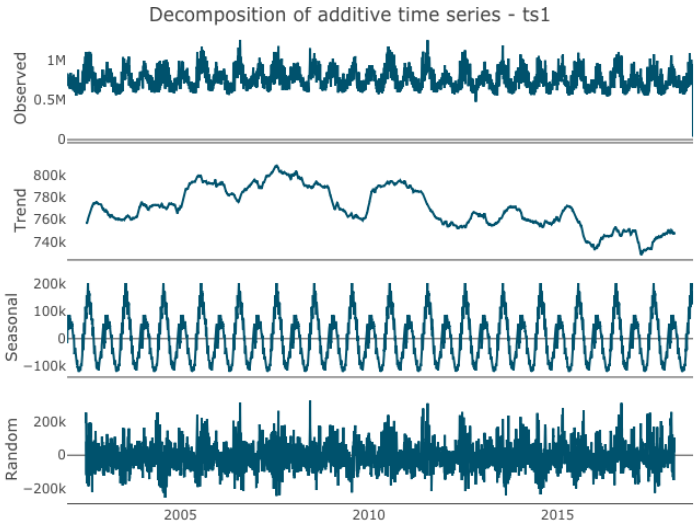


Figure 3: Decomposition of time series.

A downtrend in consumption and a clear correlation between seasons and electricity usage is observable.

Then we plot the data according to different months of the year to understand it more as shown in the figure 4.

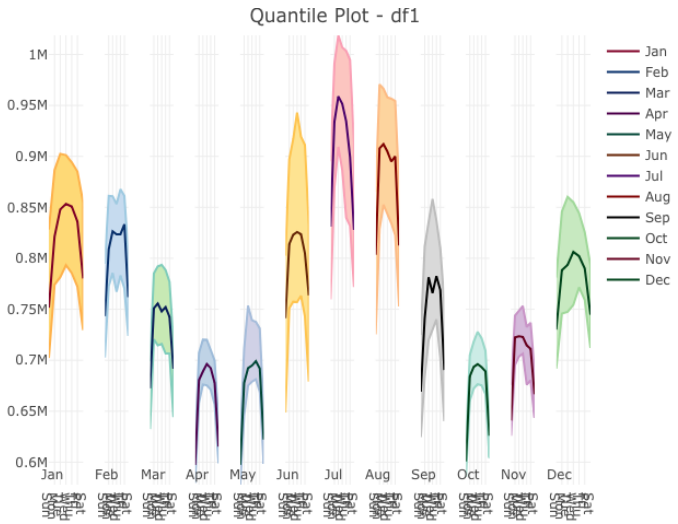


Figure 4: Monthly E.C. across 16 years.

We have a maximum usage in our data in July and minimum consumption in April.

Since we saw a lot of autocorrelation with seasonal fluctuations we also plot the AFC plot as shown in the figure 5.

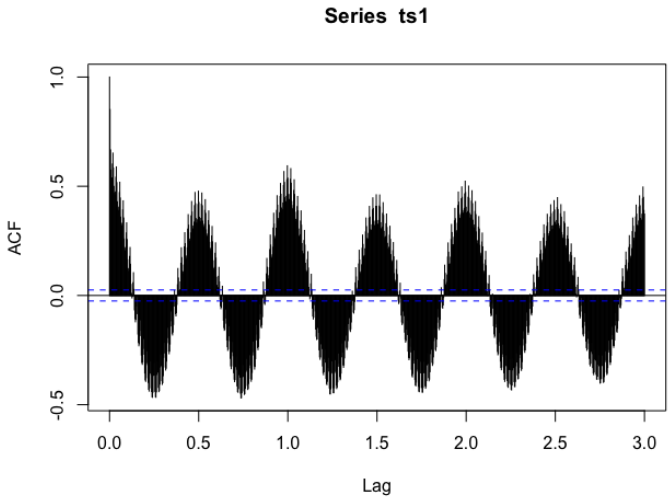


Figure 5: Serial Correlation in data.

As shown above, we can see the correlation changing from positive to negative with the change of the season. We call this seasonal lag, and we can use it for forecasting.

1.3 Pre-processing

We split the data set into two sets the training set and the testing set. The training set contains information from 2002 until the end of 2014, and the testing set incorporates the rest of the dataset.

1.4 Feature engineering

We understood that our data has a correlation with seasons, and it has a downtrend. We can now incorporate our knowledge into feature engineering of the data before feeding it to our models only for data available throughout the year.

Datetime	PJME_MW	weekday	month	lag365
2003-01-01	662489	Wed	Jan	714857
2003-01-02	770850	Thu	Jan	822277

2 Methods

We use two different models, the first one the simple linear model and then the gradient boosting model.

2.1 linear regression model

Our first model is a simple linear regression model. We train the model and then predict the test dataset. We have the following results Multiple R-squared: 0.5906 and Adjusted R-squared: 0.5889. These are good theoretical results explaining almost sixty percent of the variance in our dataset. However, we plot the model to see if it behaves well enough and whether it can be a good model for our dataset, as shown in the figure 6.

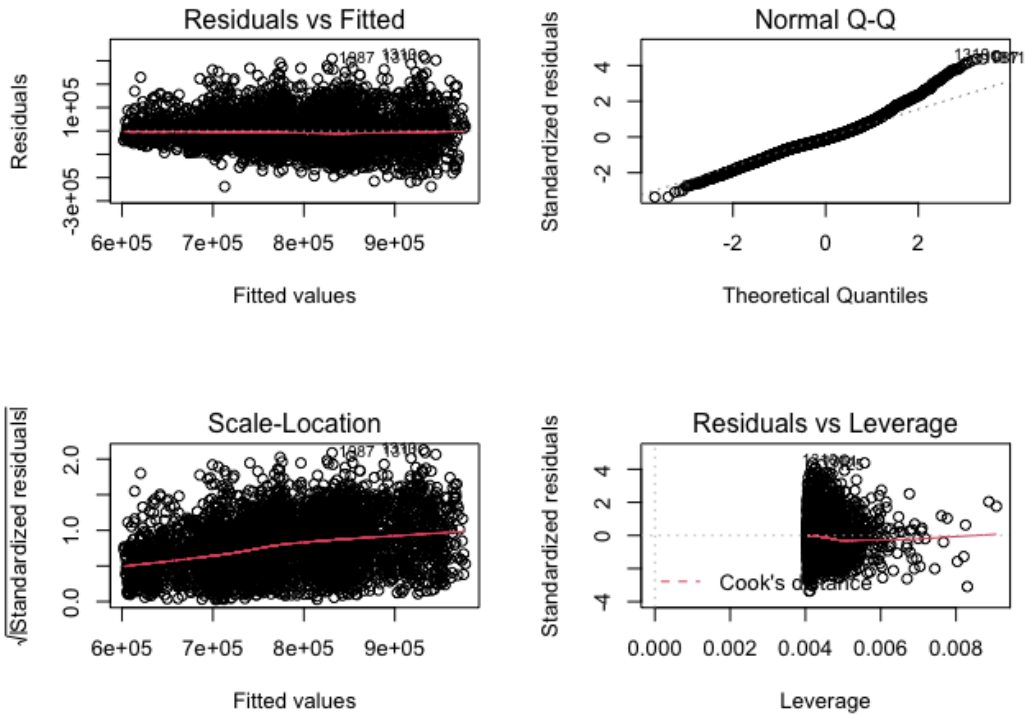


Figure 6: Linear Model.

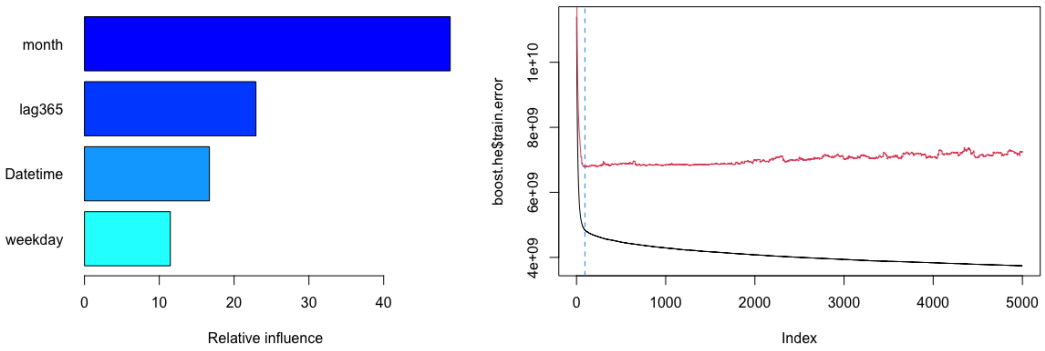
We see the sum of residual errors is almost zero, but the distribution is not normal. We can also see that the residuals are not independent of each other. This leads us to believe that we can find better models for this dataset. The RMSE metric error of this model is 7312383003.

2.2 XGBoost model

In this section, we implement gradient boosting models with different parameters, for example, interaction depths and learning rates. To see the importance of each parameter on the outcome of our model and to forecast energy consumption more realistically.

In the first model, we use the simplest model of depth one without a shrinkage parameter. The importance of variables is illustrated in the figure 7a.

We also plot the errors at the testing time vs. training time to avoid overfitting and to find the best model for predicting the response variable, as shown in the figure 7b.



(a) Feature importance

(b) Train vs Test error

Figure 7: The model with the depth of one.

We can see that the model's most effective parameter is the Month, and we have a minimum error of 6786110751, and the best model is after 88 iterations.

We perform further experiments to see which model is the best.

Depth	Shrinkage	Iterations	Min Test Error
1	Null	88	6786110751
4	Null	35	6962399594
1	0.01	1987	6791631725
4	0.01	478	6963849366
Linear	Null	0	7312383003

We can see that the model with depth one and no shrinkage achieves better results, and the model with depth 4 converges after 35 iterations.

8

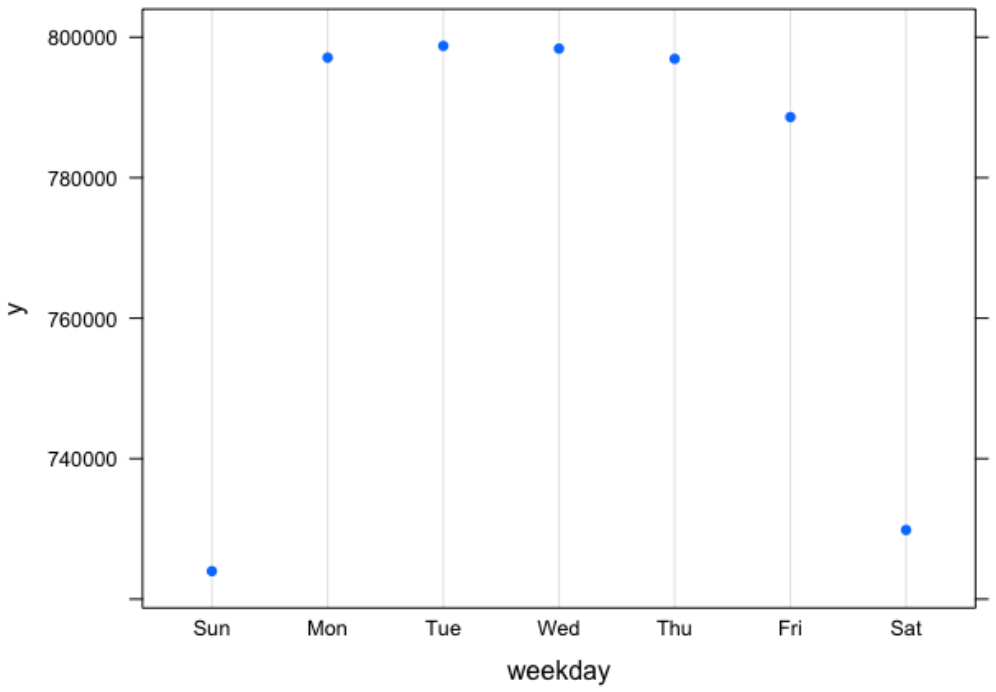


Figure 8: The effect of weekdays on the prediction.

We can see that during the weekend we have less consumption.

3 Conclusion

In this project, we went through the energy consumption of a region in the United States. We analyzed the data to see trends and correlations to better predict the future needs of this region. We came across some connections between seasons, days of the year, weekdays, and energy consumption. We implemented a model to forecast the future needs of the grid and to understand better the variables affecting them.

We saw the most important feature across our different models was the month of the year. Additionally, we saw that gradient boosting with depth one and 88 iterations gave us minimum test error, thus more realistic than our other models.

https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption