ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

Title: ETM 510: final project report

Course Title: ETM energy forecasting

Course Number: 510

Instructor: Dr. Tim Anderson and Dr. Sule Balkan

Term: Winter 2016

Year: 2017

Author(s): Michael Davis

Introduction

This class was a special reduced class time hands on course that centered around the Global Energy Forecasting Competition which was an international forecasting competition hosted Dr. Tao Hong using real data on electricity demand. The New England region was the point of origin for the real data and was broken up into several regions including: Maine, New Hampshire, Vermont, Connecticut, Rhode Island, 3 regions of Massachusetts, and the combined region ISO NE CA. The competition was broken up into two brackets based on the amount of data that the participant intended to use which was either two years of data or an unconstrained amount of data so that the participants could utilize all presently available data. The participants could also choose to work in a team or not. I choose to work with my fellow classmate, Mohamed Almusallam, with whom I had collaborated successfully in the past. We choose to utilize the constrained data set of two years.[This report mentions this collaboration solely so that the reviewer of this report will take cognizance that all efforts in this report share only a partial account of the work done this term and represent in most cases my own first attempts to create energy forecasts that in the later cases(especially the neural network) were further improved with help of my collaborator]. With that mentioned, this report will summarize and examine the energy forecast attempts that I put forth over the term.

Historical data

Before jumping into the discussion of the various forecasts made over the term, something must be said about the data. Figures 1 and 2 show the the historical data real time(RT) energy demand that the competition afforded us for the month of March which was the focus month for the first few forecasts. Notice in figure 1 that the demand in the early part of the month dips to zero. This is due to an unusual blackout that occurred and was a statistical outlier so that it was common practice in this class to simply remove this data point so as to not skew the data.

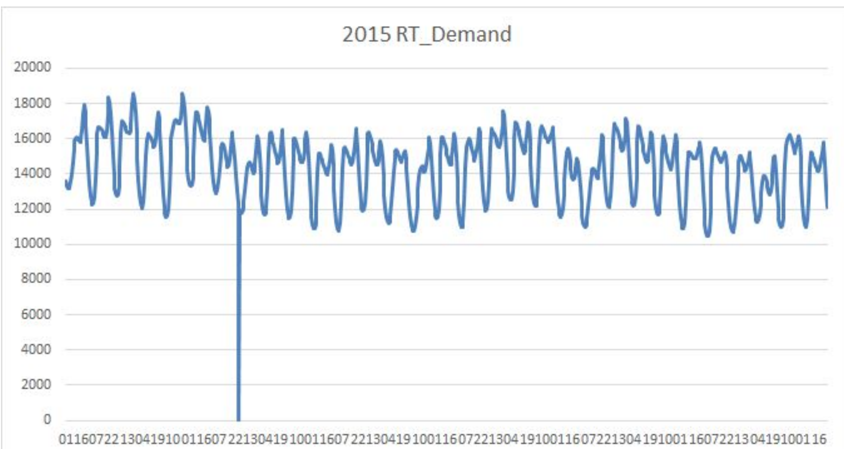

Figure 1: 2015 RT demand

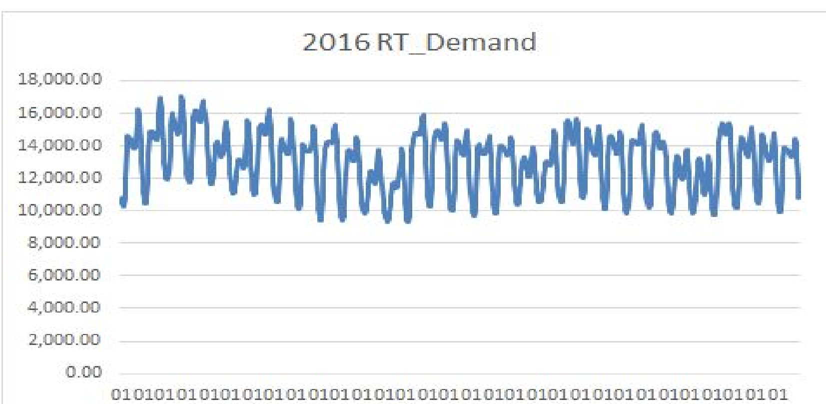

Figure 2: 2016 RT demand

<u>First method: Naive method</u>

This method is only appropriate for time series data and seemed appropriate for RT-Demand vs. time. All forecasts are set to be the value of the last known observation in the time period under consideration. That is, the forecasts of all future values are set to be y_T , where y_T is that last observed value. Notice how the forecast is simply a straight line. This is the limit of the Naive Method. The first attempt used only March 2016 data(See figure 3) and the second attempt(See figure 4) used an average of March 2015 and March 2016 data to make the forecast. In R, the forecast package was used and the naive function was used.

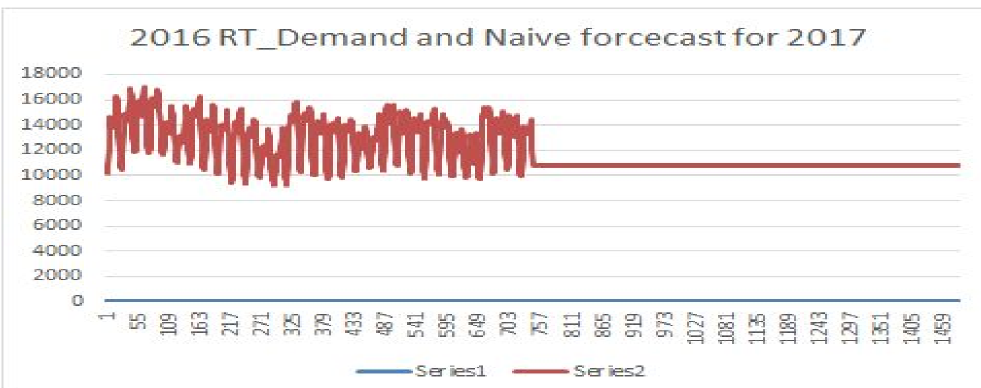

Figure 3:

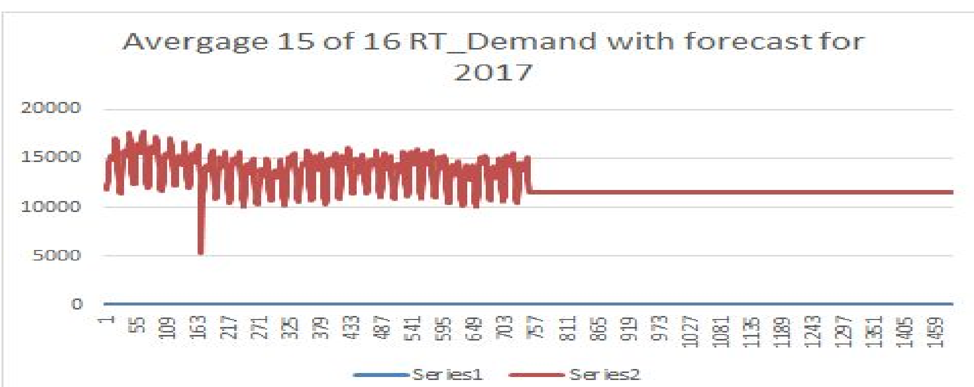

Figure 4:

This method fails to capture seasonality or daily trends so without a detailed further analysis it's obvious that it’s limited as a forecasting methodology. Nevertheless, this method could have been more effective if one was to limit the amount of data used. For instance if you were to restrict your historical data time length entry to a single day and then use that to build your monthly forecast with a succession of such days one would be left with a fairly good forecast that would track both daily trends and seasonality. This could especially be true if one were to use the average of the last two years data for each day and then use the result to build your monthly forecast with a succession of such days.

Second method: averaging method

After seeing the limitations of the naive method, a new approach was taken which was to build a forecast with simply the average of the last two data sets. It performed better analytically than the naive method because by averaging the prior two year data sets it was able to capture any daily trend that was present along with any general trend between the two years.

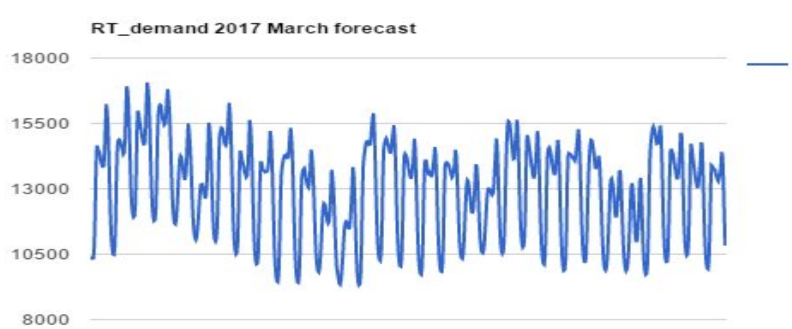

Figure 5: Averaged demand forecast.

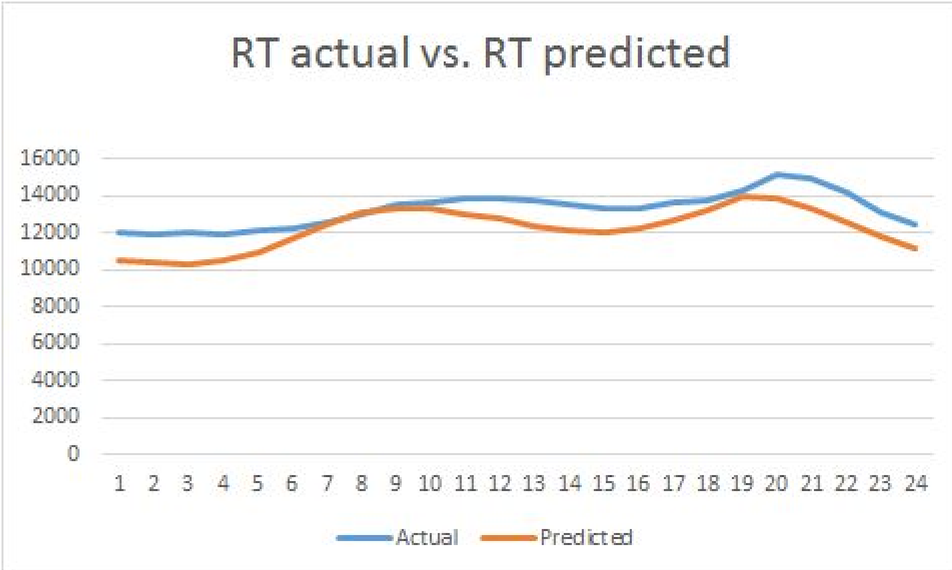

Figure 6: Actual march 19th data vs. predicted march 19th data

Analyzing the two data sets , we can see that this method yielded a Mean Absolute Percent Error(MAPE) range from a low 0.66% to a high of 13.8%. So that it was fairly accurate forecast.

Third method: exponential smoothing

The third attempt used an exponential smoothing(ES) method. The ES model attaches more weight to recent observations than to past observations so that the forecast is simply calculated using weighted averages where the weights decrease exponentially as observations come from further in the past [1]. To use the ES, the data needed to be treated before insertion into the function. In this case the data needs to be differenced to ensure data is stationary(A stationary time series is one whose properties do not depend on the time at which the series is observed so that any time series with trends, or seasonality, are not stationary). In R, the ETS function in the forecast package was used which selected the optimal exponential smoothing method based on the data(Note: I did not retain the original code that I showed in class so I have recreated the forecast and showing the data in Excel. See figure 7)

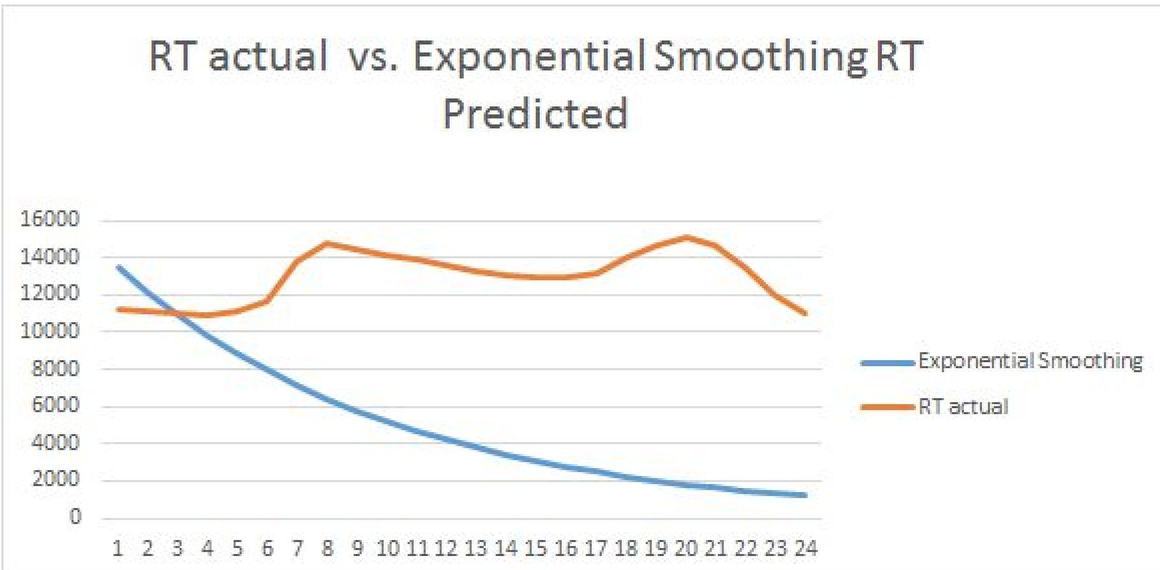

Figure 7: Exponential smoothing prediction

With an $\alpha = 0.1$, this was the prediction for march 20th 2017 . You can see that this prediction was not very accurate. Its MAPE ranges from a low of 0.6 % to an absurd high of 89.6% .

Fourth method: Neural network

A neural network can be thought of as a network of neurons organized in layers. The inputs form the bottom layer, and the forecasts form the top layer.[2]

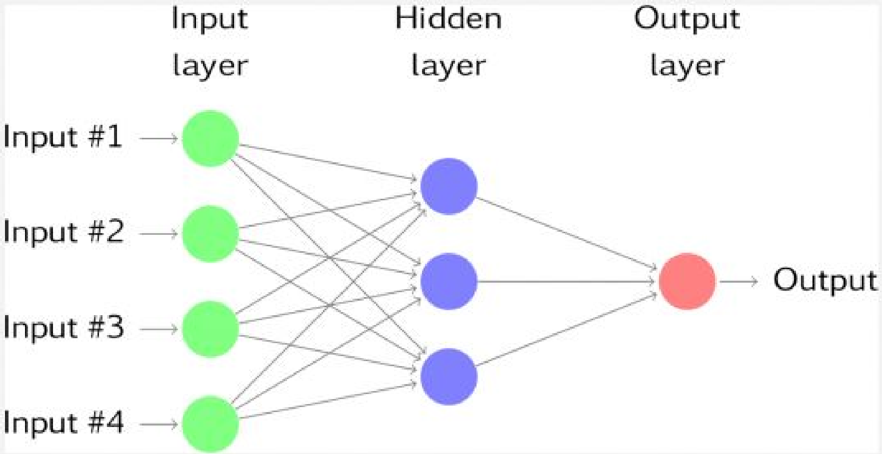

Figure 9.10: A neural network with four inputs and one hidden layer with three hidden neurons.

Figure 8: Neural Network layers [2]

Utilizing this framework, the fourth method was created using the nnetar function which fits a NNAR model to the data. I used only one week of data at a time to avoid extraneous error in my model and added the successive weeks(See figure 9) together to create the April forecast. Looking at the different weeks in figure 9, notice that the green portion of the data is the training set where the neural network(NN) analyzed the past data to create the forecast which is featured in blue. You can note that the NN was able to capture daily trends such as weekend most times(observe the lower two peaks in each week which correspond to Saturday and Sunday) .

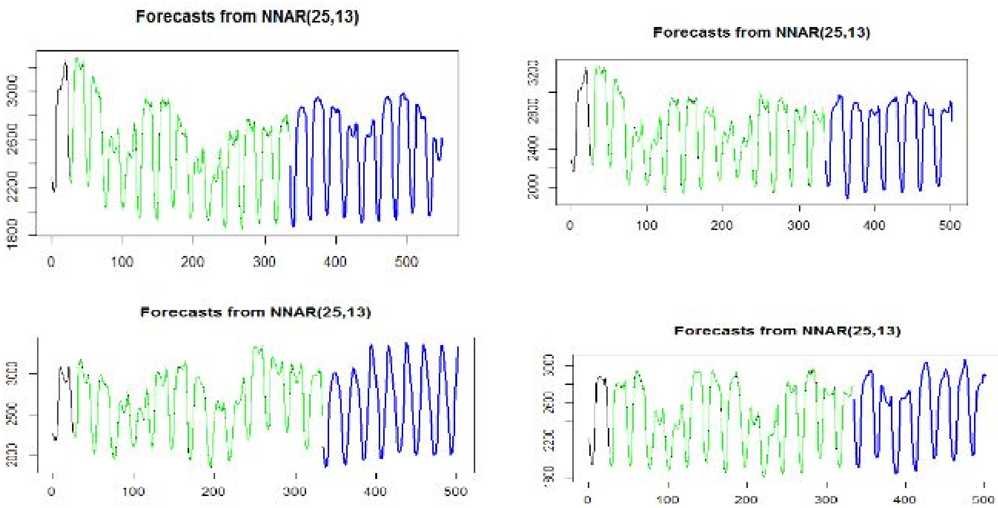

Figure 9: Weekly Neural Network forecast for April

However looking at the combined forecast(See figure 10), it should be noted that the third week shows an abnormal skew toward higher values of demand. It is unknown why that week skews high but this was one of the deciding factors that played into the decision to move into a Neural network that included extraneous variables in addition to demand(which I will leave to my collaborator to discuss in greater detail) .

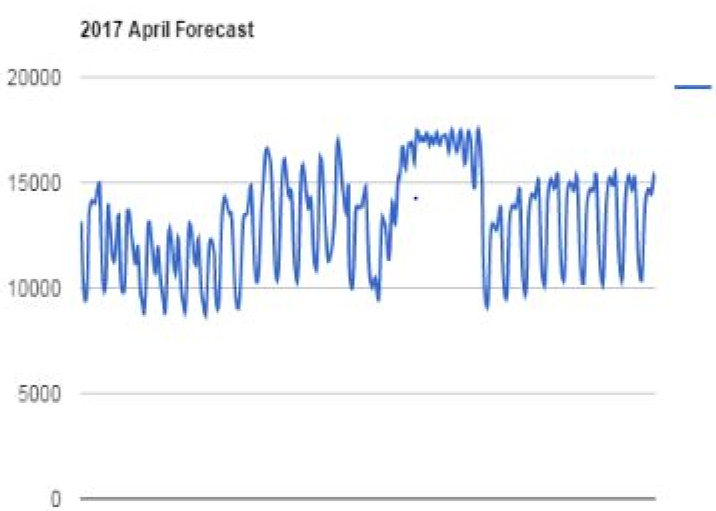

Figure 10: Combined Neural Network forecast for April

Conclusion

Overall, this was an exciting class that expanded my knowledge of how energy demand forecasts are created.

Reference

[1] Hyndman, Rob J., and George Athanasopoulos. "Section 7.1." Forecasting: Principles and Practice. Heathmont: OTexts, 2016. N. pag. Print.

[2] Hyndman, Rob J., and George Athanasopoulos. "Section 9.3." Forecasting: Principles and Practice. Heathmont: OTexts, 2016. N. pag. Print.

<u>Appendix-Code:</u>

Neural network:

```
fit.day <- nnetar((DayDemand.day))

plot(forecast(fit.day,h=24))

points(1:length(DayDemand.day),fitted(fit.day),type="l",col="green")


oo<-1

k<-1

b<-1


DayDemand.week<-c()

for(oo in 1:334){

  if (oo <= 167){

    DayDemand.week[oo] <- usedata2015.week[b,1]

    b <- b+1

  }

  if (oo > 167){

    DayDemand.week[oo] <- usedata2016.week[k,1]

    k <- k+1
```

```
  }
}

  fit.week <- nnetar((DayDemand.week))
  fit.week1<-forecast(fit.week,h=168)
  plot(forecast(fit.week,h=168))
  points(1:length(DayDemand.week),fitted(fit.week),type="l",col="green")


  usedata2016.week<-ts(X2016_NEMA[2185:2353,2])
  usedata2015.week<-ts(X2015_NEMA[2161:2329,2])
```