From: **McMenamin, Stuart** Stuart.McMenamin@itron.com
Subject: Research Data
Date: May 13, 2024 at 10:22 PM
To: jakemkosakoff@gmail.com

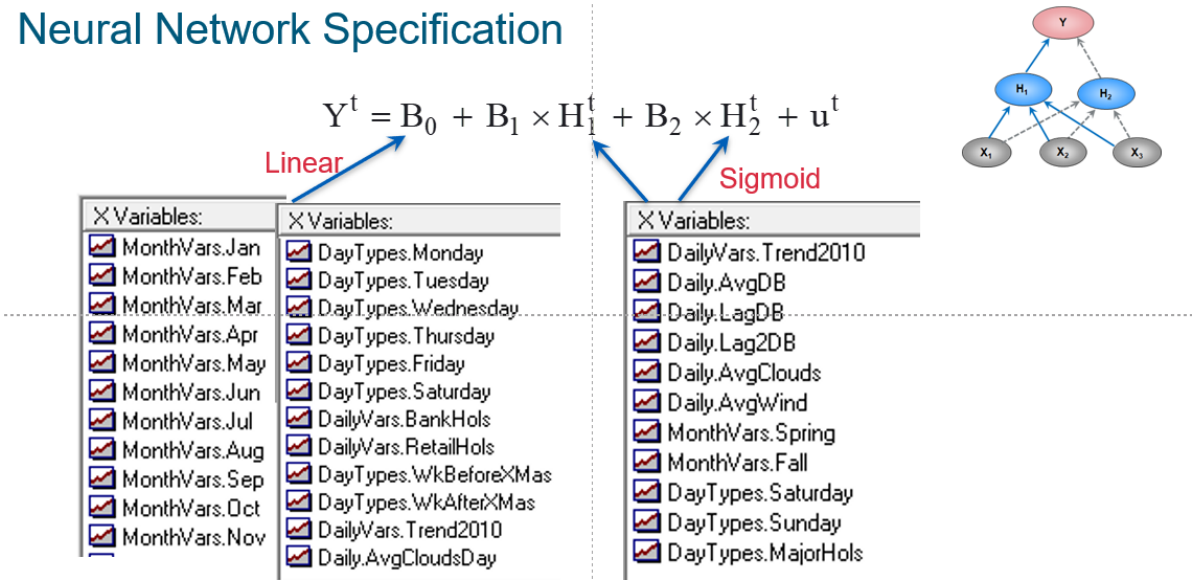
SM

Jake:

Here you go.
Please respond so I know you got the email.

In the spreadsheet, the first tab (RegData) is the data we used for regression models with all the features specified.
The second tab (NetData) has the X variables we used in the Neural Network models.
The last tab has the X variables for the Boosting step.
These are simpler more aggregated data, like day of the week (1 to 7 where 1 is Monday).
There is another tab (AggData that is the same as BoostData but day of week is like Excel with 1 for Sunday (doesn't work as well).
Nothing proprietary about the data.
The Load data are available from FERC (government) and PJM (the big east coast ISO).
The weather data are available from NOAA (government) although we got these specific data from a weather vendor (who got it from NOAA).

Here is a picture of how we used these variables in a neural network.
It is a 3 layer feed forward neural network.
In the middle (hidden) layer, the first node is linear (B0 = linear in the X variables).
The remaining nodes are nonlinear and all include the same variables and use a sigmoid (logistic) transfer function.
Usually 2 to 5 nonlinear nodes works the best.



Energy is daily energy (the sum of hourly loads) for Virginia Power, a subsidiary of Dominion Energy.
AvgDB is average drybulb temperature computed from hourly data for 4 weather stations.
AvgWind is average windspeed in MPH.
AvgClouds is average daytime clouds (8 am to 8 pm) in Octas (0 to 8) where 0 is clear

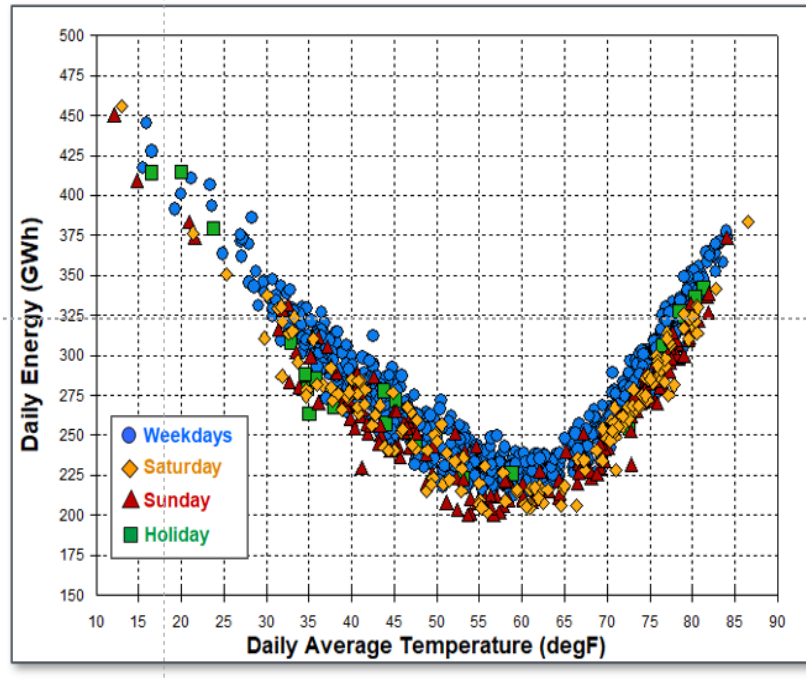
AvgClouds is average daytime clouds (8 am to 8 pm) in Octas (0 to 8) where 0 is clear and 8 is very cloudy.

I included a couple of Jupyter files, one for Regression and Boosting and one for Nets and boosting.
The regression one shows how to set it up with a loop that includes regression model estimation followed by residual boosting.
The Net one reads in the neural network residuals from 100 train/test/split runs and then applies the boosting step.
We used 10% as the out of sample fraction for the runs.
All the performance stats (in sample and out of sample) are computed following estimation.

Just so you know what the data look like, here is a scatter plot of the 1,095 observations.
Average temperature on the X axis, daily energy on the Y axis, color coded by day type.

Overview of Data

- » Daily energy in GWh
  - PJM Data for Dominion
  - Daily data for 2017 to 2019
  - N=1,095 observations
  - All pre COVID
- » Hourly weather data
  - Dry bulb temperature (deg F)
  - Wind speed (mph)
  - Cloud cover index (0 to 8)
- » Calendar variables
  - Month
  - Day type
  - Holidays



Let me know if you have any questions.
I will be interested to see what you come up with using Python Nets.

Dom_NetBoost-checkp...t.ipynb
Dom_RegAndBoost.ipynb
DomDaily.xlsx