EN.553.753
Junzhou Lin & Liangzi Zhu

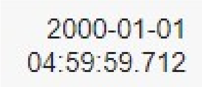
Project 2[Detail is in the coding provided]

Part One:

We have encountered two problems about the data during this process.

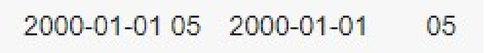
The first problem is when converting “Date(HourEnding)” to a data frame, the time variable will not be at an exact hour because it might have some milliseconds off. We created a function to round the time to an exact hour.

Prior



2000-01-01 04:59:59.712

After

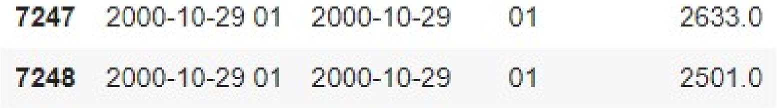


2000-01-01 05	2000-01-01	05

The second problem is that we found even the length of two data frames are the same, but one is ending on 2018/08/08 7pm and another is ending on 2018/08/10 0am. With this been said, there are at least 29 rows of error data.

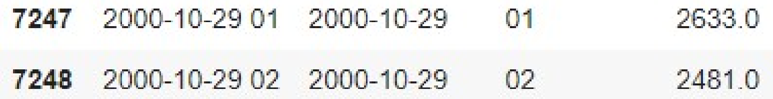
For the ‘Load’ data frame, there are 18 rows have repeated ‘Date(HourEnding)’ but with a different ‘Load’ variable, we just delete the repeated one.

Prior



7247	2000-10-29 01	2000-10-29	01	2633.0
7248	2000-10-29 01	2000-10-29	01	2501.0

After

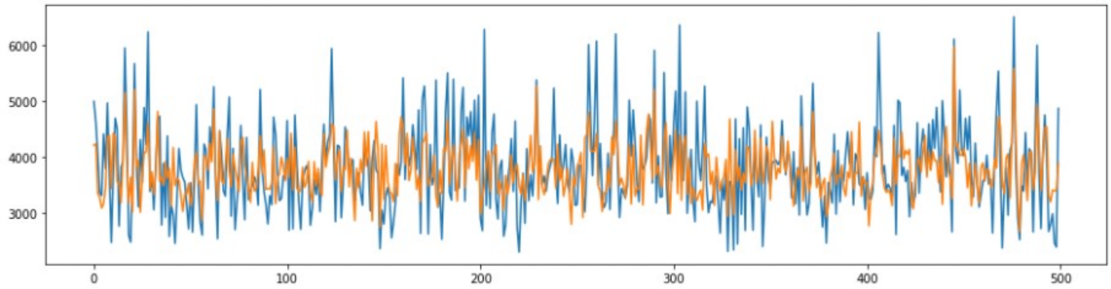


7247	2000-10-29 01	2000-10-29	01	2633.0
7248	2000-10-29 02	2000-10-29	02	2481.0

For the ‘Temperature’ data frame, not every hourly data is provided. We add one hour for ‘Date(HourBegining)’ to match with the ‘Date(HourEnding)’ then Merge two data frame on “Date(Hour)” so the whole row will be dropped if any data with a same ‘Date(Hour)’ is not provided in both data frame .

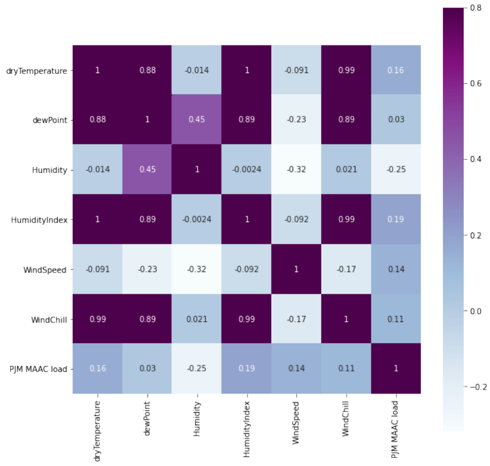
Part Two:

At the first try, we were trying to build the model through a linear regression without thinking too much. But only ended up with a 0.4047 R^2.



After ploting the Load vs. Temperature graph, we realized that nonlinear regression model might be the correct approach to this project. The graph shows an quadratic trend since warm temperature in the middle usually have a small amount of load and the more extreme temprature on either side have a higher amount of load.

Ploting a heat map is also helpful to see the correlation between each parameters and the load, since we want to consider the possibility of taking out useless (or not that useful) parameters.



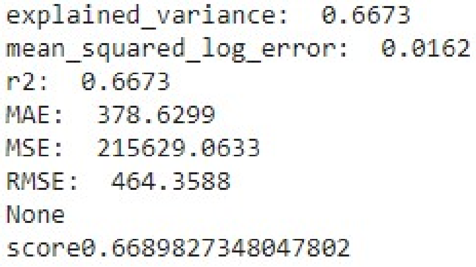
	dryTemperature	dewPoint	Humidity	HumidityIndex	WindSpeed	WindChill	PJM MAAC load
dryTemperature	1	0.88	-0.014	1	-0.091	0.99	0.16
dewPoint	0.88	1	0.45	0.89	-0.23	0.89	0.03
Humidity	-0.014	0.45	1	-0.0024	-0.32	0.021	-0.25
HumidityIndex	1	0.89	-0.0024	1	-0.092	0.99	0.19
WindSpeed	-0.091	-0.23	-0.32	-0.092	1	-0.17	0.14
WindChill	0.99	0.89	0.021	0.99	-0.17	1	0.11
PJM MAAC load	0.16	0.03	-0.25	0.19	0.14	0.11	1

Part three:

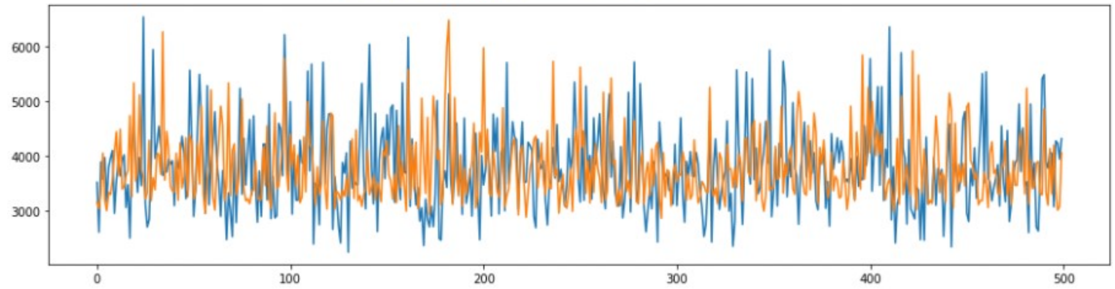
With y as the “Load”, we tried different combination of parameters for x. In conclusion, the model with all parameters included will give the best result (hightest R^2)

Our model is built through pipiline with LASSO as a type of Regularized regression. Most importantly, we used Scikit-learn PolynomialFeatures for automatically generate polynomial features from a set of linear features, with degree range from 2 to 4. (Note: Quadratic function is obviously the most fitted function in this case by looking at the graph, we’re not sure why degree of 3 performs better than degree of 2 in our model) Higher the degree will ends up with a better fitted regression model but will consume enormously more amount of time. Degree of 4 is our best choice time and efficient wise.

Following is our best fitted result: R^2=0.6673 [With all parameters included]

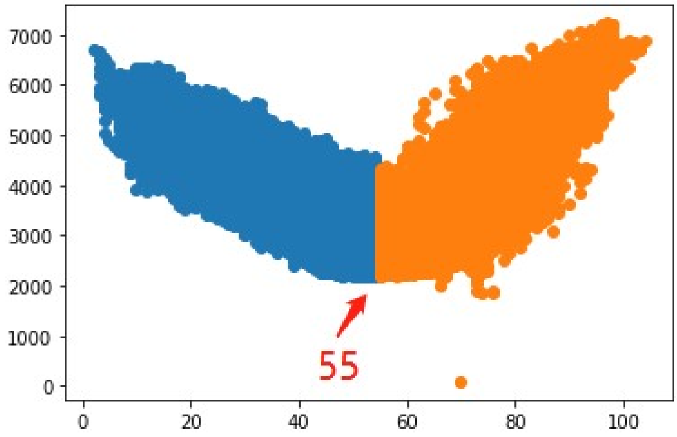


```
explained_variance:  0.6673
mean_squared_log_error:  0.0162
r2:  0.6673
MAE:  378.6299
MSE:  215629.0633
RMSE:  464.3588
None
score0.6689827348047802
```

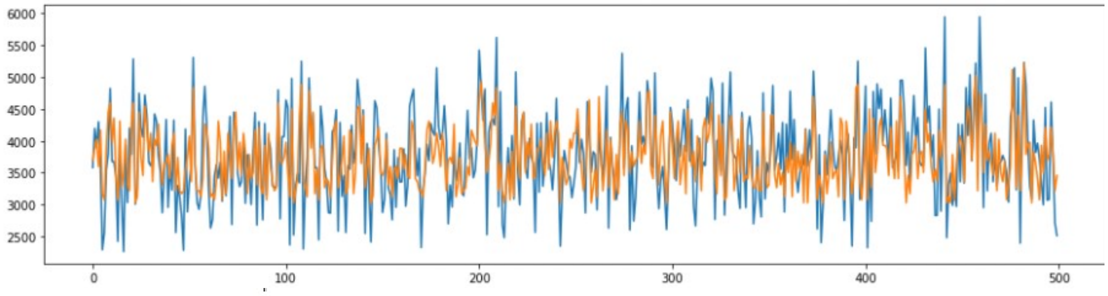


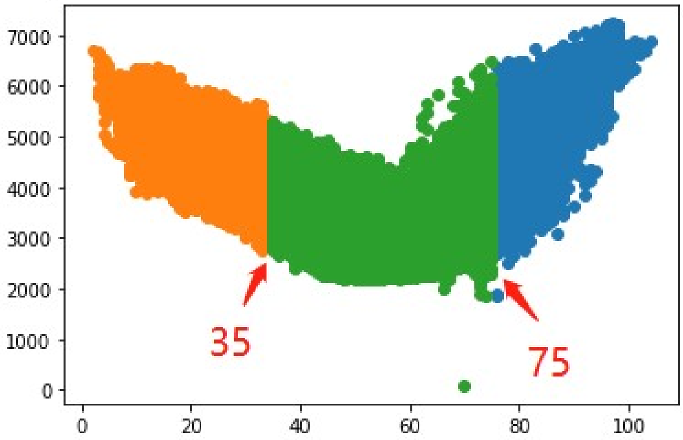
Extra:

Another way we have tried is to build models after split the dataframe on “Temperature”, so ‘splitted’ data frames would exhibit a different type of trend.

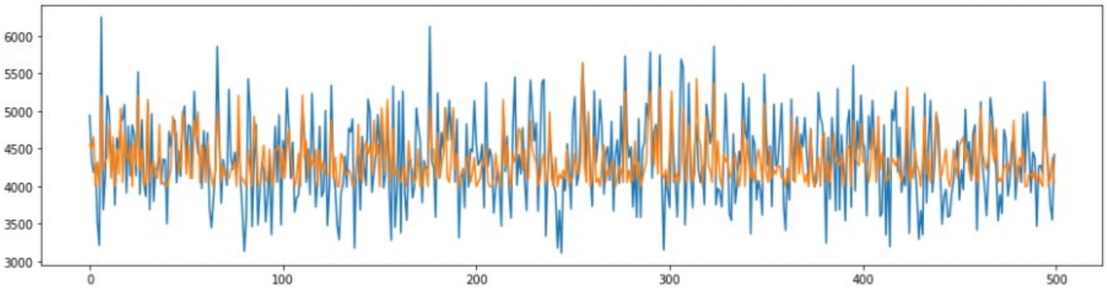


Splitting the data frame from the middle where drytemperature=55, so both data frames will exhibit an linear trend. R^2 we got from this model is 0.5948 which lower than our best fitted model.





Splitting the data frame into three parts, a nonlinear trend with data that satisify 35<=drytemperature<=75 and two linear trend with data that either drytemperature>35 or <75. Unforunately, the result R^2 = 0.642 which is lower than our best model.


6000
5500
5000
4500
4000
3500
3000
0
100
200
300
400
500