INFO 7390_Assignment3

Zhao Wang

Lezi Wang

Qiaoyi He

Content

1. Problem Statement & Requirements

Nowadays, energy consumption has become a critical requirement in multiple industry fields. Therefore, we are planning to explore the datasets, which contained the power consumption of 9 different industries, and 58 different locations in City of Boston, and then develop short-term prediction models (kWh, Power Factor, and KVarh) for each location, in order to meet following requirements:

1.Supply and demand balancing: forecast the future energy usage and peak leveling to avoid overload.

2.Market making: help government to set power price.

3.Electricity grid operational optimization: government can response according to the demand for avoids energy waste.

4.Fault and anomaly record detection.

2. Business Goal & Solutions

2.1. Business Goal

Predict short-term power consumption for each location in City of Boston, in order to

1) Balance supply and demand;

2) Predict the daily peak leveling for avoid some emergency event like power failure, and lowest leveling for avoid the power waste;

3) Help different industries to plan for future power consumption, reduce power cost and save energy.

2.2. Solutions-Micro Analysis

Implement Micro Analysis to predict following power consumption for one day in future.

1) Total kWh for one day.

2) Average Power Factor for one day.

3) Total KVarh for one day.

3. Process Steps & Technologies

Based on our goal and requirements, we build our web service with various technology tools in following steps. We will explain details for each step later in the report.

1. Explore Dataset→R

2. Choose Dataset

3. Pre-process Data→R

4. Add Temperature Data→ R

5. Visualization Data→ Tableau Public

6. Build Prediction Model→ Azure Machine Learning

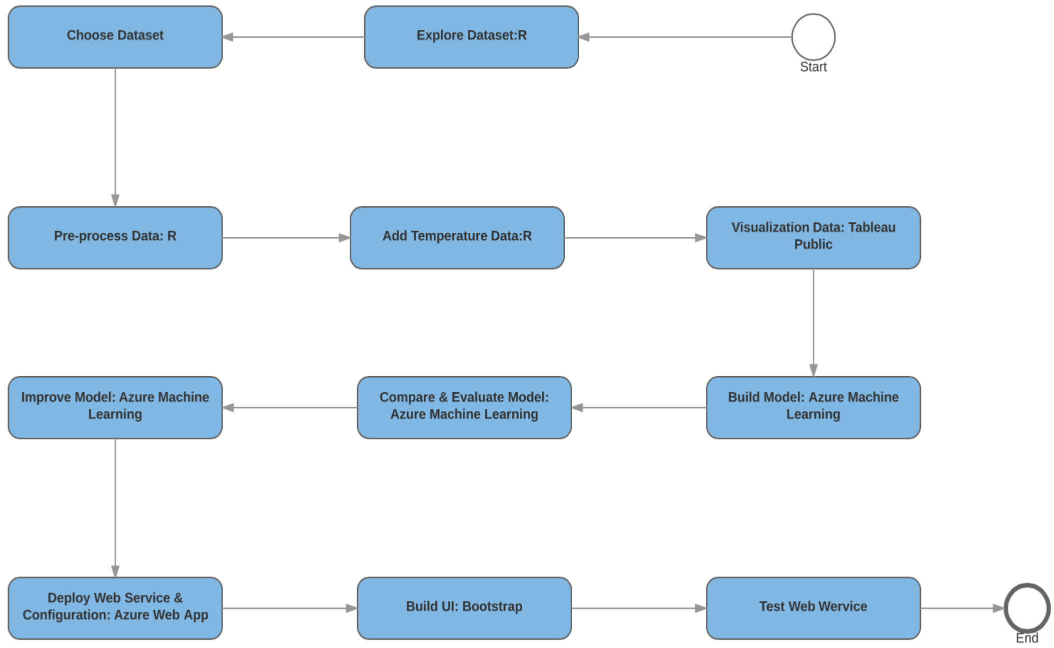
7. Compare & Evaluate Model→ Azure Machine Learning

8. Improve Model→ Azure Machine Learning

9. Deploy Web Service & Configuration→ Azure Web App

10. Build UI→Bootstrap

11. Test Web Service


Start
Explore Dataset:R
Choose Dataset
Pre-process Data: R
Add Temperature Data:R
Visualization Data: Tableau Public
Build Model: Azure Machine Learning
Compare & Evaluate Model: Azure Machine Learning
Improve Model: Azure Machine Learning
Deploy Web Service & Configuration: Azure Web App
Build UI: Bootstrap
Test Web Wervice
End

4. Pre-process Data

4.1. Explore the dataset

Firstly, we combine all data sets into a single data set, and add two parameters – location and industry to distinct different datasets. The following picture is the R code which we use to combine the data.

```
setwd("/Users/lucy/Desktop/12/")
temp<- list.files(pattern="*.csv")
energyset <- do.call(rbind, lapply(temp, fread))
write.csv(energyset,"/Users/lucy/Desktop/energyset.csv")
```

4.2. Chose the dataset

After we combine all data set and explore it, we find out that dataset which has the different industry and location's parameter has different energy consumption's pattern. So we think it's will be better to allow the customer choose their own energy consumption dataset and use the dataset to develop personality model instead of build a general model base on a single combining dataset. So we decide keep data sets separately and restructure representative datasets. When we choose the representative dataset, we think the gap in energy consumption between different industries seems greater in different locations. So we should cover different industries as much as possible. Finally, we choose 22 data sets which covering 8 different industries and 22 locations.

4.3. Restructure the dataset

3.1 Calculate the value

As for short-term prediction, we decide to predict the total energy for next 24 hours. Since we need develop three different models to predict the kwh, power factor and kvarh respectively, we need calculate those three parameters, since the raw data collects the data each 5 minute. But we need data which can representative each single day.

3.1.1 KWH

KWH is used as abbreviations for kilowatt hour, a unit of energy. So for this parameter, we need the calculate the sum of kWh. And add the total amount into data set as a now column. The following picture is the R code which we use to process the KWH relevant data.

```
#select the kWh value
subset1<-subset(data1,Units=="kWh")
#calculate the sum of kWh
temp<-subset1[,5:292]
kWh<-apply(temp,1,sum)
#add total kWh as a now column
energydata<-data.frame(subset1,kWh)
```

3.1.2 Power Factor

Power factor of an AC electrical power system is defined as the ratio of the real power flowing to the load to the apparent power in the circuit, and is a dimensionless number in the closed interval of -1 to 1. So for this parameter, we need the calculate the mean of power factor instead of the sum. And add the mean into data set as a now column. The following picture is the R code which we use to process the power factor relevant data.

```
#select the power factor value
subset2<-subset(data1,Units=="Power Factor")
#calculate the mean of power factor
temp<-subset2[,5:292]
power_factor<-apply(temp,1,mean)
#add mean of power factor as a now column
energydata<-data.frame(energydata,power_factor)
```

3.1.3 Kvarh

Kvarh is a unit of energy as same as kWh. The different between the kWh and kVARh is the W refers to watts and the VAR refers Volts times Amps (reactive). So the method used in Kvarh is as same as the method used in kWh - we should calculate the sum of kWh then add the total amount into data set as a now column. The following picture is the R code which we use to process the Kvarh relevant data.

```
#select the KVARH value
subset3<-subset(data1,Units=="kVARh")
#calculate the sum of kwh
temp<-subset3[,5:292]
kVARh<-apply(temp,1,sum)
#add sum of kVARh as a now column
energydata<-data.frame(energydata,kVARh)
```

3.2 Add temperature parameter

After thorough explore the data, we think temperature has a great impact on energy consumption, so we add three more parameters relatively the temperature into dataset – maximum temperature, minimum temperature and average temperature.

We got the Daily Temperature Record 2014 from following URL and calculate the average temperature based it:

http://www.usclimatedata.com/climate/boston/massachusetts/united-states/usma0046/2014/1

The following picture is the R code which we use to Add temperature relatively parameters into raw dataset.

```
#import the temperature date
datatem<-read.csv("~/Desktop/temperature.csv")
#Add temperature data into dataset
tem<-datatem[,2:4]
energydata<-data.frame(energydata,tem)
energydata<-energydata[,-c(4:292)]
```

3.3 Clean the data

After we restructure the data set, we need to clean abnormal or missing data for improving the usability and stability of data. The following picture is the R code which we use to process clean the data. We delete rows which contain “0” value or missing value.

```
#clear 0 and NA row
energydata[energydata == 0] = NA
energydata<-na.omit(energydata)
```

3.4 Final data structure

The following picture shows the part of data set after restructure. The new data sets include 11 columns – Account, Date, channel, industry, location, kWh, highest temperature (high), lowest temperature (low), Average temperature (average), power factor and kVARh.

	A	B	C	D	E	F	G	H	I	J	K
1	Account	Date	Channel	Industry	Location	kWh	high	low	average	power_facto	kVARh
2	########	01/01/2014	605106493	BPL	Harvard.st	497.052	29.1	19.2	24.15	0.95061441	168.381
3	2.6812E+10	01/02/2014	605106493	BPL	Harvard.st	570.789	26.2	2.1	14.15	0.96164955	163.458
4	2.6812E+10	01/03/2014	605106493	BPL	Harvard.st	465.48	14.2	2.1	8.15	0.95305582	154.134
5	2.6812E+10	01/04/2014	605106493	BPL	Harvard.st	571.365	27.1	2.1	14.6	0.96432472	155.538
6	2.6812E+10	01/05/2014	605106493	BPL	Harvard.st	346.239	37.9	24.3	31.1	0.96655633	91.836
7	2.6812E+10	01/06/2014	605106493	BPL	Harvard.st	575.172	55.9	29.1	42.5	0.97101479	136.359
8	2.6812E+10	01/07/2014	605106493	BPL	Harvard.st	549.918	29.1	9.1	19.1	0.96259224	156.348
9	2.6812E+10	01/08/2014	605106493	BPL	Harvard.st	632.259	22.1	7.2	14.65	0.96290678	176.049
10	2.6812E+10	01/09/2014	605106493	BPL	Harvard.st	666.963	30.2	19.2	24.7	0.96754126	170.145
11	2.6812E+10	01/10/2014	605106493	BPL	Harvard.st	584.991	37	18.1	27.55	0.96294836	163.116
12	2.6812E+10	01/11/2014	605106493	BPL	Harvard.st	486.441	59	34	46.5	0.97746106	101.025
13	2.6812E+10	01/12/2014	605106493	BPL	Harvard.st	407.187	54	37	45.5	0.96874735	103.356
14	2.6812E+10	01/13/2014	605106493	BPL	Harvard.st	673.947	51.1	31.1	41.1	0.96602897	177.282
15	2.6812E+10	01/14/2014	605106493	BPL	Harvard.st	595.521	50	44.1	47.05	0.96454975	160.317
16	2.6812E+10	01/15/2014	605106493	BPL	Harvard.st	628.929	48	35.1	41.55	0.96249629	173.925
17	2.6812E+10	01/16/2014	605106493	BPL	Harvard.st	595.656	39	33.1	36.05	0.96492173	159.759
18	2.6812E+10	01/17/2014	605106493	BPL	Harvard.st	583.767	46	33.1	39.55	0.96534736	156.771
19	2.6812E+10	01/18/2014	605106493	BPL	Harvard.st	558.342	37.9	32	34.95	0.96432248	153.189
20	2.6812E+10	01/19/2014	605106493	BPL	Harvard.st	338.724	35.1	30.2	32.65	0.96552862	91.287
21	2.6812E+10	01/20/2014	605106493	BPL	Harvard.st	508.23	41	26.2	33.6	0.95015148	172.431
22	2.6812E+10	01/21/2014	605106493	BPL	Harvard.st	643.932	26.2	11.1	18.65	0.96242735	179.541
23	2.6812E+10	01/22/2014	605106493	BPL	Harvard.st	721.566	18.1	7.2	12.65	0.95321607	219.771
24	2.6812E+10	01/23/2014	605106493	BPL	Harvard.st	734.337	22.1	6.3	14.2	0.95355806	220.338
25	2.6812E+10	01/24/2014	605106493	BPL	Harvard.st	723.951	19.2	7.2	13.2	0.95272836	220.671
26	2.6812E+10	01/25/2014	605106493	BPL	Harvard.st	715.806	37.9	15.3	26.6	0.95323208	215.955
27	2.6812E+10	01/26/2014	605106493	BPL	Harvard.st	605.61	26.2	16.2	21.2	0.94513609	209.196
28	2.6812E+10	01/27/2014	605106493	BPL	Harvard.st	728.334	46.9	21.2	34.05	0.95508175	216.9
29	2.6812E+10	01/28/2014	605106493	BPL	Harvard.st	723.429	21.2	13.1	17.15	0.95236007	221.463
30	2.6812E+10	01/29/2014	605106493	BPL	Harvard.st	727.596	27.1	15.3	21.2	0.95475169	215.685

COB-BPL. HARVARD. ST. 2014

3.4. Upload datasets

After pre-process, we upload all data set through COE serve. So customers can easy change the data source by change the URL of reader. You can find all representative datasets from following URL:

http://www1.ece.neu.edu/~zwang3/datatest/

Index of /~zwang3/datatest

Name	Last modified	Size	Description
Parent Directory		-	
COB-BCYF.CURLEY.CMTY.CTR.2014.csv	2016-04-15 14:44	38K	
COB-BFD.HEADQUARTERS.2014.csv	2016-04-15 14:44	35K	
COB-BPD.DUDLEY.SQ.2014.csv	2016-04-15 14:44	33K	
COB-BPD.HEADQUARTERS.D.2014.csv	2016-04-15 14:44	35K	
COB-BPD.STATION.11.AREA.C.2014.csv	2016-04-15 14:44	37K	
COB-BPL.COPLEY.SQUARE.2014.csv	2016-04-15 14:44	38K	
COB-BPL.HARVARD.ST.2014.csv	2016-04-15 14:44	32K	
COB-BPL.HYDE.PARK2014.csv	2016-04-15 14:44	34K	
COB-BPL.MATTAPAN.2014.csv	2016-04-15 14:44	33K	
COB-DND.HEMENWAY.2014.csv	2016-04-15 14:44	33K	
COB-PROP.MGMT.43.HAWKINS.2014.csv	2016-04-15 14:44	34K	
COB-PROP.MGMT.1481TREMONT.2014.csv	2016-04-15 14:44	34K	
COB-PROP.MGMT.CITY.HLL.SQ.2014.csv	2016-04-15 14:44	37K	
COB-PROP.MGMT_1481TREMONT_TOBIN.COMMUNITY.CENTER.csv	2016-04-15 14:44	37K	
COB-PWD.BLDG.2014.csv	2016-04-15 14:44	37K	
COB-SCH.Arts.2014.csv	2016-04-15 14:44	34K	
COB-SCH.BOSTON.LATIN.ANNX.2014.csv	2016-04-15 14:44	35K	
COB-SCH.CAMPBELL.2014.csv	2016-04-15 14:44	35K	
COB-SCH.CENTRAL.KITCHEN.2014.csv	2016-04-15 14:44	34K	
COB-SCH.COURT.ST.2014.csv	2016-04-15 14:44	36K	
COB-SCH.DEVER.MCCORMICK.2014.csv	2016-04-15 14:44	36K	
COB-SCH.DORCHESTER.HIGH.2014.csv	2016-04-15 14:44	35K	

Apache/2.4.7 (Ubuntu) Server at www1.ece.neu.edu Port 80

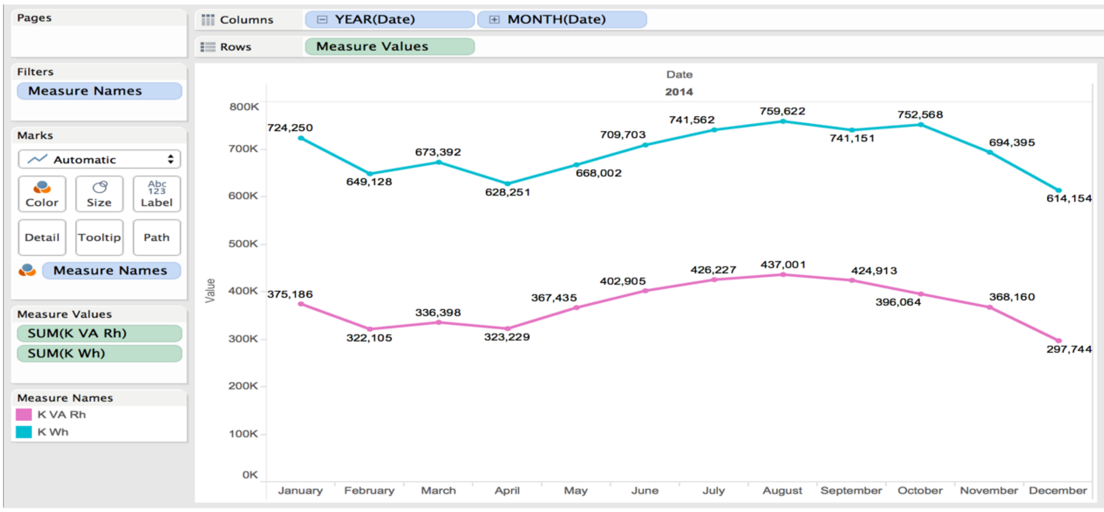
5. Visualization Data

For Visualization, we use Tableau Public tools.

We choose 22 datasets in total, and we do visualization for each dataset separately. Here we just show two typical analysis for **BPL at Copley Square** and **SCH Arts**, and **all others can be viewed on our website**, under visualization tab.

5.1 Visualization for BPL at Copley Square Energy Consumption

5.1.1. Explore monthly total energy consumption.



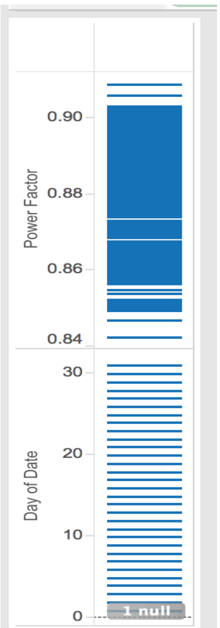
1) Analyze monthly total kWh and KVarh trend.

As shown in above picture, the total trend of monthly kWh and KVarh are similar to each other. The energy consumption trend decreases from January to April, and then increases from April to August, and then decreases again from August to December.

2) Compare kWh and KVarh.

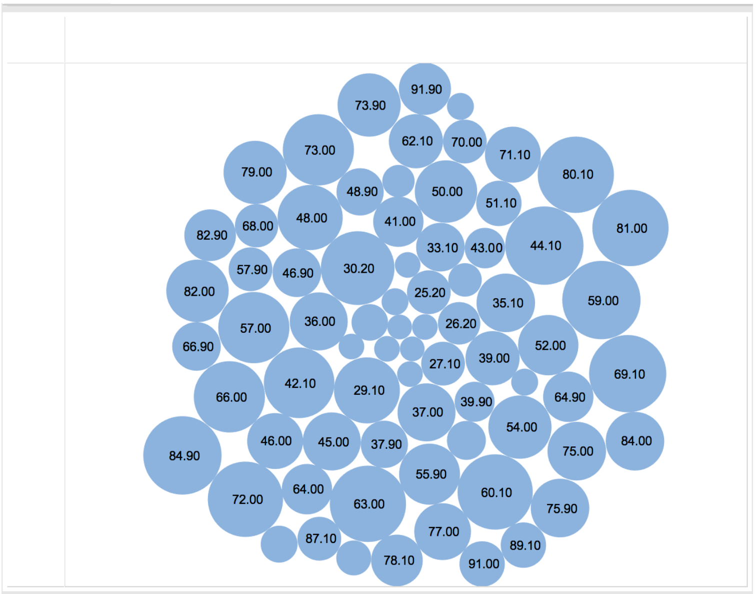
As shown in the above picture, the monthly total KWh amount is higher than monthly total KVarh amount; the monthly total KWh amount is almost twice of the monthly total KVarh amount.

5.1.2. Explore daily average Power Factor distribution



As shown in the picture, the daily average power factor distributes asymmetrically, and is pretty stable among the whole year. Power factors for most days distribute between 0.84 and 0.90.

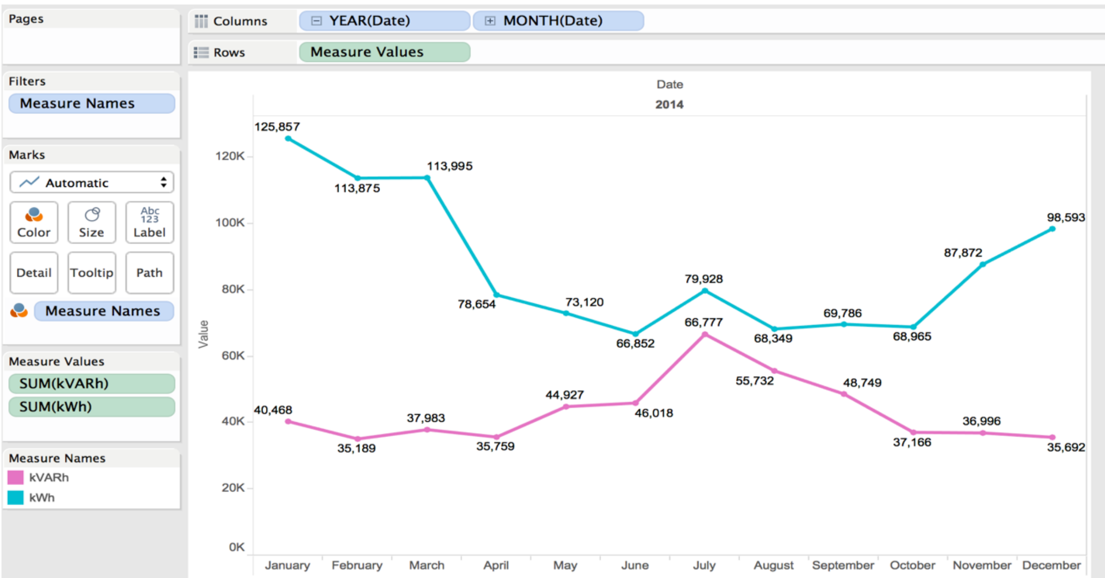
5.1.3. Explore relationship between highest temperature and average daily total kWh



AS shown in the left picture, the daily total kWh consumption tends to be a little high, when the daily highest temperature are higher than 70, or lower than40. But the trend is not very obvious.

5.2 Visualization for SCH Arts Energy Consumption

5.2.1. Explore monthly total energy consumption.



1) Analyze monthly total kWh and KVarh trend.

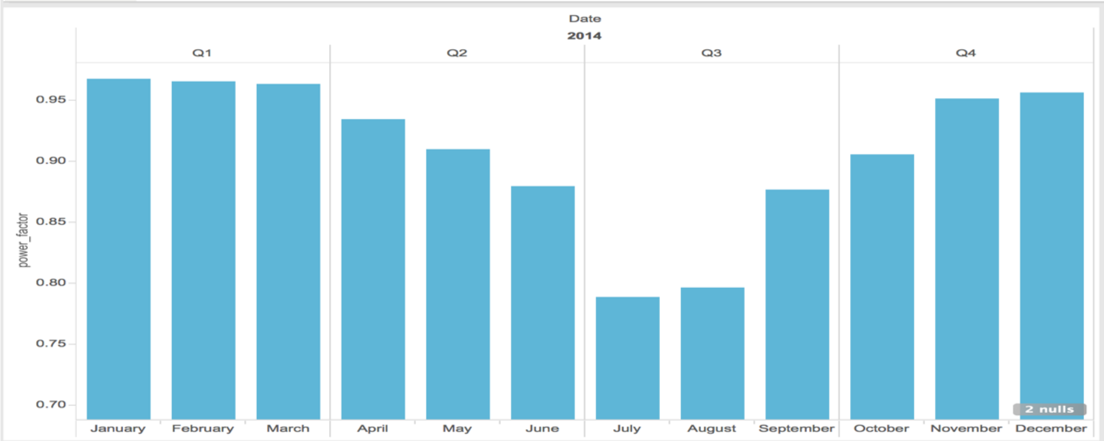
As shown in above picture, the total trend of monthly kWh and KVarh are similar to each other. The energy consumption trend decreases from January to April, and then increases from April to July, and then decreases again from July to October, and then increases from October to December.

2) Compare kWh and KVarh.

As shown in the above picture, the monthly total KWh amount is higher than monthly total KVarh amount in January to April, and the gap between KWh and KVarh becomes smaller from April to July, and then the gap increases again from August to December.

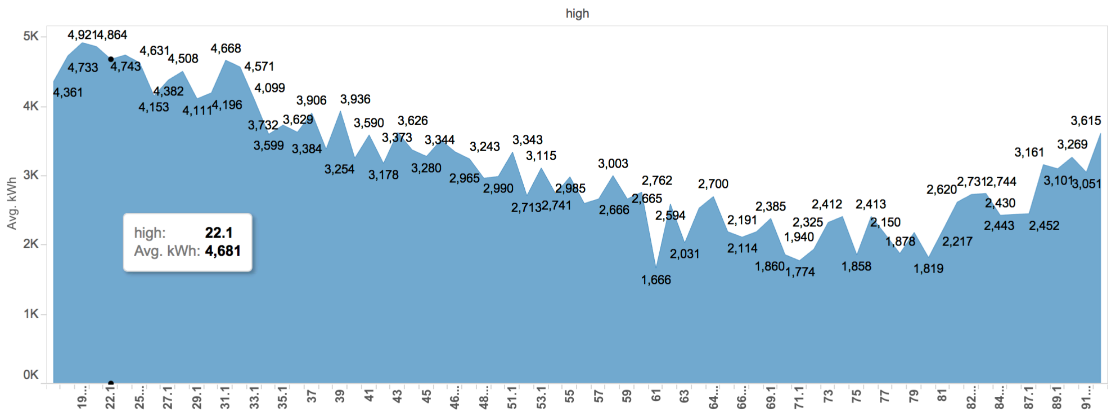
5.2.2. Explore daily average Power Factor trend

As shown in the picture below, the daily average power factor is pretty stable in first, second and forth season, and it is a little low in the third season. Power factors for most days distribute between 0.80 and 0.95.



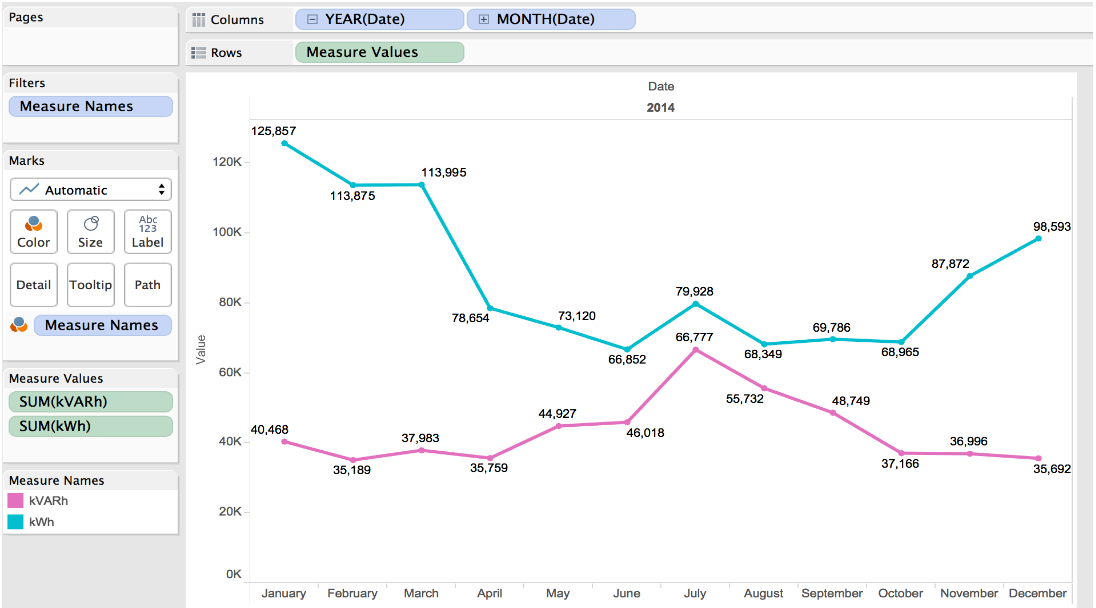
5.2.3. Explore relationship between highest temperature and average daily total kWh

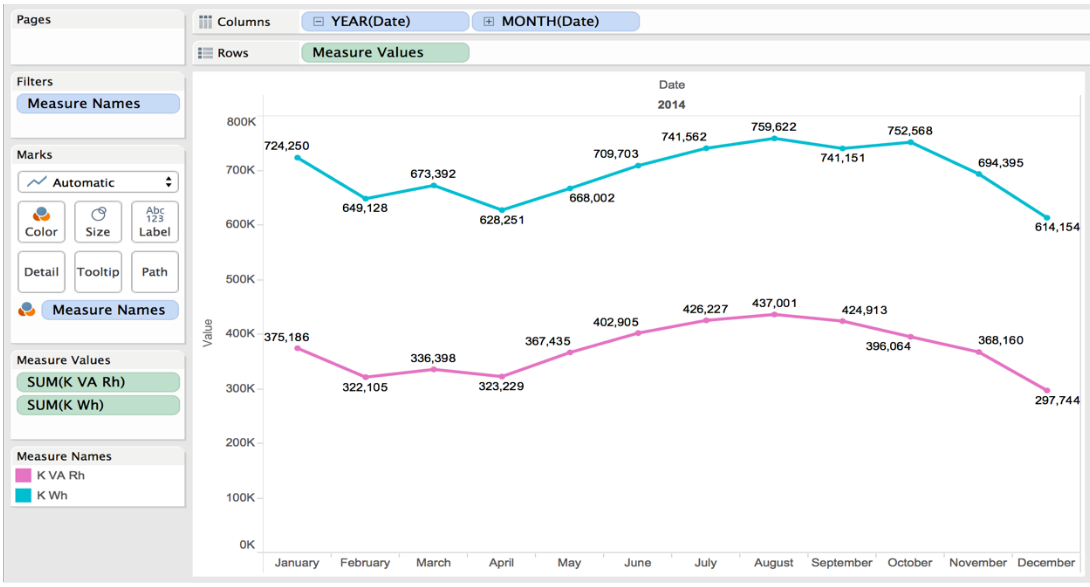
As shown in the picture below, as the daily highest temperature increases, the daily total kWh decreases from 19 (F) to 61(F), and increases slightly from 61(F) to 91(F).



5.3 Comparison Between SCH and BPL

As shown in the picture below, the energy consumption trends for different locations are extremely different. Therefore, we stick to our decision that analyzing different locations' energy consumption separately.





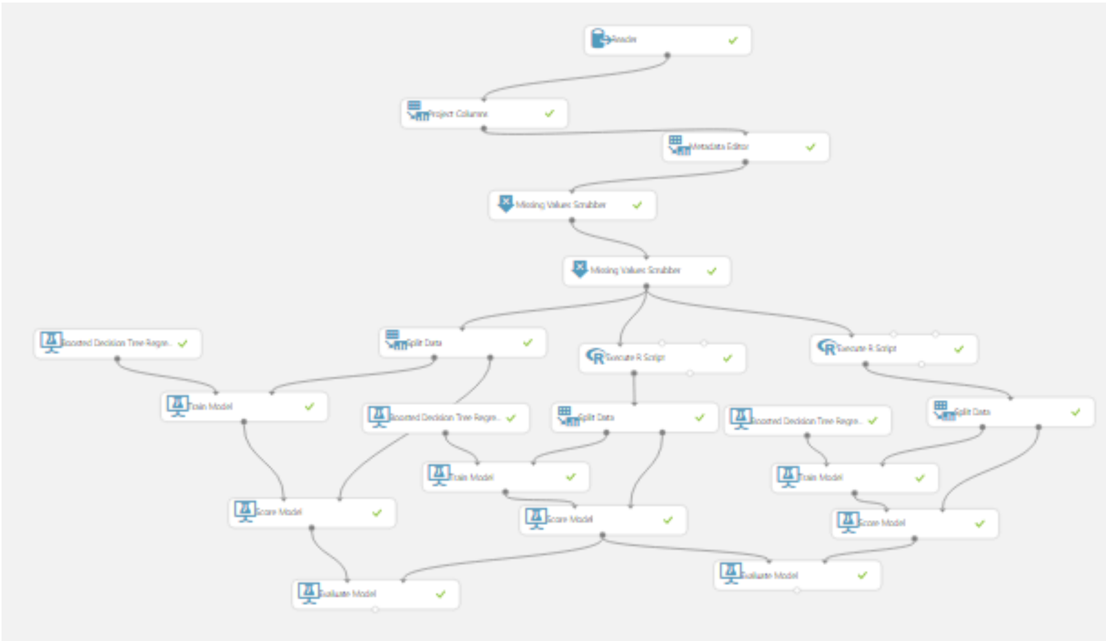
6. Create Predictive Model for kWh (Compared Variables Subsets)

We created machine learning models in Azure Machine Learning Workspace. In our project, the website can provide user three predictive values. The first is the total kWh value for one day which users can input into the Date text filed, the second is the average power factor for one day which uses can input into the Data text filed, and the third is the total Kvarh for that day.

Thus, we created three models for each result in Azure Machine Learning. In first experiment, we created three models to select the best variables subset. In the second experiment, we created two models to choose a better performance algorithm. In the third experiment, we use the best variables subset and a better performance algorithm.

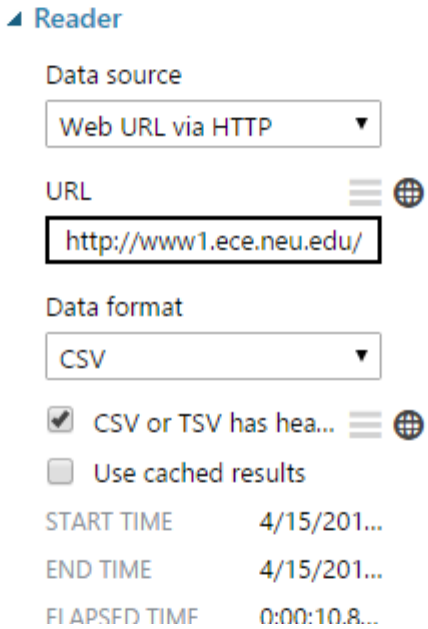
6.1 Predictive Model Overview

We post our model screen shot here, and in later several chapters, we will descript this model step by step very details.



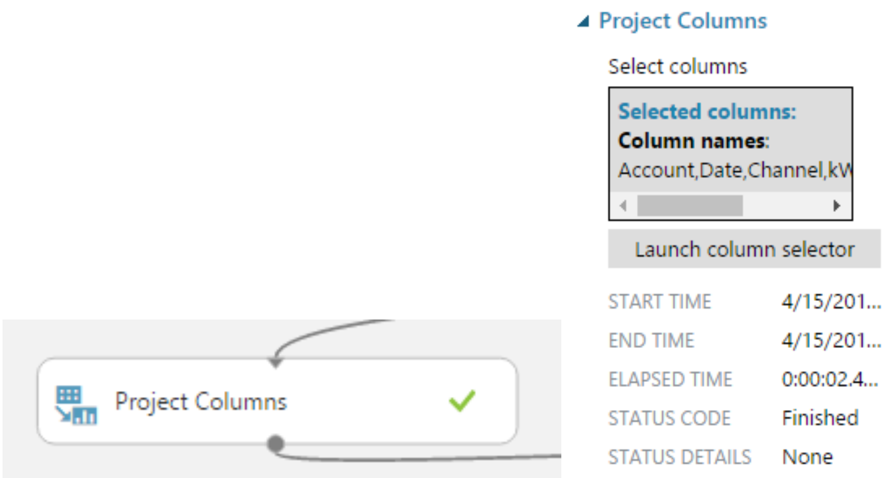
6.2 Get Dataset Using Reader Instead of Using Saved Data

For user Convenience, in Azure Workspace, we use Reader console to get the input dataset instead of using saved dataset. The reason is, if we used the dataset saved in Azure, the dataset is fixed, the model is only for this dataset, and user only can use this model for only one time. However, if we use Reader-Web URL via HTTP, we can input a dataset URL. Also, we can set URL as the Web Service Parameter. In this way, user can input the URL of one dataset they want to use, and use our models to predict values based on their preferred dataset. Our models are more flexible and more useful.



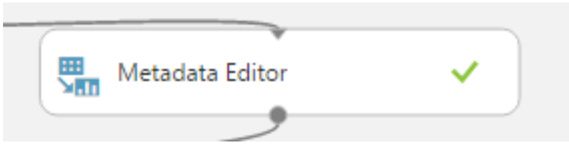
6.3 Drop Columns

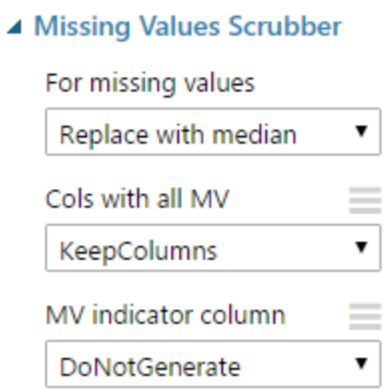
Because we plan to let user choose one of dataset among dozens of datasets professor gave us. Thus, for each dataset, the Industry, Account and Location features are same because each dataset is only for one location. For this reason, we dropped these three features in our first step.

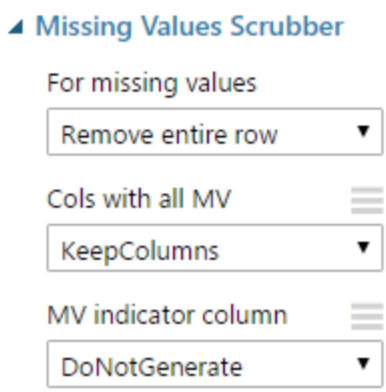


6.4 Clean Missing Data and Split

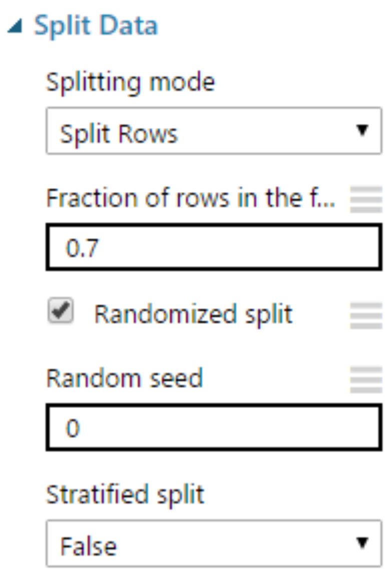
In this step, we found each row which includes missing data. At the first, we replace the missing data used the median value of that column. If there is still missing data. We delete that entire row.







After cleaning data, We Split the dataset into 30% and 70% as Validation set and Train set.



6.5 Create Models

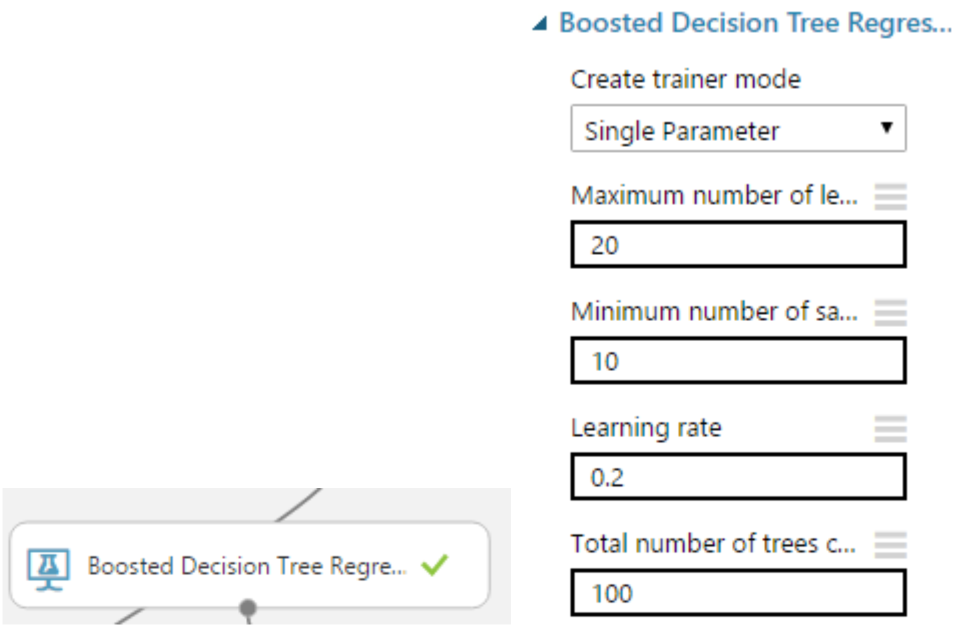
In this experiment, we created three models which use same machine learning algorithm- Boosted Decision Tree Regression as Training Algorithm. But we use three different variables subsets for each compared models.

Boosted Decision Tree Regression enables to create an ensemble of regression trees using boosting. Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage. This regression method is a supervised learning method.

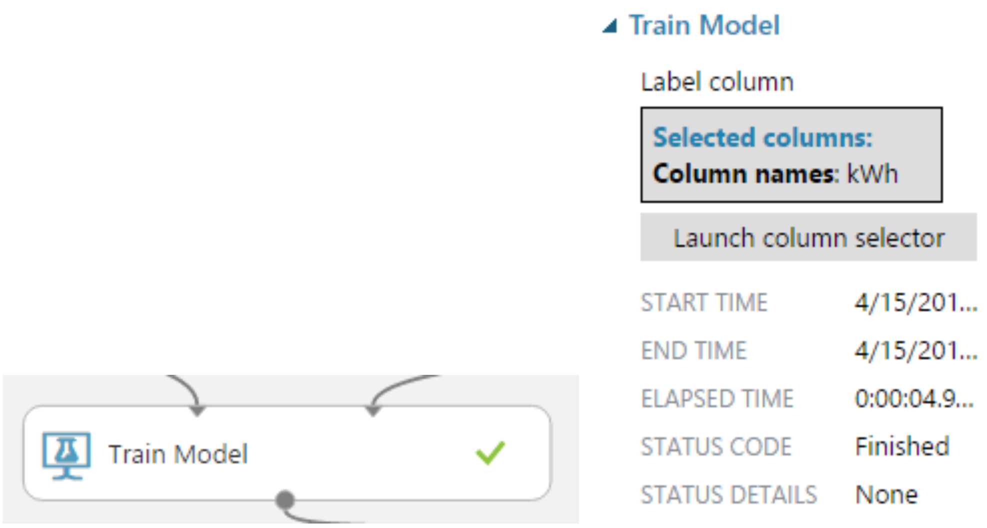
The three training models that we constructed were all based on the same input data, but we added different additional features to each training set. Because this model is for predicting kWh value, so the output of this model is kWh.

- Set A = Date + Highest Temperature + Lowest Temperature+ Average Temperature
- Set B = Date + Average Temperature
- Set C = Date + Highest Temperature +Lowest Temperature

Picture about Boosted Decision Tree Regression:



Picture about train model:



6.6 Score Model and Evaluate Models

Channel	kWh	high	low	average	Scored Labels
6051064931 kWh	660.609	64.9	43	53.95	602.406616
6051064931 kWh	1092.419997	86	71.1	78.55	972.445251
6051064931 kWh	659.565	50	37.9	43.95	637.893982
6051064931 kWh	622.197	64	46	55	730.296082
6051064931 kWh	666.963	30.2	19.2	24.7	530.106445

As you can see, the scored Labels' values are closed to the predictive feature-- kWh. So the first model which includes all features can predict value accurately.

kWh	average	Scored Labels
660.609	53.95	564.803162
1092.419997	78.55	986.786438
659.565	43.95	605.001465
622.197	55	682.236633
666.963	24.7	543.966431

As you can see, the second model which only includes average temperature can predict kWh properly, but not good as the first model.

kWh	high	low	Scored Labels
			
660.609	64.9	43	649.233215
1092.419997	86	71.1	943.994385
659.565	50	37.9	606.191772
622.197	64	46	670.73114
666.963	30.2	19.2	527.052673

As you can see, the third model which includes highest temperature and lowest temperature has best performance.

Of course, we cannot say it is the best only based on the score value. So we evaluate each two models below.

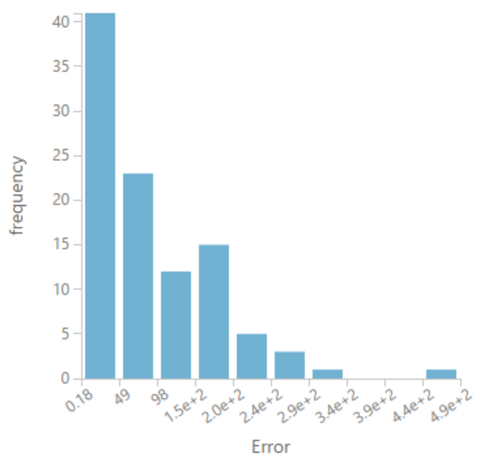
Metrics

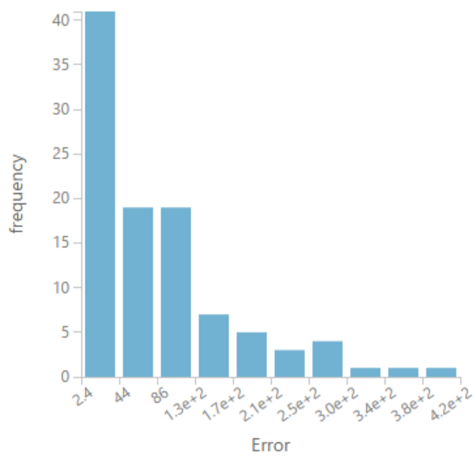
Mean Absolute Error	91.863546
Root Mean Squared Error	124.553907
Relative Absolute Error	0.905576
Relative Squared Error	0.70127
Coefficient of Determination	0.29873

Metrics

Mean Absolute Error	88.376551
Root Mean Squared Error	122.185869
Relative Absolute Error	0.871202
Relative Squared Error	0.674859
Coefficient of Determination	0.325141

Compared the first and second models, we can see the mean absolute error of first is bigger than second, that means that the error rate of first is bigger than second's, and the average difference of first model between predictive values and real values are bigger than second's. Also, the RMS of first is bigger than second's, it stands for same situation as absolute error, but just squared it.





The difference of RMS is very small, so we still need to compare the error frequency. As you can see, the average error frequency of first model is lower than second's. That means the first model is not as frequently as second model.

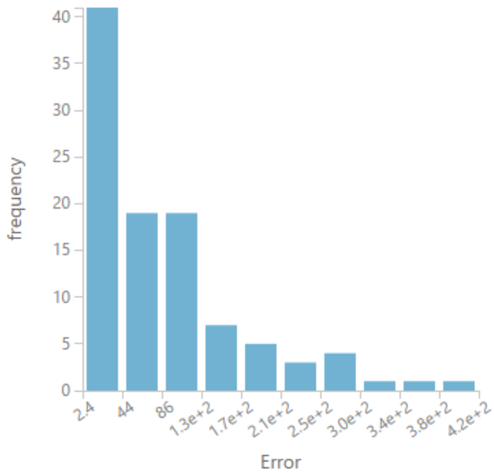
Metrics

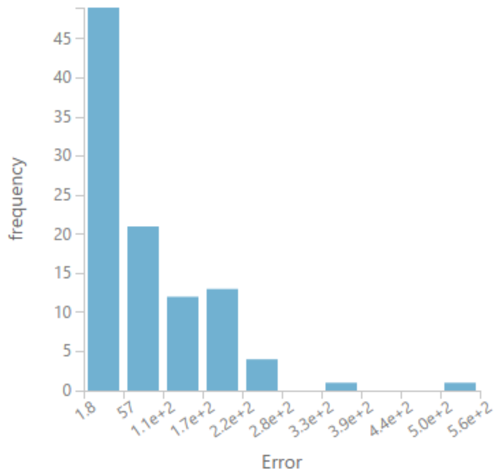
Mean Absolute Error	88.376551
Root Mean Squared Error	122.185869
Relative Absolute Error	0.871202
Relative Squared Error	0.674859
Coefficient of Determination	0.325141

Metrics

Mean Absolute Error	90.008469
Root Mean Squared Error	125.155236
Relative Absolute Error	0.887289
Relative Squared Error	0.708058
Coefficient of Determination	0.291942

Compared the second model and third model, we can see the mean absolute error of second is bigger than first, that means that the error rate of second is bigger than first's, and the average difference of second model between predictive values and real values are bigger than second's. Also, the RMS of second is bigger than first's, it stands for same situation as Absolute error, but just squared it.





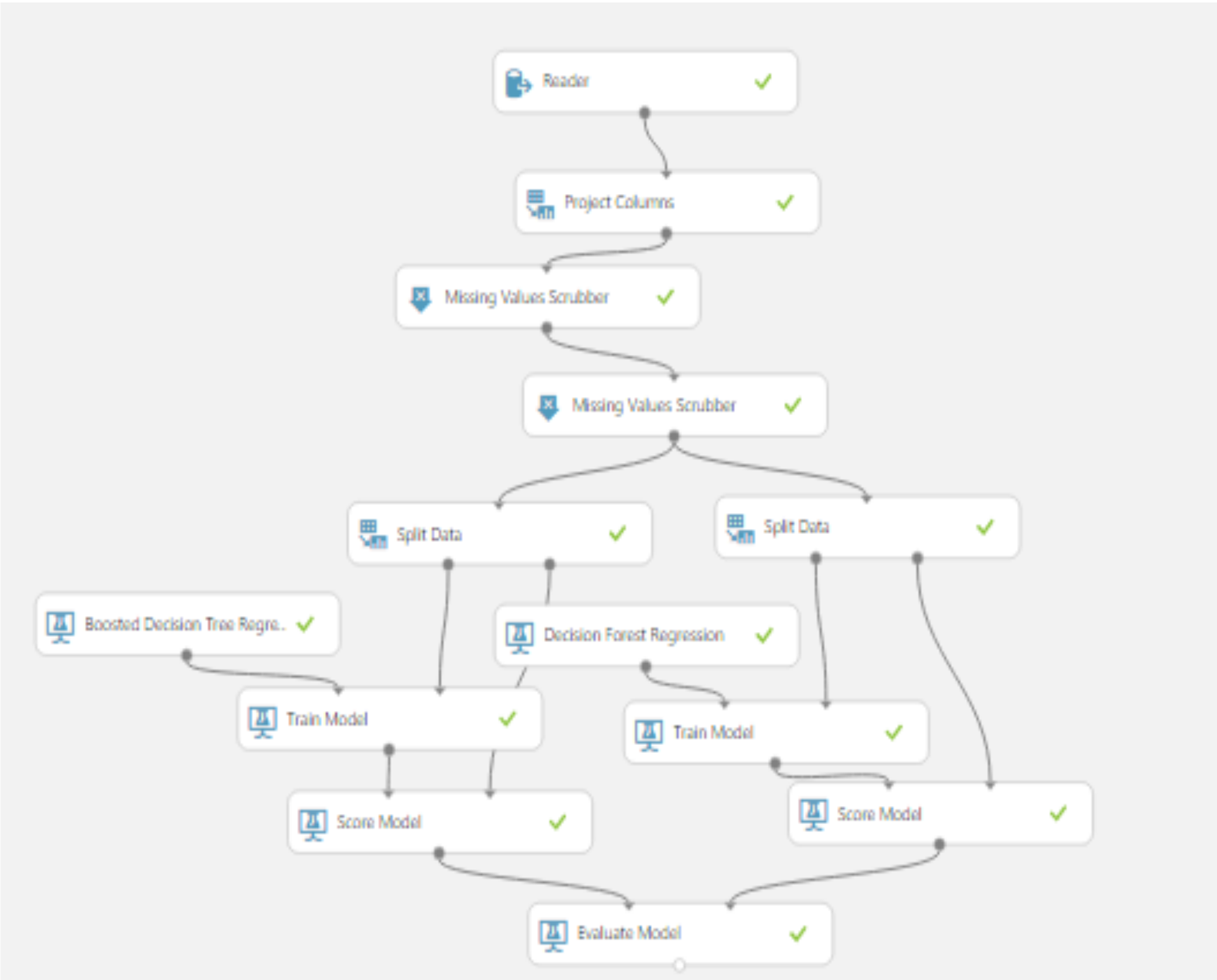
The difference of RMS is still very small, so we compared the error frequency. As you can see, the average error frequency of second model is lower than first's. That means the second model is not as frequently as second model.

For this result, we can say that first model and third model has similar performance on RMS, REMS and error Frequency, However, the third's RMS are slightly bigger than first, so we choose the third model which include the highest temperature and lowest temperature.

7. Create Predictive Model for Power Factor (Compare Algorithm)

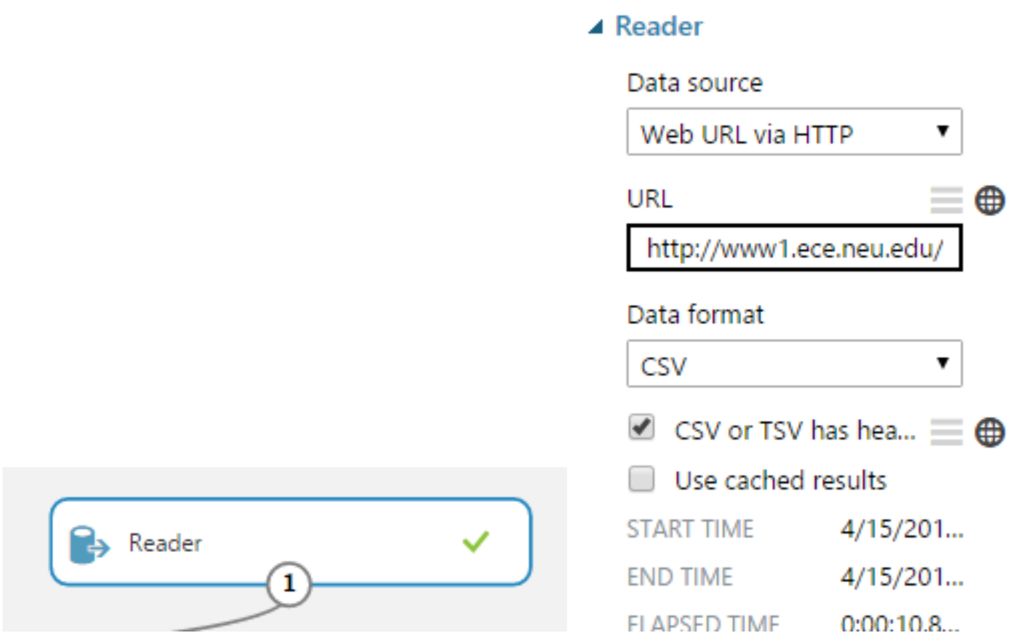
7.1 Predictive Model Overview

We post our model screen shot here, and in later several chapters, we will descript this model step by step very details.



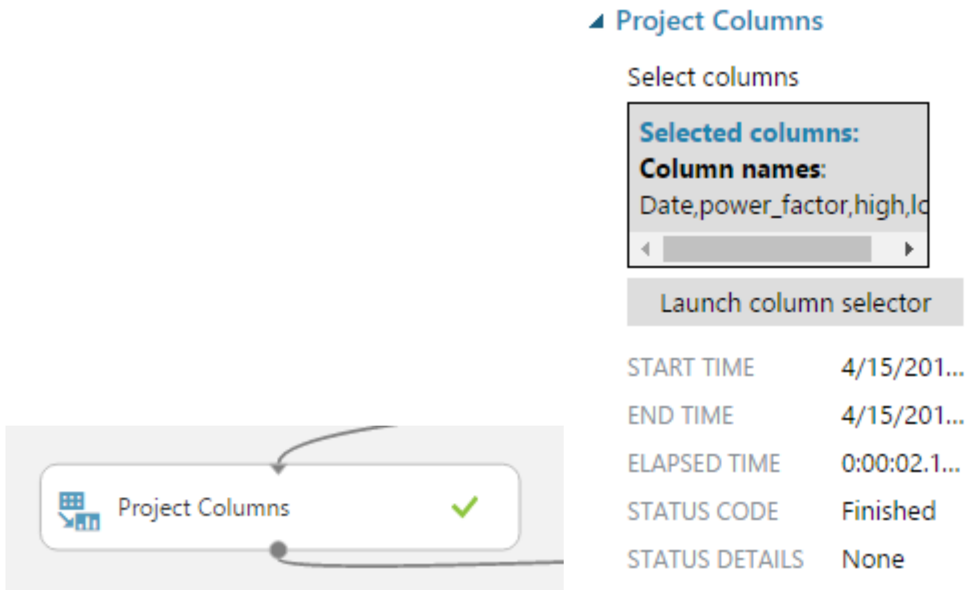
7.2 Get Dataset Using Reader Instead of Using Saved Data

In this step, we use same method with first experiment to import dataset. We used Reader console and give a URL value to Reader instead of using saved dataset. The reason is, if we used the dataset saved in Azure, the dataset is fixed, the model is only for this dataset, and user only can use this model for only one time. However, if we use Reader-Web URL via HTTP, we can input a dataset URL. Also, we can set URL as the Web Service Parameter. In this way, user can input the URL of one dataset they want to use, and use our models to predict values based on their preferred dataset. Our models are more flexible and more useful.



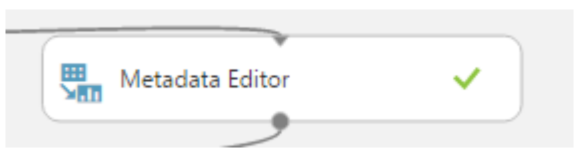
7.3 Drop Columns

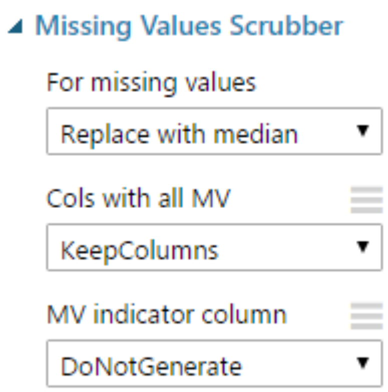
In this step, we use same variables with first experiment. Because we plan to let user choose one of dataset among dozens of datasets professor gave us. Thus, for each dataset, the Industry, Account and Location features are same because each dataset is only for one location. For this reason, we dropped these three features in our first step.

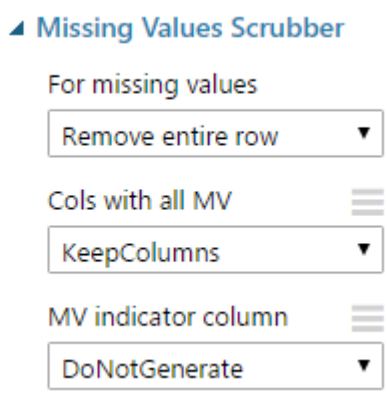


7.4 Clean Missing Data and Split

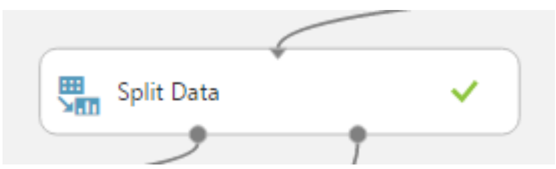
In this step, we found each row which includes missing data. At the first, we replace the missing data used the median value of that column. If there is still missing data. We delete that entire row.

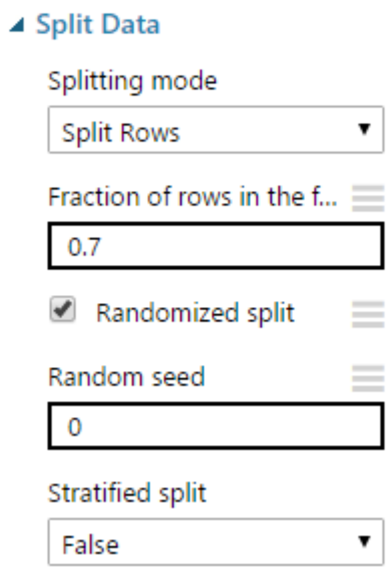






After cleaning data, We Split the dataset into 30% and 70% as Validation set and Train set.





7.5 Create Models

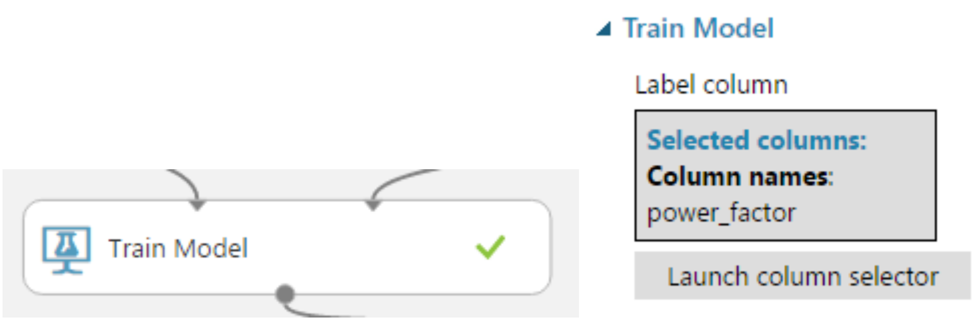
In this experiment, we created two models which use same variables subsets – Date, Highest Temperature, and Lowest Temperature. This variables subset is from the conclusion of first model. But we use two different algorithms for each compared model.

The first algorithm is Boosted Decision Tree Regression, it enables to create an ensemble of regression trees using boosting. Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage. This regression method is a supervised learning method.

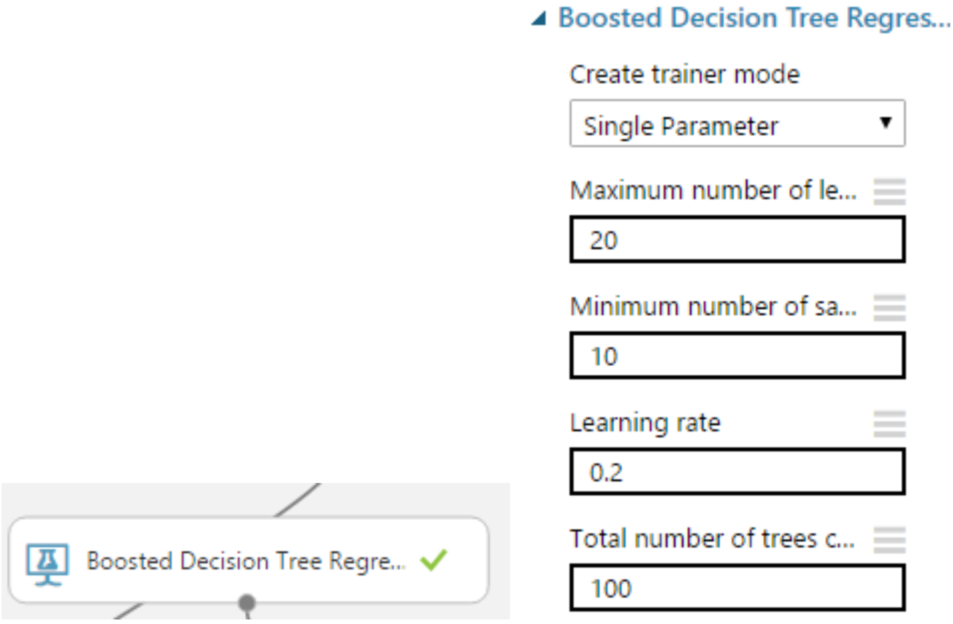
Another algorithm is Decision Forest Regression. It is used to create a regression model using an ensemble of decision trees. Decision trees are non-parametric models that perform a sequence of simple tests for each instance, traversing a binary tree data structure until a leaf node (decision) is reached.

Because this model is for predicting average Power Factor value, so the output of this model is power factor.

Picture about train model:



Picture about Boosted Decision Tree Regression:



7.6 Score Model and Evaluate Models

high	low	power_factor	Scored Labels
64.9	43	0.954262	0.95683
86	71.1	0.948253	0.950703
50	37.9	0.966182	0.963741
64	46	0.966041	0.967143
30.2	19.2	0.967541	0.961603
70	60.1	0.951639	0.952999
55.9	29.1	0.971015	0.967971
41	30.2	0.9609	0.964153

high	low	power_factor	Scored Label Mean	Scored Label Standard Deviation
64.9	43	0.954262	0.952366	0.003701
86	71.1	0.948253	0.953029	0.005941
50	37.9	0.966182	0.965036	0.001101
64	46	0.966041	0.964394	0.001991
30.2	19.2	0.967541	0.964501	0.001038

As you can see, both of these two scored Labels' values are closed to the predictive feature—power factor. So these two model which includes all features can predict value accurately. Of course, we cannot say it is the best only based on the score value. So we compared two models below.

Log Likelihood	Absolute Error	Squared Error	Absolute Error	Squared Error	Determination
Infinity	0.006345	0.009598	0.705026	0.655104	0.344896
-15.975275	0.006759	0.010133	0.751055	0.73012	0.26988

Compared the first and second models, we can see the mean absolute error of second is slightly bigger than second, that means that the error rate of second is bigger than second's, and the average difference of first model between predictive values and real values are bigger than first's. Also, the RMS of second is bigger than first's, RMS stand for same situation as absolute error, but just squared it. For Negative Log Likelihood value, the smaller the better, so infinity negative is better than -15.

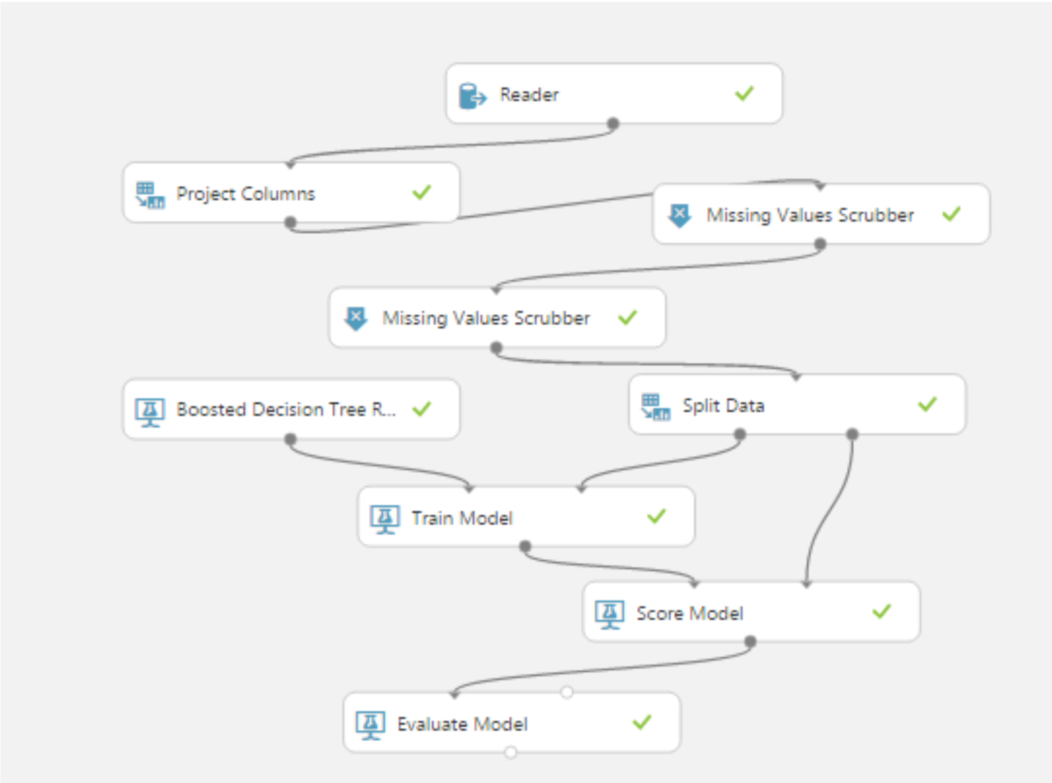
Maximum likelihood estimation optimizes likelihood function (no negative sign). The log function is monotonic and makes it easy to calculate. Some software do it in minimum that is why there a negative sign (-).Because a regression problem is defined to minimize sum of error square.

For this result, we can say that first model which used Boosted Decision Tree Algorithm is slightly better than second used Decision Forest Regression Algorithm. So we choose the Boosted Decision model.

8 Create Predictive Model for Kvarh (Improved Model)

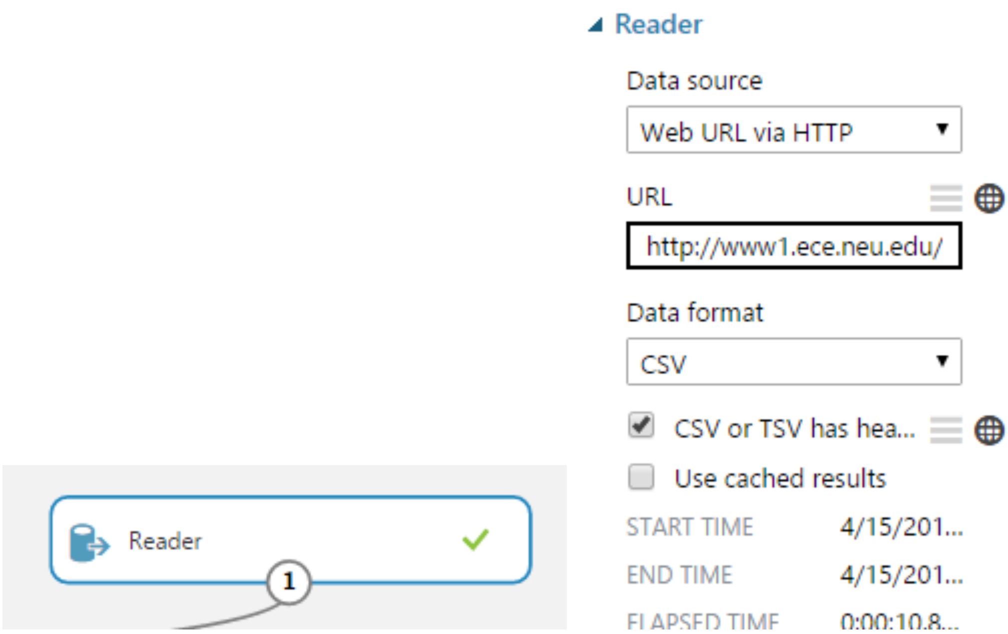
8.1 Predictive Model Overview

We post our model screen shot here, and in later several chapters, we will descript this model step by step very details.



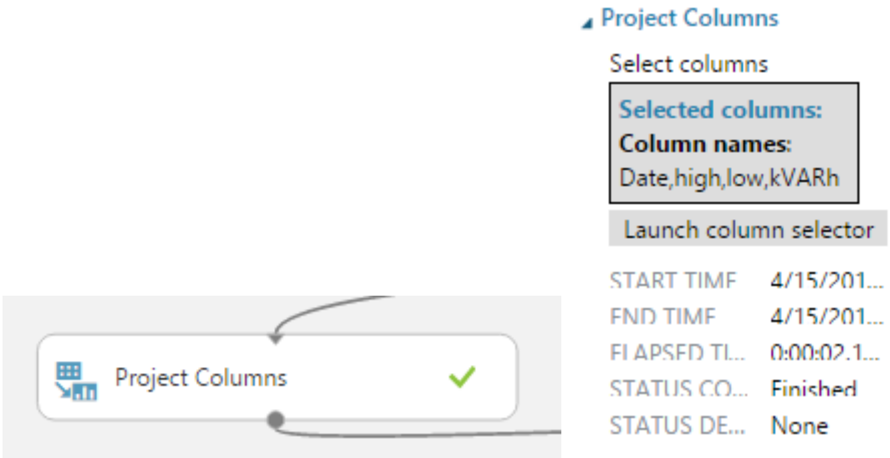
8.2 Get Dataset Using Reader Instead of Using Saved Data

In this step, we use same method with first and second experiment to import dataset. We used Reader console and give a URL value to Reader instead of using saved dataset. The reason is, if we used the dataset saved in Azure, the dataset is fixed, the model is only for this dataset, and user only can use this model for only one time. However, if we use Reader-Web URL via HTTP, we can input a dataset URL. Also, we can set URL as the Web Service Parameter. In this way, user can input the URL of one dataset they want to use, and use our models to predict values based on their preferred dataset. Our models are more flexible and more useful.



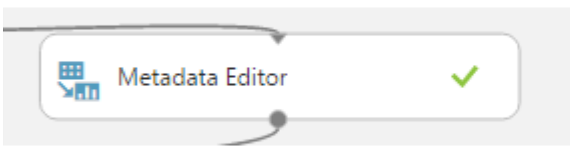
8.3 Drop Columns

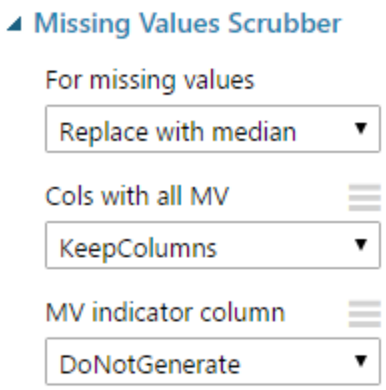
In this step, we use same variables with first and second experiment. Because we plan to let user choose one of dataset among dozens of datasets professor gave us. Thus, for each dataset, the Industry, Account and Location features are same because each dataset is only for one location. For this reason, we dropped these three features in our first step.

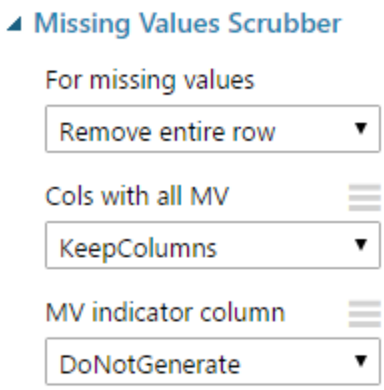


8.4 Clean Missing Data and Split

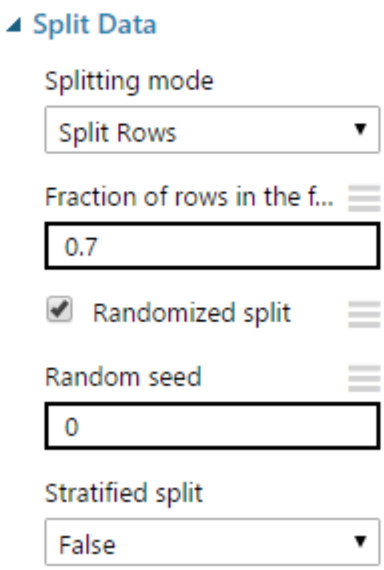
In this step, we found each row which includes missing data. At the first, we replace the missing data used the median value of that column. If there is still missing data. We delete that entire row.







After cleaning data, We Split the dataset into 30% and 70% as Validation set and Train set.



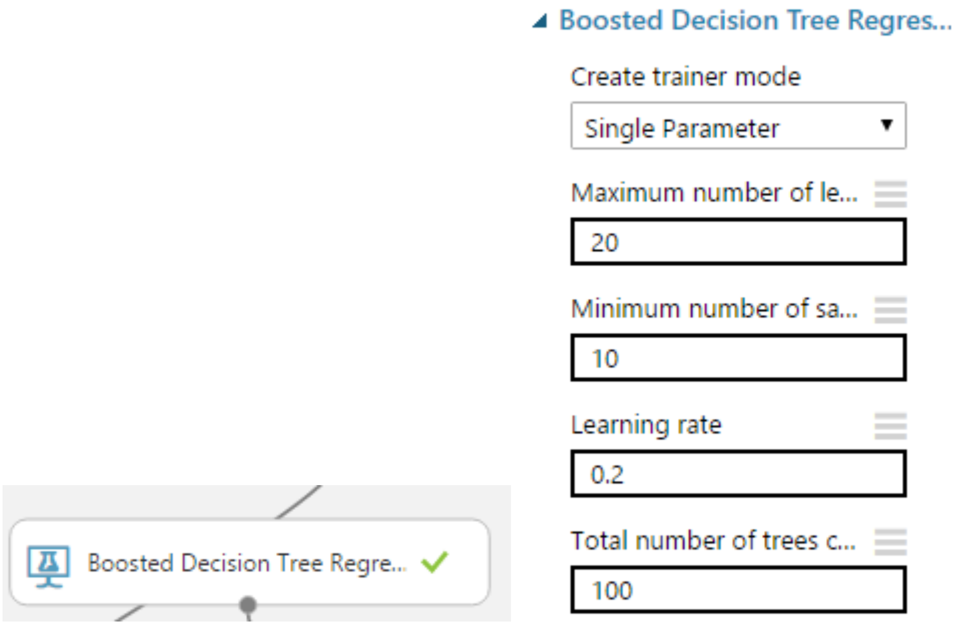
8.5 Create Models

In this experiment, we created only one model which uses Date, Highest Temperature, and Lowest Temperature. And it uses the Boosted Decision Tree Regression to train the model.

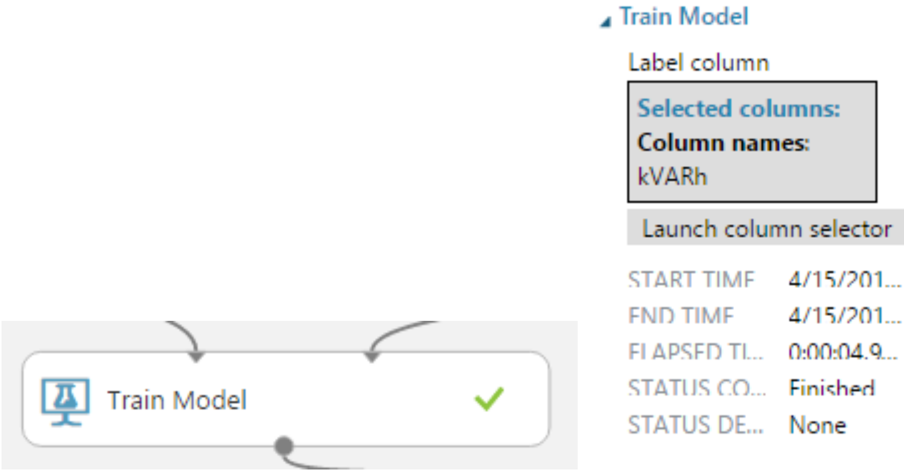
Boosted Decision Tree Regression, it enables to create an ensemble of regression trees using boosting. Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage. This regression method is a supervised learning method.

Because this model is for predicting total Kvarh value, so the output of this model is Kvarh.

Picture about Boosted Decision Tree Regression:



Picture about train model:



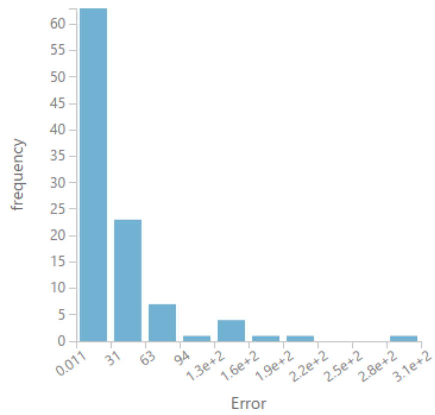
8.6 Score Model and Evaluate Models

high	low	kVARh	Scored Labels
64.9	43	175.932	165.442795
86	71.1	455.958	395.116974
50	37.9	174.078	171.563049
64	46	162	191.89798
30.2	19.2	170.145	157.90303
70	60.1	300.042	284.660034
55.9	29.1	136.359	112.822823
41	30.2	180.495	148.270798

As you can see, the scored Labels' values are closed to the predictive feature—kVarh. So this model can predict kVarh value accurately. Of course, we cannot say it only based on the score value. So we compared two models below.

Metrics

Mean Absolute Error	35.853465
Root Mean Squared Error	59.188511
Relative Absolute Error	0.534795
Relative Squared Error	0.416622
Coefficient of Determination	0.583378



The Absolute Error, RMS and RSE of this model are acceptable. That means that the difference between predictive value and real value is acceptable. So, if these values are small, that means this model can predict kVarh value accurately.

Also, the Error Frequency indicates that this model will not make an error prediction very usual.

9. Deploy Web Service & Configuration

9.1 Deploy Web Service

After creating three training models, Azure Machine Leaning Workspace can create three predictive models automatically. These three predictive models are similar with that three training models we created before.

After predictive models were created, we deployed Web Service in Azure to get models' APIs and URIs. These two values are very important for developing an integration of model. You can see the details of APIs and URI in under picture.

API key

kkkTShGdVpm7dR8HHmv2NJ6agfrh8pIsY2NzxdP3k/iHQ1fkSX50bX9n8bISEeDwOGucTyqjxsCNMv/Dbg0UFQ==

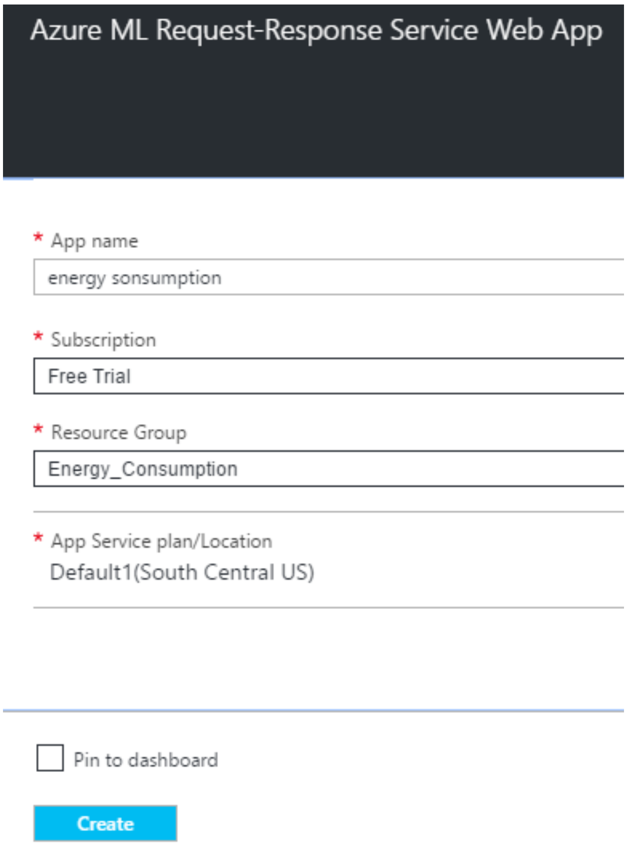
API stands for application programming interface. It can be helpful to think of the API as a way for different apps to talk to one another. For many users, the main interaction with the API will be through API keys, which allow other apps to access your account without you giving out your password.

Method	Request URI	HTTP Version
POST	https://ussouthcentral.services.azureml.net/workspaces/e6b3b1c1dac349c084190a360b36508a/services/cc2cdd8470ba4da48991360db62cf289/execute?api-version=2.0&details=true	HTTP/1.1

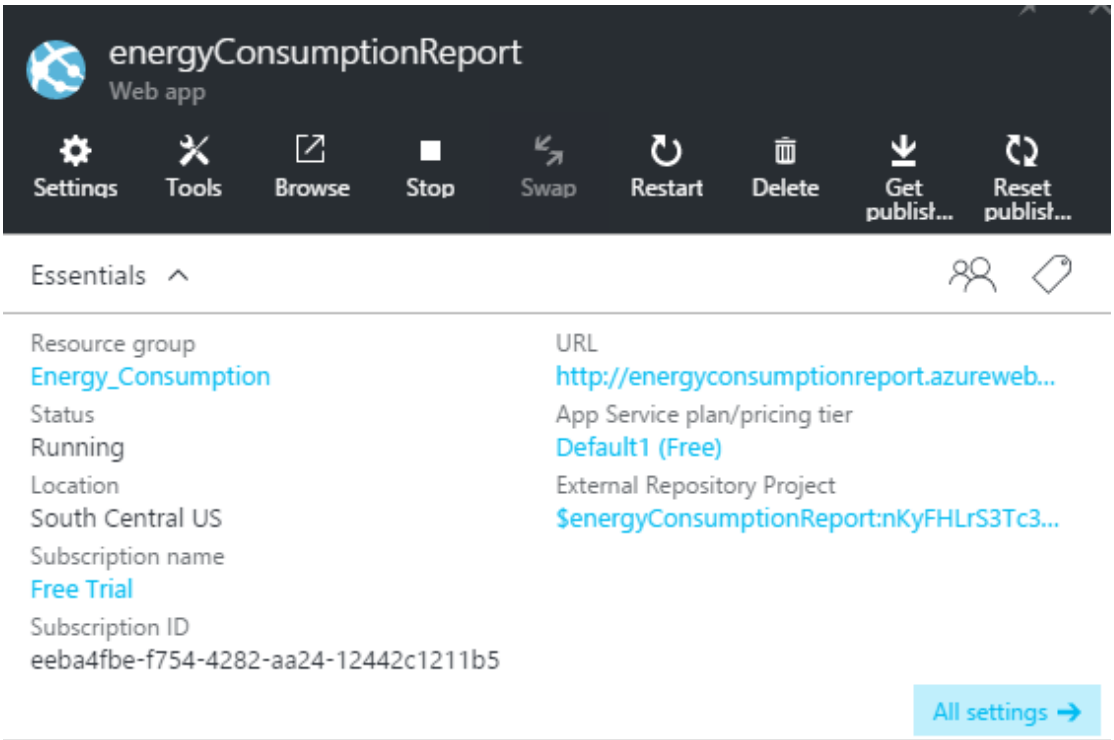
To paraphrase the World Wide Web Consortium, Internet space is inhabited by many points of content. A URI (Uniform Resource Identifier; pronounced YEW-AHR-EYE) is the way you identify any of those points of content, whether it be a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a Uniform Resource Locator (URL).

9.2 Models Integration

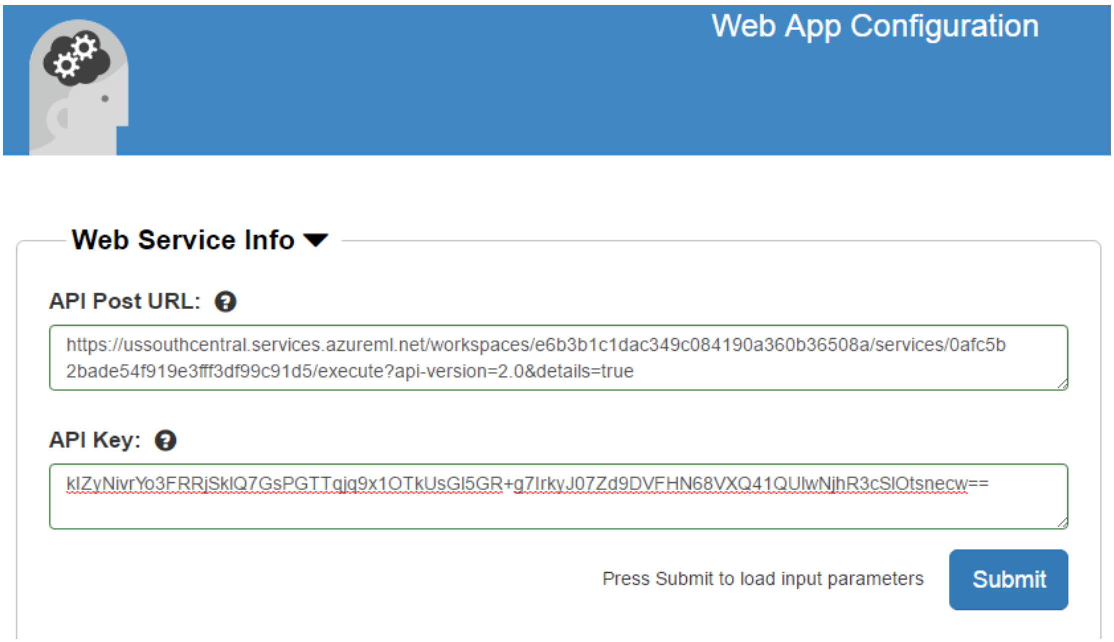
We used Azure Web Service to create Web App. In Azure studio, we create an Azure ML Request-Response Service Web App, and input the API key and URL of models.



This is the process to create a Web App in Azure Studio. After inputting name of App name and subscription, Azure will create a Web app for us. And it will give us a URL of this Web app.



Now, this page is just an empty page, and it does not have any function. We need click the URL and configure this page.



Now, we can copy the API key and URL of one of model to here to implements that model to this page.

List of Global Parameters

#	Name	Type	Alias	Description	Default	Min	Max
1	URL	string			http://www1.ece.		
2	CSV or TSV has header row	boolean			true		

After implementing models, we can set some default value to this page. In this setting page, we can change the alias, max and min value of each feature. We also can set default value to any of feature.

List of Output Parameters

#	Name	Type	Alias	Enabled
1	Date	string		ON
2	high	number		ON
3	low	number		ON
4	kVARh	number		OFF
5	Scored Labels	number		ON

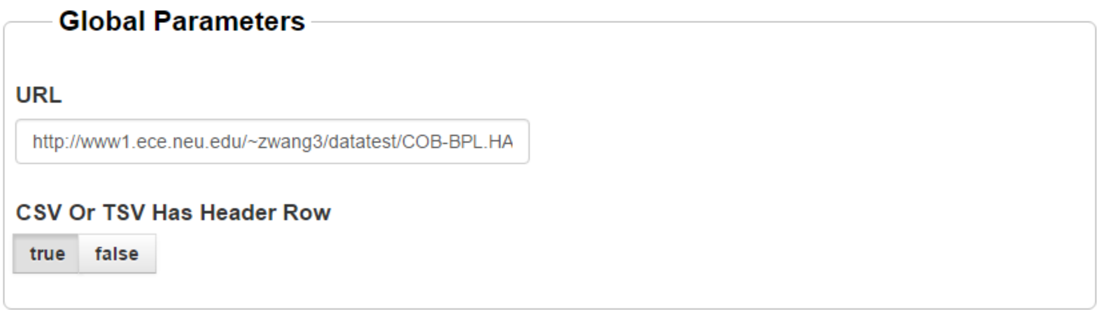
Save changes

In this setting page, we also can choose what features we want to show in the predict result table. Because this model is for predicting kVarh, so we decided to not show the original kVarh value in our result. Result will show the value of Date, High temperature, Low temperature user input, and the predictive kVarh value. Of course, we put similar setting in kWh and power factor setting page.

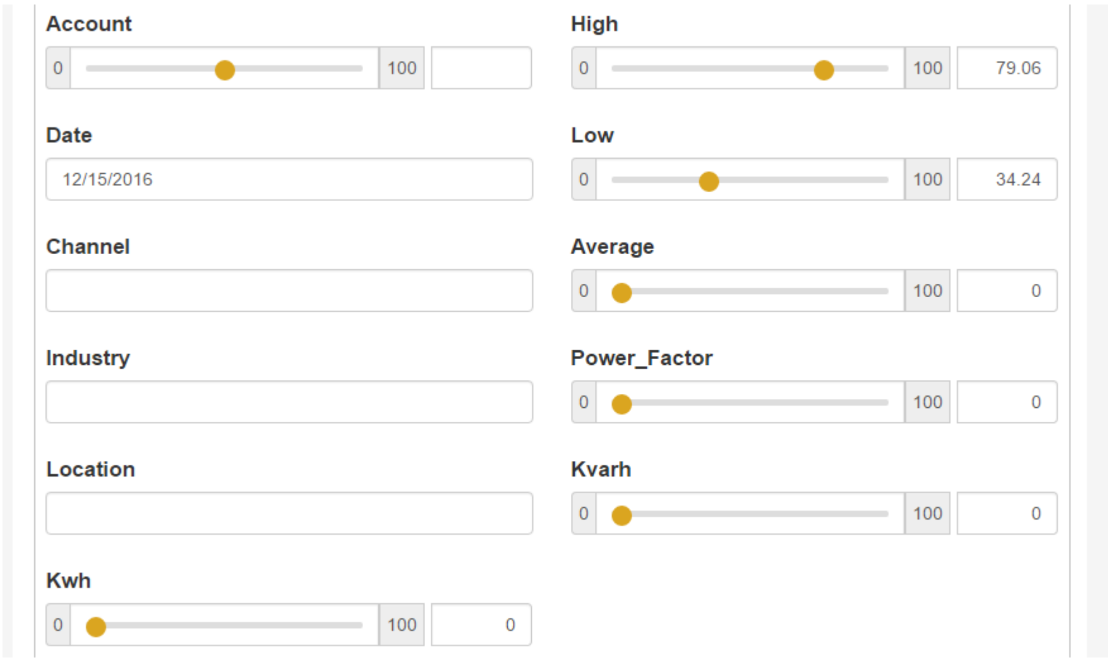
9.3 Result Testing

In this way, we created three predictive web page for kWh, power factor and kVarh prediction. The screen shots of result testing process are shown below.

9.3.1 kWh Prediction Testing



We have uploaded all datasets to COE server. Thus, each dataset has its own URL. We set one of dataset's URL as the URL default value. Of course, users can change URL value to the URL of the dataset they want to use.



Also, user should input each parameter's value. However, for each dataset, the Location, Industry, and Account are same. So, user need not input these three parameters.

Our models do not use Average Temperature and Channel feature, the reason has been explained before.

This model is for kWh prediction, so the model does not use power factor and kVarh variables. And users should not input kWh of course.

In conclusion, users only should input the Date they want to predict, the Highest Temperature and Lowest Temperature of that day.

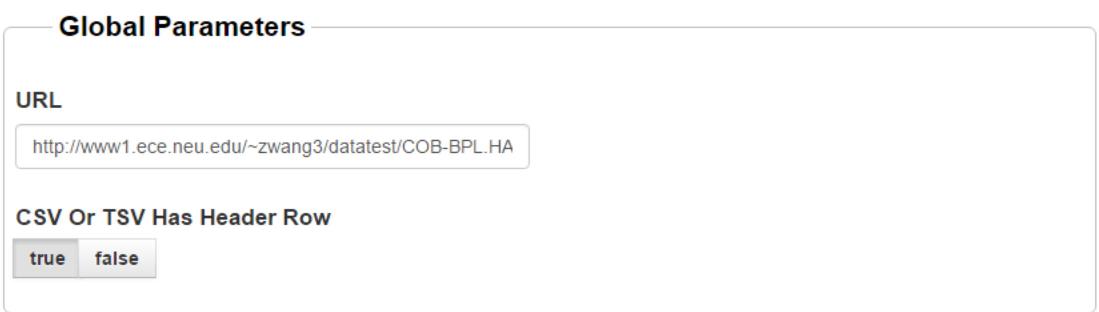
Submit

Result

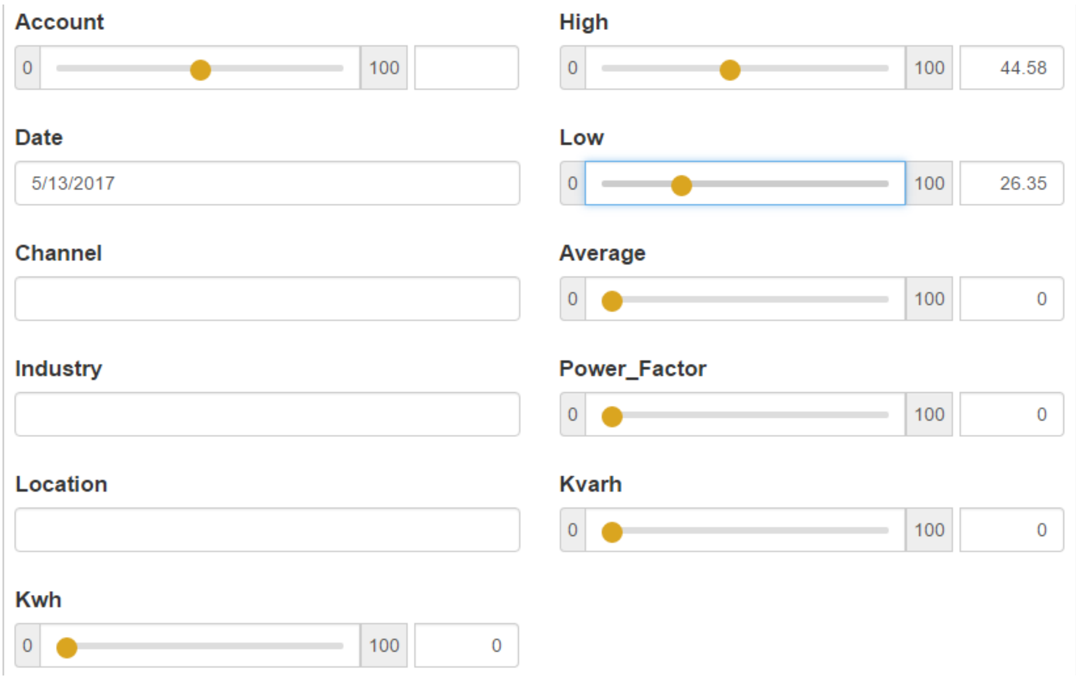
Label	Value
output1	
Date	12/15/2016 12:00:00 AM
High	79.06
Low	34.24
Predicted kWh	563.620300292969

After clicking Submit button, we can get a result table. The predicted total kWh value in 12/15/2016 was shown in the last line. This result come from the kWh prediction model created in Azure Machine Learning Workspace. About the result accuracy, we talked about it in Creating Models part before.

9.3.2 Power Factor Prediction Testing



Same URL default value with kWh page was show in URL text filed. Users can change this URL before prediction.



The Location, Industry, and Account are same for each dataset. So, user need not input these three parameters.

Our models do not use Average Temperature and Channel feature, the reason has been explained before.

This model is for power factor prediction, so the model does not use kWh and kVarh variables. And users should not input power factor of course.

In conclusion, users only should input the Date they want to predict, the Highest Temperature and Lowest Temperature of that day.

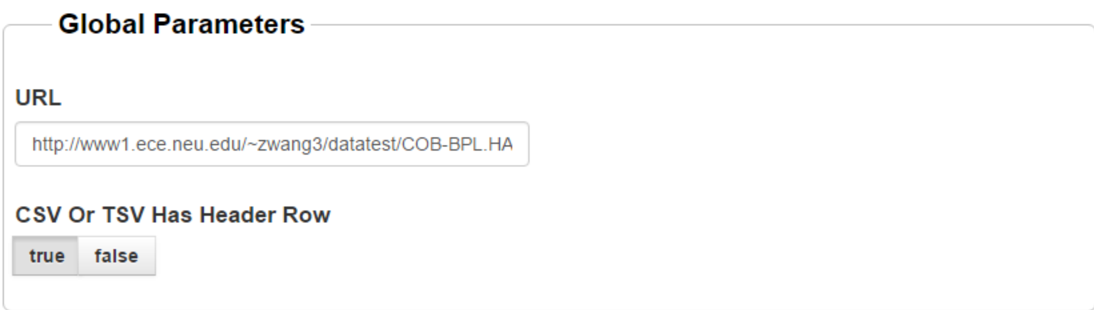
Submit

Result

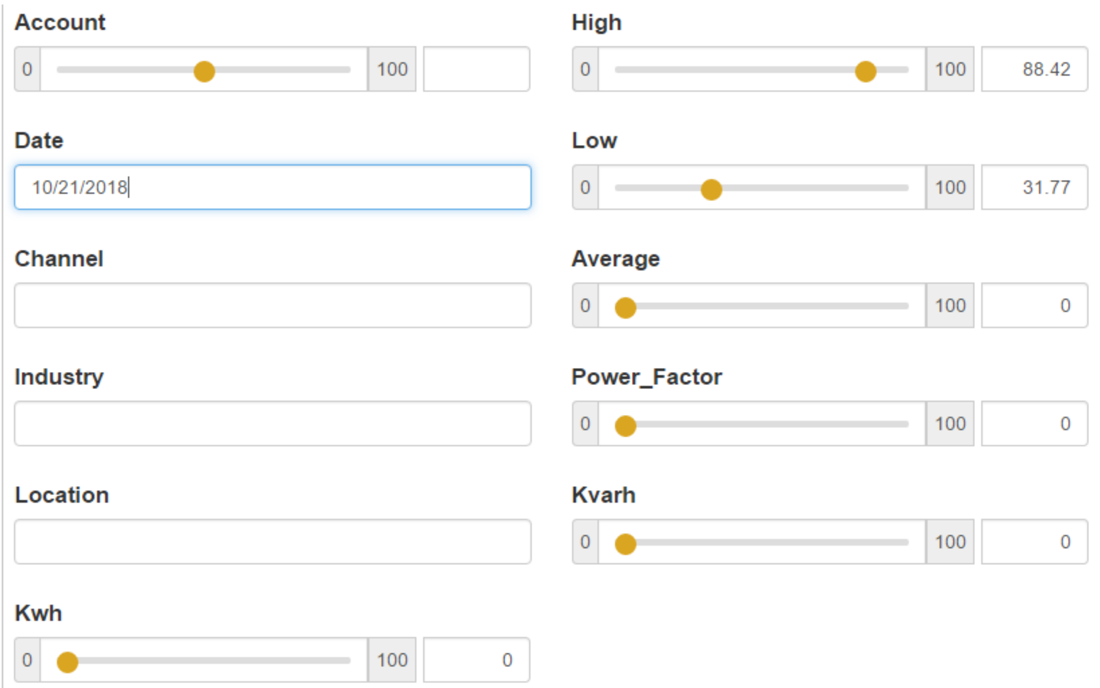
Label	Value
output1	
Date	5/13/2017 12:00:00 AM
High	44.58
Low	26.35
Predicted Power Factor	0.962981879711151

After clicking Submit button, we can get a result table. The predicted average power factor value for the 5/13/2017 was shown in the last line. This result come from the power factor prediction model created in Azure Machine Learning Workspace. About the result accuracy, we talked about it in Creating Models part before.

9.3.3 kVarh Prediction Testing



Same URL default value with kWh and power factor page was show in URL text filed. Users can change this URL before prediction.



The Location, Industry, and Account are same for each dataset. So, user need not input these three parameters.

Our models do not use Average Temperature and Channel feature, the reason has been explained before.

This model is for kVarh prediction, so the model does not use kWh and power factor variables. And users should not input kVarh of course.

In conclusion, users only should input the Date they want to predict, the Highest Temperature and Lowest Temperature of that day.

Submit

Result

Label	Value
output1	
Date	10/21/2018 12:00:00 AM
High	88.42
Low	31.77
Predicted Kvarh	131.051177978516

After clicking Submit button, we can get a result table. The predicted total kVarh value for 10/21/2018 was shown in the last line. This result come from the kVarh prediction model created in Azure Machine Learning Workspace. About the result accuracy, we talked about it in Creating Models part before.

10. Web App & UI

10.1. UI Tools

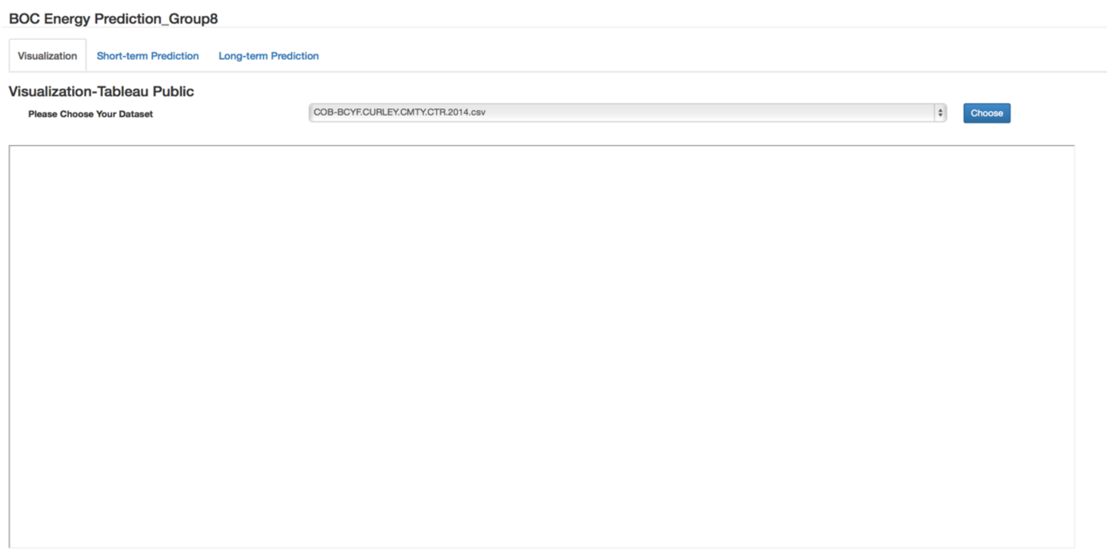
We develop our User Interface using HTML, CSS and Bootstrap. And we also combine our UI with web service UI from Azure.

10.2. Web App Explanation

1） Home Page

URL: http://www1.ece.neu.edu/~zwang3/energy_prediction_group8.html

The home page contains two parts: Visualization, Short-term prediction.

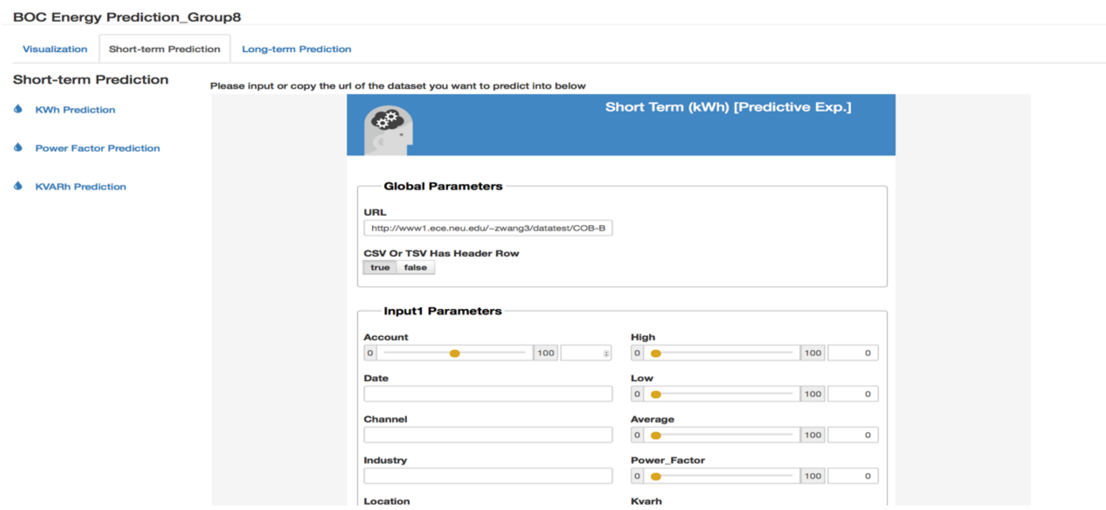


2）Visualization

After click the “Visualization” tab, user can choose the dataset he wants to analyze and predict with. After choose dataset and click “Choose” button, the visualization analysis for the chosen dataset will occur below. We use Tableau public for that.

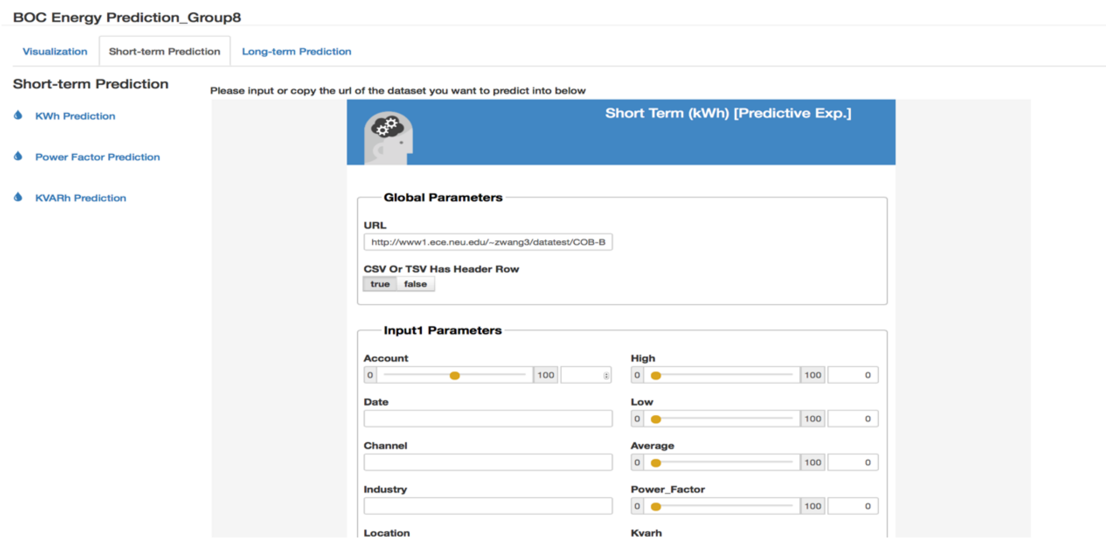
3） Short-term Prediction Page

After click “Short-term Prediction” tab, user will enter short-term prediction page. And it contains three prediction models: kWh, Power Factor, KVarh. The default one is for kWh.



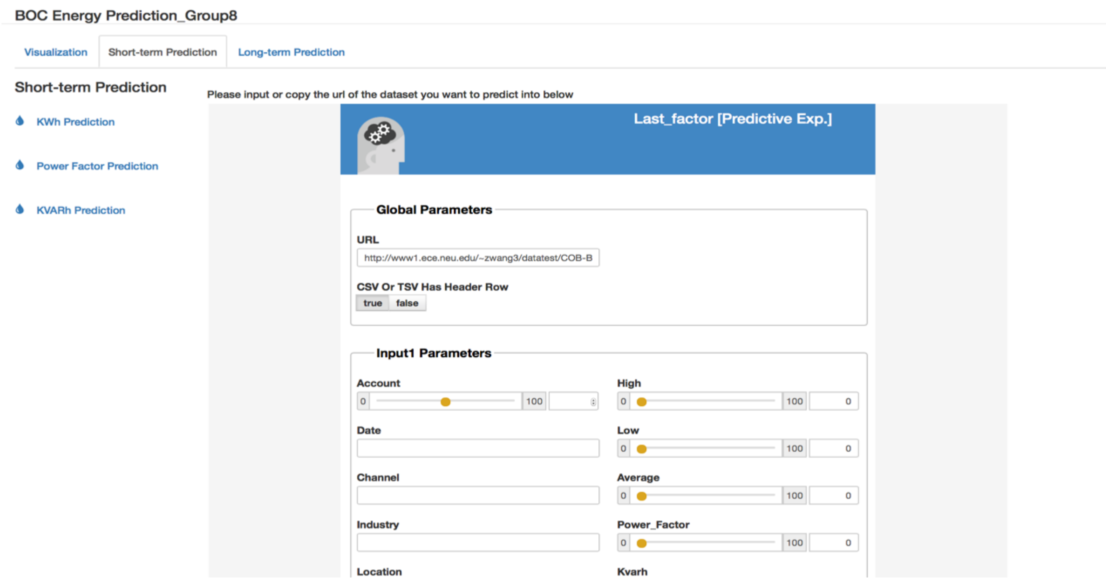
4） Short-term kWh Prediction Page

After choosing kWh prediction model, user can input or copy the URL of the dataset into input box, and then fill the prediction forms with parameters, and click “submit” button. Then the prediction output will show in the bottom.



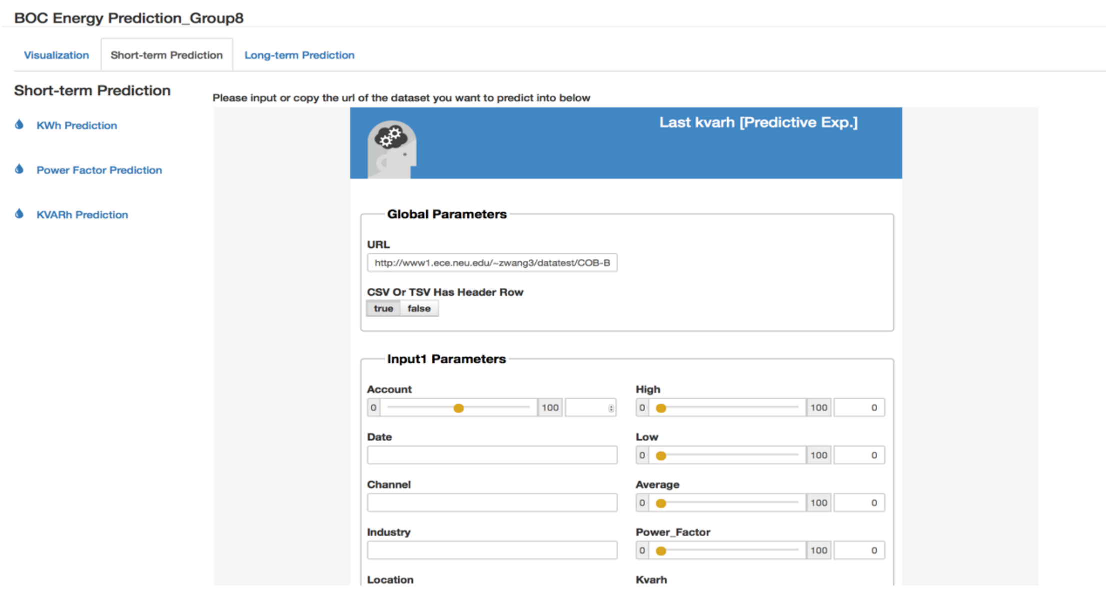
5） Short-term Power Factor Prediction Page

After choosing Power Factor prediction model, user can input or copy the URL of the dataset into input box, and then fill the prediction forms with parameters, and click “submit” button. Then the prediction output will show in the bottom.



6） Short-term KVarh Prediction Page

After choosing KVarh prediction model, user can input or copy the URL of the dataset into input box, and then fill the prediction forms with parameters, and click “submit” button. Then the prediction output will show in the bottom.



11. Challenges & Improvement in the Future

1. We should offer the long-term prediction to the customer. The Macro Analysis can predict energy consumption for future 1 week to 3 months, in order to help different industries to plan for future energy consumption, reduce power cost and save energy.

The most important challenge to complete that part is data quality and size, since the data set is proved by customers. The customer can easy change the data set by change the URL. For long-term scenarios, it is important to have high quality data that covers a span of multiple years (minimum 3 years, preferable 5-10 years). Unfortunately, data sets which we used now is only cover a single year, so the performance of the long-term predict model will decline and hard to predict the meaningful value.

2. The model which we now use can only predict a single result each time. But for both short-term and long-term prediction, it will be better, if we can predict multiple result at once. For example, for the short-term prediction, if the customer wants to predict the energy consumption for next 24 hours, the website will general the total amount of energy consumption for next 24 hours. It’s will be better if the website can predict hourly amount of energy consumption and draw a line chart to show the trend of energy consumption.

3. We should prove the better user experience by design the better user interface and offer better interactive. We should combine the visualization part with the prediction part better.