Solar power generation prediction based on weather and previous generation forecasts

Team deeplearning
Lee jung-hwan
Kim dong-hwan
Kwon ga-min
Park jun-young
Park chae-won

Index


Recap

Modeling


Final model

Recap

Recap

Background of Topic selection

가치창출대학
POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

HENERGY

Solar Power Generation Forecast Competition hosted

by POSTECH and H energy

Recap

Competition Rules Explanation - Forecast Incentive (Settlement) Calculation

Forecast incentives are calculated according to the forecast error rate for each time period

Recap

Competition Rules Explanation - Forecast Incentive (Settlement) Calculation

During the competition period, the higher the total forecast incentive (settlement amount) sum, the better the evaluation

Recap

Correlation

Professor KIM

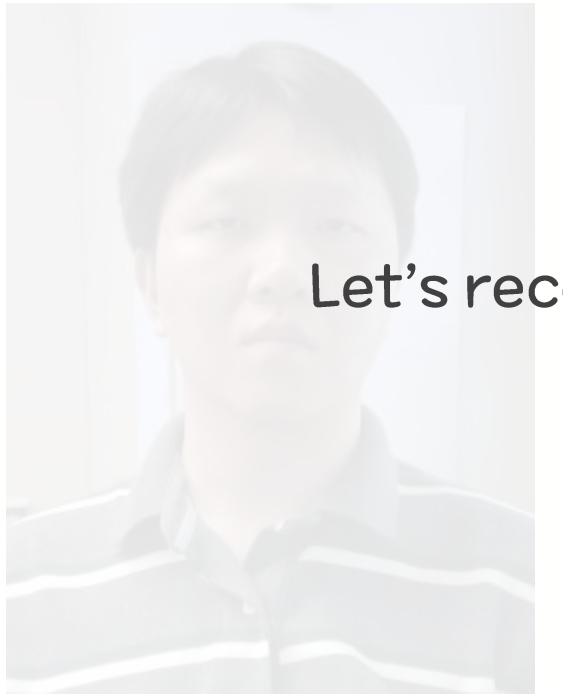
In bootstrapping, there is no guarantee that the correlation will be maintained for each sample. You must calculate the correlation for each sample

In the second semester

Statistical Data mining sample reuse method lecture

Recap

Correlation



Professor KIM

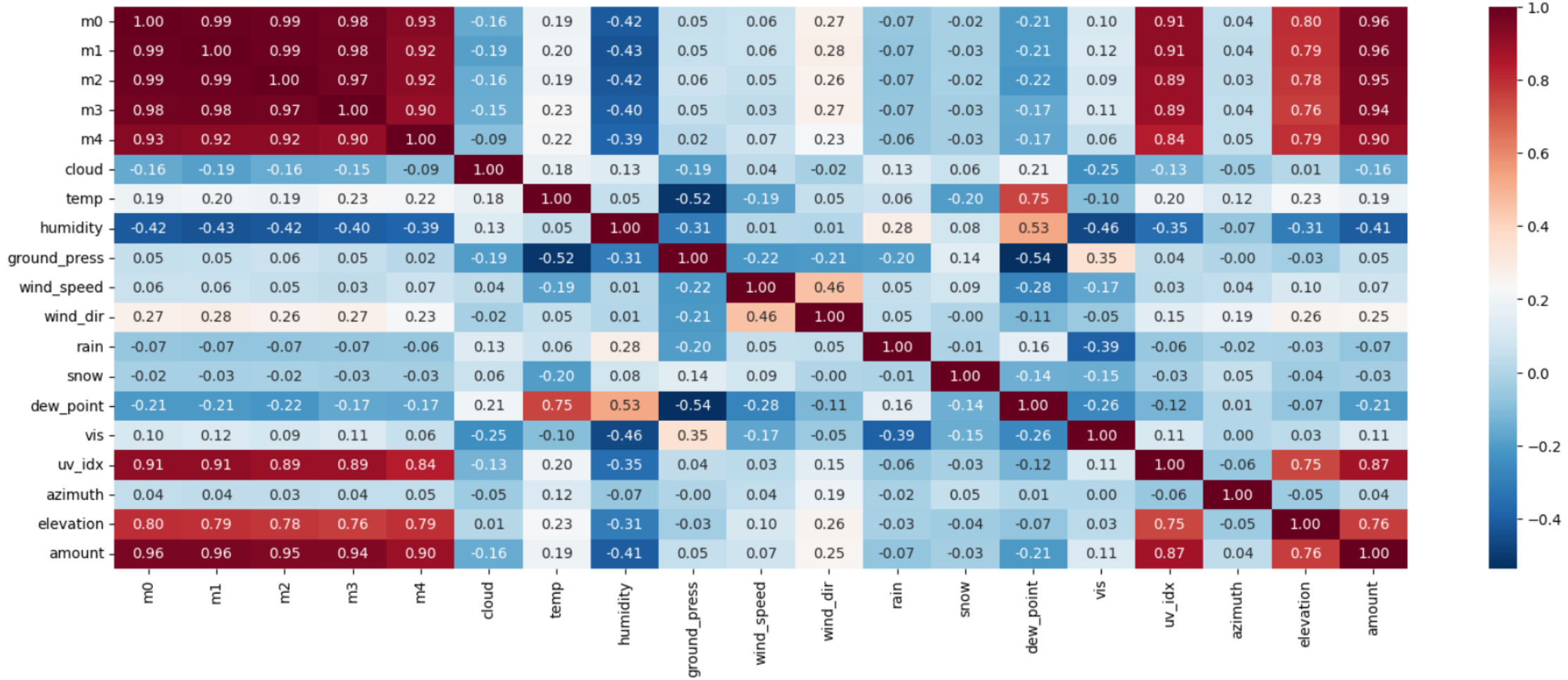
In bootstrapping, there is no guarantee that the correlation will be maintained for each sample. You must calculate the correlation for each sample

In the second semester
Statistical Data mining sample reuse method lecture

Let’s recalculate the correlation for Oct-Nov

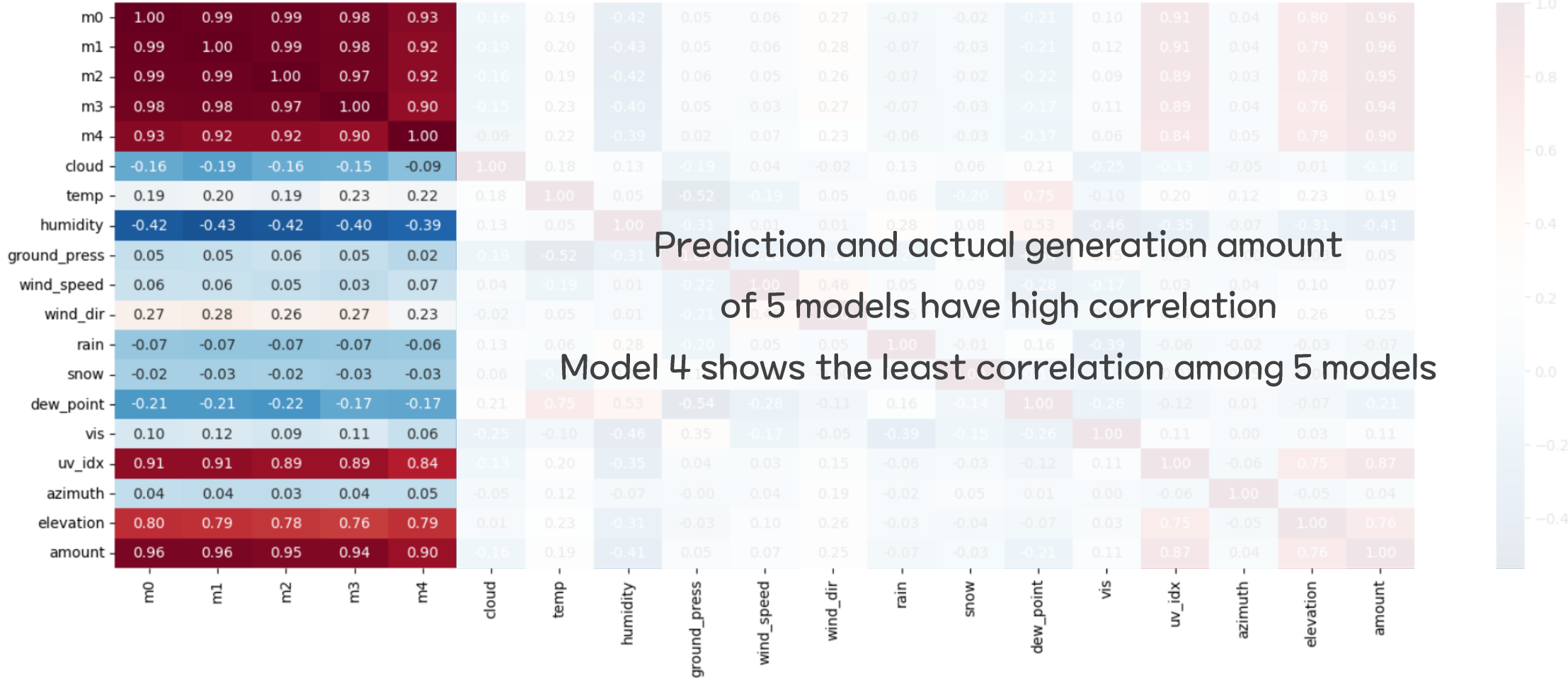
Recap

Correlation heatmap Oct-Nov



Recap

Correlation heatmap Oct-Nov



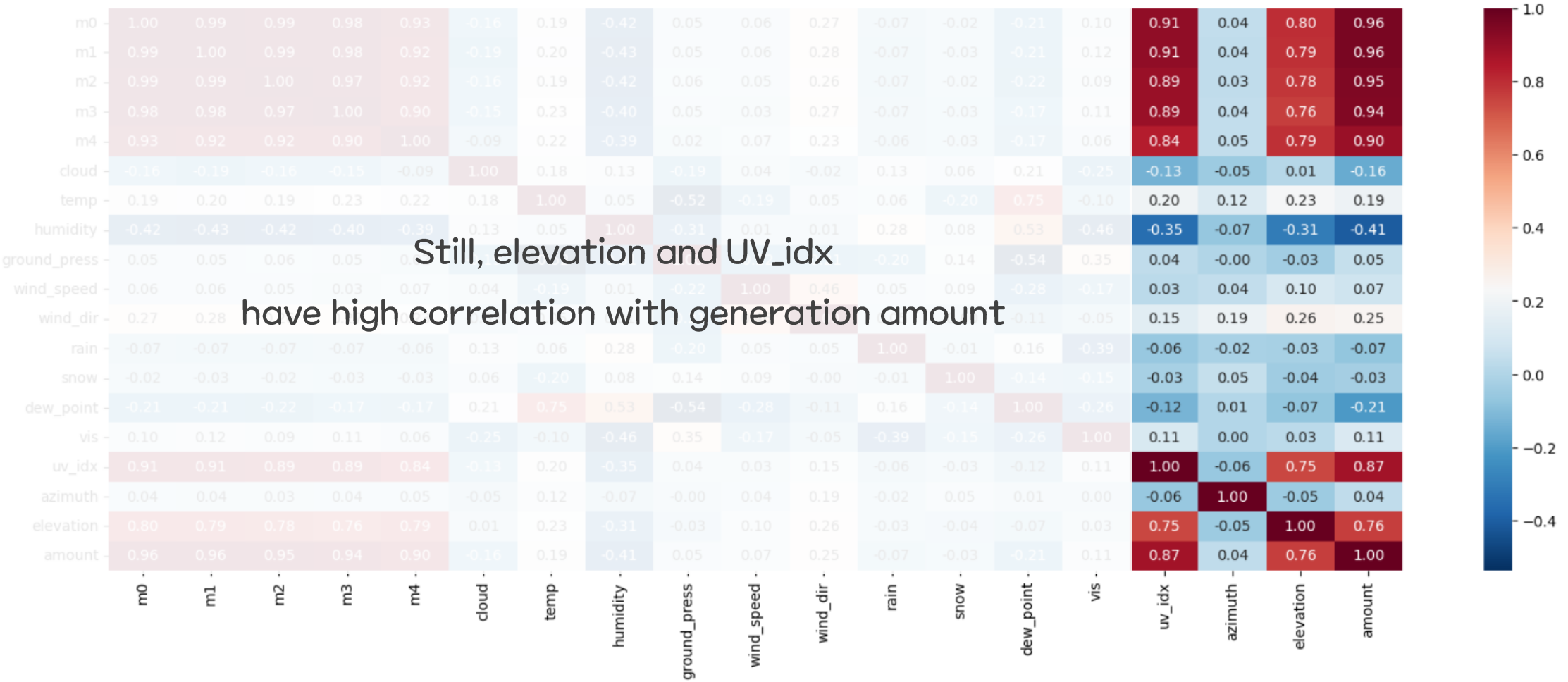
Prediction and actual generation amount

of 5 models have high correlation

Model 4 shows the least correlation among 5 models

Recap

Correlation heatmap Oct-Nov



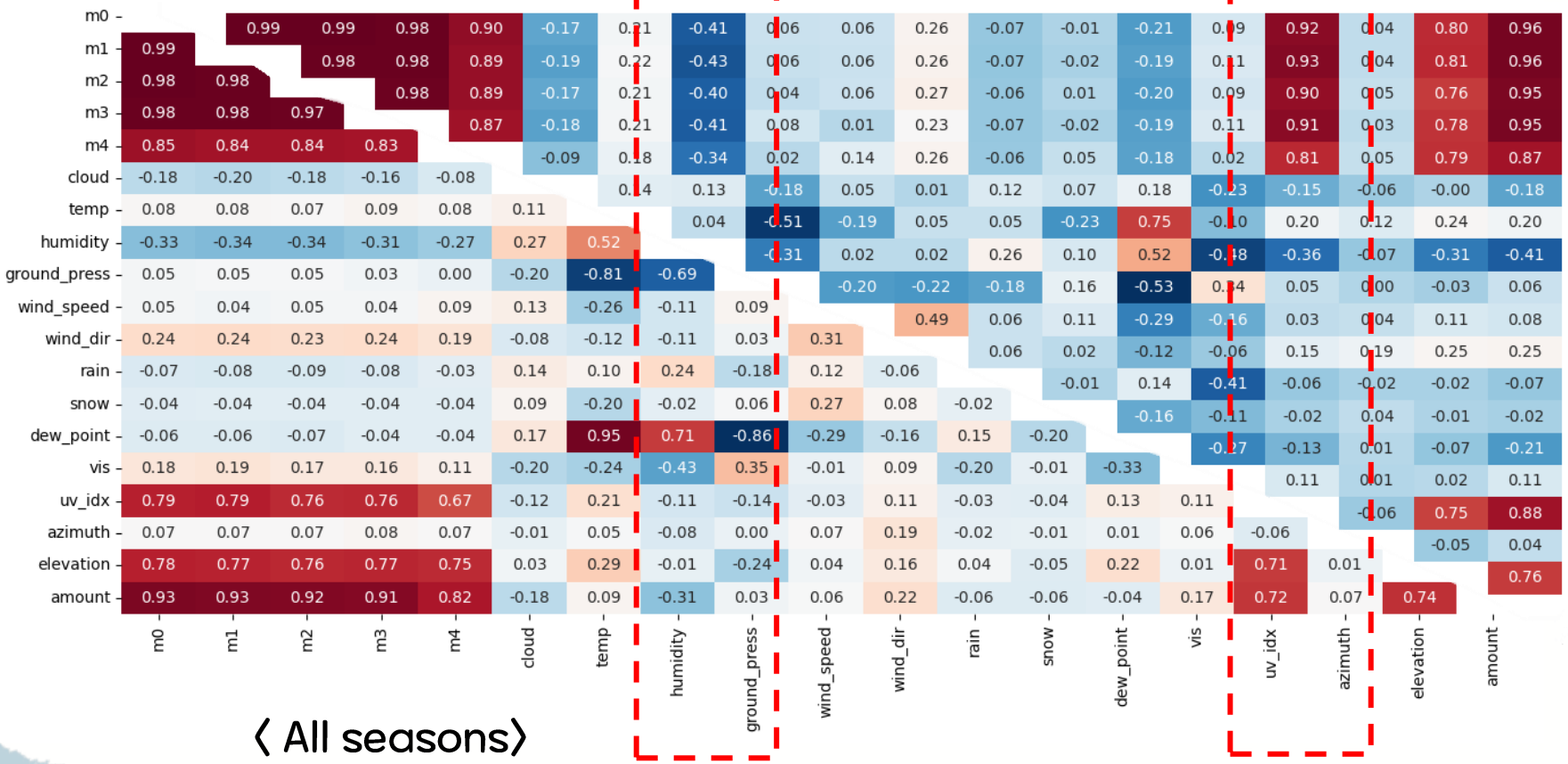
Still, elevation and UV_idx

have high correlation with generation amount

Recap

Correlation heatmap Oct-Nov

〈 Fall 〉



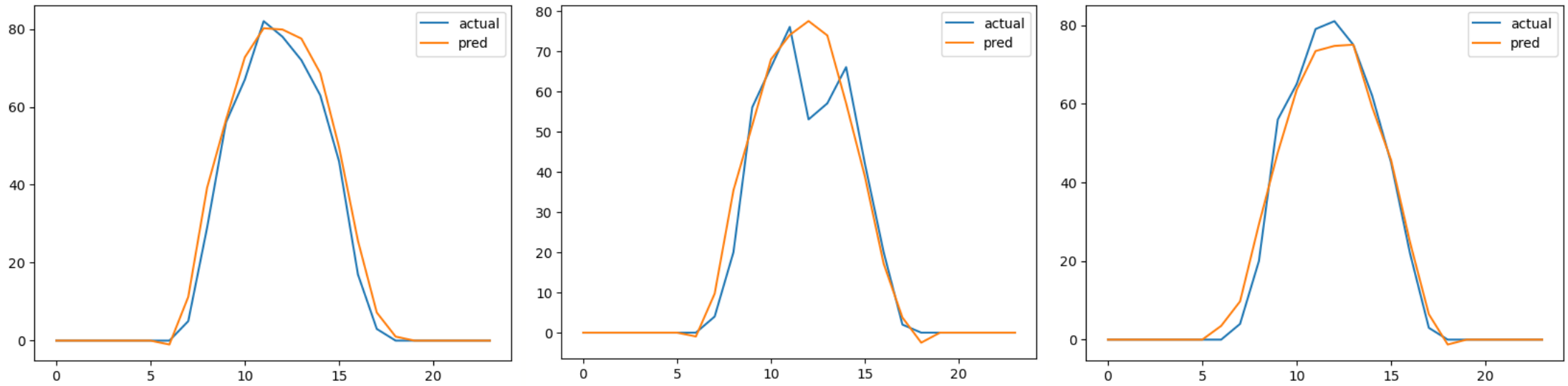
〈 All seasons〉

In conclusion, while there are slight differences in the degree of the correlation coefficients, The overall patterns are similar.

The negative correlation between humidity and generation, and the positive correlation between UV_idx and generation, are more pronounced.

Recap

Week 1 Announcements



Previously announced plans to address spike points

Time-series clustering, vector similarity etc.

Recap

Week 1 Announcements

Let's get start

Previously announced plans to address spike points

Time-series clustering, vector similarity etc.

Modeling

Modeling

Clustering

hour	model0	model1	model2	model3	model4
7:00:00	0.8955	2.7081	0.6543	1.4359	4.5149
8:00:00	2.6194	4.9056	4.4764	3.2249	7.1268
9:00:00	7.7035	8.3473	11.0274	8.0444	12.7201
10:00:00	10.2554	10.6475	13.5739	10.0637	16.9746
11:00:00	12.8063	13.2360	14.2907	12.5647	19.7866
12:00:00	13.0861	13.8327	14.3313	13.7187	23.5350
13:00:00	12.5749	12.7261	13.3266	13.7153	21.6607
14:00:00	13.0941	13.3052	13.8213	14.0544	20.8093
15:00:00	12.5095	12.7522	13.3478	13.5479	19.0190
16:00:00	9.6196	9.9271	10.4812	11.1831	15.8285
17:00:00	6.3971	6.4595	7.8366	7.6823	10.8050
18:00:00	2.4153	3.7629	2.9694	2.7476	6.6126
19:00:00	1.1412	1.9345	0.4829	1.1759	4.7196

Round1/Round2

error rate of each model

Could applying different models based on error rate patterns improve the accuracy of each model?

Modeling

Clustering

hour	model0	model1	model2	model3	model4
7:00:00	0.8955	2.7081	0.6543	1.4359	4.5149
8:00:00	2.6194	4.9056	4.4764	3.2249	7.1268
9:00:00	7.7035	7.3478	[illegible]	8.0444	12.7201
10:00:00	10.2554	10.6475	[illegible]	10.0637	16.9716
11:00:00	12.8063	13.2360	[illegible]	12.5647	19.7866
12:00:00	13.0861	13.8327	14.3313	13.7187	23.5350
13:00:00	12.5749	12.7261	13.3266	13.1158	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
15:00:00	12.5095	12.7522	13.3478	13.5479	19.0190
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
17:00:00	6.3971	6.4595	7.8366	7.6823	10.8050
18:00:00	2.4153	3.7629	2.9694	2.7476	6.6128
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Round1/Round2

error rate of each model

After clustering based on each model's error rates,

let's apply the regression model which have solid

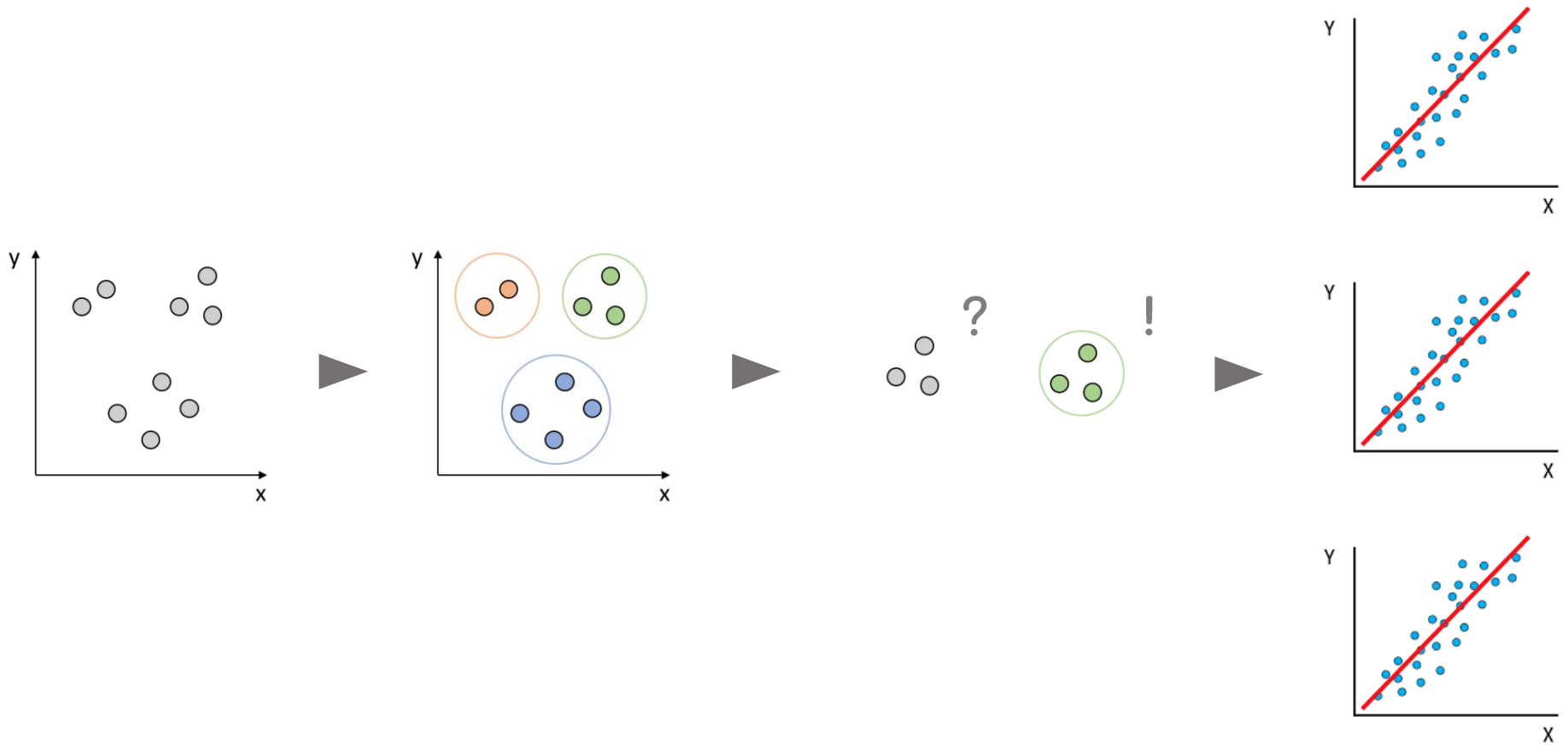
performance differently for each cluster!

Could applying different models based on error rate

patterns improve the accuracy of each model?

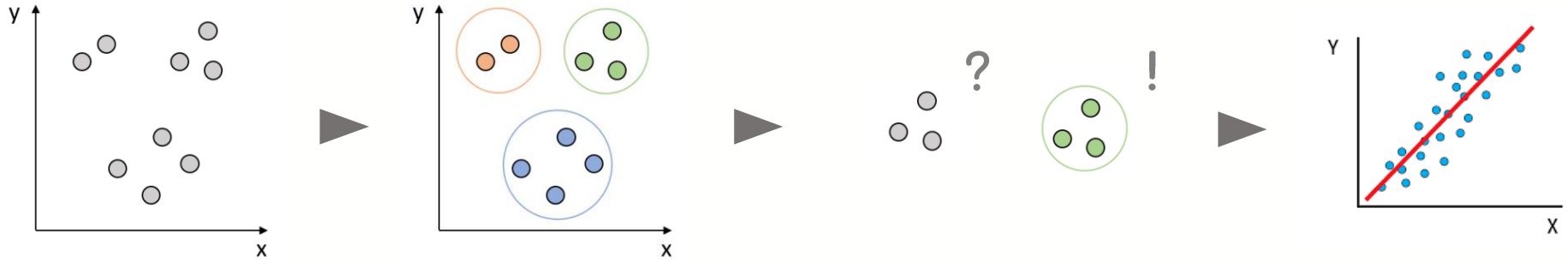
Modeling

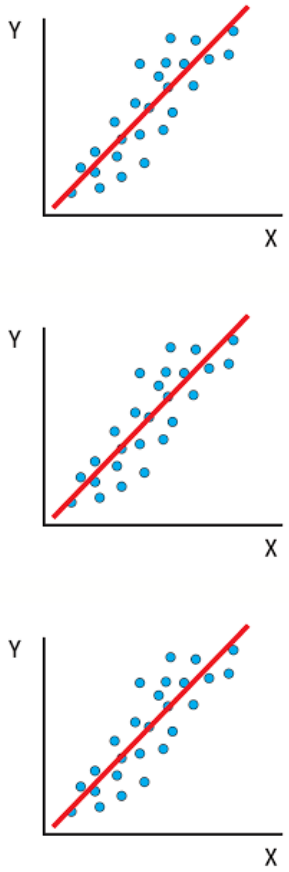
Try 1. Clustering based on error rate



Modeling

Try 1. Clustering based on error rate





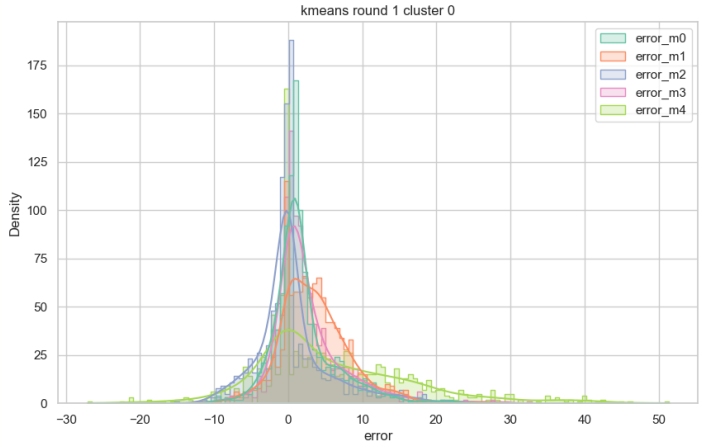
Step 1

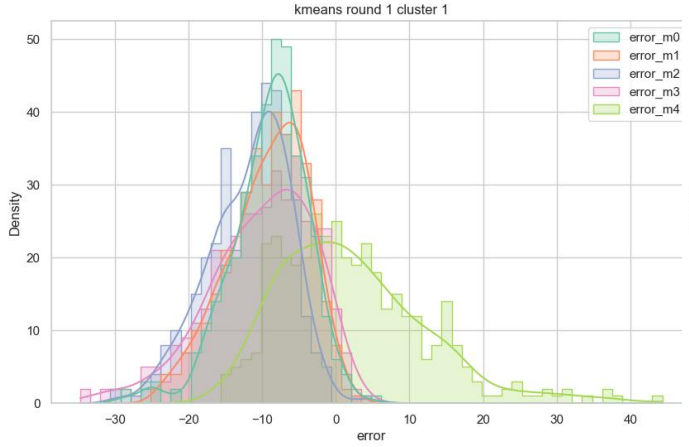
K-means Clustering

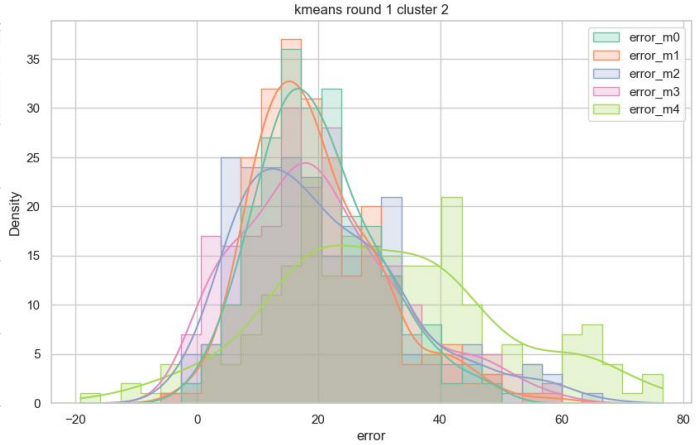
On error rate

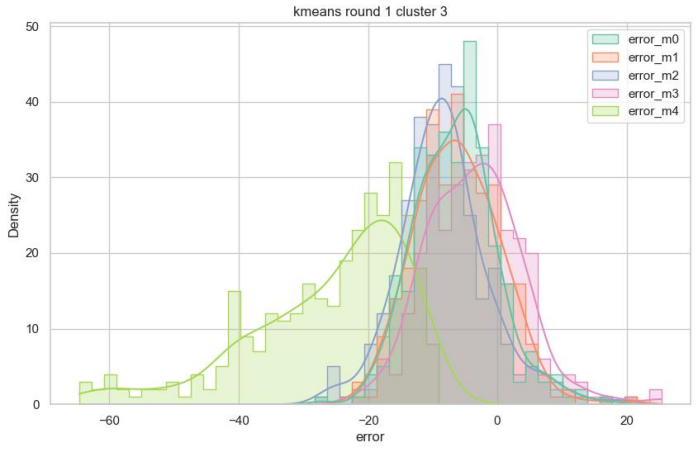
Modeling

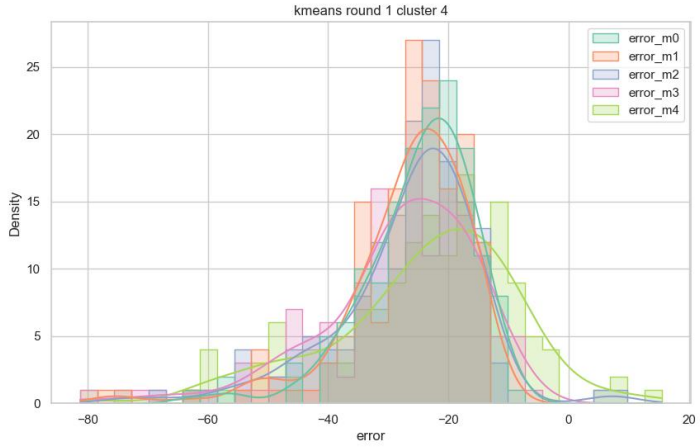
Try 1. Clustering based on error rate





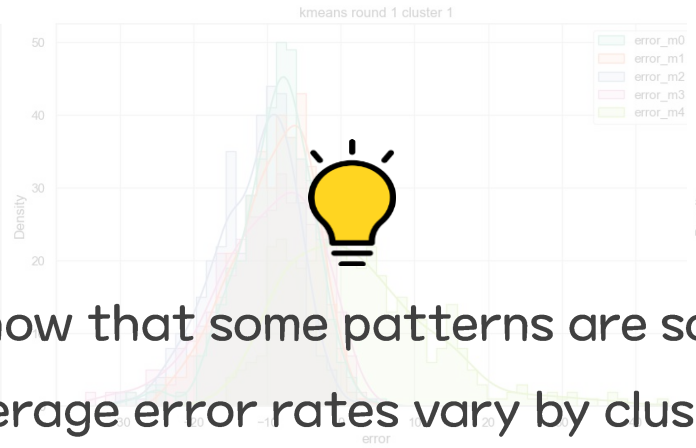


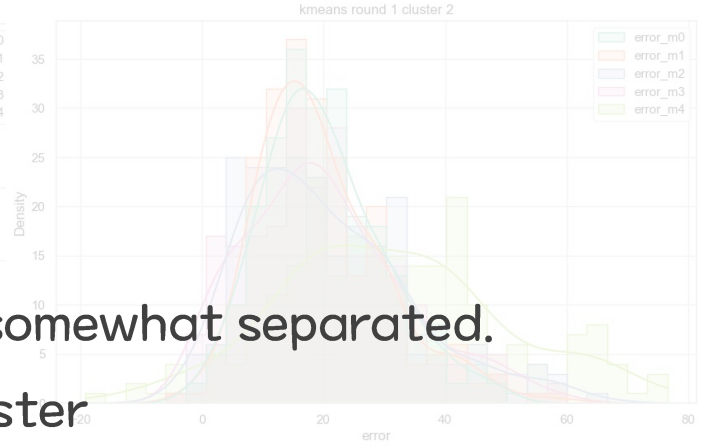




Modeling

Try 1. Clustering based on error rate



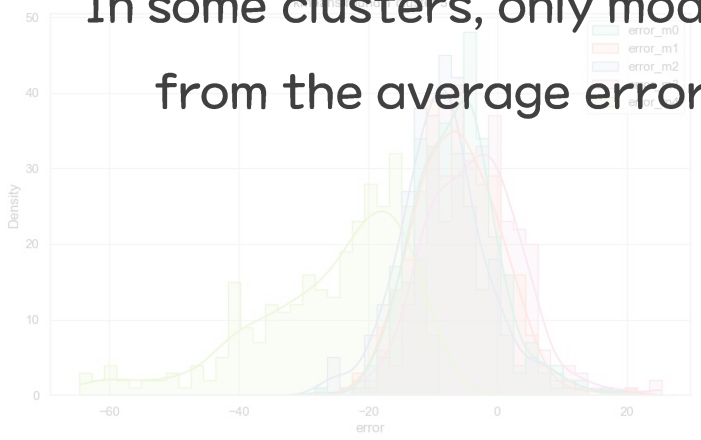


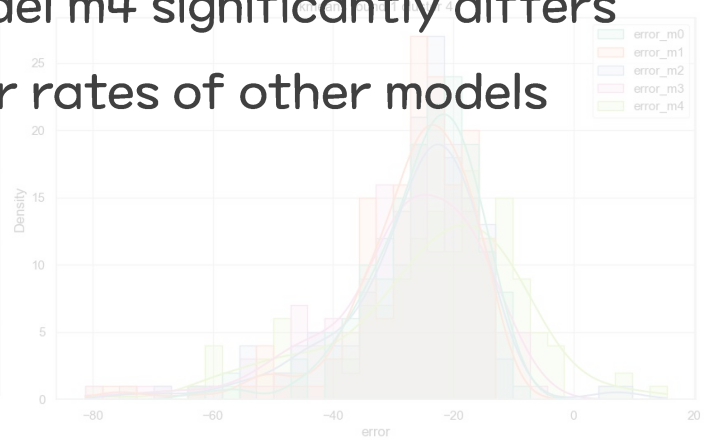
Clustering results show that some patterns are somewhat separated.

Average error rates vary by cluster

In some clusters, only model m4 significantly differs

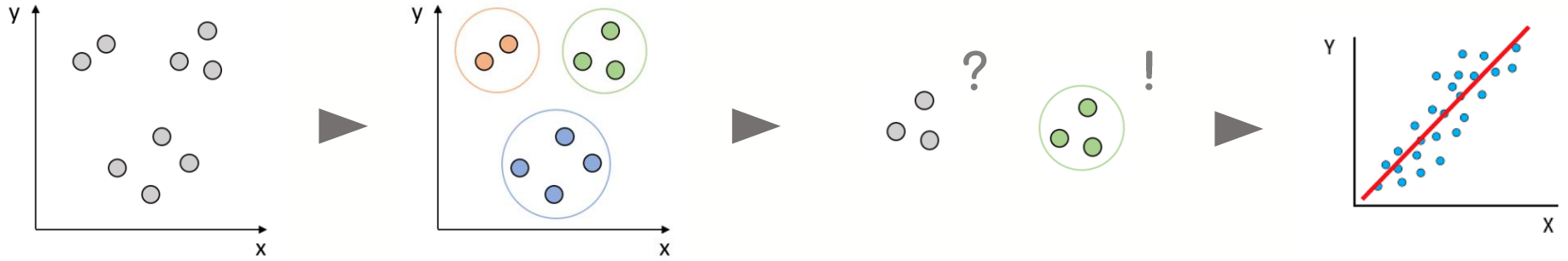
from the average error rates of other models





Modeling

Try 1. Clustering based on error rate



Step 2

Linear Regression fitting for each cluster

Modeling

Try 1. Clustering based on error rate

Coefficient by cluster

	coef
const	6.5330
m0	0.3787
m1	0.0846
m2	0.1648
m3	0.1669
m4	0.2311

Cluster 0

	coef
const	-1.6935
m0	0.0855
m1	0.2778
m2	0.2778
m3	0.1540
m4	0.1744

Cluster 1

	coef
const	24.6682
m0	0.4918
m1	-0.1858
m2	0.1414
m3	0.2227
m4	0.2427

Cluster 2

	coef
const	-12.1329
m0	0.3223
m1	0.2492
m2	-0.0857
m3	0.1465
m4	0.1476

Cluster 3

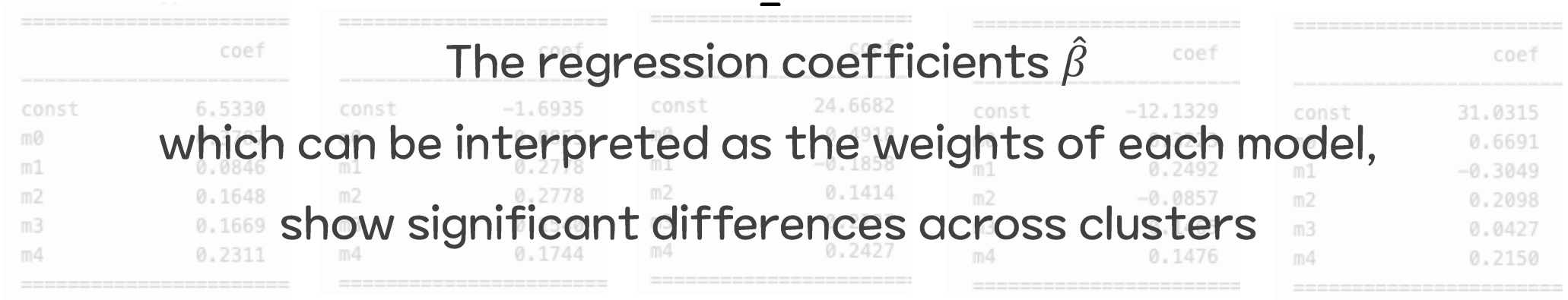
	coef
const	31.0315
m0	0.6691
m1	-0.3049
m2	0.2098
m3	0.0427
m4	0.2150

Cluster 4

Modeling

Try 1. Clustering based on error rate

Coefficient by cluster



	coef
const	6.5330
m0	[illegible]
m1	0.0846
m2	0.1648
m3	0.1669
m4	0.2311

	coef
const	−1.6935
m0	[illegible]
m1	0.2778
m2	0.2778
m3	[illegible]
m4	0.1744

	coef
const	24.6682
m0	[illegible]
m1	−0.1858
m2	0.1414
m3	[illegible]
m4	0.2427

	coef
const	−12.1329
m0	[illegible]
m1	0.2492
m2	−0.0857
m3	[illegible]
m4	0.1476

	coef
const	31.0315
m0	[illegible]
m1	0.6691
m2	−0.3049
m3	0.2098
m4	0.0427
m5	0.2150

Cluster 0 Cluster 1 Cluster 2 Cluster 3 Cluster 4

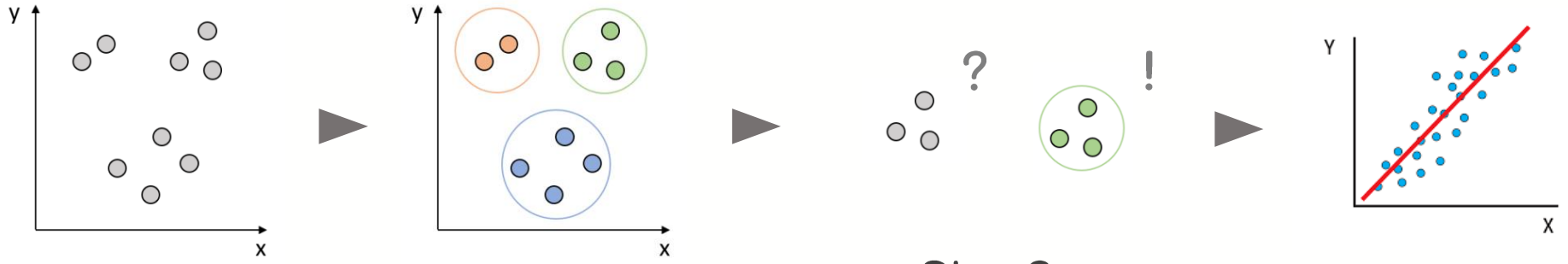
The regression coefficients $\hat{\beta}$

which can be interpreted as the weights of each model,

show significant differences across clusters

Modeling

Try 1. Clustering based on error rate



Step 3

Use a classification model

to estimate which cluster

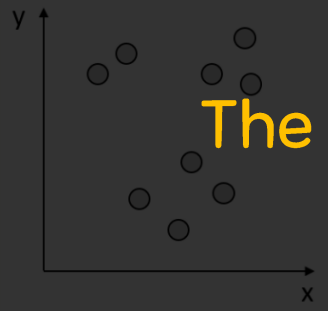
the data to be predicted belongs to

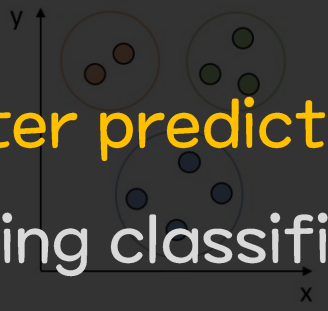
Modeling

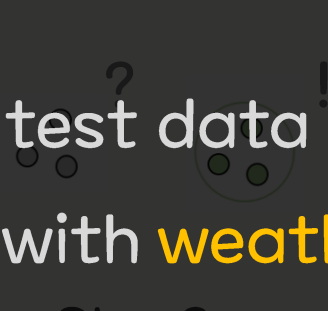
Try 1. Clustering based on error rate

Clustering was based on error rate,

But since, actual Y is need to calculate the error rate,







The cluster prediction for test data should be done

Using classification with weather data

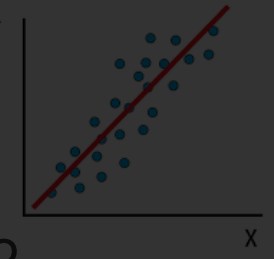
(Test data do not have Y variable)

Step 3

Use a classification model

to estimate which cluster

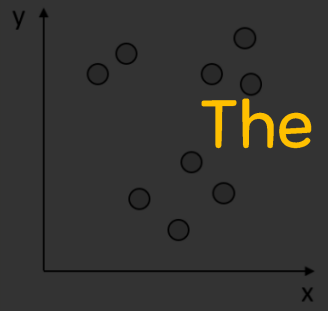
the data to be predicted belongs to

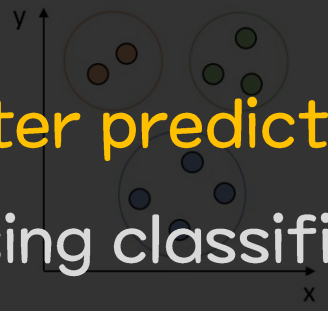


Modeling

Clustering was based on error rate,

But since, actual Y is need to calculate the error rate,

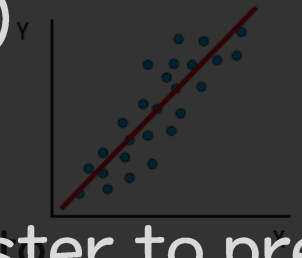




The cluster prediction for test data should be done

Using classification with weather data

(Test data do not have Y variable)

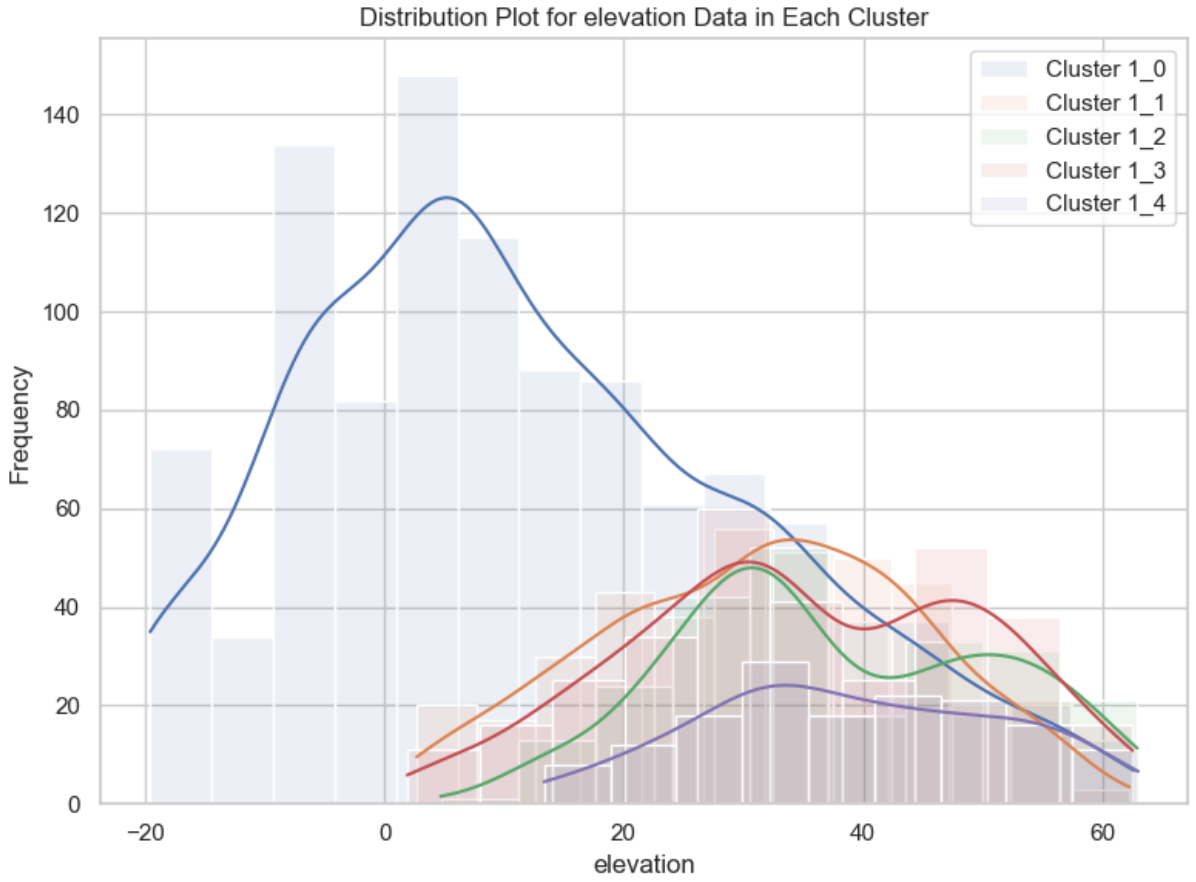


Weather features must be distinct for each cluster to predict

the cluster for the test data

Modeling

Try 1. Clustering based on error rate



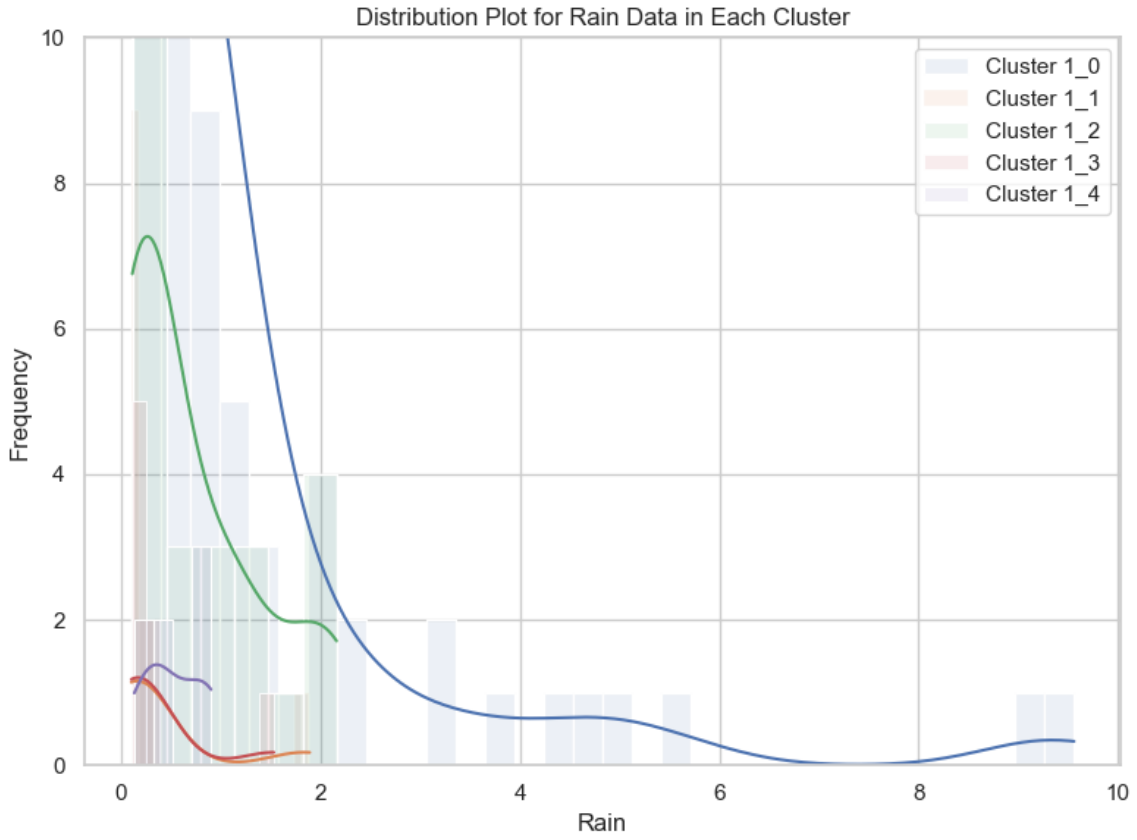
elevation:

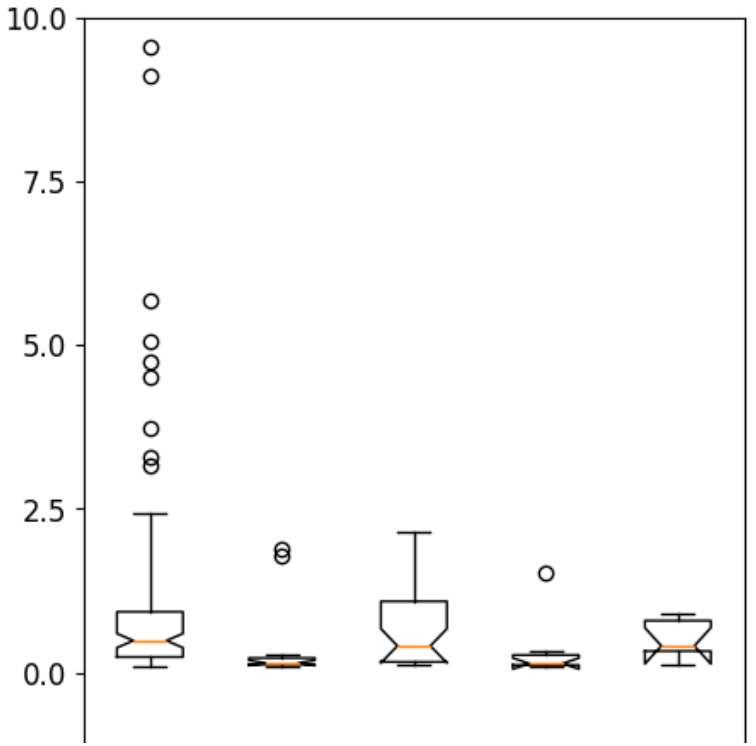
Cluster0 is clearly distinct with the other clusters.

Various patterns appeared in the afternoon time zone, so clusters at this time are considered diverse

Modeling

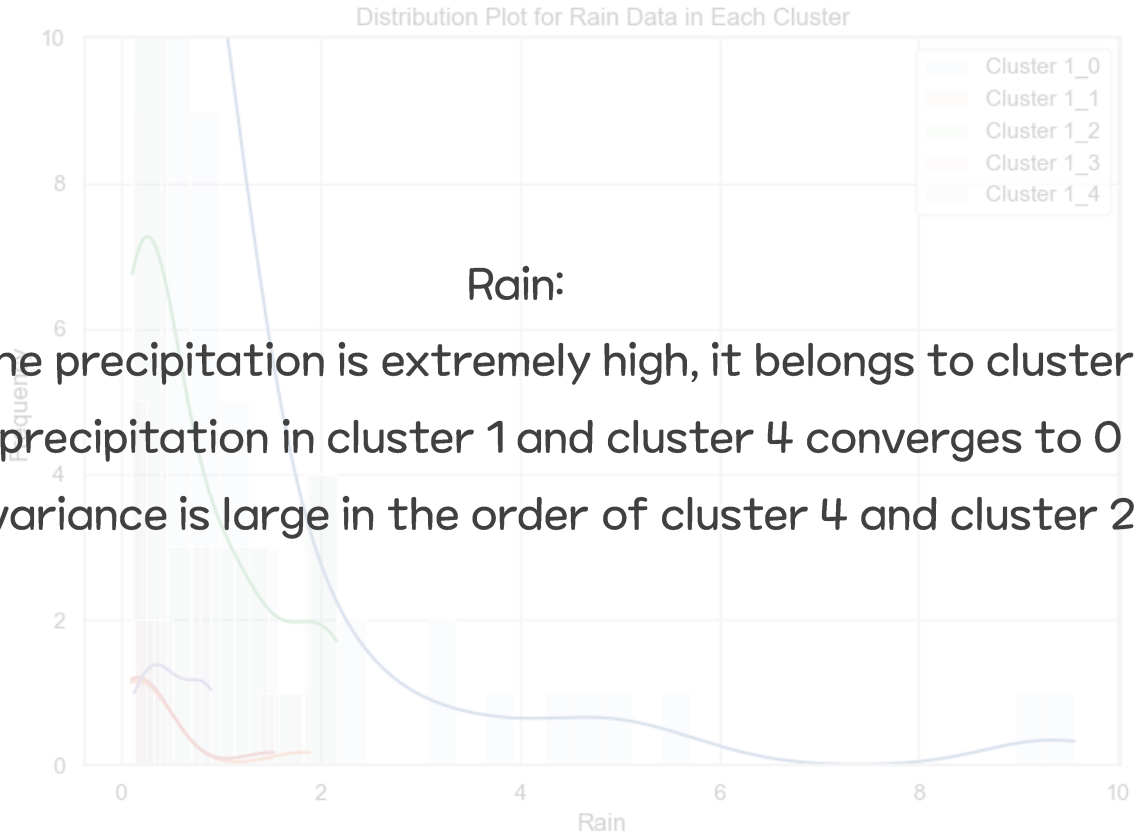
Try 1. Clustering based on error rate





Modeling

Try 1. Clustering based on error rate

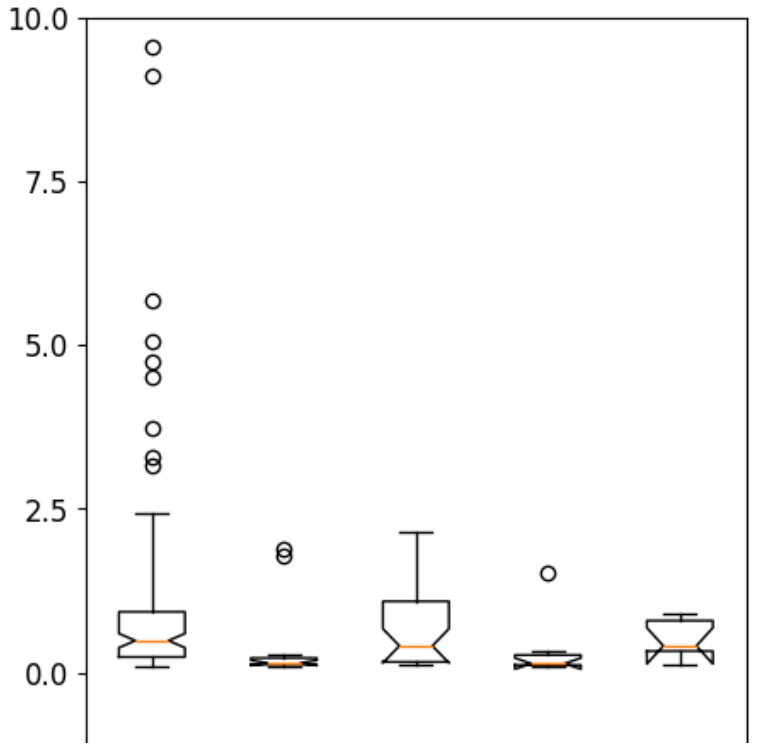


Rain:

When the precipitation is extremely high, it belongs to cluster 0.

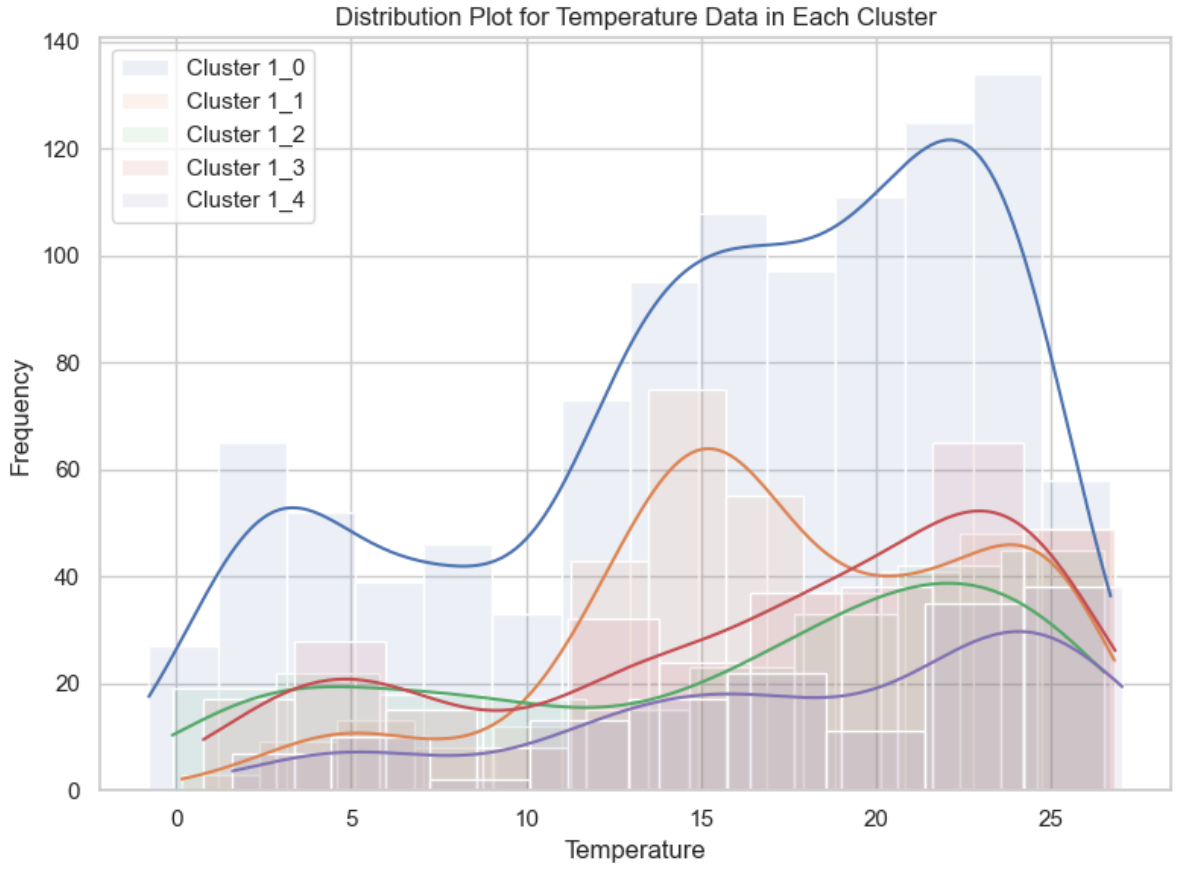
The precipitation in cluster 1 and cluster 4 converges to 0

The variance is large in the order of cluster 4 and cluster 2



Modeling

Try 1. Clustering based on error rate



Temperature:

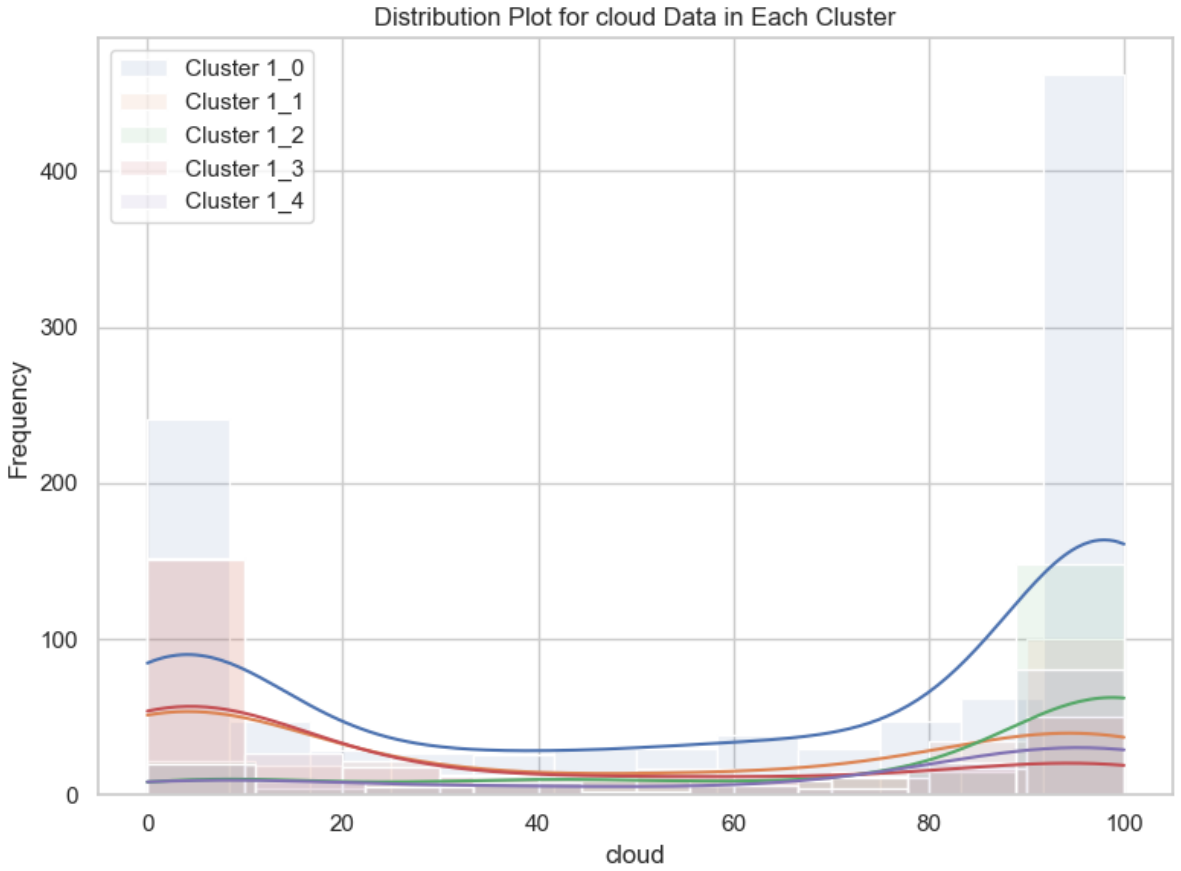
Cluster 0: Temperature is relatively evenly distributed

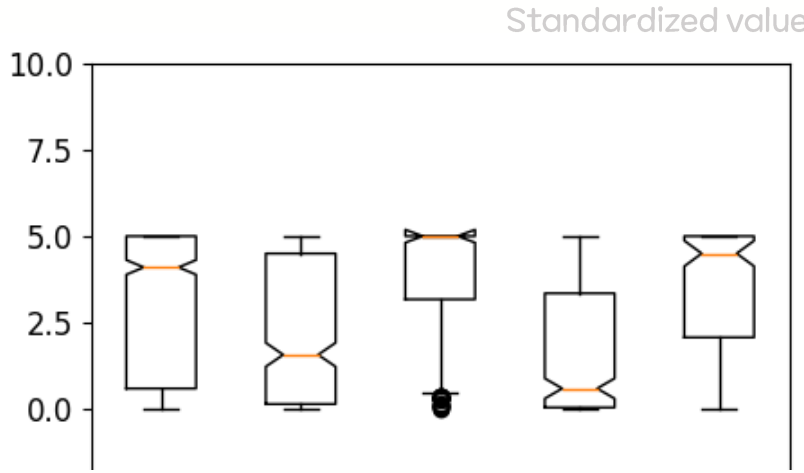
Cluster 1: Most values are between 10˚C and 20˚C

Other clusters: Most values are above 20˚C

Modeling

Try 1. Clustering based on error rate





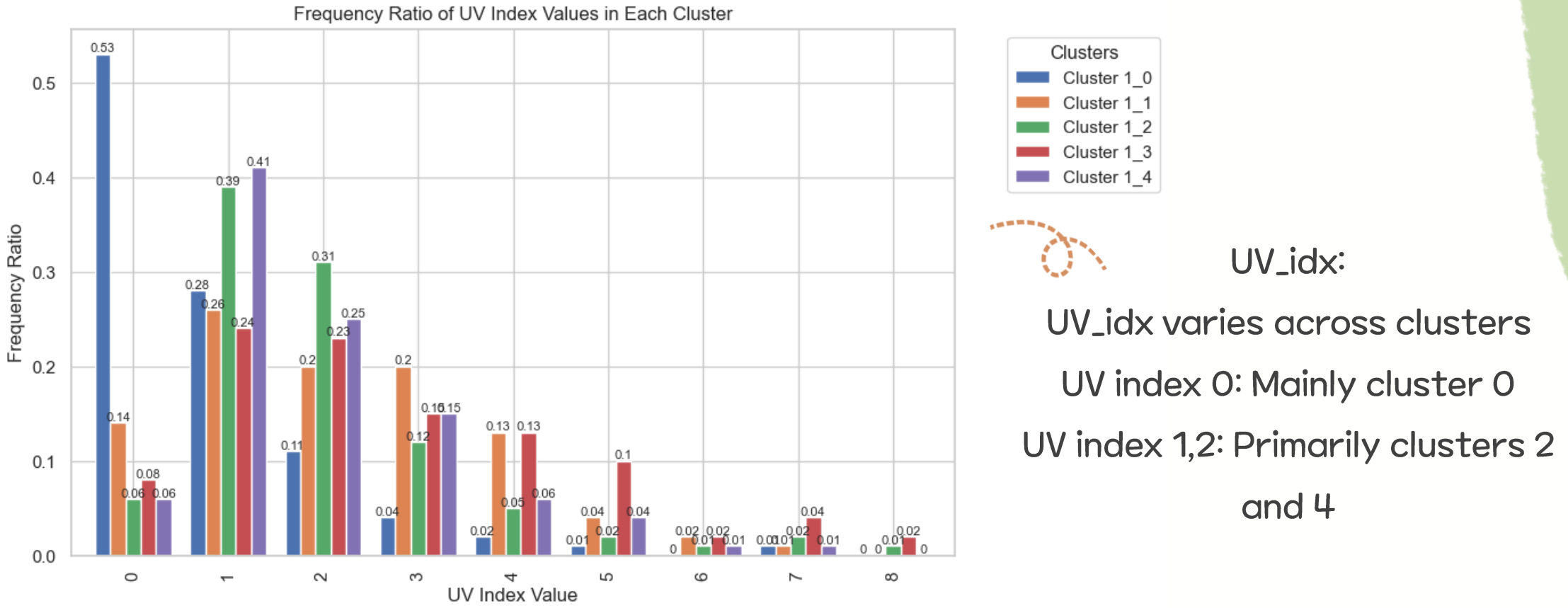
cloud:

Cluster1,4 : Median is very small

Cluster 0, 2, 4: Median is very close to maximum value

Modeling

Try 1. Clustering based on error rate



UV_idx:

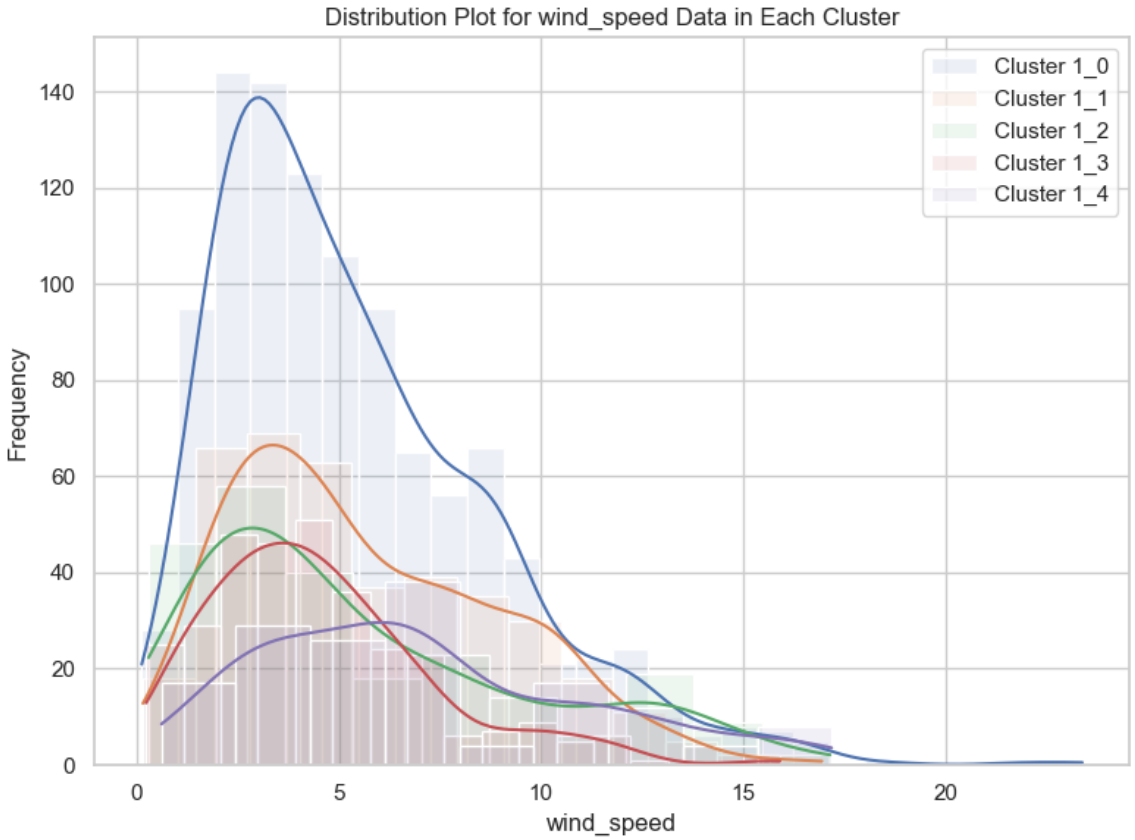
UV_idx varies across clusters

UV index 0: Mainly cluster 0

UV index 1,2: Primarily clusters 2 and 4

Modeling

Try 1. Clustering based on error rate

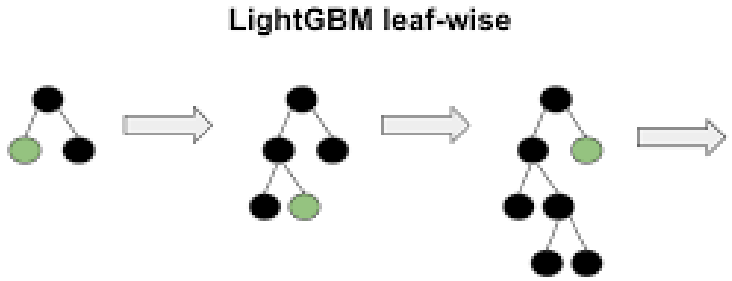


wind_speed:

No distinct features observed

Modeling

Try 1. Clustering based on error rate



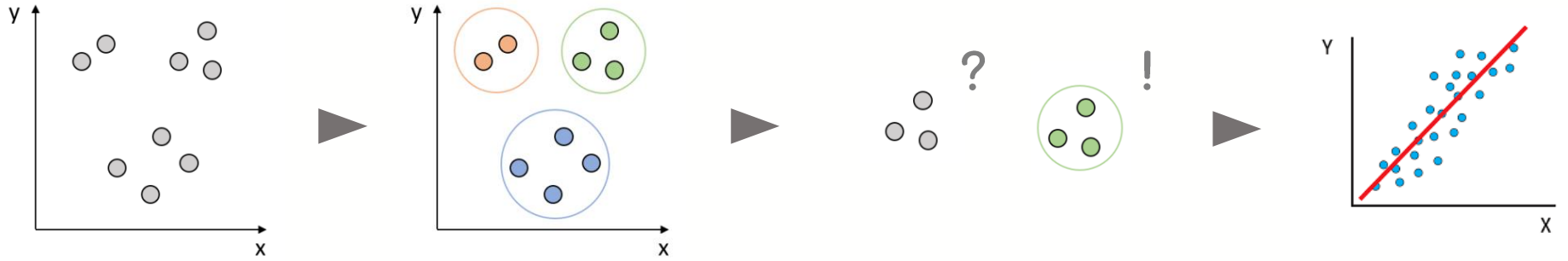

```
Accuracy: 0.67

array([[0.26618259, 0.36383844, 0.23064967, 0.1393293 ],
       [0.43677199, 0.31113253, 0.23660249, 0.01549299],
       [0.762417  , 0.09231643, 0.1070721 , 0.03819447],
       ...,
       [0.24811051, 0.69617256, 0.00520268, 0.05051425],
       [0.63704155, 0.35835819, 0.00143727, 0.00316299],
       [0.15188749, 0.30375961, 0.04706461, 0.4972883 ]])
```

Predicting which cluster each test data point belongs to using weather data, classification accuracy was low

어쩌면 당연함…

Modeling

Try 1. Clustering based on error rate

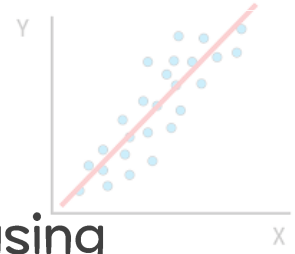


Step 4

Cluster-wise Linear Regression fitting

Modeling

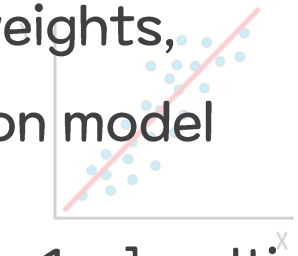
Try 1. Clustering based on error rate



Instead of classifying into specific cluster, using

probability of belongings to each clusters as weights,

due to the poor performance of the classification model



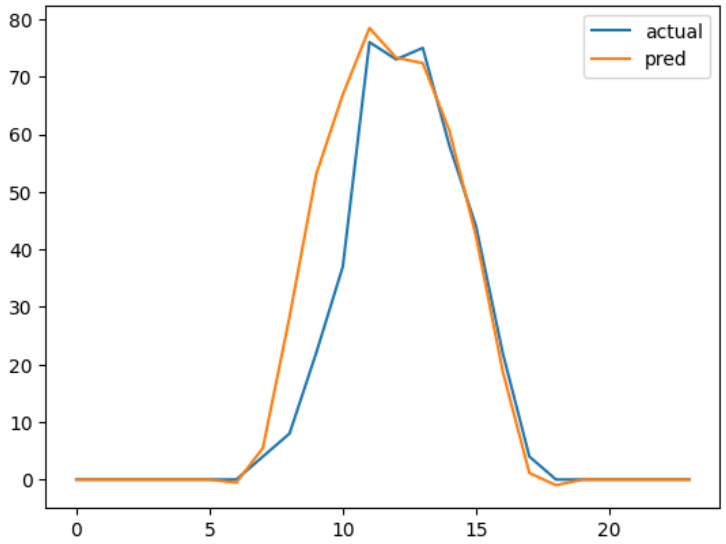
[Probability of being in Cluster 0, Probability of being in Cluster 1, ...] multiply by

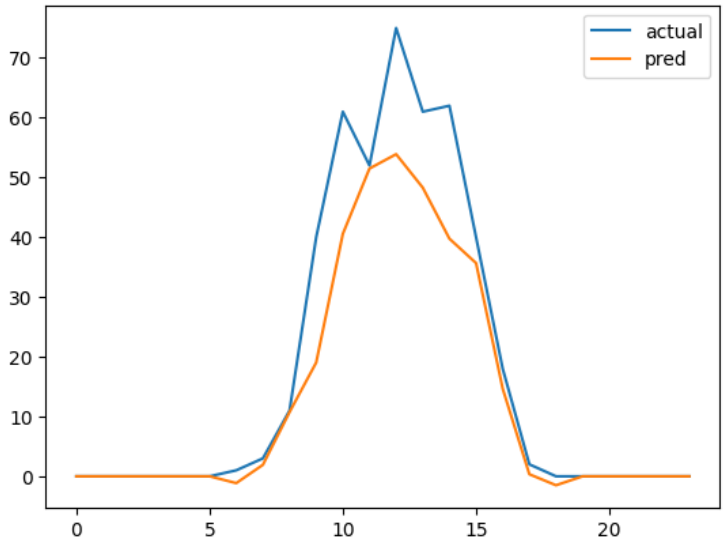
[Cluster 0 model prediction, Cluster 1 model prediction, ...]

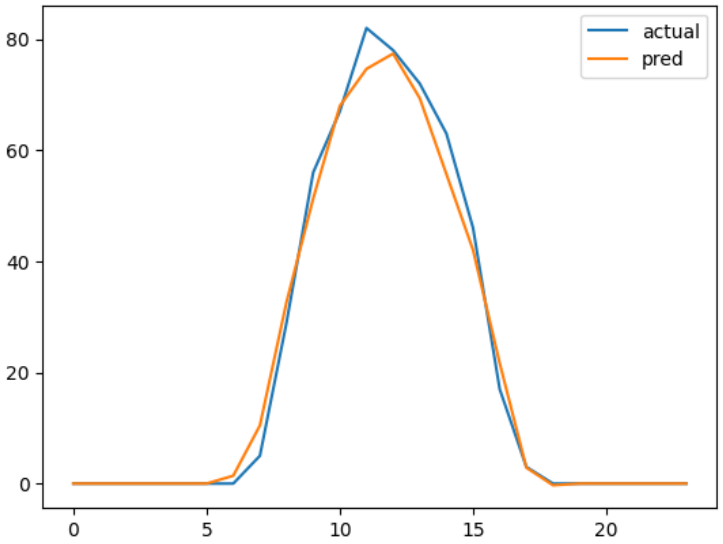
= Final prediction reflecting cluster weights

Modeling

Try 1. Clustering based on error rate



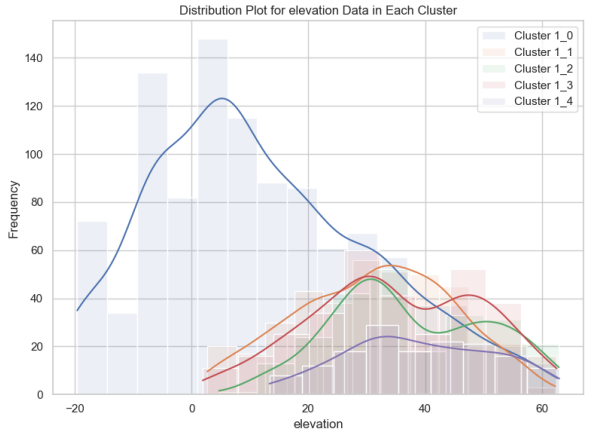


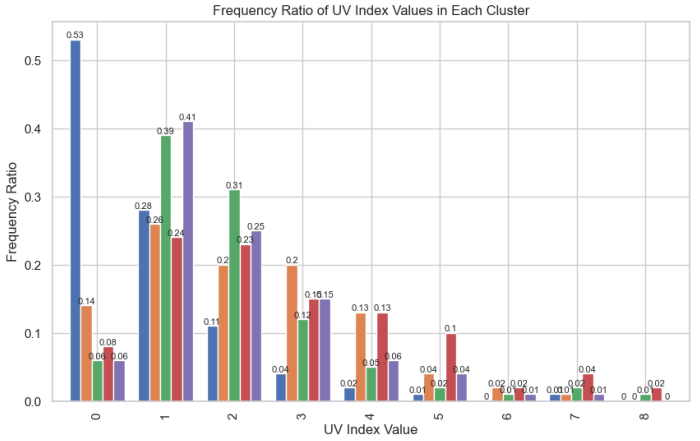


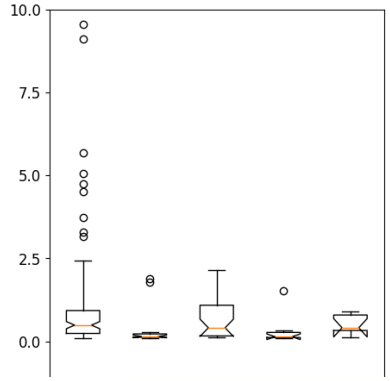
model	data	Total incentive(10.25~10.31)	notes
LGBM	error rate(M0~M4), weather forecast	8219	Classifier for test data:LGBM
Linear Regression	Predicted generation(M0~M4)		

Modeling

Try 2. Clustering based on weather







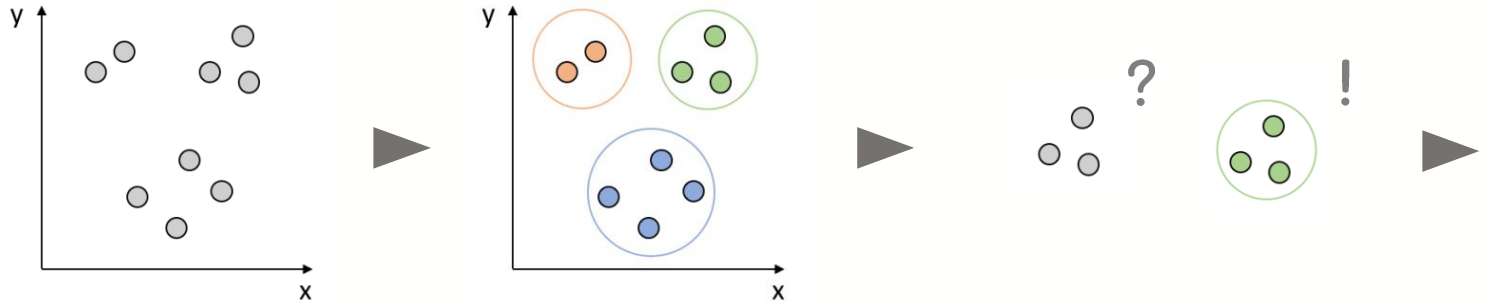
Even if clustering is based on errors, clusters for

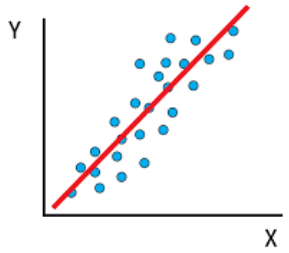
test data still need to be predicted using weather data.

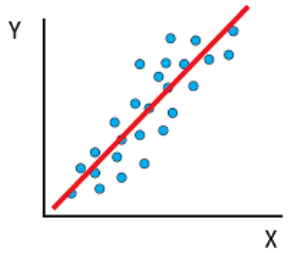
-> let's cluster by weather!

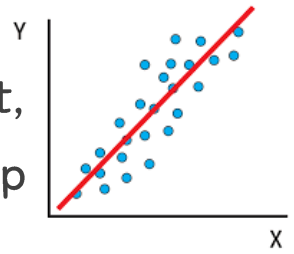
Modeling

Try 2. Clustering based on weather





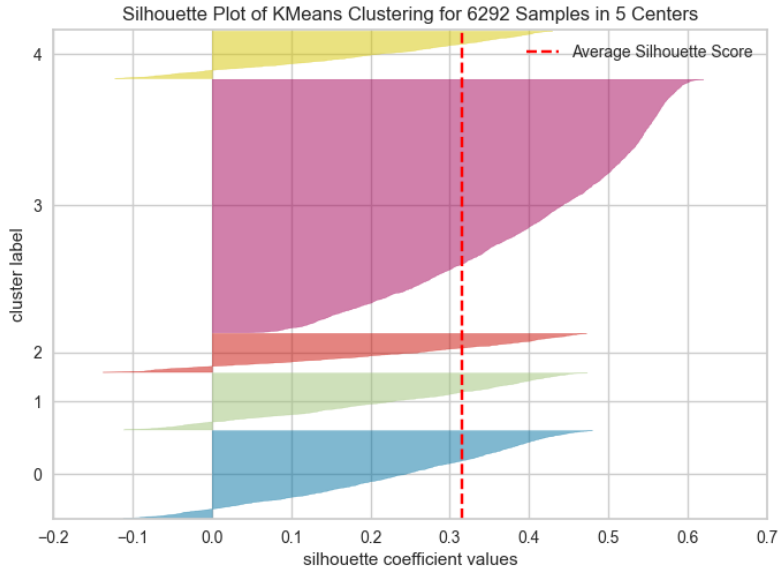


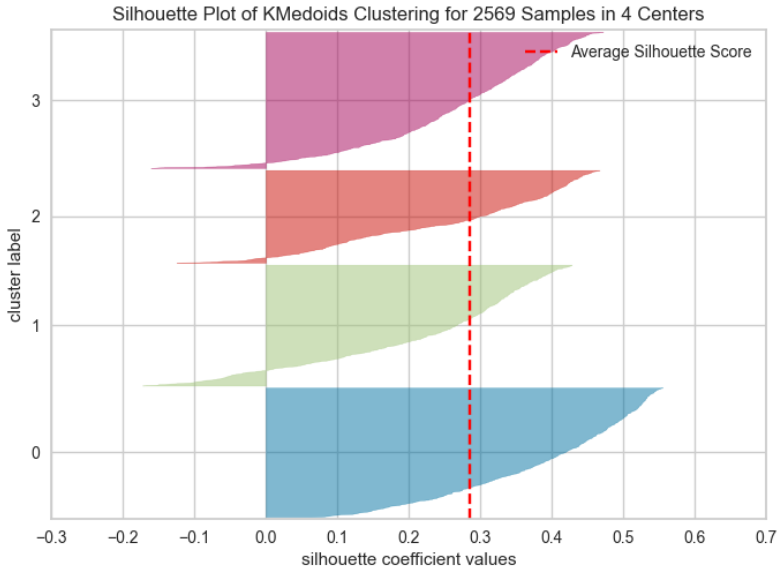


Following the same process as the previous attempt,
but using weather data for the initial clustering step

Modeling

Try 2. Clustering based on weather





Observed that it is much

improved compared to clustering based on error

Modeling

Try 2. Clustering based on weather

```
Accuracy: 0.97
```

```
array([[1.36516970e-05, 1.93583067e-06, 3.32037165e-05, 9.99951209e-01],
       [3.24360624e-06, 1.99441301e-05, 9.41990909e-07, 9.99975870e-01],
       [2.27986502e-07, 9.99998031e-01, 4.60774040e-07, 1.28015904e-06],
       ...,
       [1.09727939e-05, 2.14053386e-06, 2.38517709e-05, 9.99963035e-01],
       [9.99940237e-01, 1.58732508e-06, 5.44352759e-05, 3.74083918e-06],
       [1.75641292e-06, 9.99962672e-01, 3.83002573e-06, 3.17413815e-05]])
```

Classifier accuracy also improved

Modeling

Try 2. Clustering based on weather

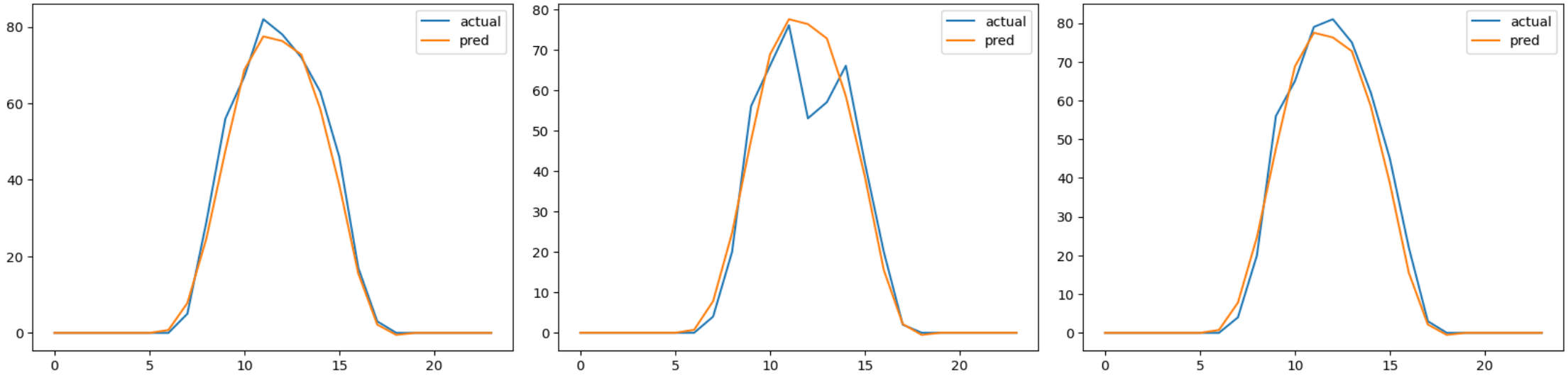
model	data	Total incentive(10.25~10.31)	notes
LGBM	error rate(M0~M4), weather forecast	8219	Classifier for test data:LGBM
Linear Regression	Predicted generation(M0~M4)		

▼

model	data	Total incentive(10.25~10.31)	notes
LGBM	Weather forecast	8810	Classifier for test data:LGBM
Linear Regression	Predicted generation(M0~M4)		

Modeling

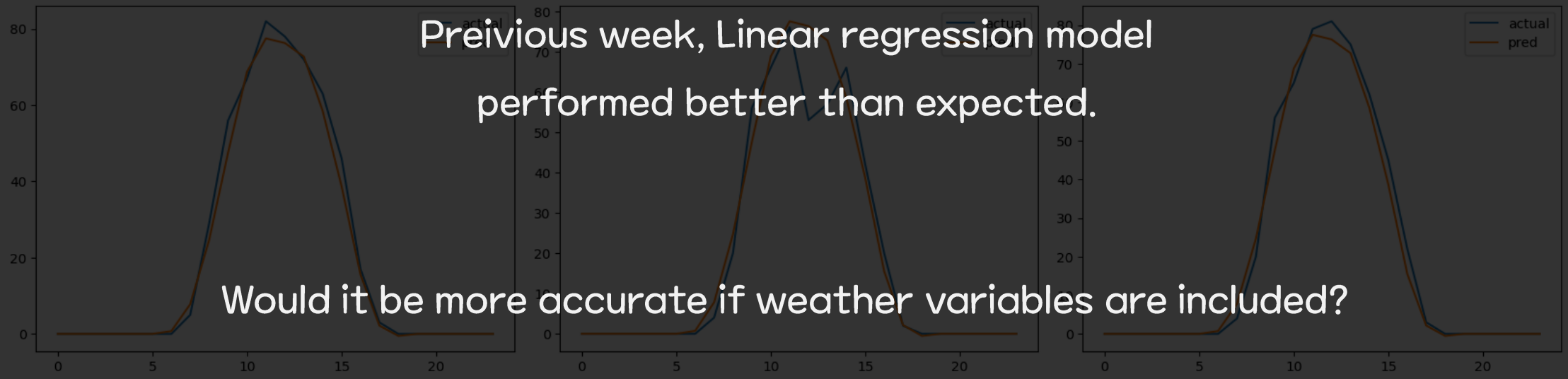
Linear regression – variable selection (divided by season)



model	data	Total incentive(10.25~10.31)	notes
Linear Regression	M0~M4	9020	Divided by seasons

Modeling

Linear regression – variable selection (divided by season)



Preivious week, Linear regression model performed better than expected.

Would it be more accurate if weather variables are included?

model	data	Total incentive(10.25~10.31)	notes
Linear Regression	M0~M4	9020	Divided by seasons

Modeling

Linear regression – variable selection (divided by season)

	round	time	model_id	amount
0	1	2022-06-19 01:00:00+09:00	0	0.0
1	1	2022-06-19 01:00:00+09:00	1	0.0
2	1	2022-06-19 01:00:00+09:00	2	0.0
3	1	2022-06-19 01:00:00+09:00	3	0.0
4	1	2022-06-19 01:00:00+09:00	4	0.0

Total of 5 variables from m0 to m4

weather_forecast

cloud	temp	humidity	ground_press	wind_speed	wind_dir
6.0	20.03	93.0	1009.0	3.01	162.0
7.0	19.88	95.0	1009.0	3.16	159.0

rain	snow	dew_point	vis	uv_idx	azimuth	elevation
0.0	0.0	18.3333	16.0934	0.0	6.70428	-31.5296
0.0	0.0	18.3333	16.0934	0.0	22.19640	-28.4404

13 variables excluding time and round

A total of 18 variables (m0~m4 and weather variables)

In linear regression, having many variables

can cause issues like multicollinearity and overfitting

Modeling

Linear regression – variable selection (divided by season)

```
y1_fall$amount ~ m0 + m1 + m2 + m3 + temp + humidity + wind_speed +
    wind_dir + snow + dew_point + vis + uv_idx + elevation
```

	Df	Sum of Sq	RSS	AIC
<none>			95895	7258.1
- m2	1	160.0	96055	7259.1
- vis	1	167.7	96063	7259.3
+ cloud	1	21.0	95874	7259.7
+ azimuth	1	17.9	95878	7259.7
+ rain	1	13.6	95882	7259.8
+ ground_press	1	0.0	95895	7260.1
+ m4	1	0.0	95895	7260.1
- elevation	1	223.9	96119	7260.3
- snow	1	261.9	96157	7261.0
- dew_point	1	327.3	96223	7262.3
- temp	1	349.8	96245	7262.7
- humidity	1	363.1	96259	7263.0
- m3	1	505.1	96401	7265.7
- wind_dir	1	894.7	96790	7273.0
- wind_speed	1	1379.3	97275	7282.1
- uv_idx	1	1491.0	97386	7284.2
- m1	1	3220.6	99116	7316.3
- m0	1	4381.2	100277	7337.6

```
y2_fall$amount ~ m0 + m1 + m3 + m4 + temp + humidity + wind_speed +
    wind_dir + dew_point + vis + uv_idx
```

	Df	Sum of Sq	RSS	AIC
<none>			97270	7208.5
- m4	1	124.5	97394	7208.8
+ m2	1	66.1	97204	7209.2
+ snow	1	43.6	97226	7209.6
+ cloud	1	28.9	97241	7209.9
+ elevation	1	25.2	97245	7210.0
+ rain	1	6.3	97264	7210.3
+ azimuth	1	0.6	97269	7210.4
+ ground_press	1	0.0	97270	7210.5
- m3	1	218.1	97488	7210.5
- vis	1	240.3	97510	7210.9
- humidity	1	258.4	97528	7211.2
- dew_point	1	311.9	97582	7212.2
- uv_idx	1	332.0	97602	7212.6
- temp	1	400.2	97670	7213.9
- wind_speed	1	730.9	98001	7219.9
- wind_dir	1	1136.7	98407	7227.4
- m1	1	1853.5	99123	7240.5
- m0	1	5751.5	103022	7309.9

Use step-wise method for variable selection

Modeling

Linear regression – variable selection (divided by season)

Round1

m0, m1, m2, m3

temp

humidity

wind_speed

wind_dir

snow

vis

dew_point

uv_idx

elevation

Round2

m0, m1, m3, m4

temp

humidity

wind_speed

wind_dir

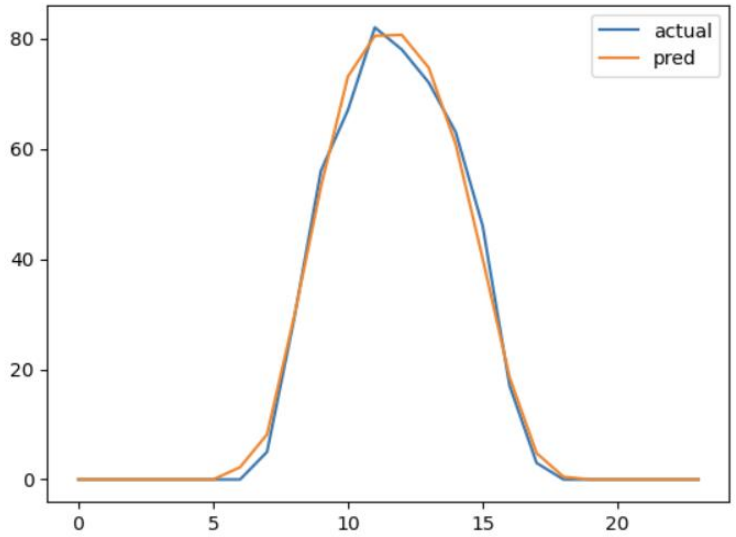
dew_point

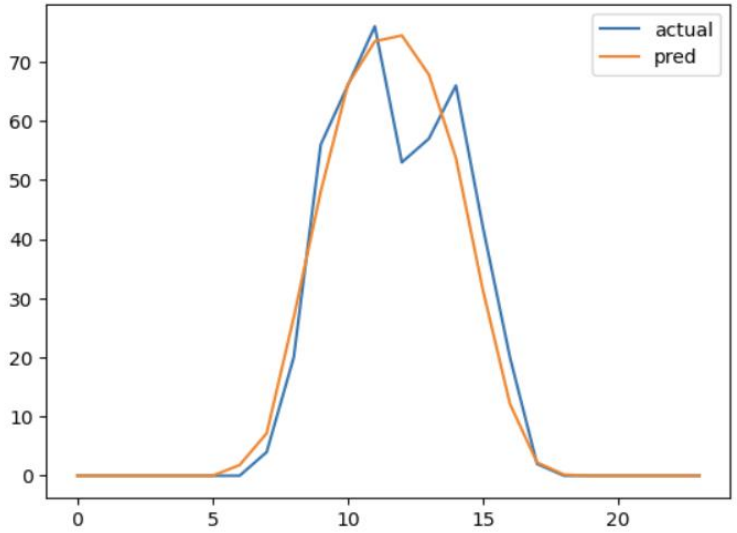
vis

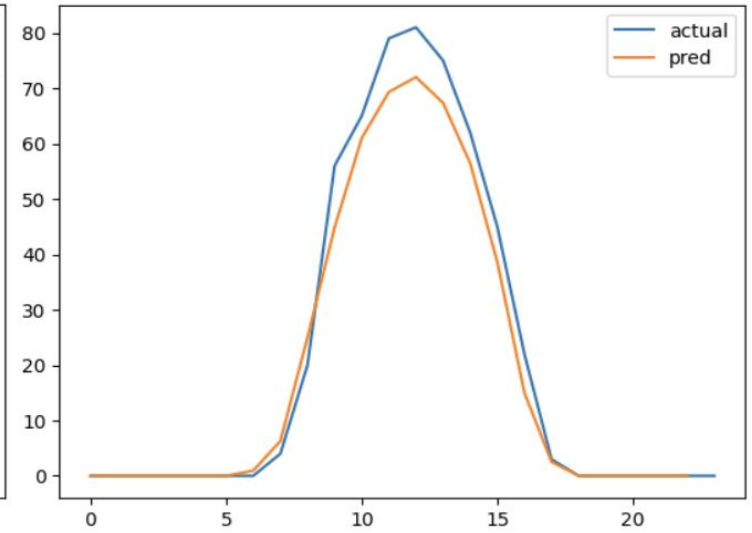
uv_idx

Modeling

Linear regression – variable selection (divided by season)



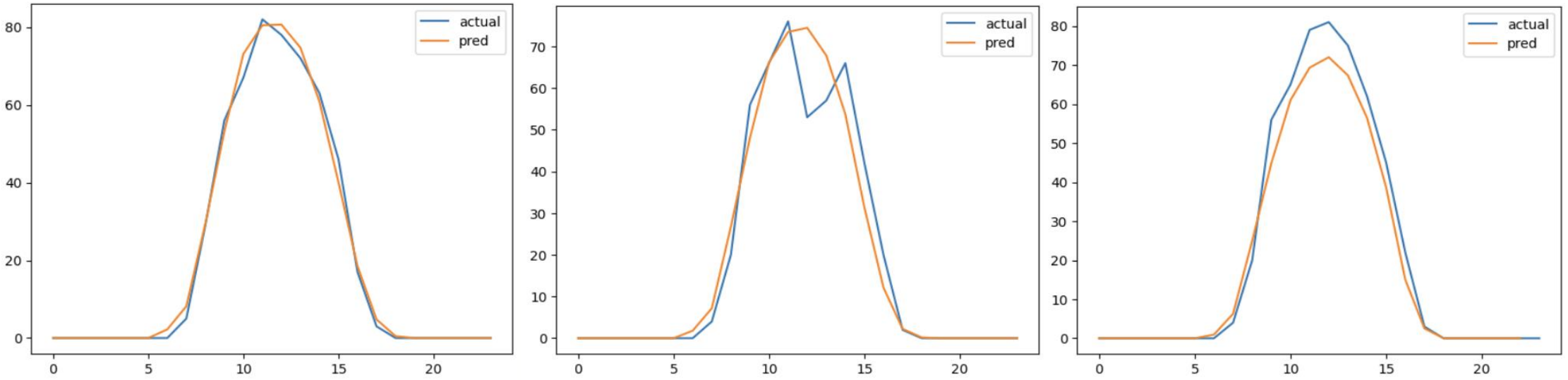




model	data	Total incentive(10.25~10.31)	notes
Linear Regression	“See previous slide”	7667	Divided by seasons

Modeling

Linear regression – variable selection (divided by season)



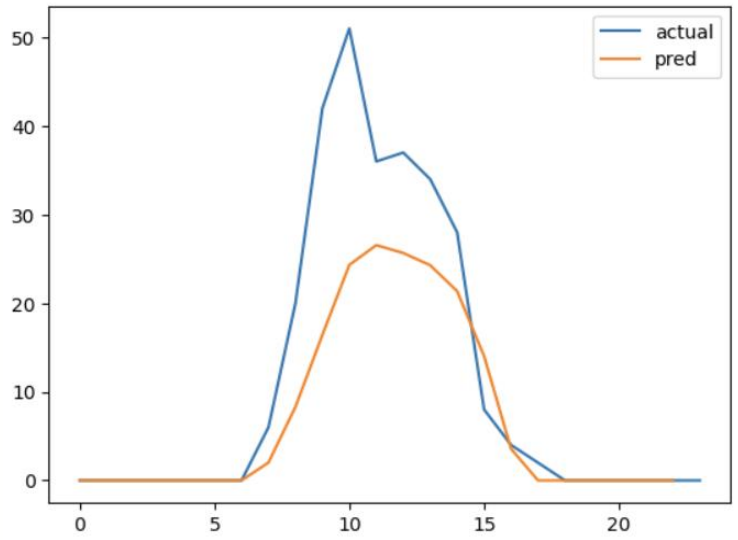
Bad predictions for Spike points

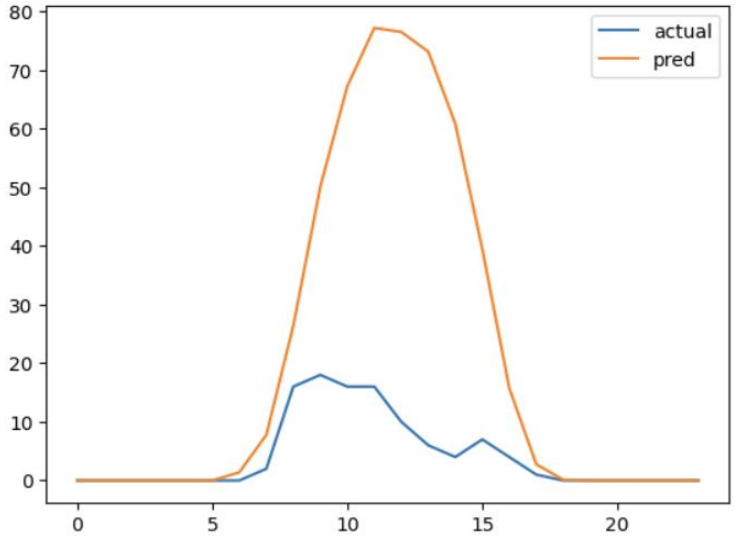
Generally predict well, but sometimes

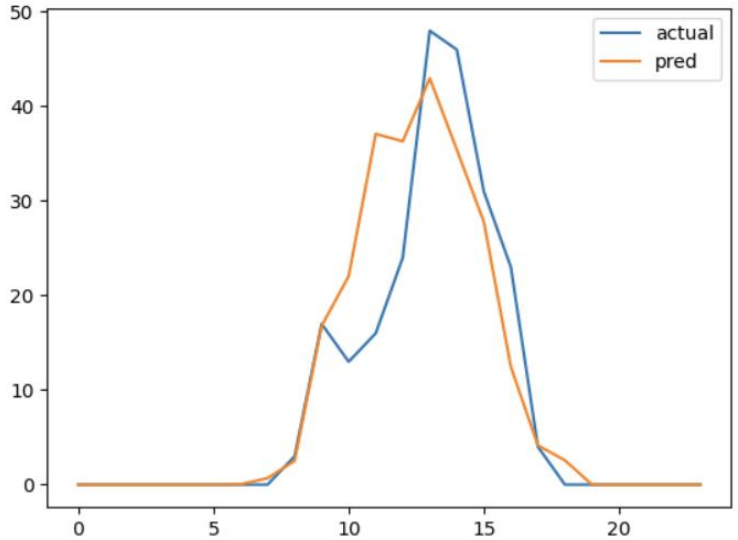
fails to predict the midday part

Modeling

Linear regression – variable selection (divided by season)



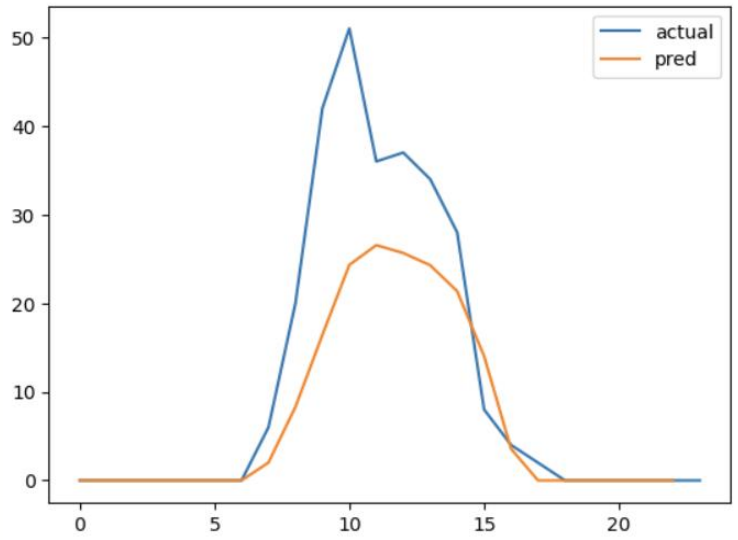


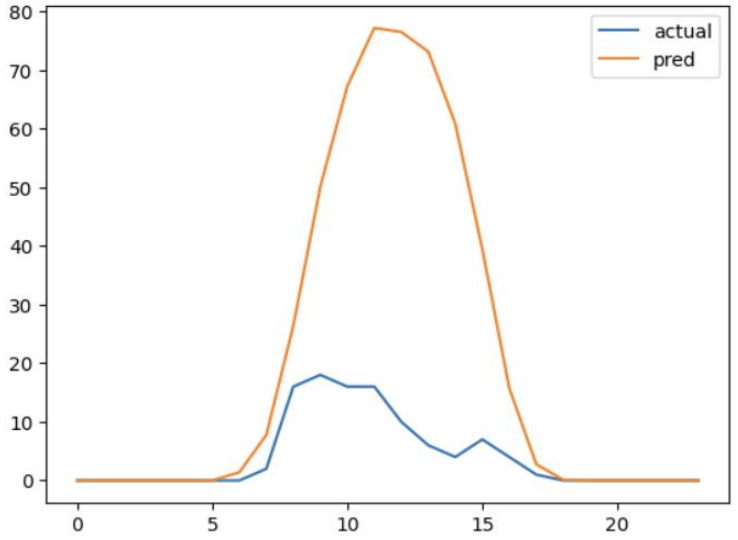


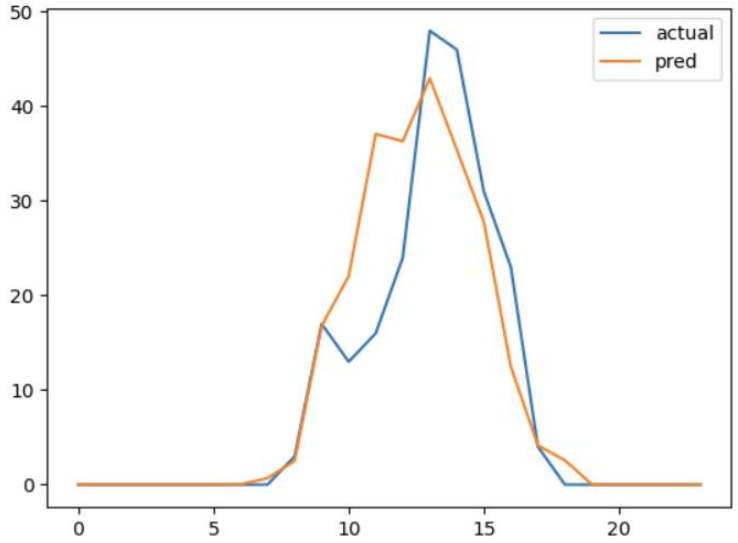
model	data	Total incentive(11.01~11.07)	notes
Linear Regression	“See previous slide”	4374	Divided by seasons

Modeling

Linear regression – variable selection (divided by season)







Overall, it shows bad performance

Modeling

MLP + LGBM – Only fall data

	Every seasonal data	Only fall data
LR	6956	9020
LR + LGBM	7139	7221
MLP + LGBM	9156	다음주 공개

There are some case where the model performance improved with fall data

Let’s also apply this to MLP+LGBM, which currently shows the best performance

Modeling

MLP + LGBM – Only fall data

	Every seasonal data	Only fall data
LR	6956	9020
LR + LGBM	7139	7221
MLP + LGBM	9156	8491

Totall incentive slightly decreased when using only fall data

This might because deep models typically

perform better with larger datasets

Modeling

MLP + LGBM – Only fall data

Let's increase the data size

For the complex models,
typically the more data the
better the performance

We will add more data for
model training

Modeling

MLP + LGBM – with more data

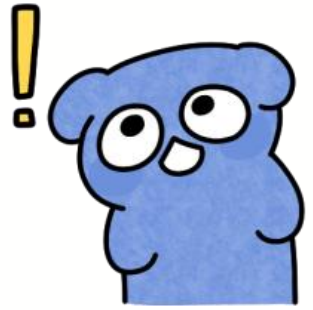
Train model with additional data from 2023/10/20~11/12

Test on data from 2023-11-13

Time incentive was earned

Train data set	8a.m.	1 p.m.	4 p.m.	5 p.m.	6 p.m.	total
Original	20	0	168	0	4	192
Additional	0	224	0	100	0	324

Slightly increased



Modeling

MLP + LGBM – with more data

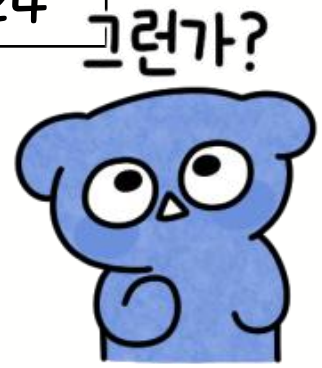
Train model with additional data from 2023/10/20~11/12

Test on data from 2023-11-13

Time incentive was earned

Train data set	8a.m.	1 p.m.	4 p.m.	5 p.m.	6 p.m.	total
Original	20	0	168	0	4	192
Additional	0	224	0	100	0	324

Slightly increased



Will adding the actual power generation from the same time of the previous day improve performance?

Modeling

MLP + LGBM – with more data

Train model with additional data from 2023/10/20~11/12

Test on data from 2023-11-13

Time incentive was earned

Train data set	8a.m.	1 p.m.	4 p.m.	5 p.m.	6 p.m.	total
Original	20	0	168	0	4	192
Additional	0	224	0	100	0	324
+yesterday gen	20	224	0	0	0	244

It was not…

Modeling

SCINet

Sample Convolution and Interaction Network

Time series prediction model based on CNN

State-of-the-Art in Time Series Prediction

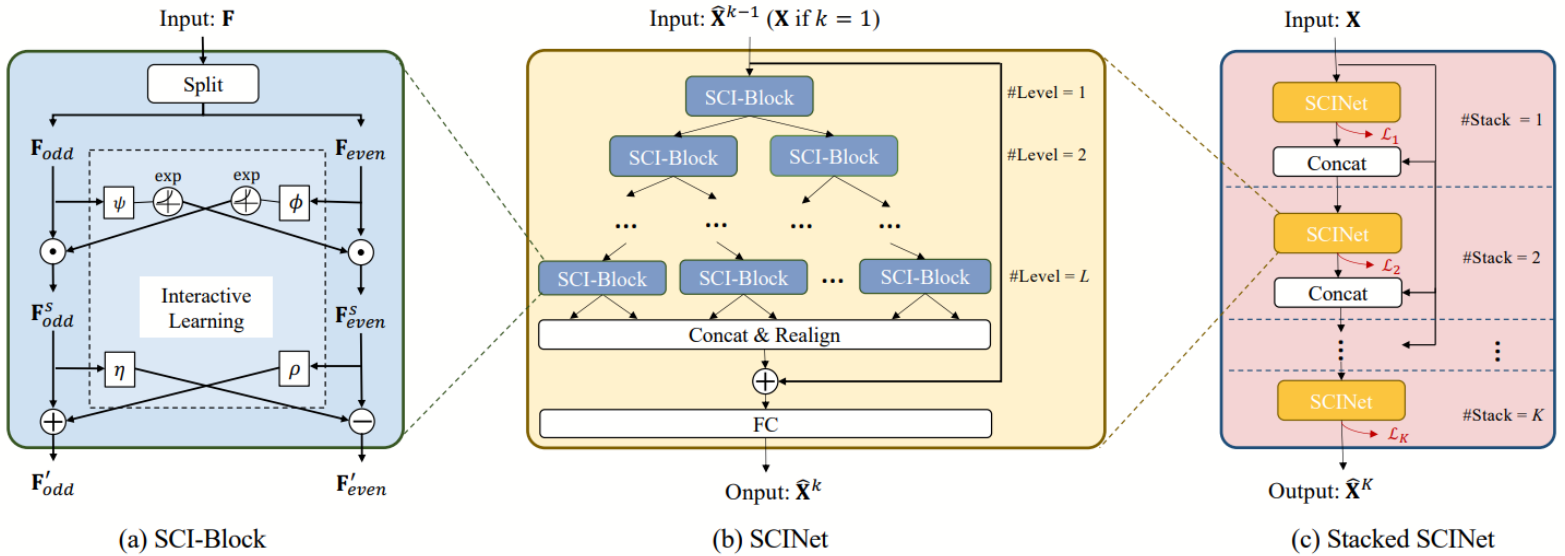


Figure 2: The overall architecture of Sample Convolution and Interaction Network (SCINet).

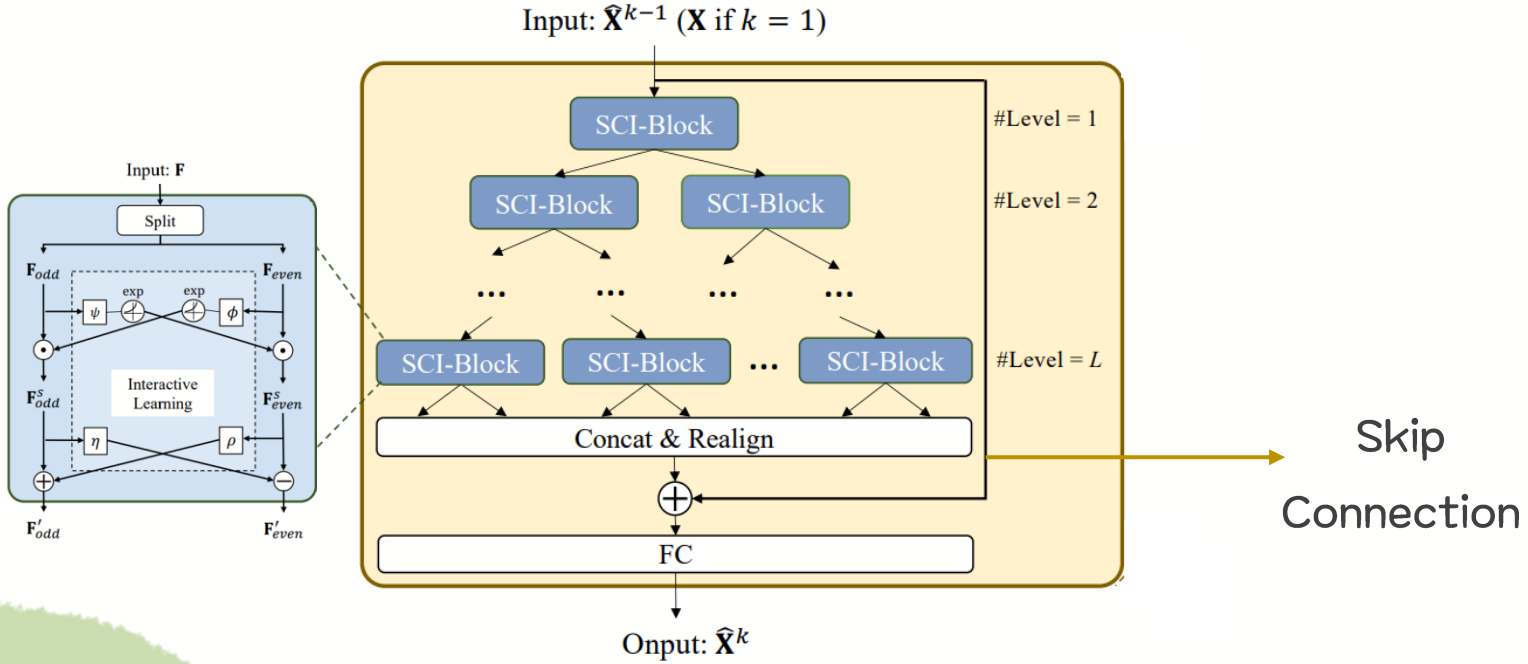
Modeling

SCINet

Role as encoder

SCI-Net

Separate the output of SCI-Block repeatedly, then finally rearrange and predict time-series output through FC Layer



Skip Connection

Skip Connection

Idea proposed by ResNet

Preserve the input information by adding the input vector to the output features

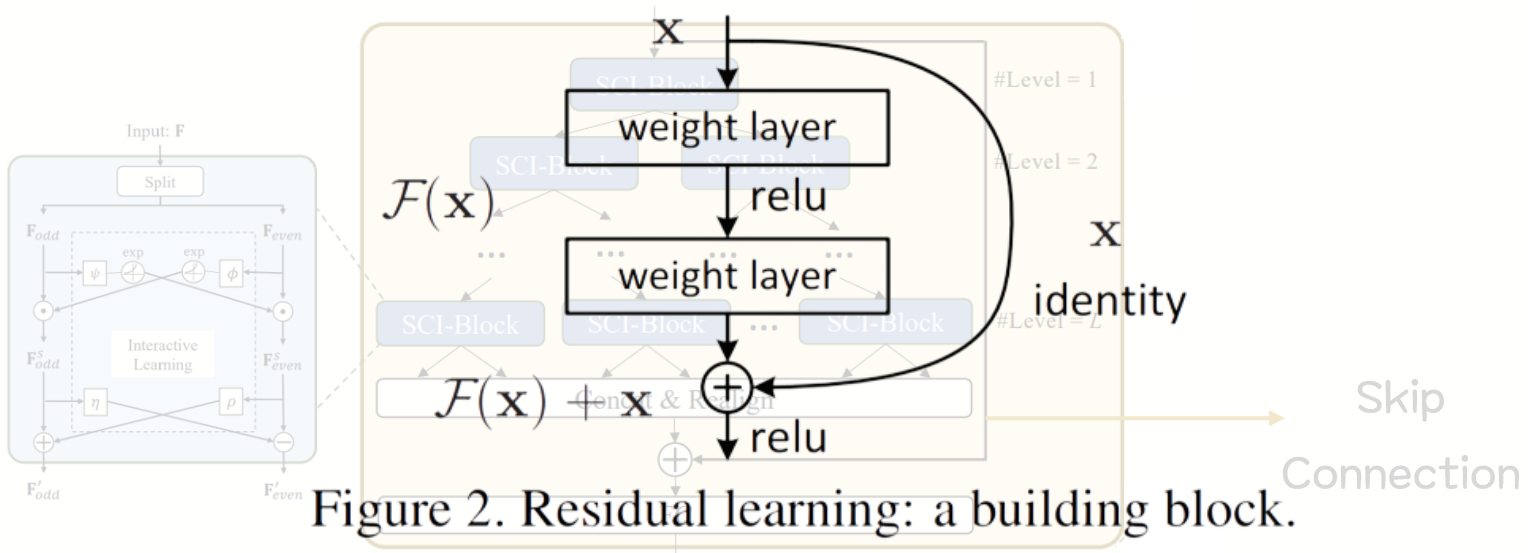


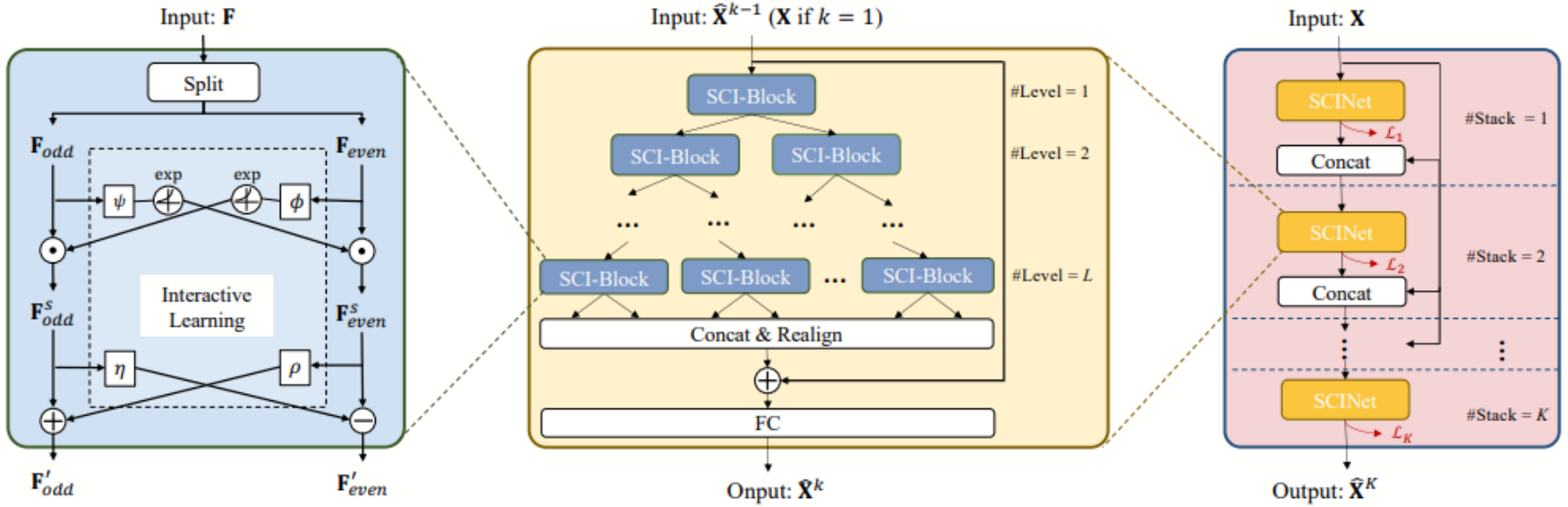
Figure 2. Residual learning: a building block.

Modeling

SCINet

Stacked SCINet

SCINet stacked in multiple layers with repeated application of skip connections during the stacking process



Modeling

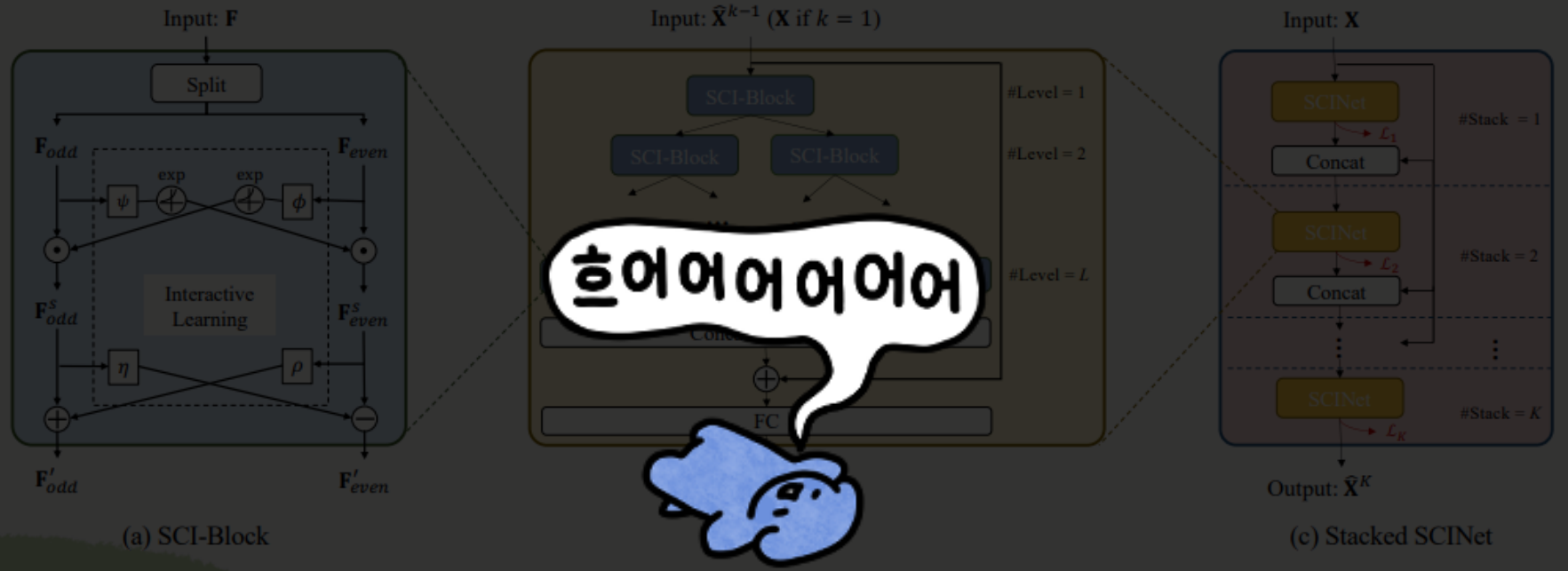
SCINet

Fail

Stacked SCINet

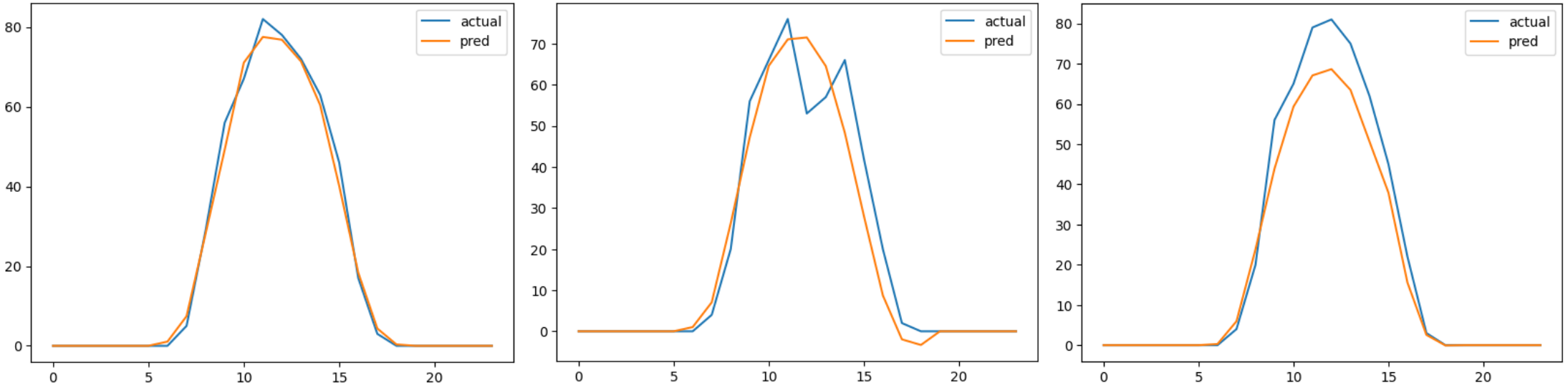
SCINet stacked in multiple layers with repeated

Predict similar values for all time point



Modeling

Time-series clustering



If we can separate these patterns and train the model
on them individually, wouldn't that improve performance?

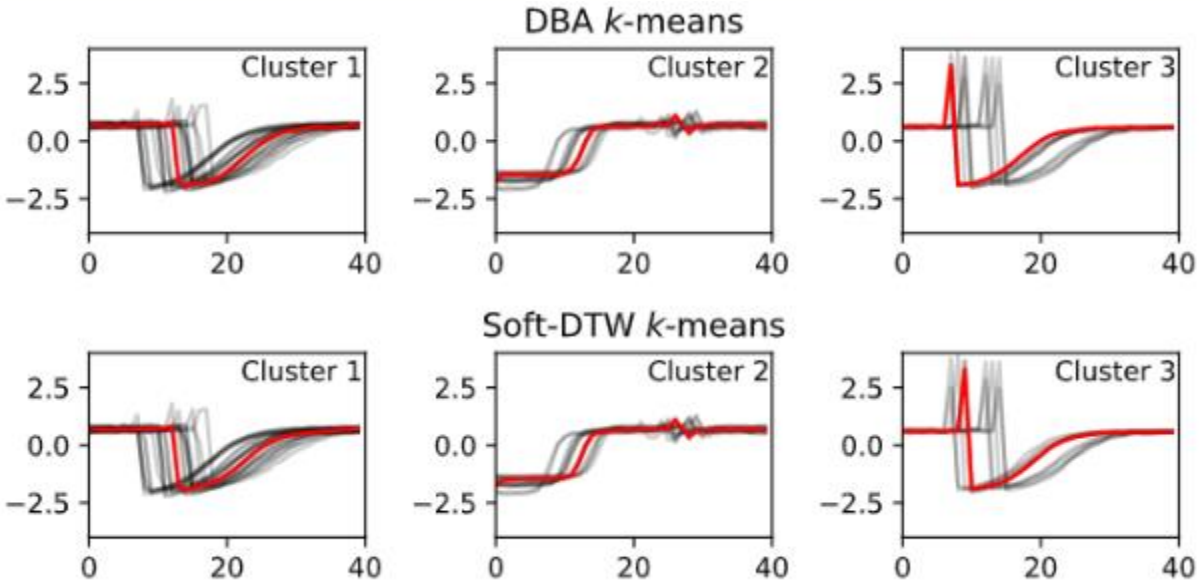
Modeling

Time-series clustering

Time-series clustering

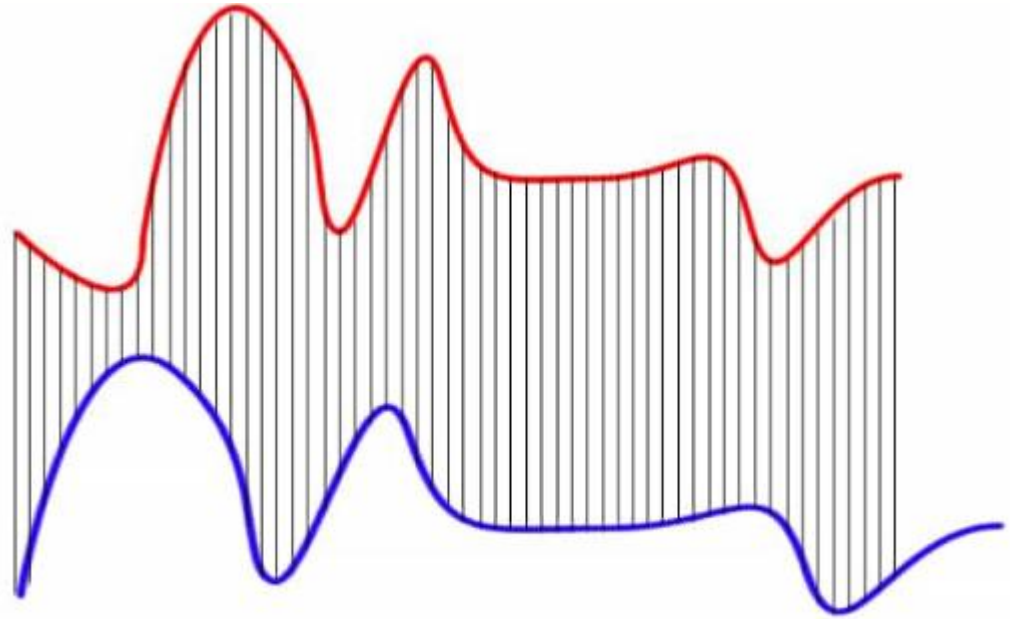
A method for grouping data with temporal characteristics

Adding time-series properties to conventional clustering

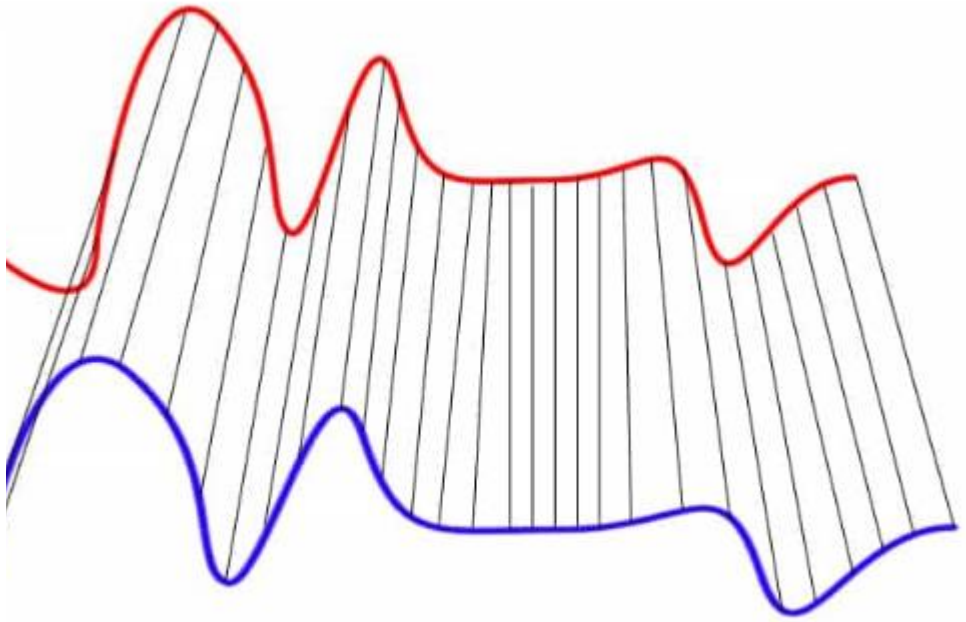


Modeling

Methods of time-series clustering



Euclidean Matching

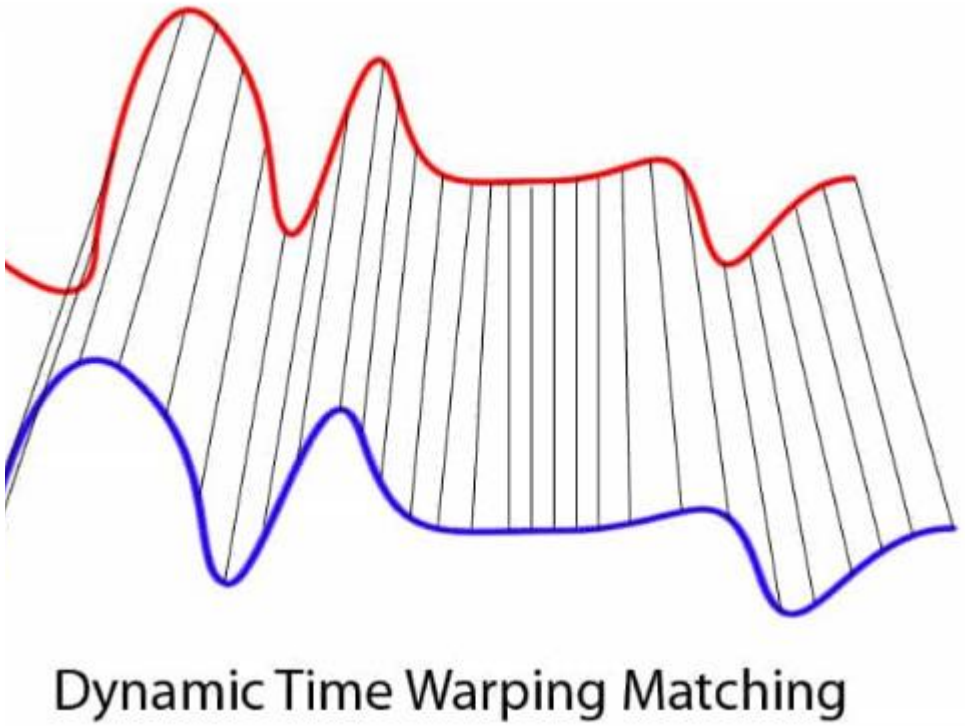


Dynamic Time Warping Matching

Modeling

Methods of time-series clustering

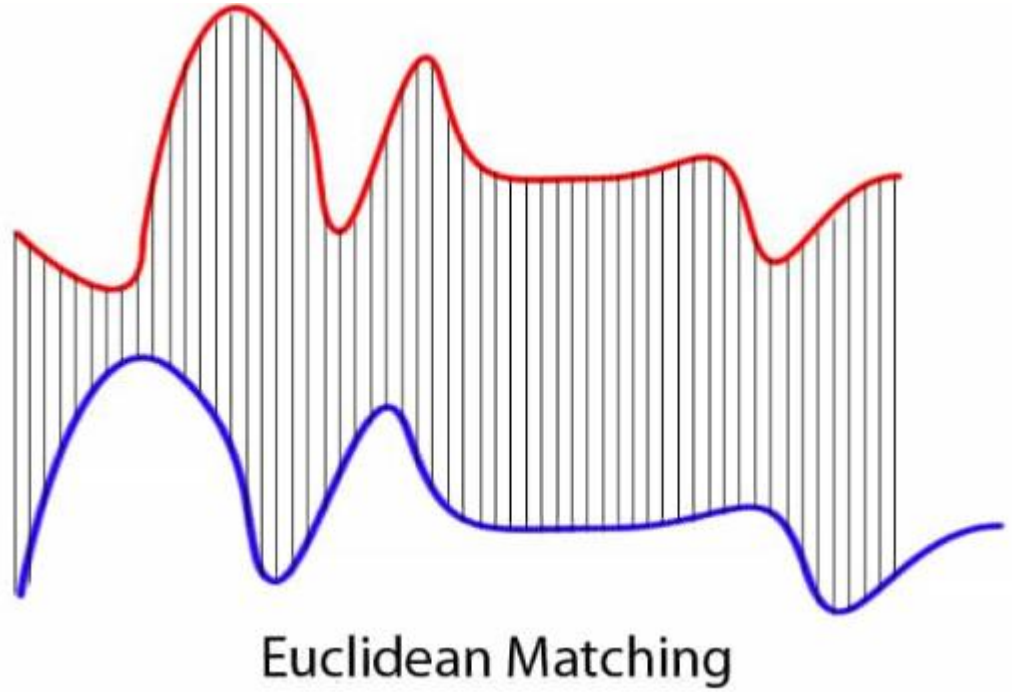
A method to measure the similarity between two time sequences that do not exactly match in time, speed, or length



Dynamic Time Warping Matching

Modeling

Methods of time-series clustering



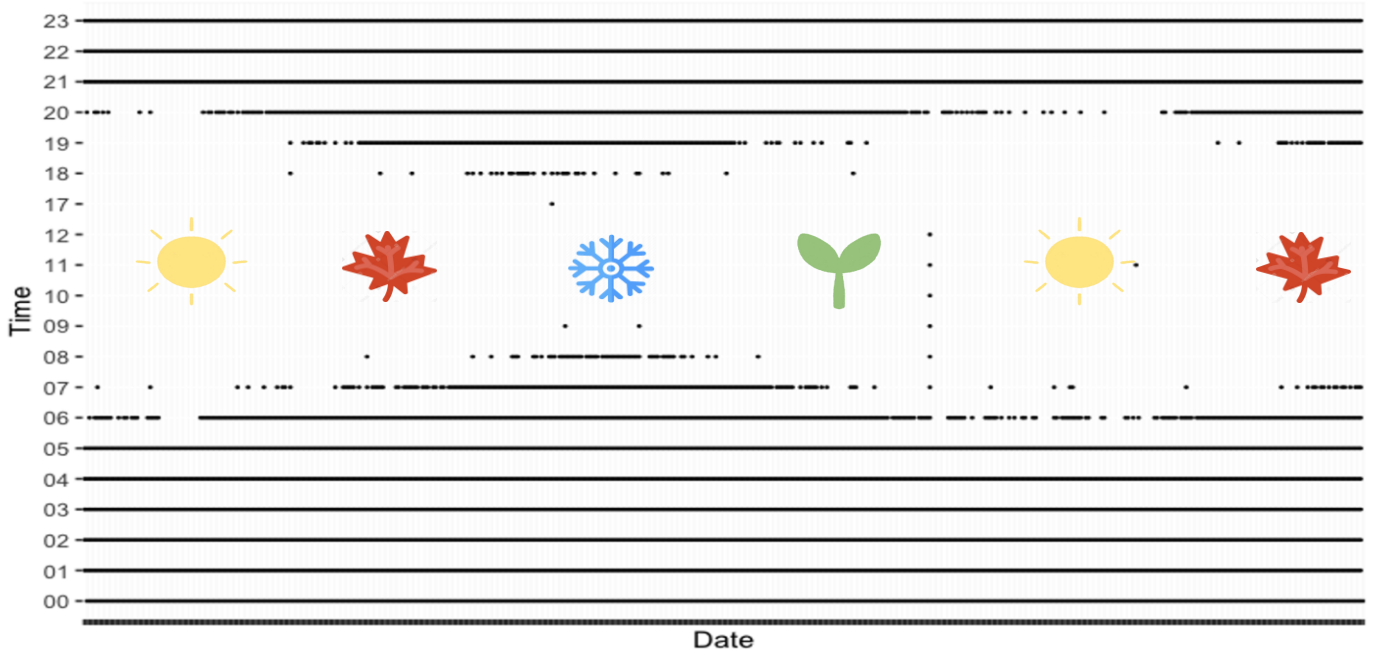
Euclidean Matching

Method for creating clusters based on Euclidean distance, Commonly used when time ranges are same

Dynamic Time Warping Matching

Modeling

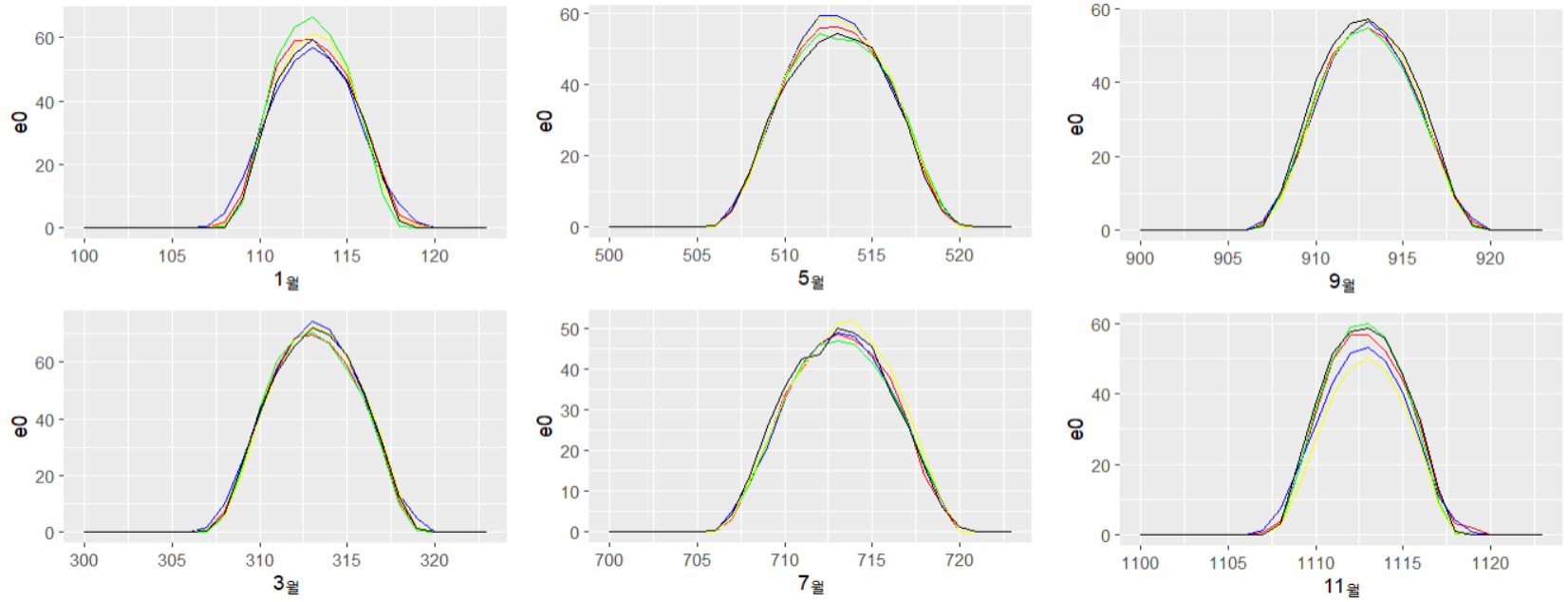
Methods of time-series clustering



In the previous week, there was a difference in time period with zero power generation depending on the season

Modeling

Methods of time-series clustering



There is a difference in the start and end times

of power generation depending on the season

Decided to consider both DTW and Euclidean method

Modeling

time-series clustering evaluation index

tsclust in R

Sil : Silhouette index (Rousseeuw (1987); to be maximized)

D : Dunn index (Arbelaitz et al. (2013); to be maximized)

COP : COP index (Arbelaitz et al. (2013); to be minimized)

DB : Davies-Bouldin index (Arbelaitz et al. (2013); to be minimized)

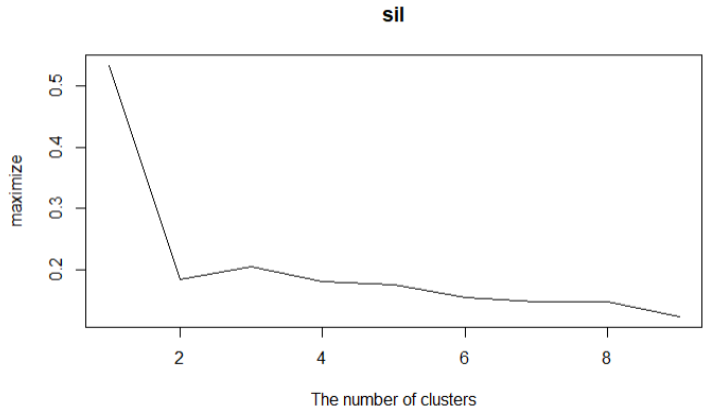
Dbstar : Modified Davies-Bouldin index (DB*) (Kim et al.(2005); to be minimized)

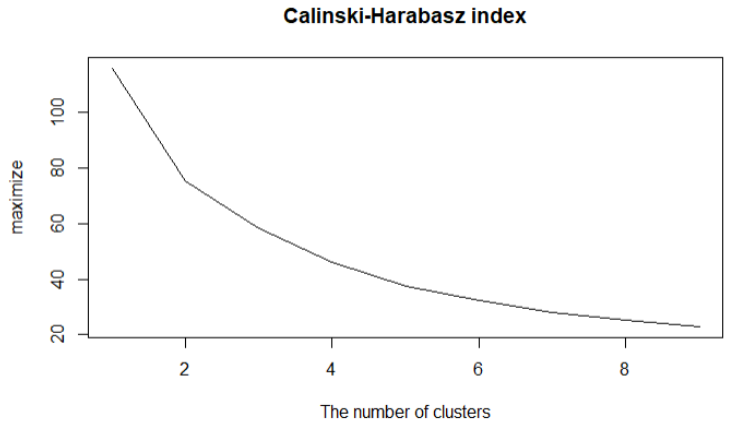
CH : Calinski-Harabasz index (Arbelaitz et al. (2013); to be maximized)

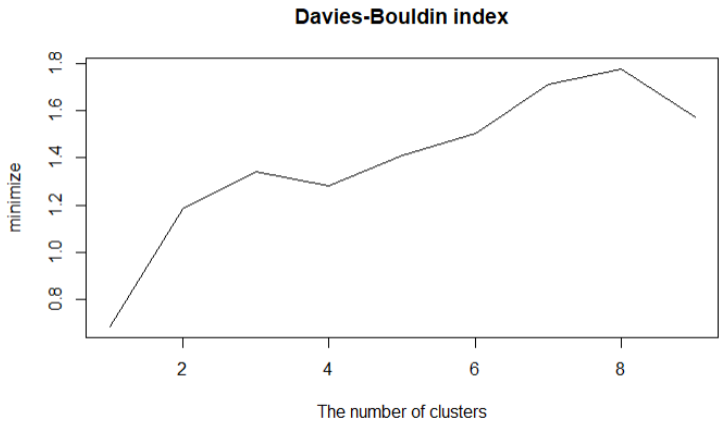
SF : Score Function (Saitta et al. (2007); to be maximized; see notes)

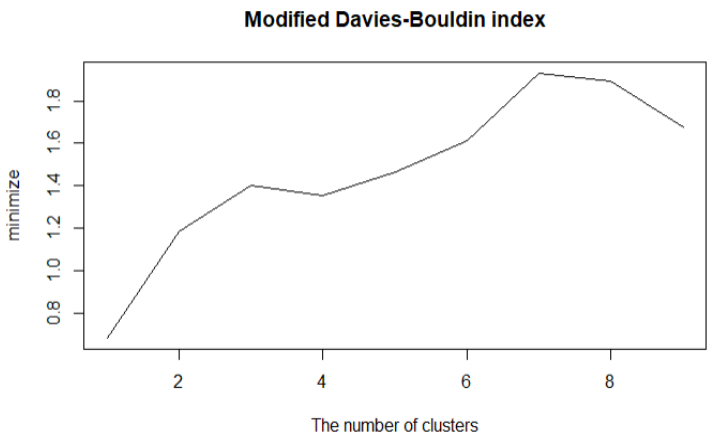
Modeling

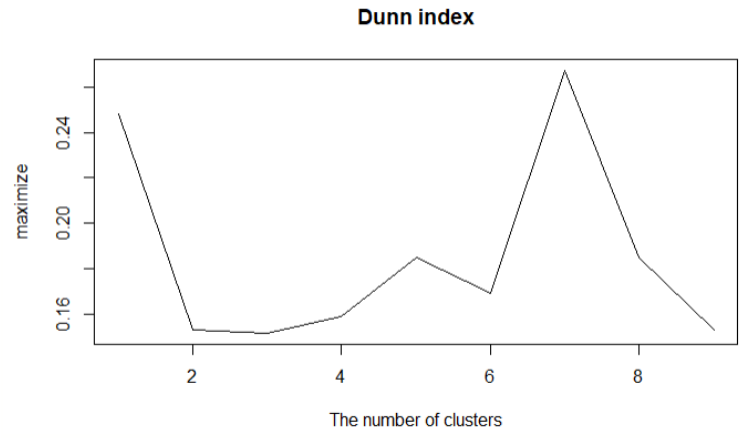
time-series clustering evaluation index (DTW)

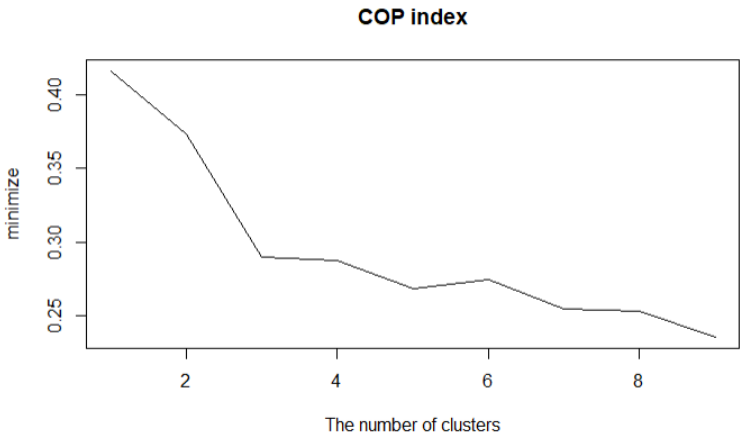






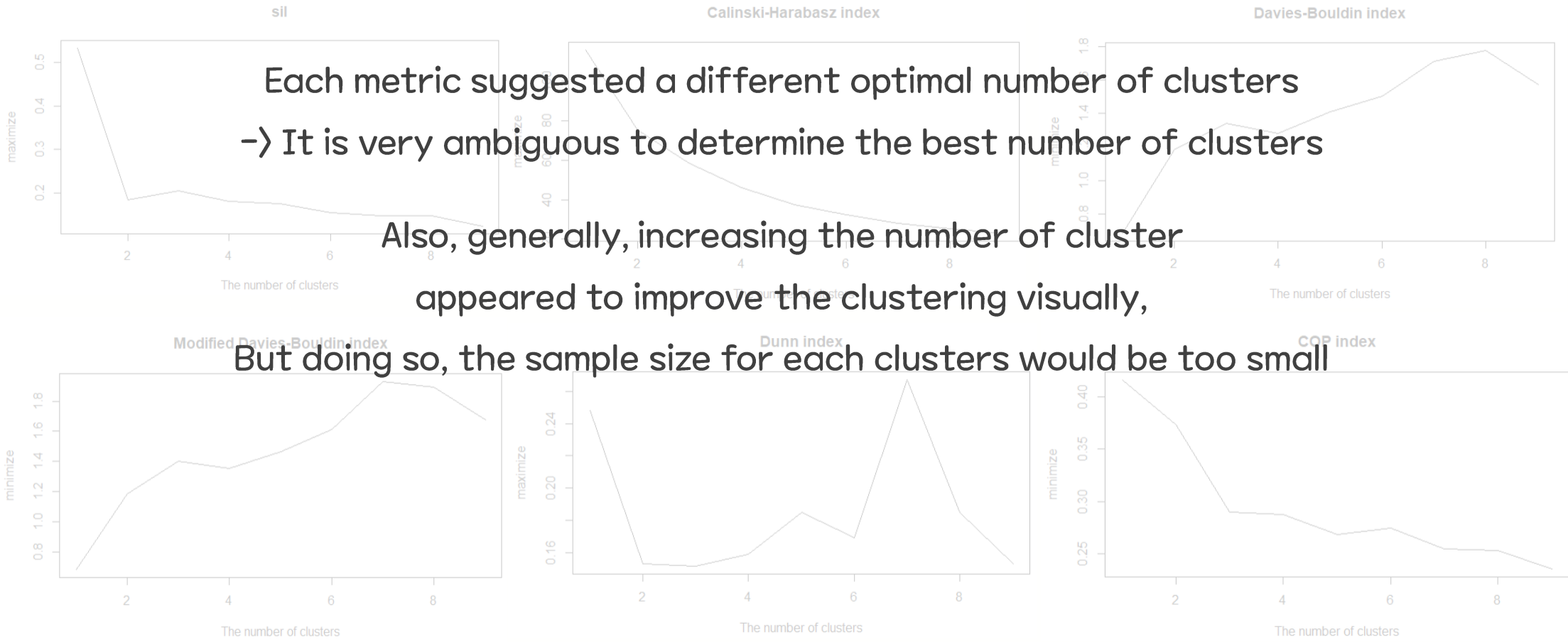






Modeling

time-series clustering evaluation index (DTW)



Each metric suggested a different optimal number of clusters

-> It is very ambiguous to determine the best number of clusters

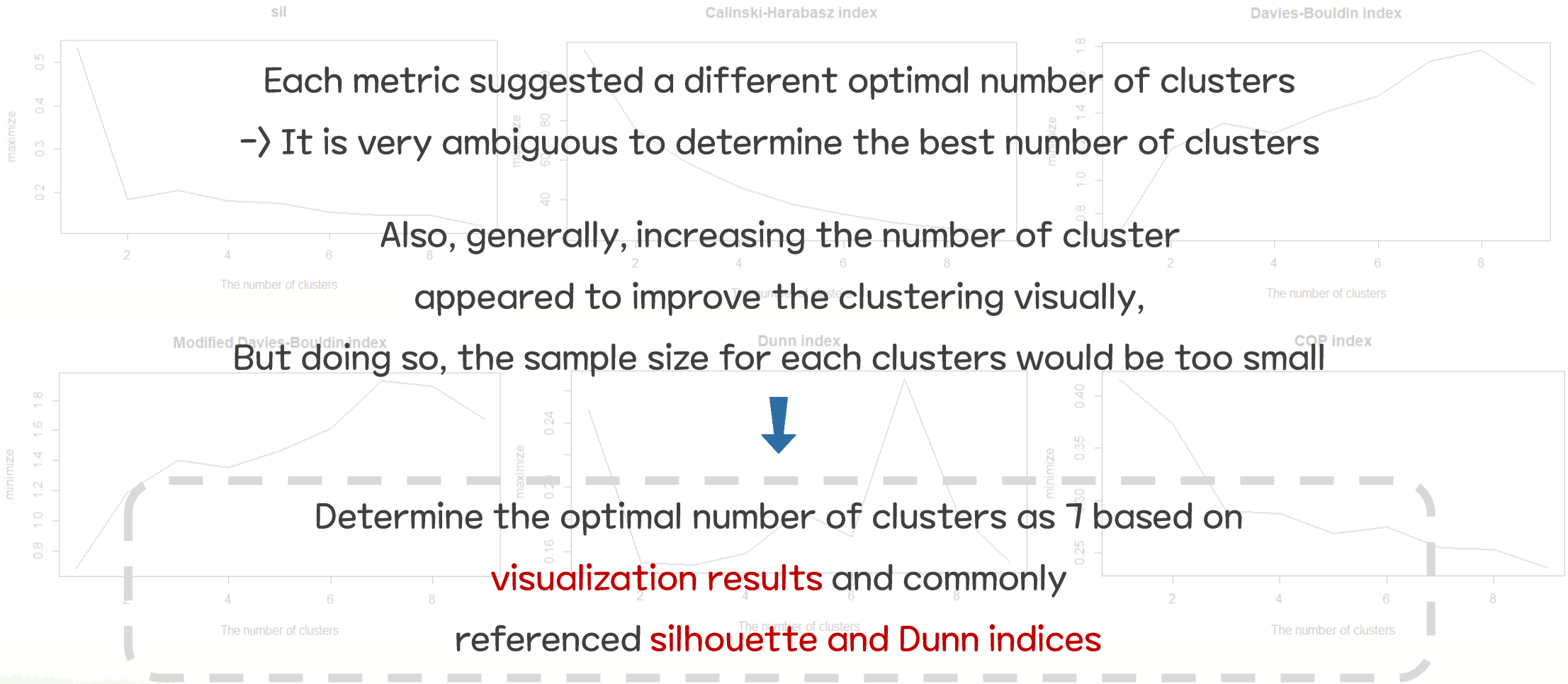
Also, generally, increasing the number of cluster

appeared to improve the clustering visually,

But doing so, the sample size for each clusters would be too small

Modeling

time-series clustering evaluation index (DTW)



Each metric suggested a different optimal number of clusters

-> It is very ambiguous to determine the best number of clusters

Also, generally, increasing the number of cluster

appeared to improve the clustering visually,

But doing so, the sample size for each clusters would be too small

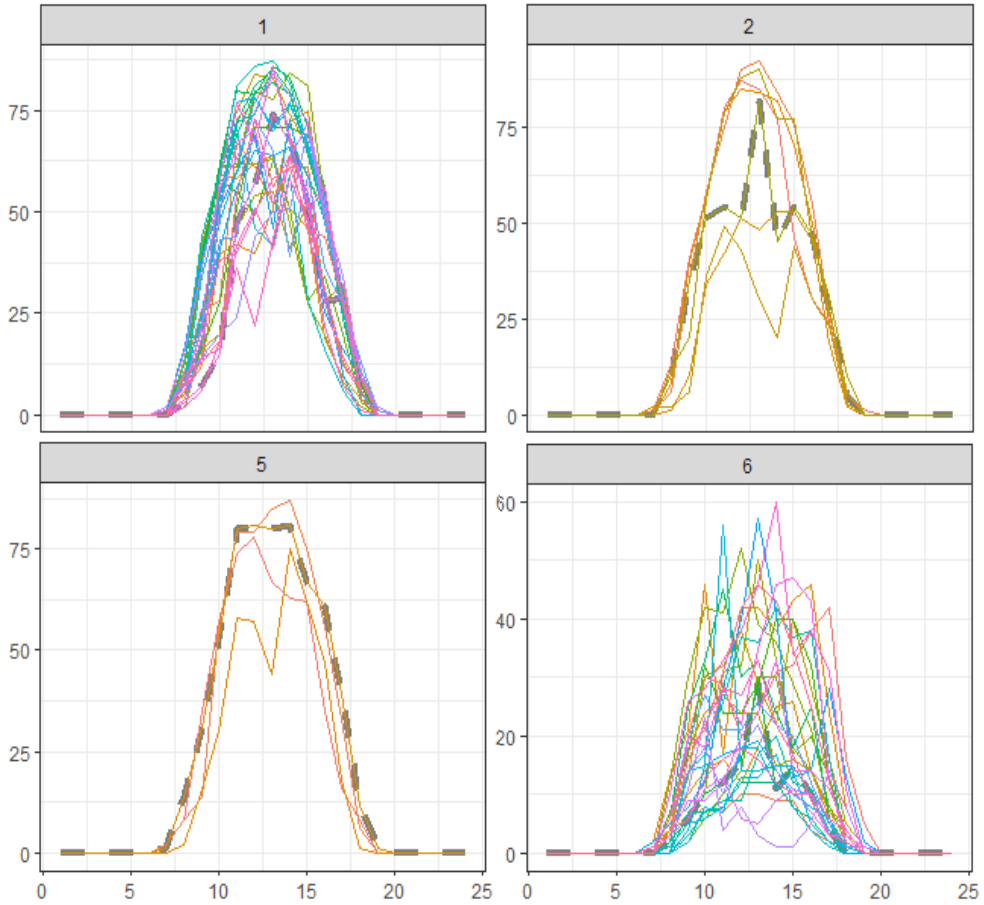
Determine the optimal number of clusters as 7 based on

visualization results and commonly

referenced silhouette and Dunn indices

Modeling

time-series clustering result



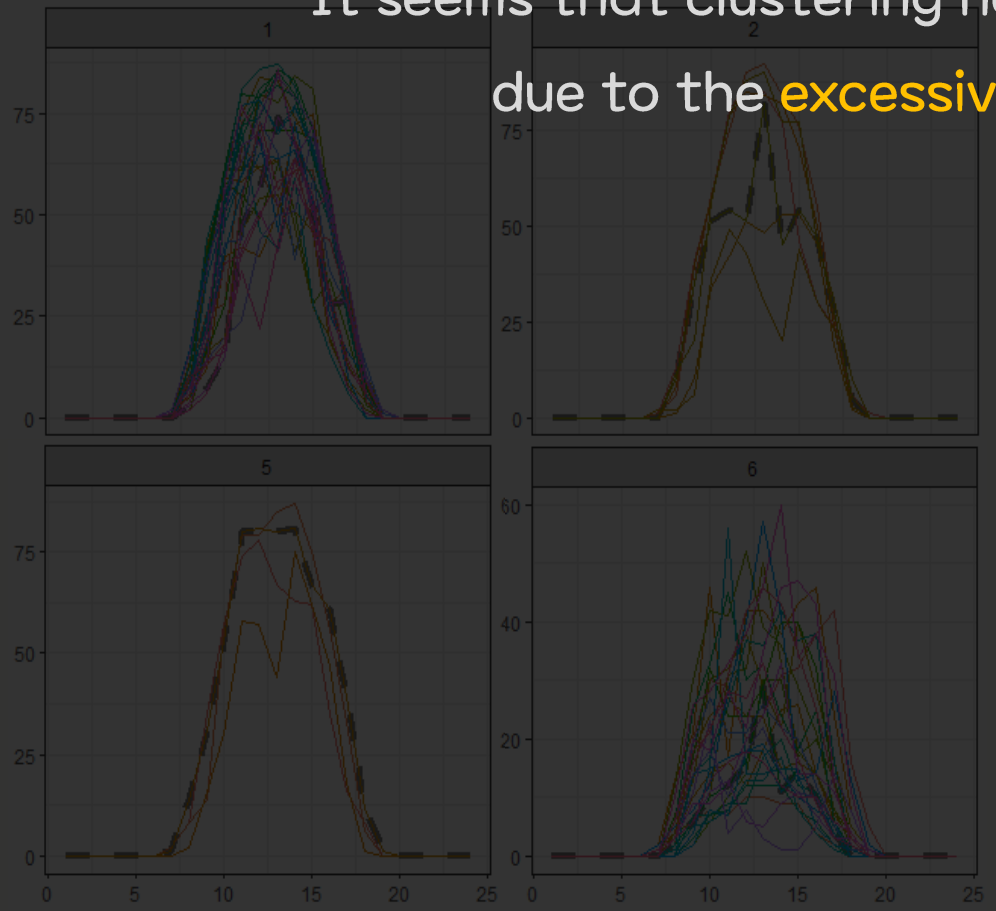
Clusters like 2 and 5 have very few samples

For clusters 1 and 6, the patterns are not distinct enough to be considered separated clusters

Changing the number of clusters, whether increasing or decreasing, leads to issues.

Modeling

time-series clustering result



It seems that clustering has not been properly achieved

due to the excessive diversity of patterns

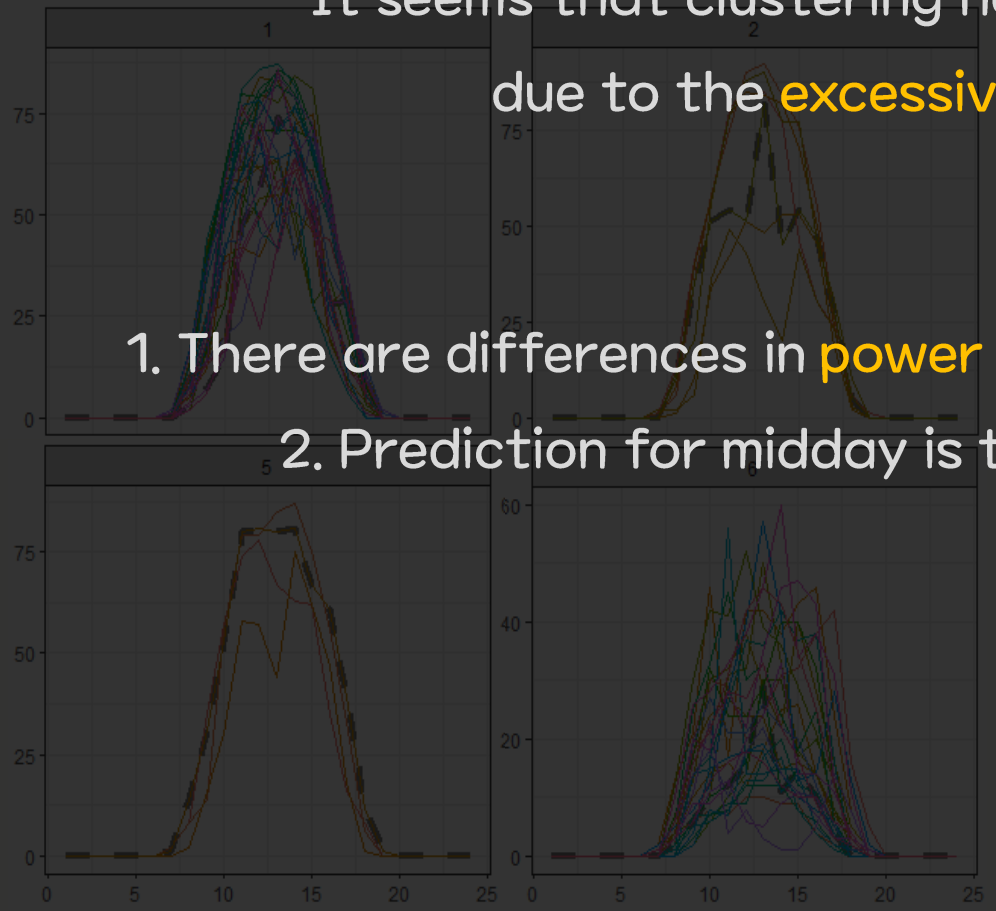
Clusters like 2 and 5 have very few samples

For clusters 1 and 6, the patterns are not distinct enough to be considered separated clusters

Changing the number of clusters, whether increasing or decreasing, leads to issues.

Modeling

time-series clustering result



It seems that clustering has not been properly achieved

due to the excessive diversity of patterns

1. There are differences in power generation patterns across weather clusters
2. Prediction for midday is the most crucial (high incentive)

Clusters like 2 and 5 have very few samples

For clusters 1 and 6, the patterns are not distinct enough to be considered separated

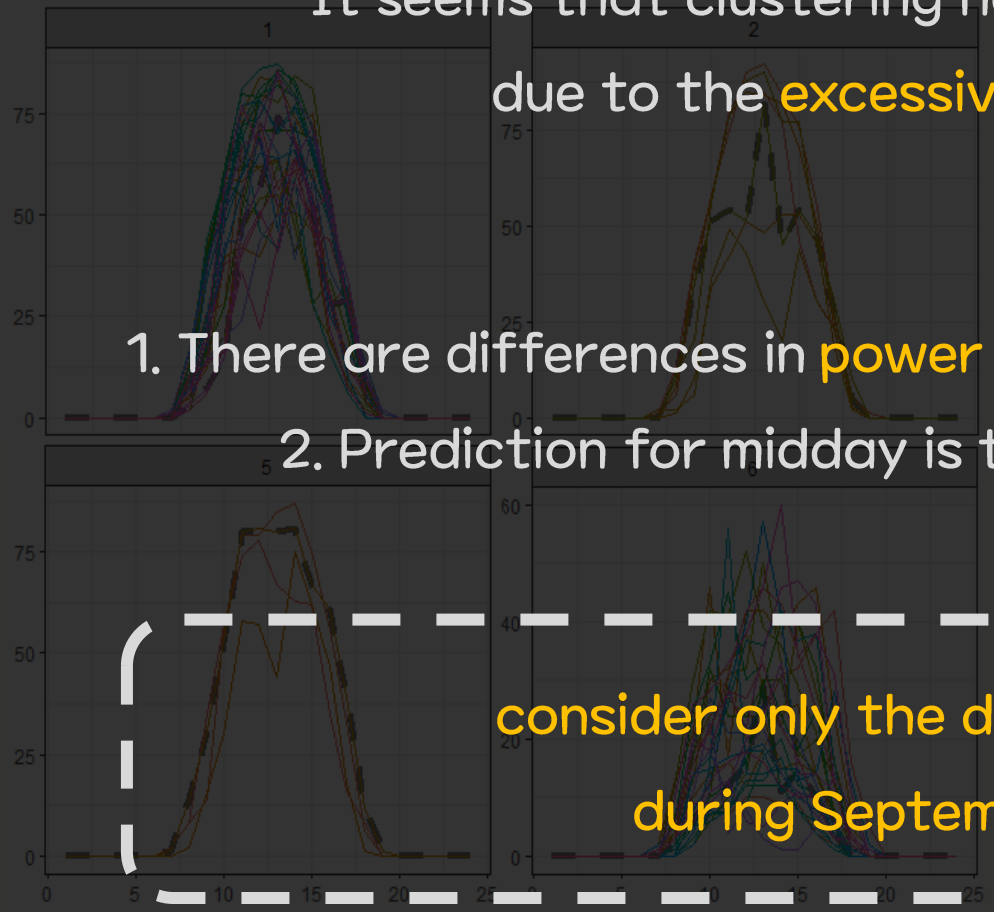
Changing the number of clusters, whether increasing or decreasing, leads to issues.

Modeling

time-series clustering result

It seems that clustering has not been properly achieved

due to the excessive diversity of patterns



Clusters like 2 and 5 have very few samples

For clusters 1 and 6, the patterns are not distinct enough to be considered separated clusters

1. There are differences in power generation patterns across weather
2. Prediction for midday is the most crucial (high incentive)

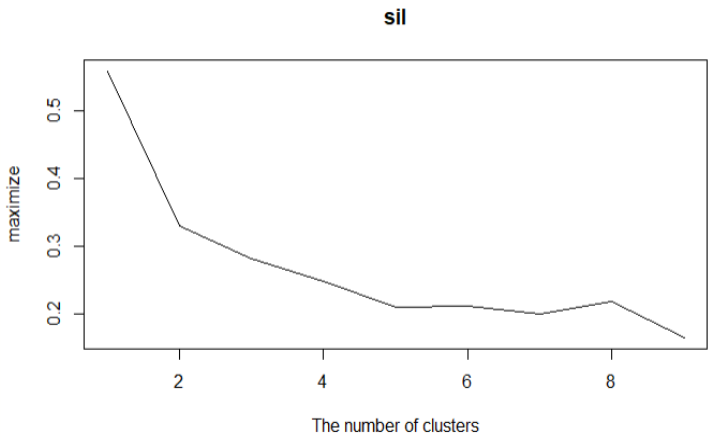
consider only the data from 11 AM to 3 PM

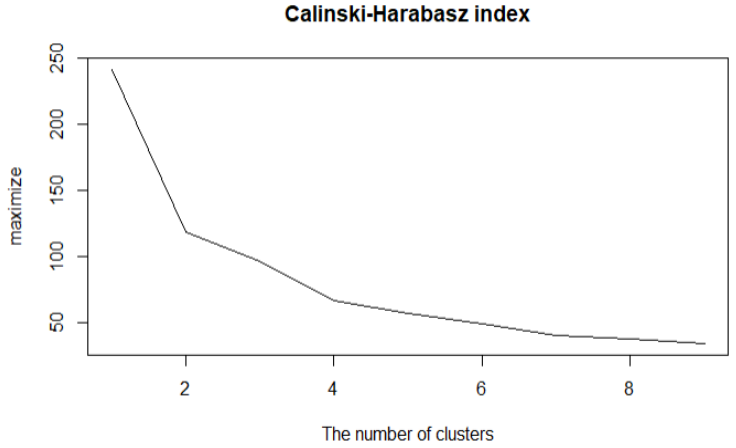
during September to November!

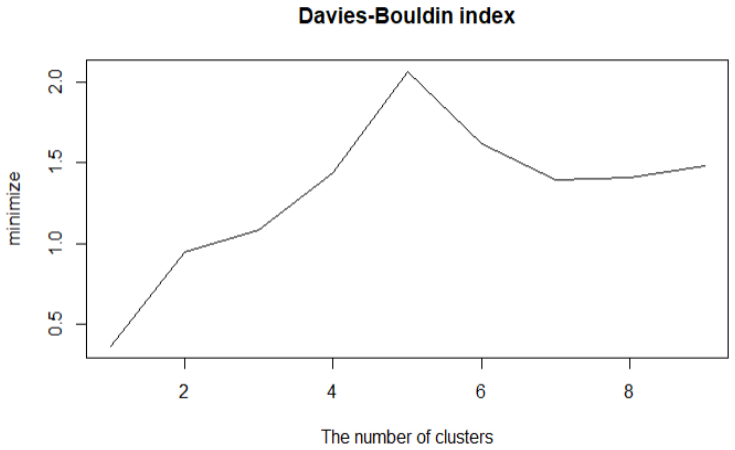
Changing the number of clusters, whether increasing or decreasing, leads to issues.

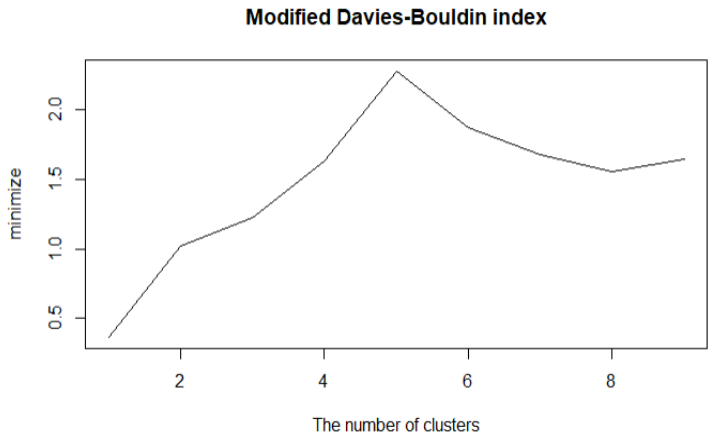
Modeling

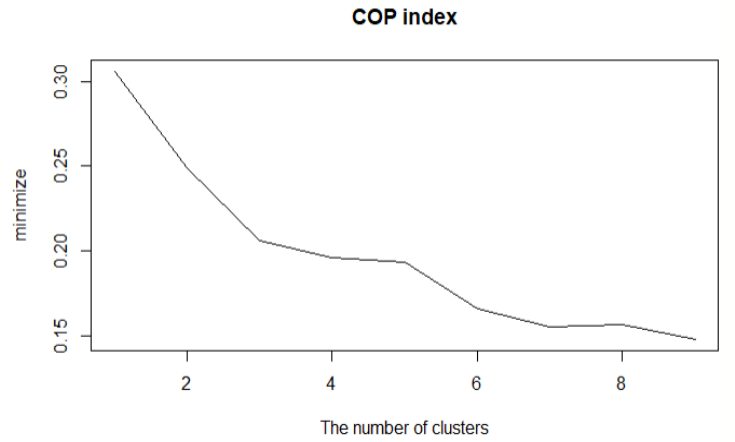
time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)





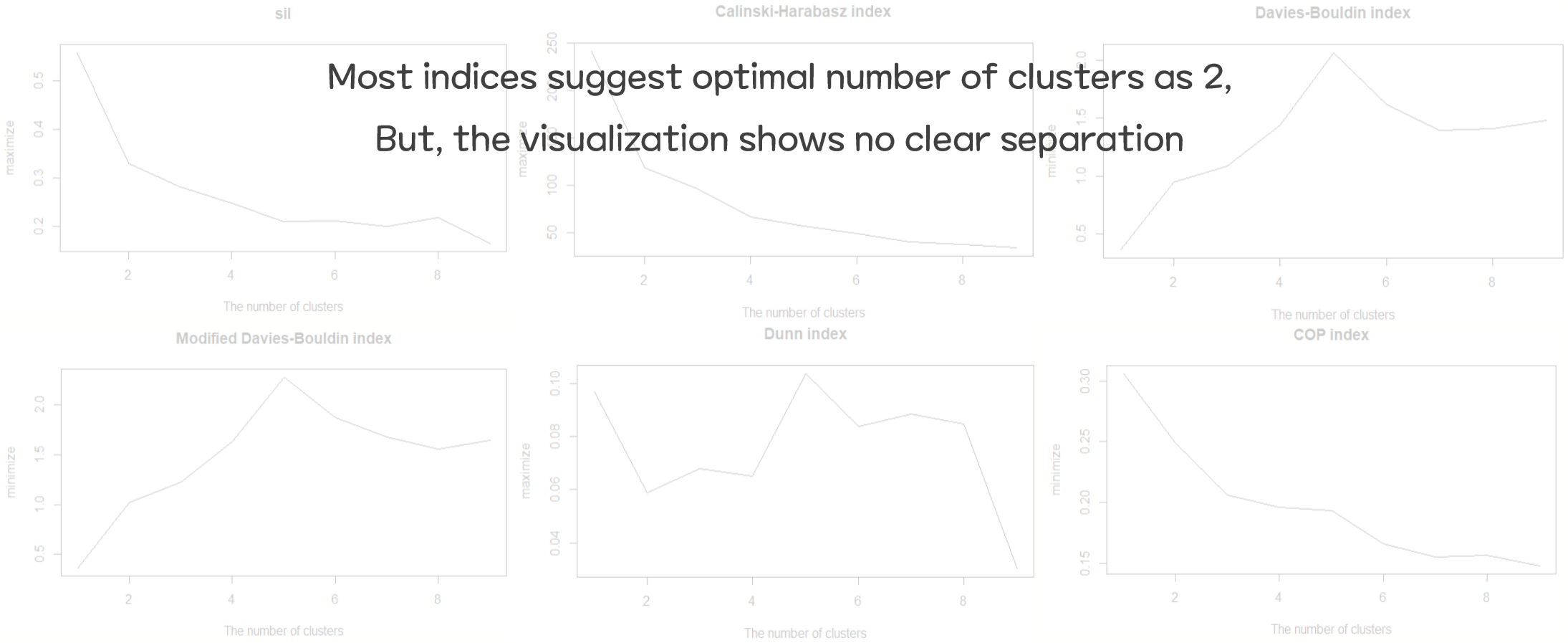






Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)

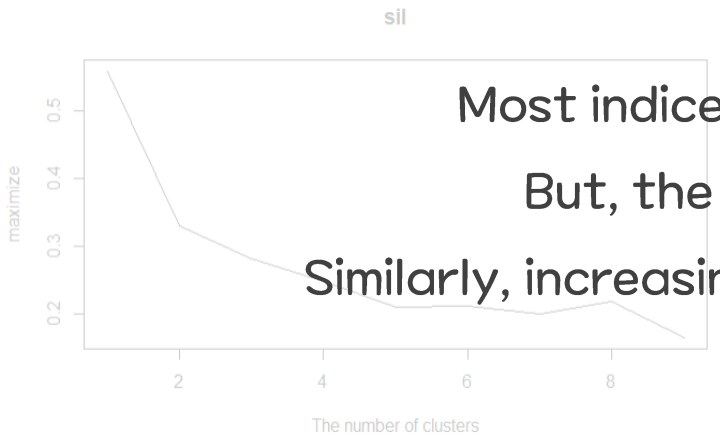


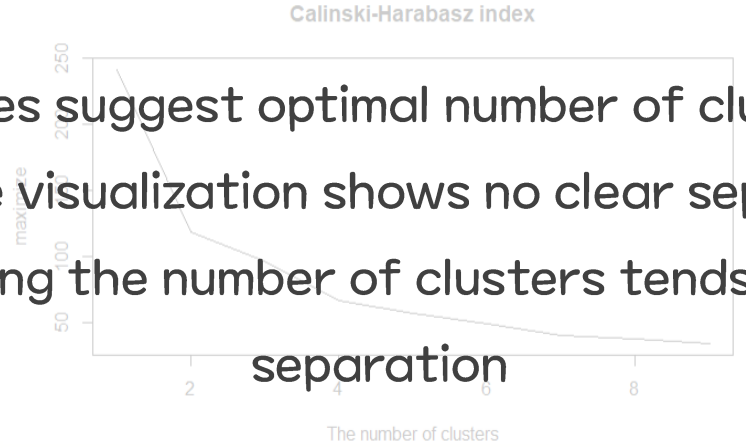
Most indices suggest optimal number of clusters as 2,

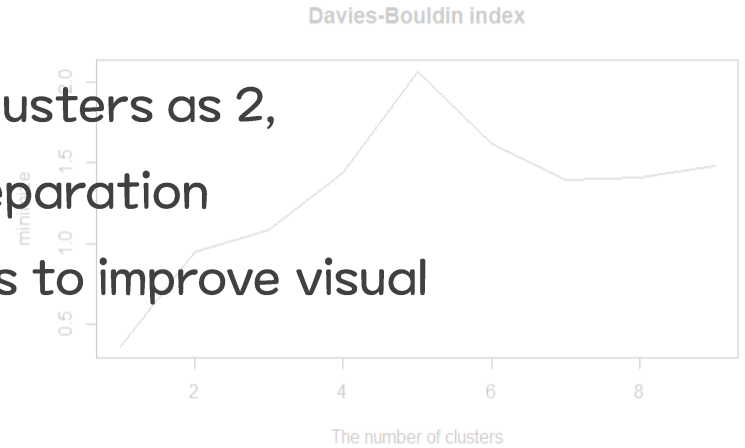
But, the visualization shows no clear separation

Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)







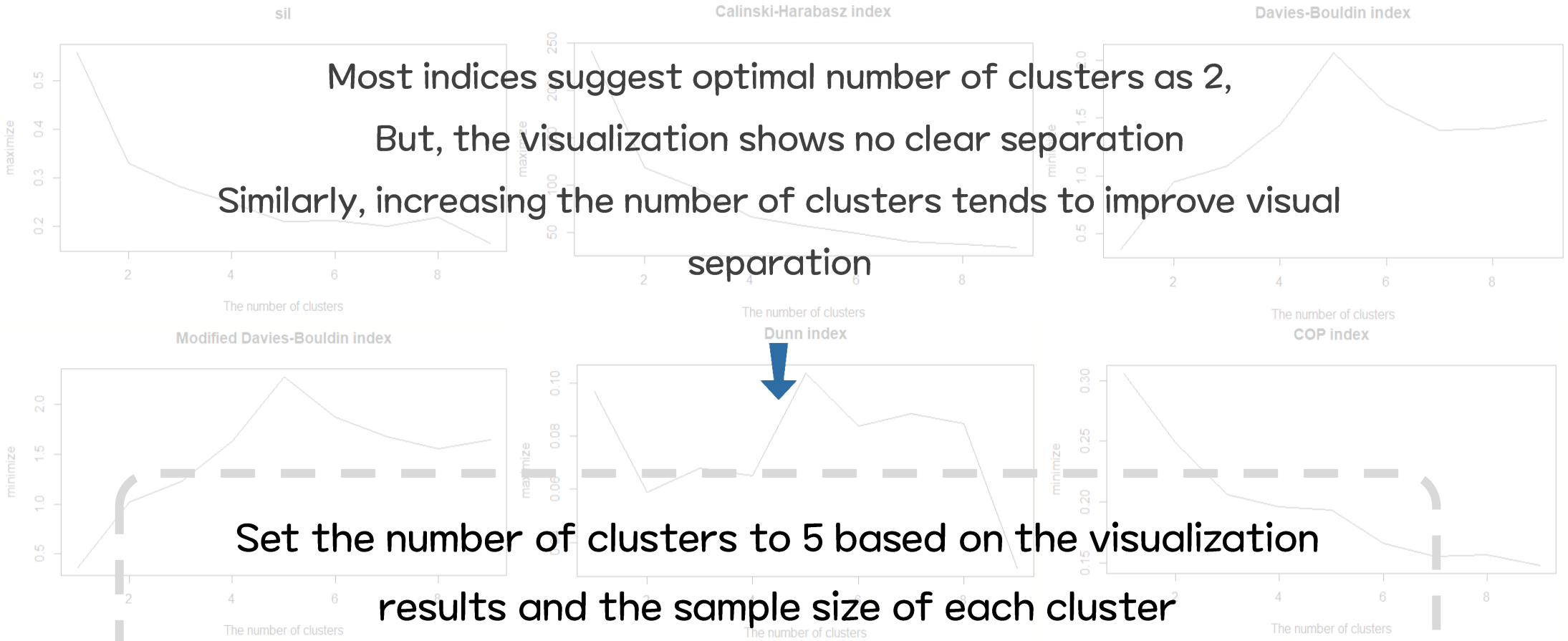
Most indices suggest optimal number of clusters as 2,

But, the visualization shows no clear separation

Similarly, increasing the number of clusters tends to improve visual separation

Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)



Most indices suggest optimal number of clusters as 2,

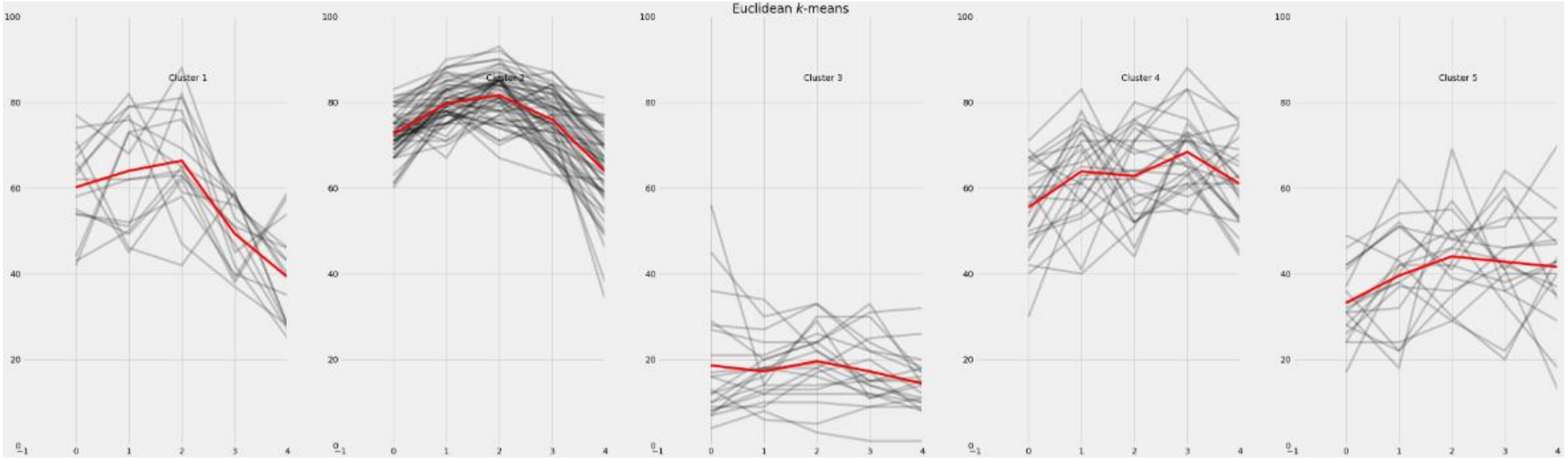
But, the visualization shows no clear separation

Similarly, increasing the number of clusters tends to improve visual separation

Set the number of clusters to 5 based on the visualization results and the sample size of each cluster

Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)



Likewise, the patterns are too diverse

Although it appears that clustering has

somewhat been achieved, it actually has not.

Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)

Despite the poor clustering results,

it was decided to attempt modeling

~~decided to go ahead since It seemed a waste not to use the work already done~~

Likewise, the patterns are too diverse

Although it appears that clustering has

somewhat been achieved, it actually has not.

Modeling

time-series clustering evaluation index (Euclidean/11~3/Sep~Nov)

Regression-Enhanced Random Forest Algorithm

Step 1 : Extend the p-dimensional predictor X to a (p+q)-dimensional predictor X* by adding higher-order, interaction or other known parametric functions of X

Step 2 : Run Lasso of Y on X* with a pre-specified penalty parmeter λ. Let $\hat{\beta}_\lambda$ be the estimated coefficient, and ε^λ = Y − X* $\hat{\beta}_\lambda$ be the residual from Lasso. Create a new training dataset C^λ = {C^λ = (X_i, ε_i^λ):i=1,⋯,N}.

Step 3 : Build a random forest...

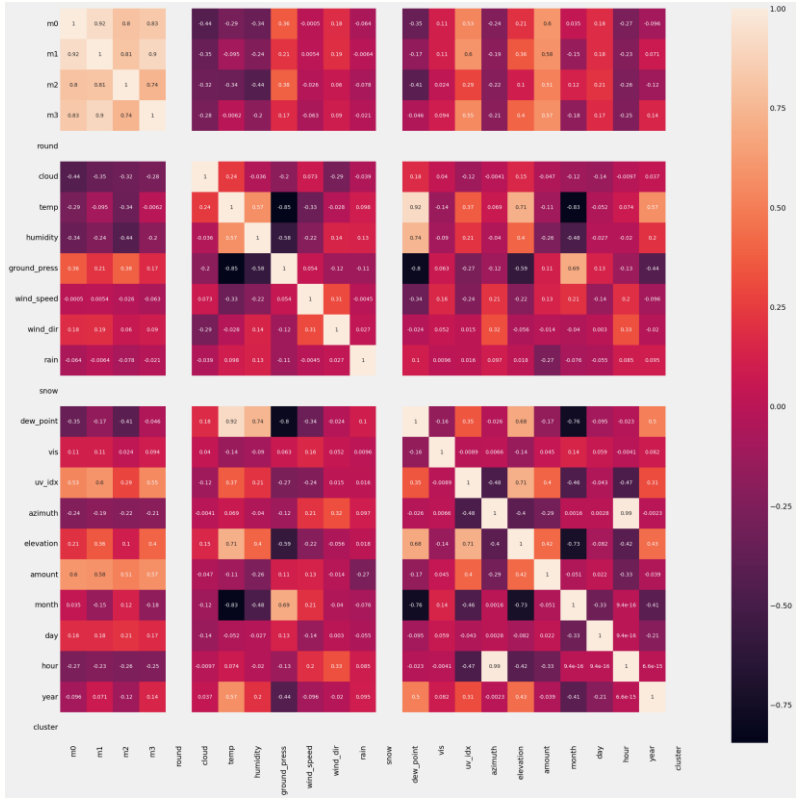
⋮

Haozhe, Z., Dan , N., Zhengyuan, Z. (2019). *Regression-Enhanced Random Forests*, 4. doi:10.48550/arXiv.1904.10416

In the first week, to address the linear relationship between dependent and independent variables, LR and MLP were applied to variables with high correlation (Regression-enhanced method)

Modeling

time-series clustering result (Euclidean /11~3/Sep~Nov)



When calculating correlation between power generation and variables for each clusters,
The correlation with UV/elevation/predicted generation was very high(between 0.8-0.9) before clustering
However, after dividing into clusters,
it sharply decreased to between 0.2-0.5

Modeling

time-series clustering result (Euclidean /11~3/Sep~Nov)

When calculating correlation between power generation and variables for each clusters, The correlation with UV/elevation/predicted generation was very high(between 0.8-0.9) before clustering

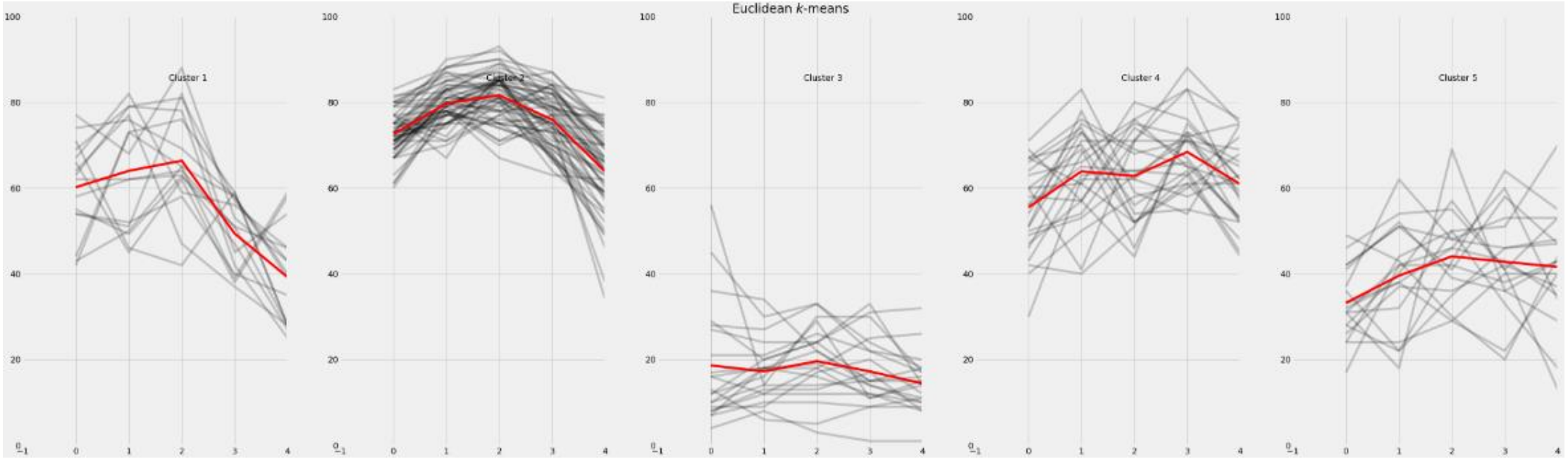
However, after dividing into clusters, it sharply decreased to between 0.2-0.5

Since the correlation is not extremely high,

1) Regression enhanced method 2) model with full variables were tried

Modeling

Time-series clustering+LR+LGBM

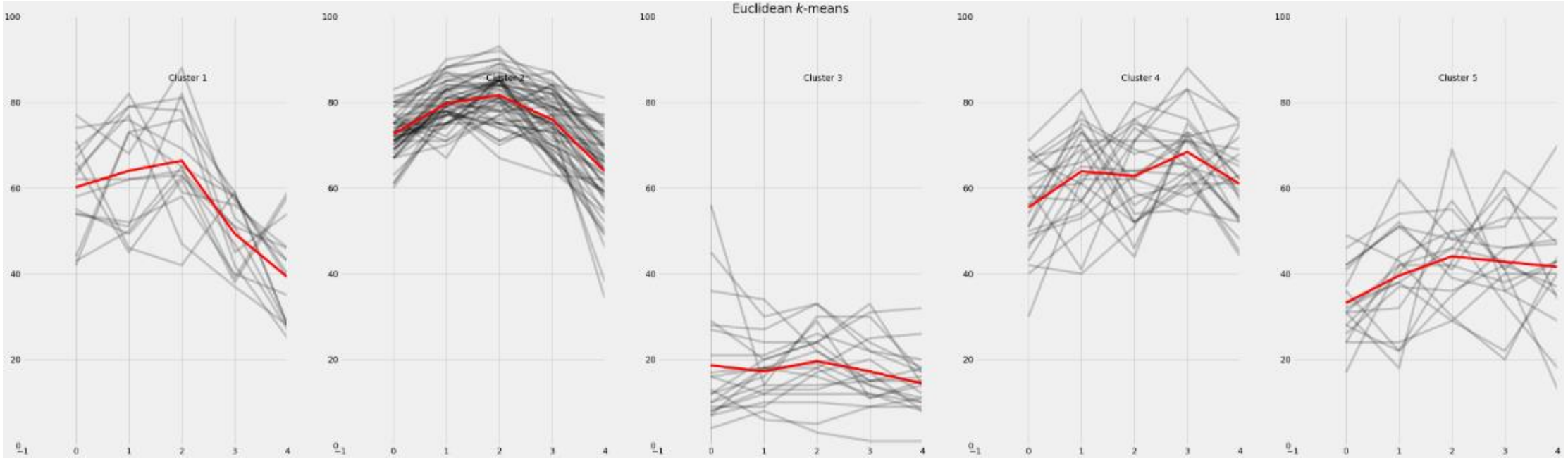


To make a prediction, it is necessary to determine in advance

which cluster the target test data belongs to

Modeling

Time-series clustering+LR+LGBM



It was decided to use the same approach as previously used

Modeling

Time-series clustering+LR+LGBM

	m0	m1	m2	m3	m4	time	round	cloud	temp	humidity	...	vis	uv_idx	azimuth	elevation	amount	month	day	hour	year	cluster
1786	52.1409	52.3106	55.2198	52.0965	45.8437	2022-09-01 11:00:00+09:00	1	100.0	24.32	74.0	...	16.0934	7.0	135.254	55.8898	36.0	9	1	11	202	2
1787	51.8298	53.6636	61.3145	48.4306	40.6616	2022-09-01 12:00:00+09:00	1	100.0	25.02	71.0	...	16.0934	8.0	161.559	62.4236	34.0	9	1	12	202	2
1788	60.7819	72.4139	74.5632	70.9034	53.2792	2022-09-01 13:00:00+09:00	1	100.0	25.58	70.0	...	16.0934	6.0	194.207	62.8751	22.0	9	1	13	202	2
1789	59.6068	69.6676	73.3340	65.7185	33.1650	2022-09-01 14:00:00+09:00	1	100.0	25.96	68.0	...	16.0934	5.0	221.728	56.9884	15.0	9	1	14	202	2
1790	58.0700	61.6508	65.0286	58.8161	19.0536	2022-09-01 15:00:00+09:00	1	99.0	25.97	70.0	...	16.0934	4.0	240.326	47.3516	18.0	9	1	15	202	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11578	57.8762	59.3476	52.5556	55.5518	51.8007	2023-10-15 11:00:00+09:00	1	12.0	18.98	68.0	...	16.0934	4.0	152.542	42.9224	54.0	10	15	11	202	0
11579	60.4846	63.2077	52.5277	58.5800	57.0589	2023-10-15 12:00:00+09:00	1	10.0	19.20	67.0	...	16.0934	4.0	172.860	46.6011	51.0	10	15	12	202	0
11580	61.0122	66.6243	52.7246	64.0328	63.2603	2023-10-15 13:00:00+09:00	1	8.0	19.37	67.0	...	16.0934	4.0	194.339	45.8045	82.0	10	15	13	202	0
11581	62.4748	63.3071	49.9163	63.1707	56.2874	2023-10-15 14:00:00+09:00	1	10.0	19.47	68.0	...	16.0934	3.0	213.572	40.7522	45.0	10	15	14	202	0
11582	51.3055	51.1993	41.5603	53.4420	42.0570	2023-10-15 15:00:00+09:00	1	10.0	19.53	69.0	...	16.0934	2.0	229.064	32.5767	54.0	10	15	15	202	0

Add a cluster column to each time point and fit it

as the Y variable in a classification model

Modeling

Time-series clustering+LR+LGBM

Tried both methods:

arranging data by day as rows

arranging data by hour as rows

	m0	m1	m2	m3	m4	time	round	cloud	temp	humidity	...	vis	uv_idx	azimuth	elevation	amount	month	day	hour	year	cluster
1786	52.1409	52.3106	55.2198	52.0965	45.8437	2022-09-01 11:00:00+09:00	1	100.0	24.32	71.0	...	16.0934	7.0	135.254	55.8898	36.0	9	1	11	202...	2
1787	51.8298	53.6636	61.3145	48.4306	40.6616	2022-09-01 12:00:00+09:00	1	100.0	25.02	71.0	...	16.0934	8.0	161.559	62.4236	34.0	9	1	12	202...	2
1788	60.7819	72.4139	74.5632	70.9034	53.2792	2022-09-01 13:00:00+09:00	1	100.0	25.58	70.0	...	16.0934	6.0	194.207	62.8751	22.0	9	1	13	202...	2
1789	59.6068	69.6676	73.3340	65.7185	33.1650	2022-09-01 14:00:00+09:00	1	100.0	25.81	70.0	...	16.0934	4.0	221.728	56.9884	15.0	9	1	14	202...	2
1790	58.0700	61.6508	65.0286	58.8161	19.0536	2022-09-01 15:00:00+09:00	1	99.0	25.97	70.0	...	16.0934	4.0	240.326	47.3516	18.0	9	1	15	202...	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11578	57.8762	59.3476	52.5556	55.5518	51.8007	2023-10-15 11:00:00+09:00	1	12.0	18.98	68.0	...	16.0934	4.0	152.542	42.9224	54.0	10	15	11	202...	0
11579	60.4846	63.2077	52.5277	58.5800	57.0589	2023-10-15 12:00:00+09:00	1	10.0	19.20	67.0	...	16.0934	4.0	172.860	46.6011	51.0	10	15	12	202...	0
11580	61.0122	66.6243	52.7246	64.0328	63.2603	2023-10-15 13:00:00+09:00	1	8.0	19.37	67.0	...	16.0934	4.0	194.339	45.8045	82.0	10	15	13	202...	0
11581	62.4748	63.3071	49.9163	63.1707	56.2874	2023-10-15 14:00:00+09:00	1	10.0	19.47	68.0	...	16.0934	3.0	213.572	40.7522	45.0	10	15	14	202...	0
11582	51.3055	51.1993	41.5603	53.4420	42.0570	2023-10-15 15:00:00+09:00	1	10.0	19.53	69.0	...	16.0934	2.0	229.064	32.5767	54.0	10	15	15	202...	0

Add a cluster column to each time point and fit it

as the Y variable in a classification model

Modeling

Time-series clustering+LR+LGBM

Tried both methods:

arranging data by day as rows

arranging data by hour as rows

Both methods had an accuracy between 0.5 and 0.55,

-> poor performance

	m0	m1	m2	m3	m4	time	round	cloud	temp	humidity	...	vis	uv_idx	azimuth	elevation	amount	month	day	hour	year	cluster
1786	52.1409	52.3106	55.2198	52.0965	45.8437	2022-09-01 11:00:00+09:00	1	100.0	24.32	71.0	...	16.0934	7.0	135.254	55.8898	36.0	9	1	11	202...	2
1787	51.8298	53.6636	61.3145	48.4306	40.6616	2022-09-01 12:00:00+09:00	1	100.0	25.02	71.0	...	16.0934	8.0	161.559	62.4236	34.0	9	1	12	202...	2
1788	60.7819	72.4139	74.5632	70.9034	53.2792	2022-09-01 13:00:00+09:00	1	100.0	25.58	70.0	...	16.0934	6.0	194.207	62.8751	22.0	9	1	13	202...	2
1789	59.6068	69.6676	73.3340	65.7185	33.1650	2022-09-01 14:00:00+09:00	1	100.0	25.97	70.0	...	16.0934	4.0	221.728	56.9884	15.0	9	1	14	202...	2
1790	58.0700	61.6508	65.0286	58.8161	19.0536	2022-09-01 15:00:00+09:00	1	99.0	25.97	70.0	...	16.0934	4.0	240.326	47.3516	18.0	9	1	15	202...	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11578	57.8762	59.3476	52.5556	55.5518	51.8007	2023-10-15 11:00:00+09:00	1	12.0	18.98	68.0	...	16.0934	4.0	152.542	42.9224	54.0	10	15	11	202...	0
11579	60.4846	63.2077	52.5277	58.5800	57.0589	2023-10-15 12:00:00+09:00	1	10.0	19.23	67.0	...	16.0934	4.0	172.860	46.6011	51.0	10	15	12	202...	0
11580	61.0122	66.6243	52.7246	64.0328	63.2603	2023-10-15 13:00:00+09:00	1	8.0	19.37	67.0	...	16.0934	4.0	194.339	45.8045	82.0	10	15	13	202...	0
11581	62.4748	63.3071	49.9163	63.1707	56.2874	2023-10-15 14:00:00+09:00	1	10.0	19.47	68.0	...	16.0934	3.0	213.572	40.7522	45.0	10	15	14	202...	0
11582	51.3055	51.1993	41.5603	53.4420	42.0570	2023-10-15 15:00:00+09:00	1	10.0	19.53	69.0	...	16.0934	2.0	229.064	32.5767	54.0	10	15	15	202...	0

Add a cluster column to each time point and fit it

as the Y variable in a classification model

Modeling

Time-series clustering+LR+LGBM

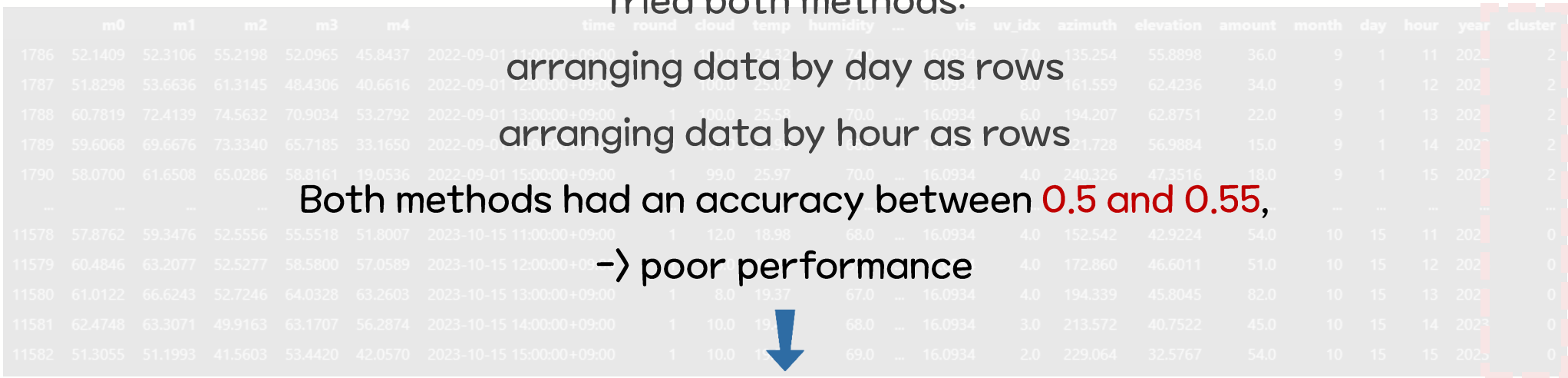
Tried both methods:

arranging data by day as rows

arranging data by hour as rows

Both methods had an accuracy between 0.5 and 0.55,

-> poor performance



	m0	m1	m2	m3	m4	time	round	cloud	temp	humidity	...	vis	uv_idx	azimuth	elevation	amount	month	day	hour	year	cluster
1786	52.1409	52.3106	55.2198	52.0965	45.8437	2022-09-01 11:00:00+09:00	1	100.0	24.32	71.0	...	16.0934	7.0	135.254	55.8898	36.0	9	1	11	202...	2
1787	51.8298	53.6636	61.3145	48.4306	40.6616	2022-09-01 12:00:00+09:00	1	100.0	25.02	71.0	...	16.0934	8.0	161.559	62.4236	34.0	9	1	12	202...	2
1788	60.7819	72.4139	74.5632	70.9034	53.2792	2022-09-01 13:00:00+09:00	1	100.0	25.58	70.0	...	16.0934	6.0	194.207	62.8751	22.0	9	1	13	202...	2
1789	59.6068	69.6676	73.3340	65.7185	33.1650	2022-09-01 14:00:00+09:00	1	100.0	25.97	70.0	...	16.0934	4.0	221.728	56.9884	15.0	9	1	14	202...	2
1790	58.0700	61.6508	65.0286	58.8161	19.0536	2022-09-01 15:00:00+09:00	1	99.0	25.97	70.0	...	16.0934	4.0	240.326	47.3516	18.0	9	1	15	202...	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11578	57.8762	59.3476	52.5556	55.5518	51.8007	2023-10-15 11:00:00+09:00	1	12.0	18.98	68.0	...	16.0934	4.0	152.542	42.9224	54.0	10	15	11	202...	0
11579	60.4846	63.2077	52.5277	58.5800	57.0589	2023-10-15 12:00:00+09:00	1	8.0	19.37	67.0	...	16.0934	4.0	172.860	46.6011	51.0	10	15	12	202...	0
11580	61.0122	66.6243	52.7246	64.0328	63.2603	2023-10-15 13:00:00+09:00	1	8.0	19.37	67.0	...	16.0934	4.0	194.339	45.8045	82.0	10	15	13	202...	0
11581	62.4748	63.3071	49.9163	63.1707	56.2874	2023-10-15 14:00:00+09:00	1	10.0	19...	68.0	...	16.0934	3.0	213.572	40.7522	45.0	10	15	14	202...	0
11582	51.3055	51.1993	41.5603	53.4420	42.0570	2023-10-15 15:00:00+09:00	1	10.0	1...	69.0	...	16.0934	2.0	229.064	32.5767	54.0	10	15	15	202...	0

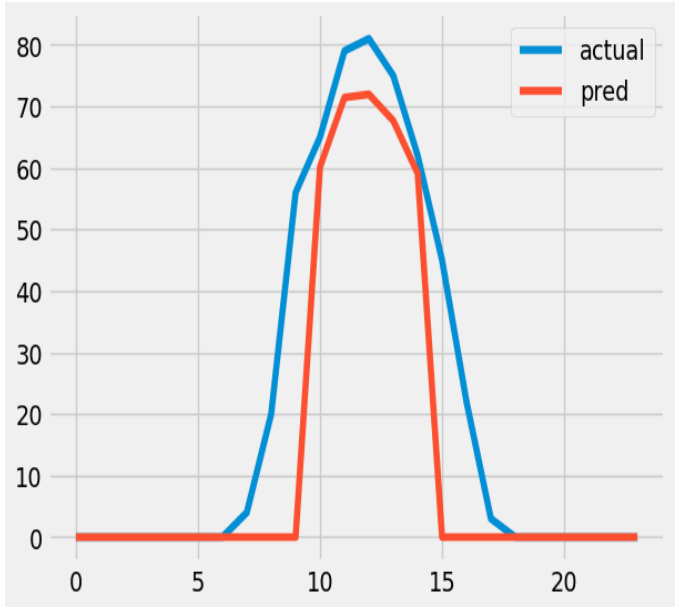
Instead of classifying into a specific cluster

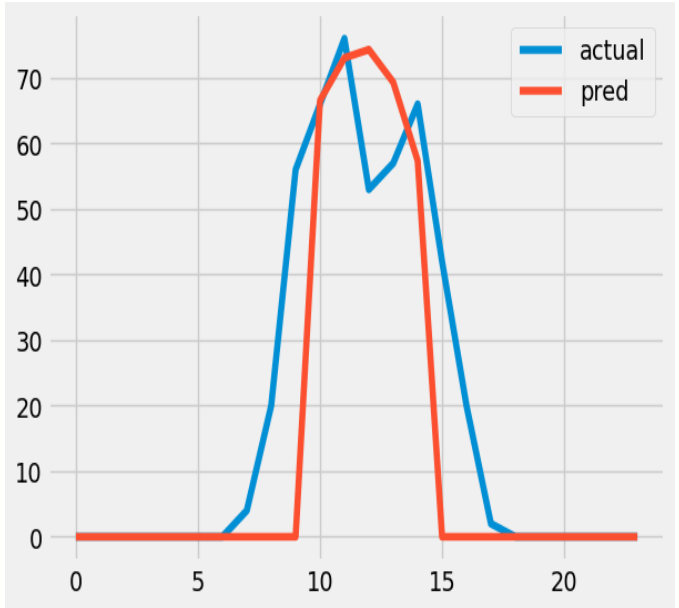
Decided to weight the predicted values by the

probability of belonging to each cluster

Modeling

Time-series clustering+LR+LGBM result





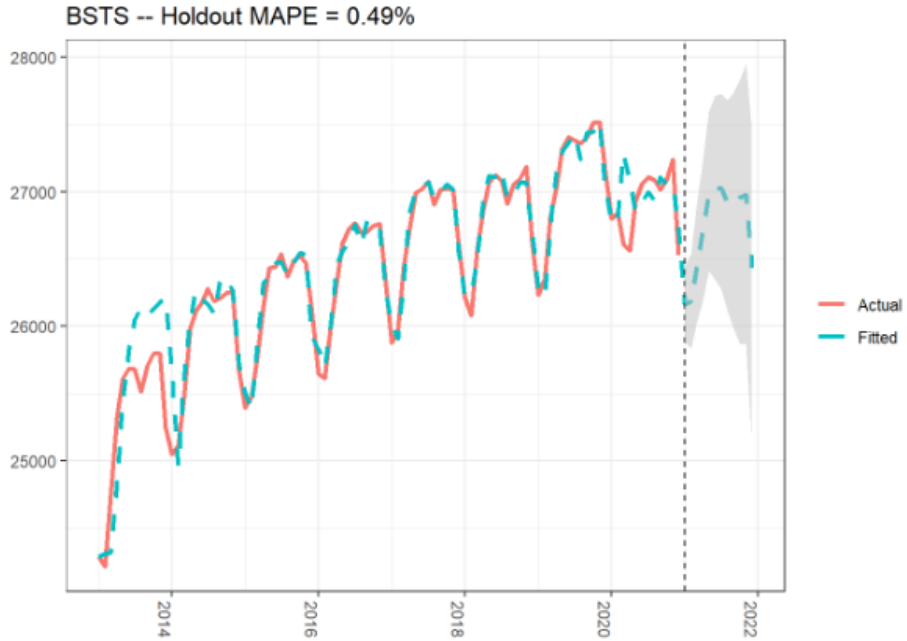
There is no significant performance improvement

Modeling

Bayesian Structural Time Series (BSTS)

Bayesian Structural Time Series

A time-series model that not only reflects trends and seasonality
but also captures correlations among
different time-series data through regression operators



Modeling

Bayesian Structural Time Series (BSTS)

$$y_t = \mu_t + \epsilon_t$$
$$y_t = \mu_t + S + \epsilon_t$$
$$y_t = \mu_t + x_t\beta + S + \epsilon_t$$

μ_t : trend
S : seansonality
β: regression coef

Users can selectively utilize trend, seasonality, and regression components to account for the structure of the data

Modeling

Bayesian Structural Time Series (BSTS)

Spike-Slab prior

it is used for variable selection or determining variable importance in a model. Particularly useful in regression models to identify which of the many variables significantly affect the model

Modeling

Bayesian Structural Time Series (BSTS)

Spike-Slab prior

it is used for variable selection or determining variable importance in a model. Particularly useful in regression models to identify which of the many variables significantly affect the model

$$Let\ r_k = 1\ if \beta_k \neq 0, and\ r_k = 0\ if \beta_k = 0.$$
$$Let\ \beta_r\ denote\ the\ subset\ of\ elements\ of \beta\ where \beta_k \neq 0.$$
$$p(\beta, r, \sigma_\epsilon^2) = p(\beta_r | r, \sigma_\epsilon^2) p(\sigma_\epsilon^2 | r) p(r)$$

In BSTS, prior distributions are set for the regression components, and MCMC is used to estimate time-varying coefficients

Modeling

Bayesian Structural Time Series (BSTS)

model	data	Total incentive(10.25~10.31)	notes
Linear Regression	M0~M4	9020	Divided by seasons

The regression model, which unexpectedly showed the second-best performance incorporates trends and seasonality while reflecting different coefficients at each time point?

Modeling

Bayesian Structural Time Series (BSTS)

model	data	Total incentive(10.25~10.31)	notes
Linear Regression	M0~M4	9020	Divided by seasons

The regression model, which unexpectedly showed the second-best performance incorporates trends and seasonality while reflecting different coefficients at each time point?

Decided to try it immediately

Modeling

Condition for Adding Regression Components

All X variables must meet stationarity
The ADF test result shows that temp variable
does not satisfy stationarity, so differencing was applied before use

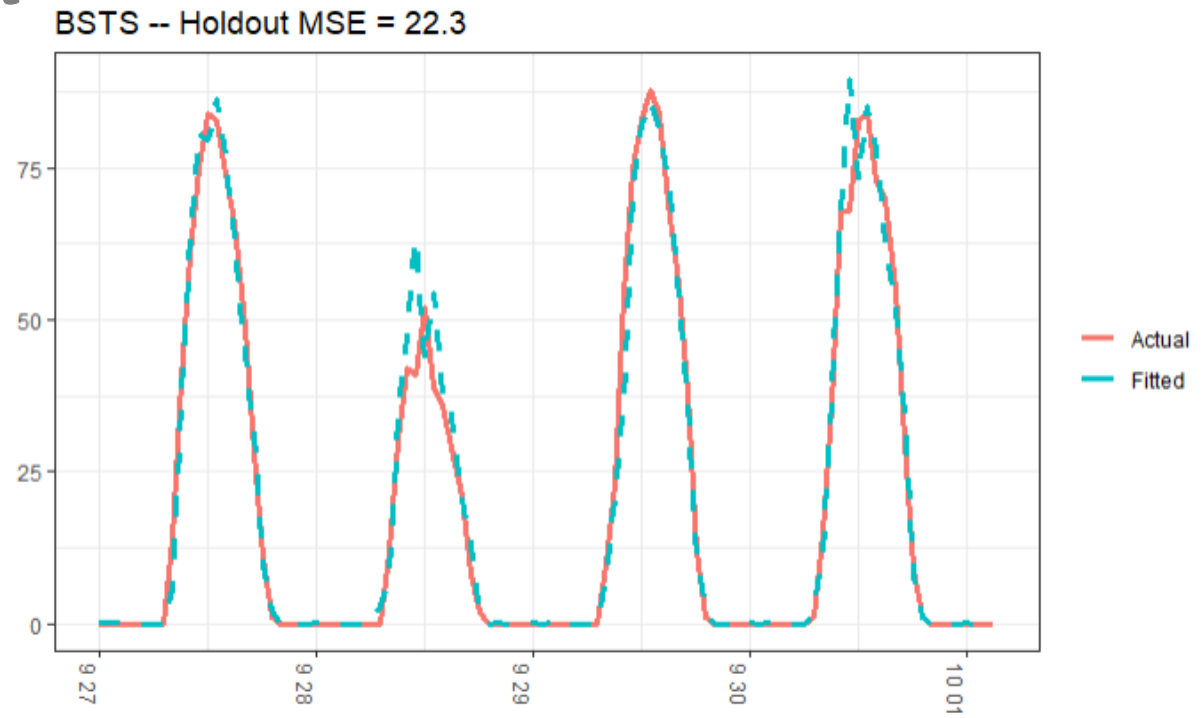
$$Let\ r_k = 1\ if \beta_k \neq 0, and\ r_k = 0\ if \beta_k = 0.$$
$$Let\ \beta_r\ denote\ the\ subset\ of\ elements\ of \beta\ where \beta_k \neq 0.$$

$$p(\beta, r, \sigma_\epsilon^2) = p(\beta_r | r, \sigma_\epsilon^2) p(\sigma_\epsilon^2 | r) p(r)$$

Temp	Diff(Temp)
P-value : 0.548188	P-value < 0.01

Modeling

BSTS result



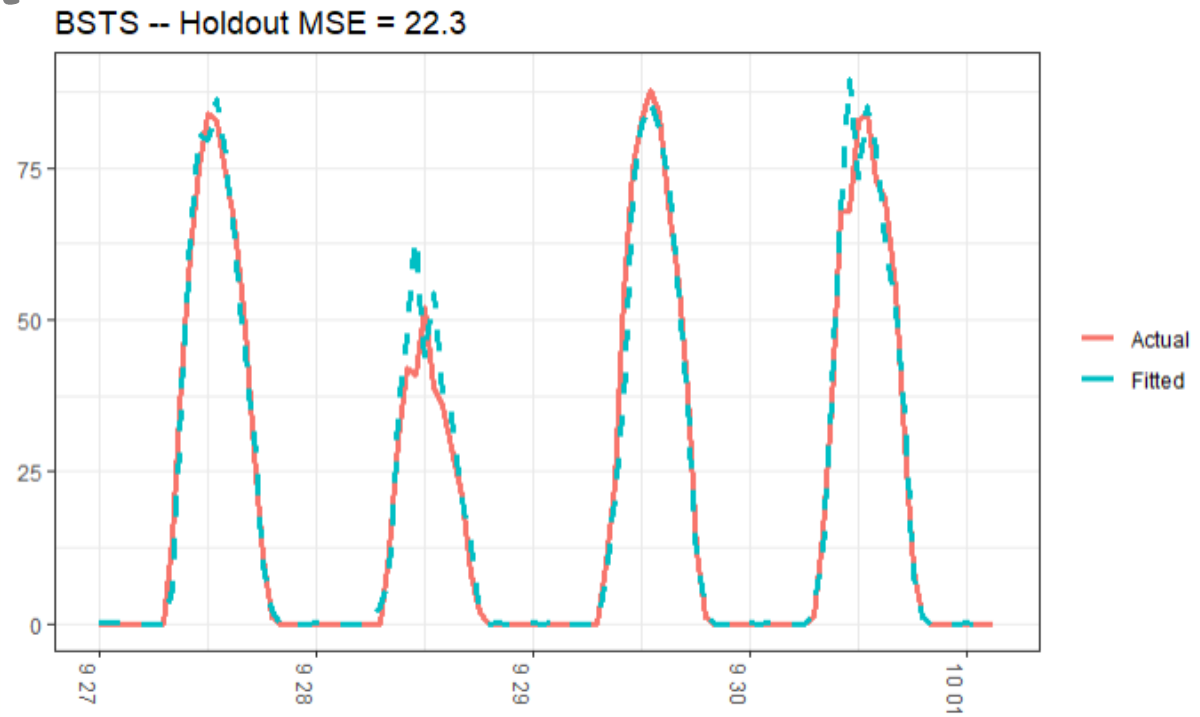
model seems to capture the spikes point

the actual results show an error rate of over 8%,

resulting in an incentive value of 0

Modeling

BSTS result



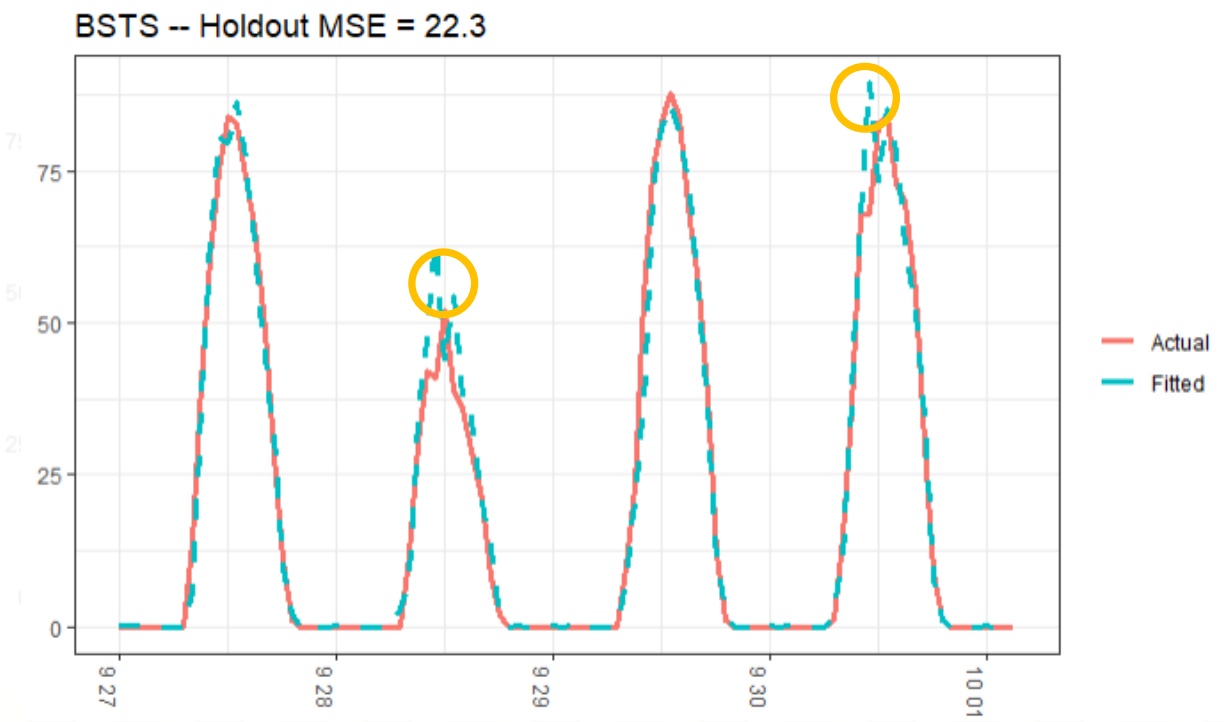
The model fits the smooth shape very well,

but not significantly better than other models

it has the issue of large error rates at spike points

Modeling

BSTS result



Could the model be overestimating due to certain regression coefficients being too large, especially in sharp patterns?

Modeling

BSTS result

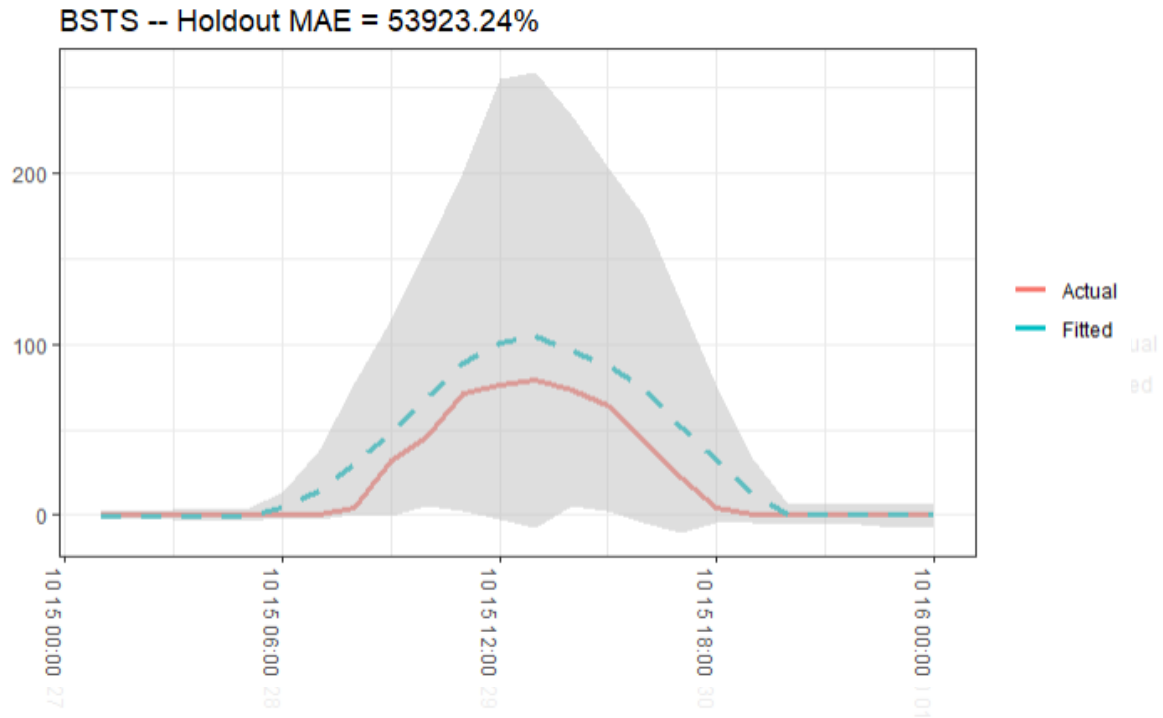
```
AddDynamicRegression(
    state.specification,
    formula,
    data,
    model.options = NULL,
    sigma.mean.prior.DEPRECATED = NULL,
    shrinkage.parameter.prior.DEPRECATED = GammaPrior(a = 10, b = 1),
    sigma.max.DEPRECATED = NULL,
    contrasts = NULL,
    na.action = na.pass)
```

Adjusting the parameters of the prior can control the rate at which regression coefficients converge to zero

However, no significant difference observed

Modeling

BSTS result



Also, it can be observed that the daily prediction results show a high variance

sharp peaks result in a very high error rate, leading to zero incentive

Due to the uncertainty of power generation patterns during the competition,

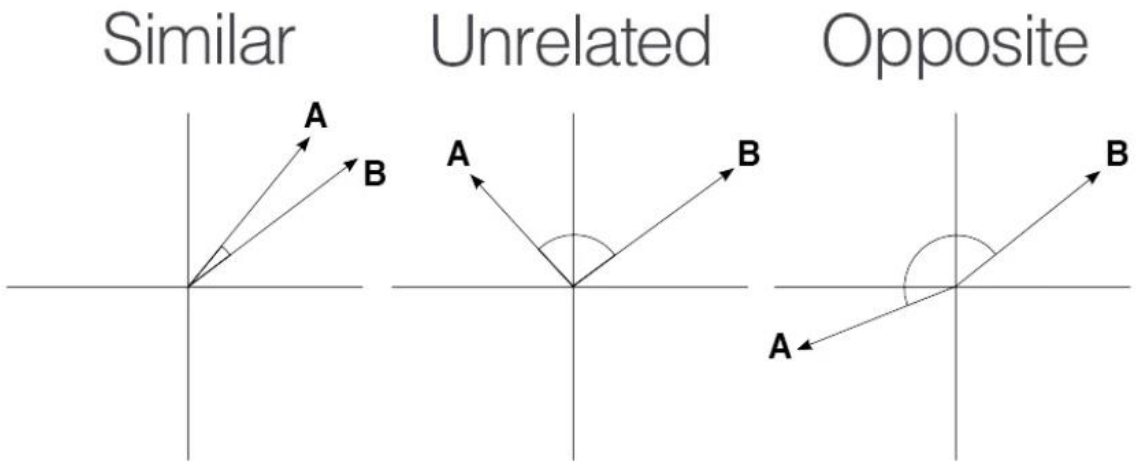
this model was excluded from the candidates

Modeling

Vector similarity

Vector Similarity

A metric for measuring the similarity between two vectors, indicating how similar the two data sets are



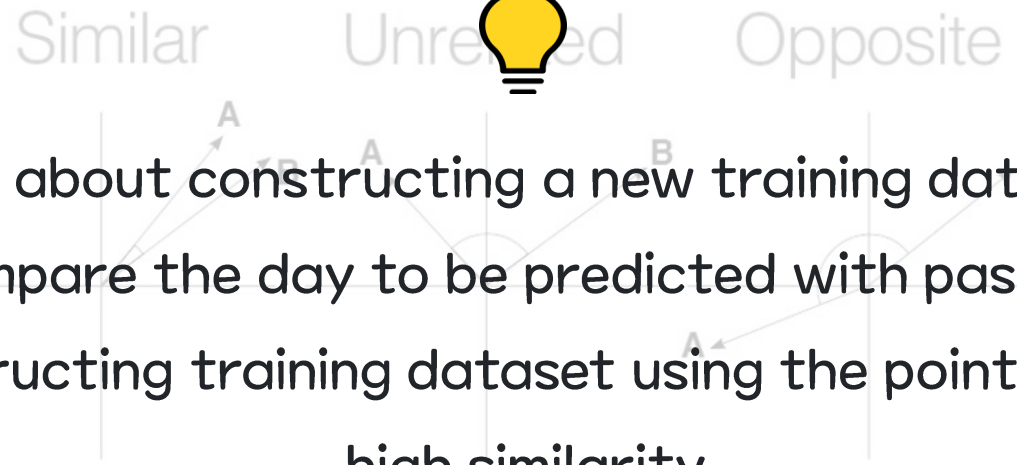
Modeling

Vector similarity

Vector Similarity

A metric for measuring the similarity between two vectors,

indicating how similar the two data sets are

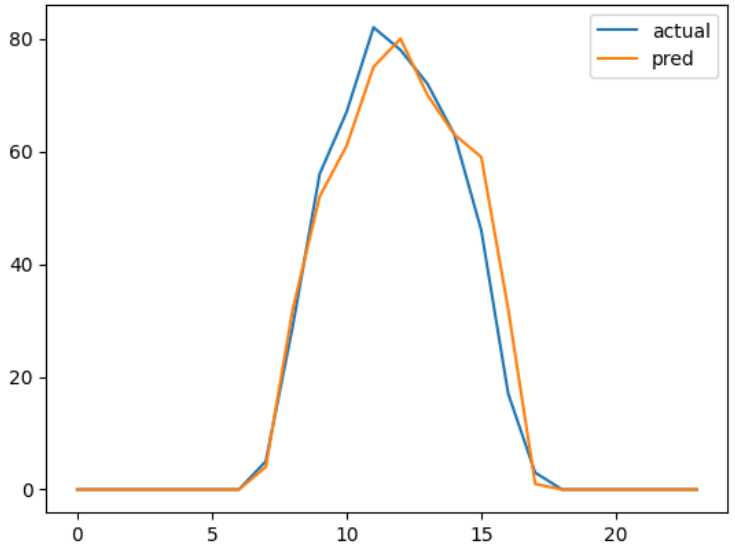


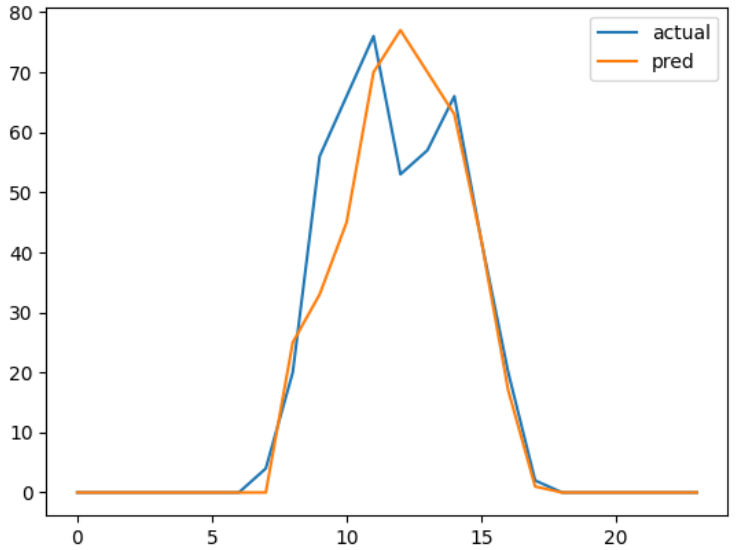
How about constructing a new training dataset?

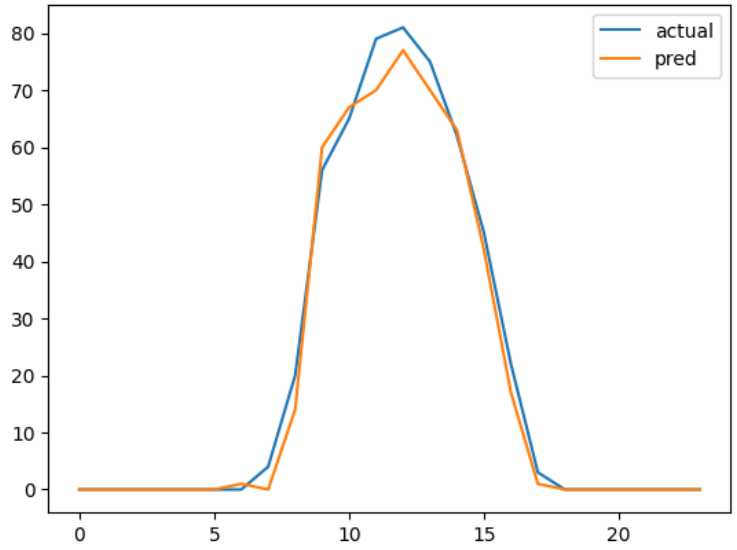
1. Compare the day to be predicted with past data.
2. Constructing training dataset using the points with the high similarity

Modeling

Vector similarity – Euclidean distance



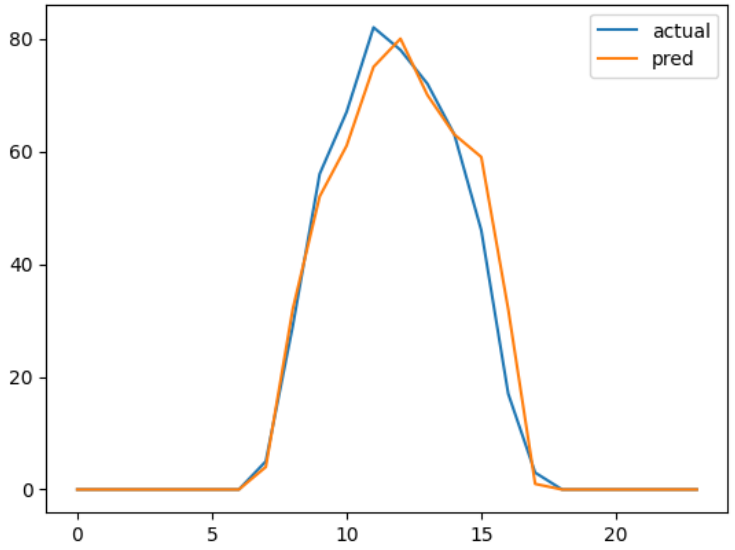


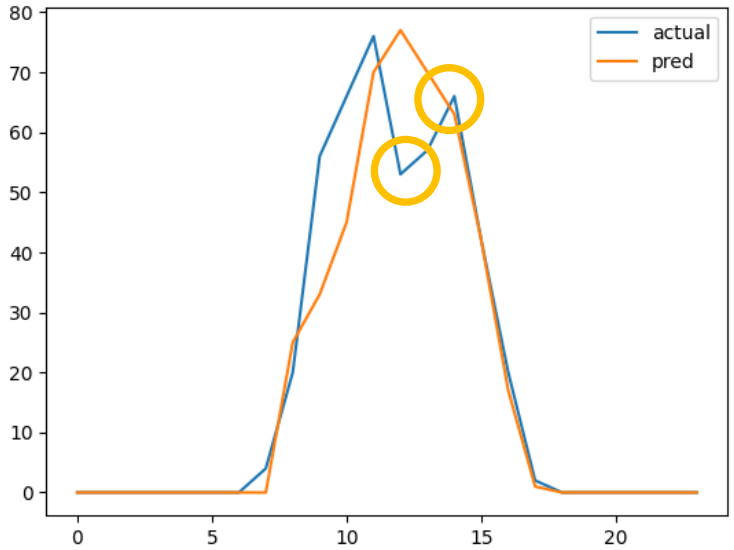


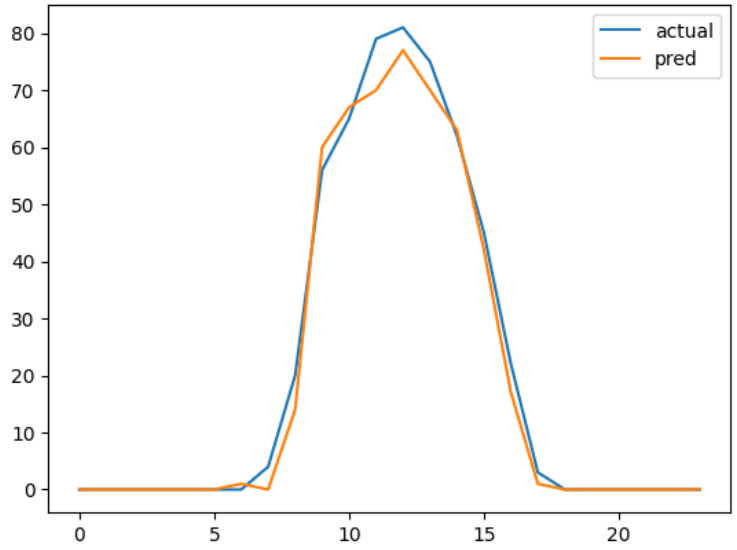
model	data	Total incentive(10.25~10.31)	notes
Euclidean Similarity	Weather data(m0~m4)	8826	Train data : Oct ~ Nov

Modeling

Vector similarity – Euclidean distance





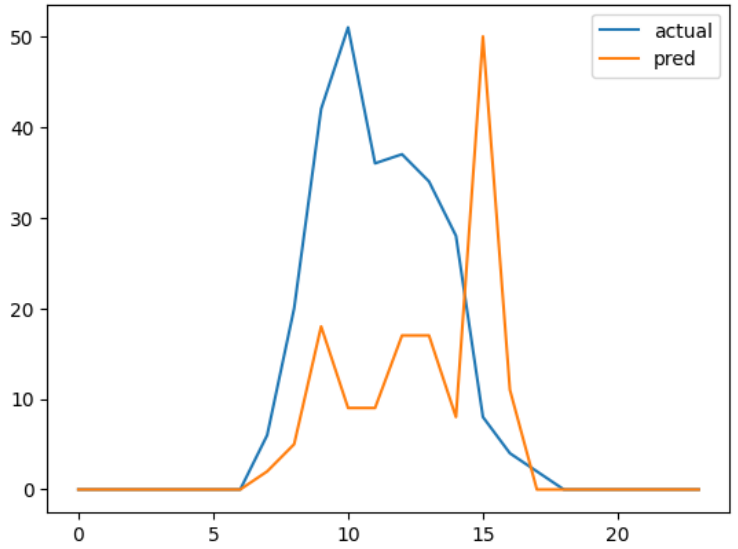


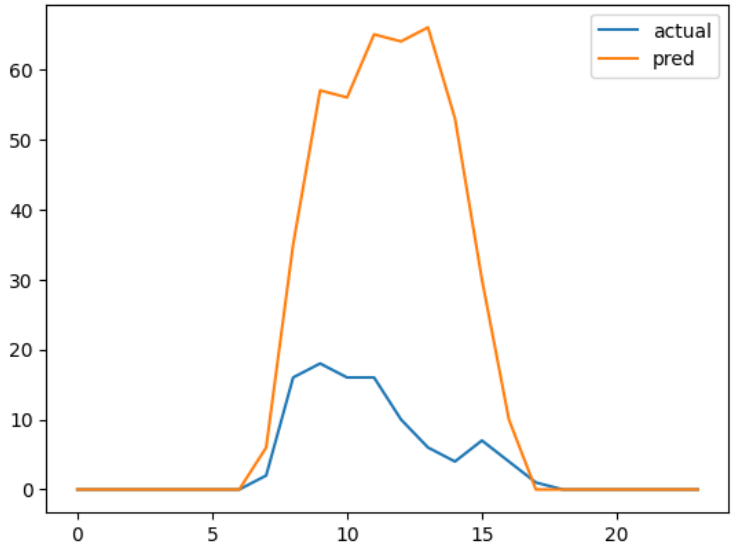
Days failing to predict spikes receive lower incentives

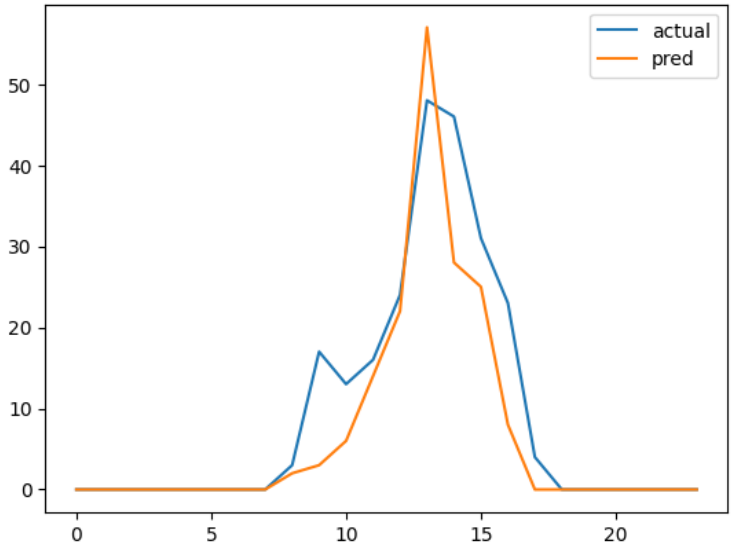
predictions are generally accurate, some days miss the midday point

Modeling

Vector similarity – Euclidean distance



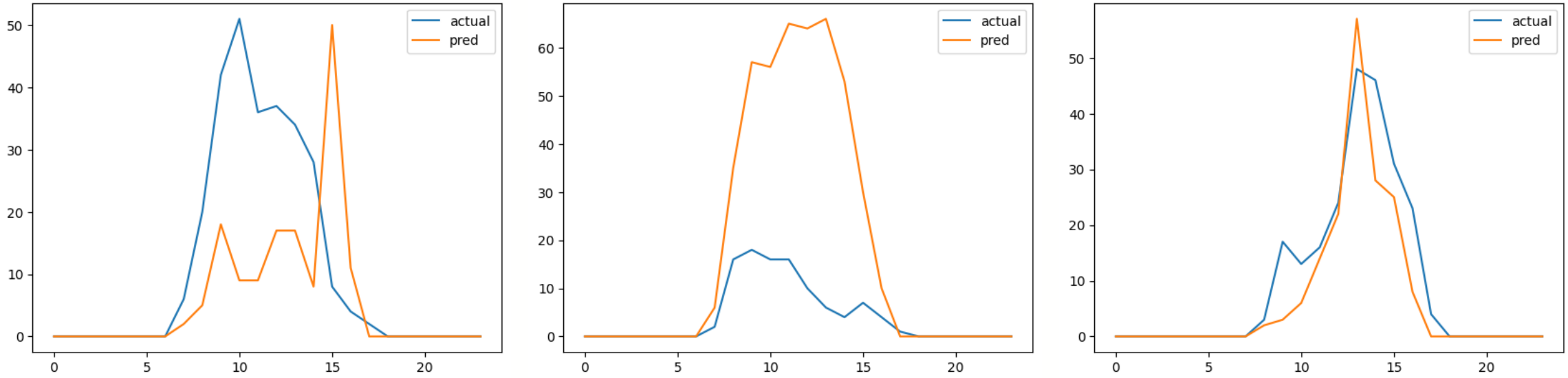




model	data	Total incentive(11.01~11.07)	notes
Euclidean Similarity	Weather data(m0~m4)	1424	Train data : Oct ~ Nov

Modeling

Vector similarity – Euclidean distance

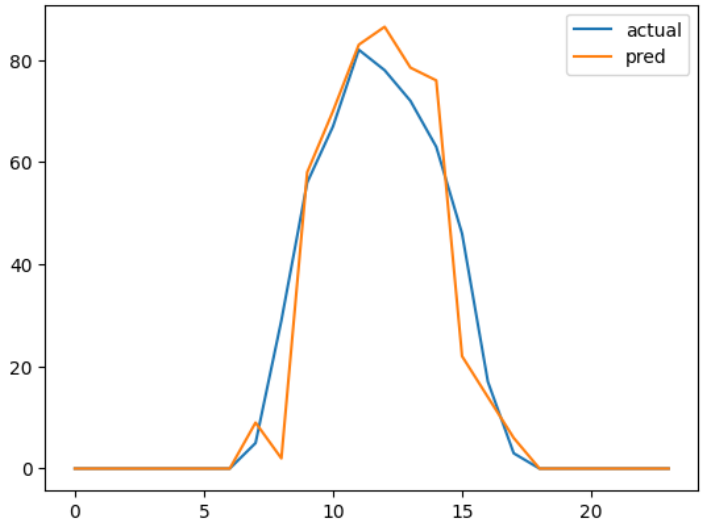


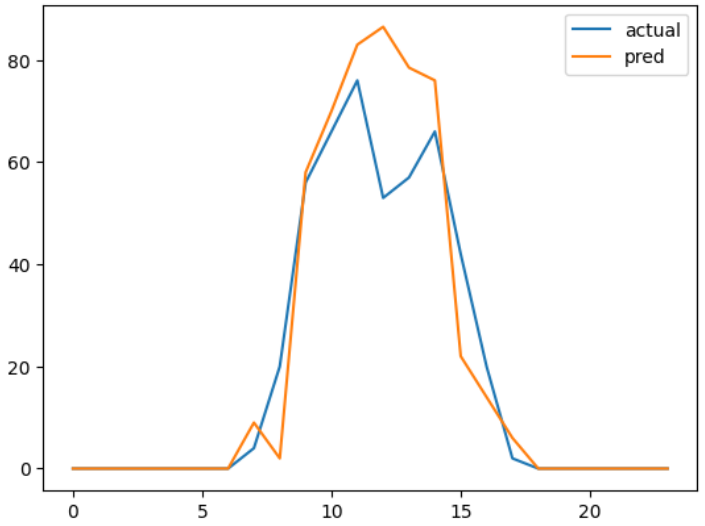
Overall prediction accuracy dropped in November

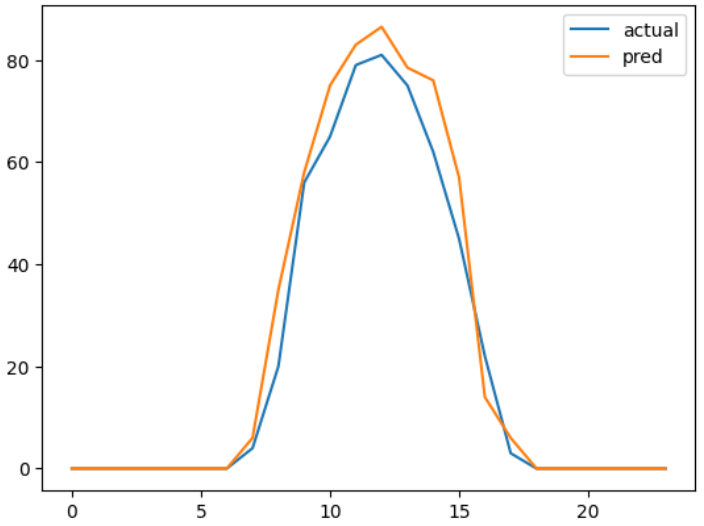
due to frequent cloudy weather

Modeling

Vector similarity – Cosine distance



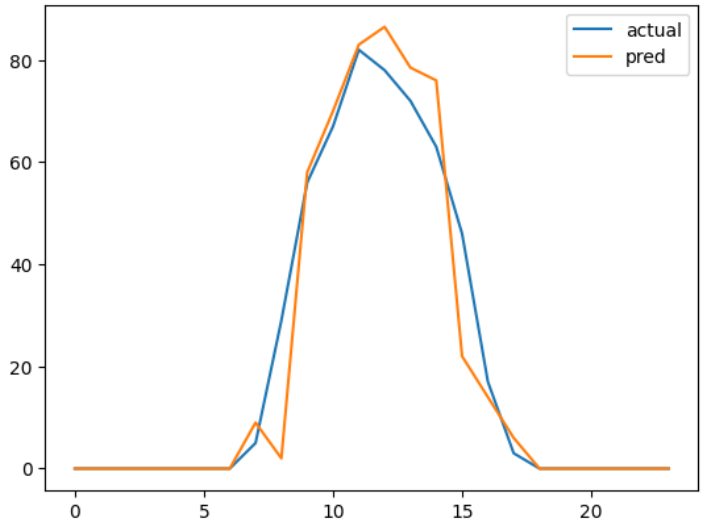


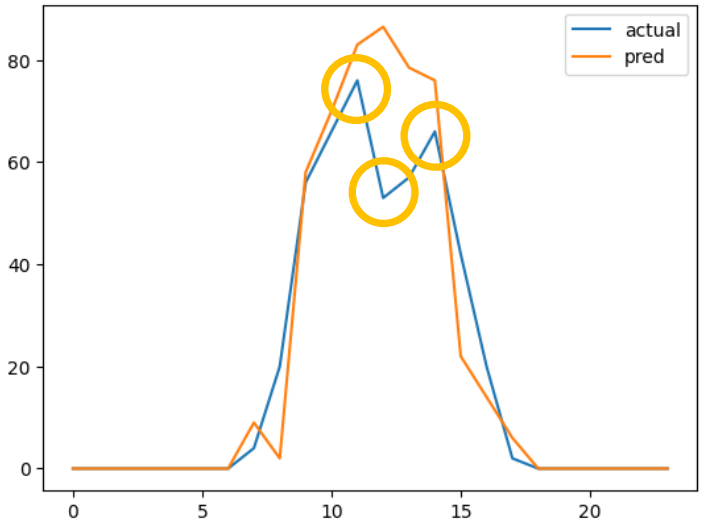


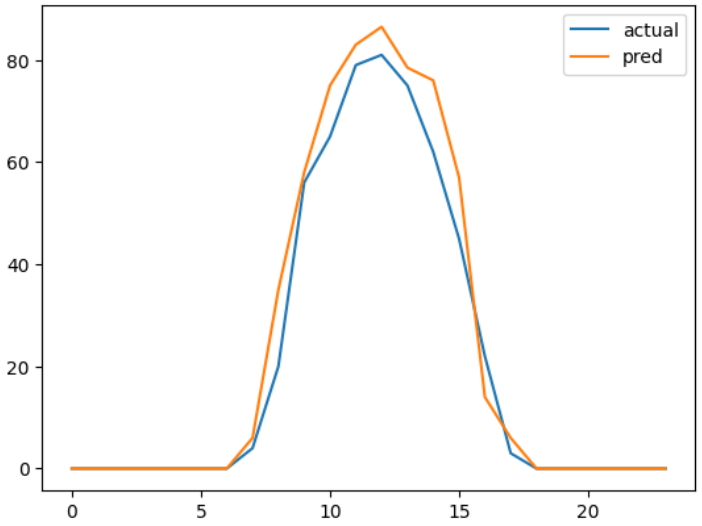
model	data	Total incentive(10.25~10.31)	notes
Cosine Similarity	Weather data(m0~m4)	4964	Train data : Oct ~ Nov

Modeling

Vector similarity – Cosine distance





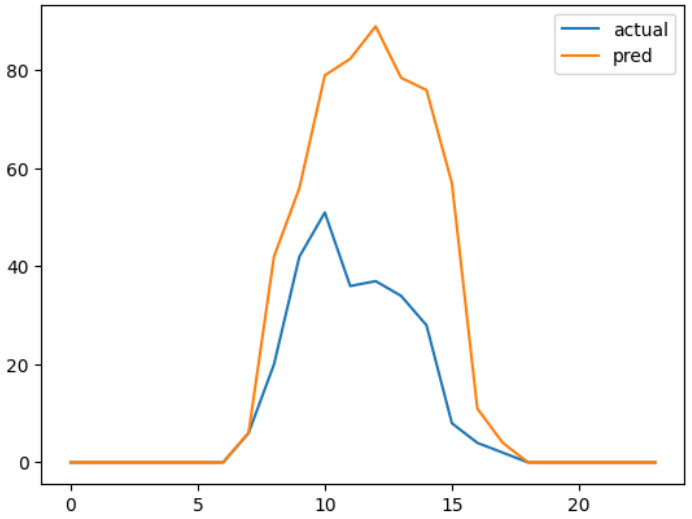


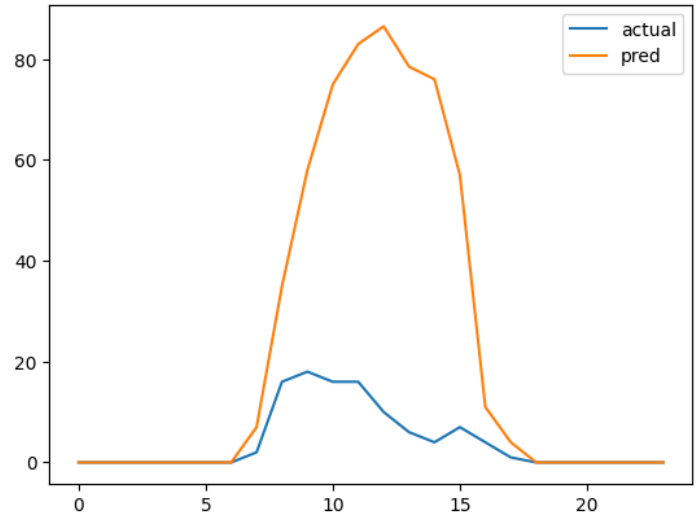
Generally has lower performance compared to Euclidean

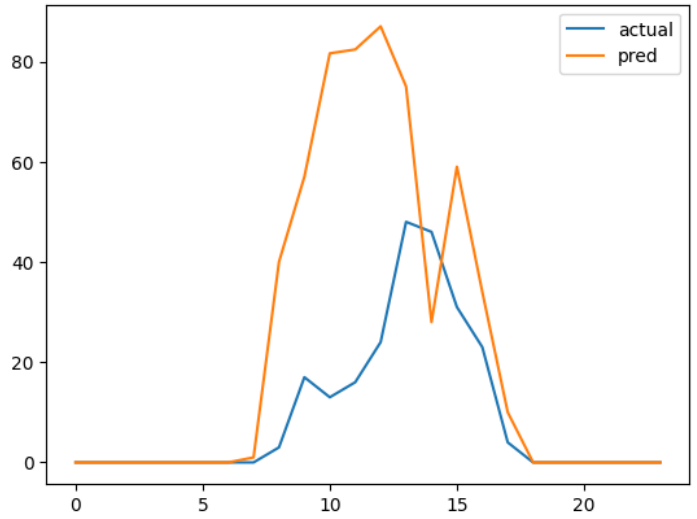
Poor prediction for spike points

Modeling

Vector similarity – Cosine distance



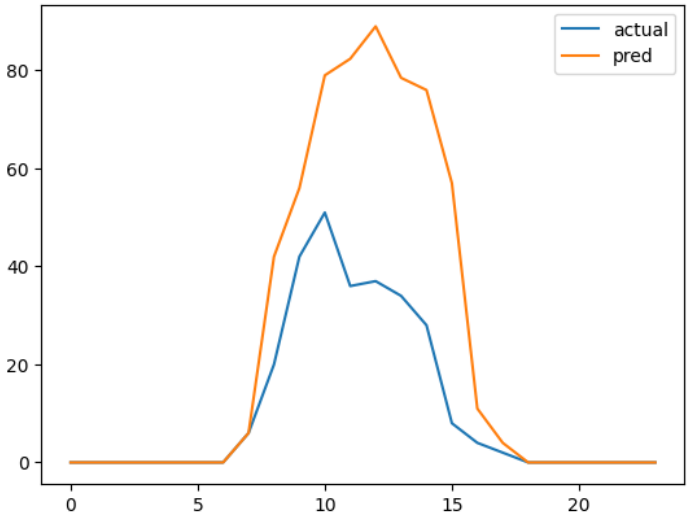


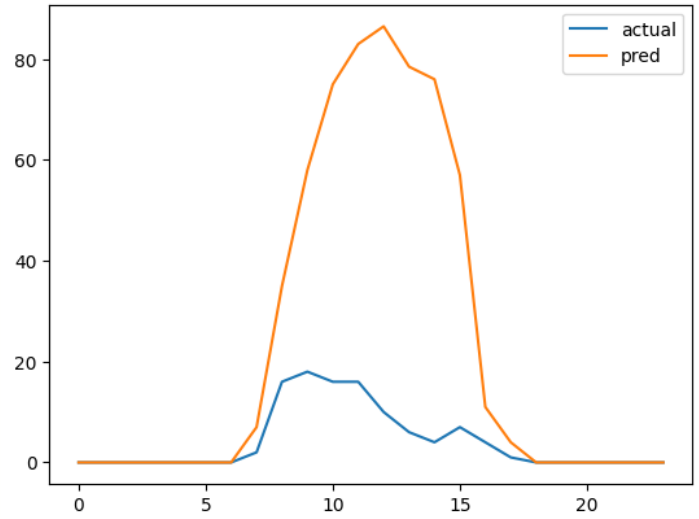


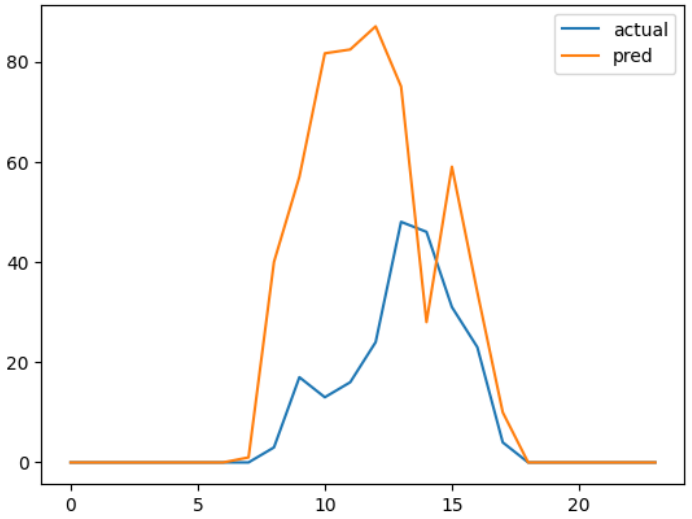
model	data	Total incentive(11.01~11.07)	notes
Cosine Similarity	Weather data(m0~m4)	1065	Train data : Oct ~ Nov

Modeling

Vector similarity – Derived variable





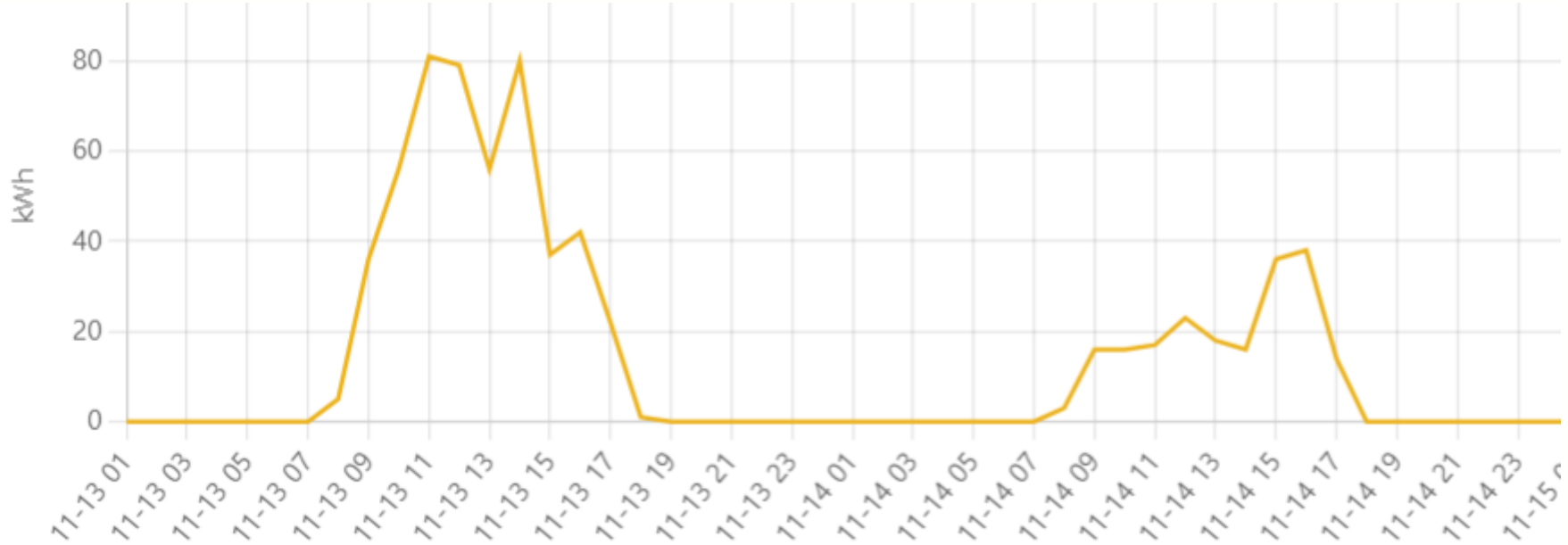


Predicting based only on similarity results in too large error

Decided to use it as a new variables

Modeling

Vector similarity – Derived variable

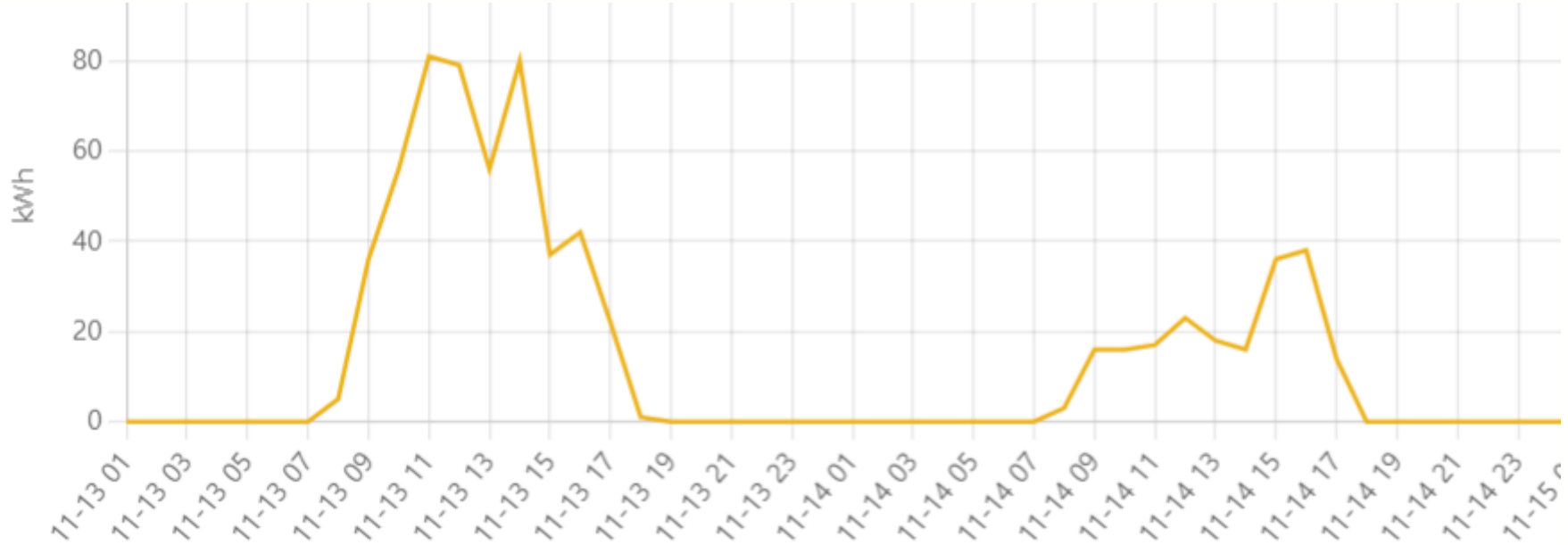


The generation patterns in the first two days of the competition were highly unusual

Performance significantly decreased in conditions of high cloud, rain, or cold weather

Modeling

Vector similarity – Derived variable

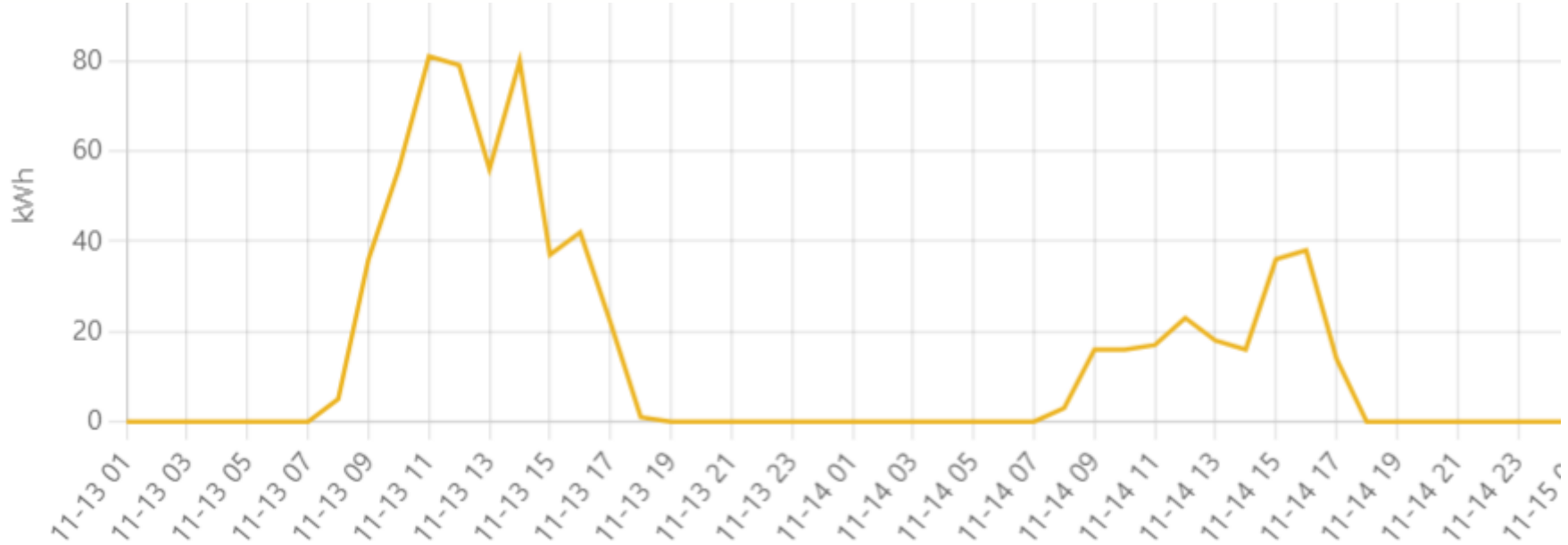


Model performance was validated using data

from mid-October to early November

Modeling

Vector similarity – Derived variable

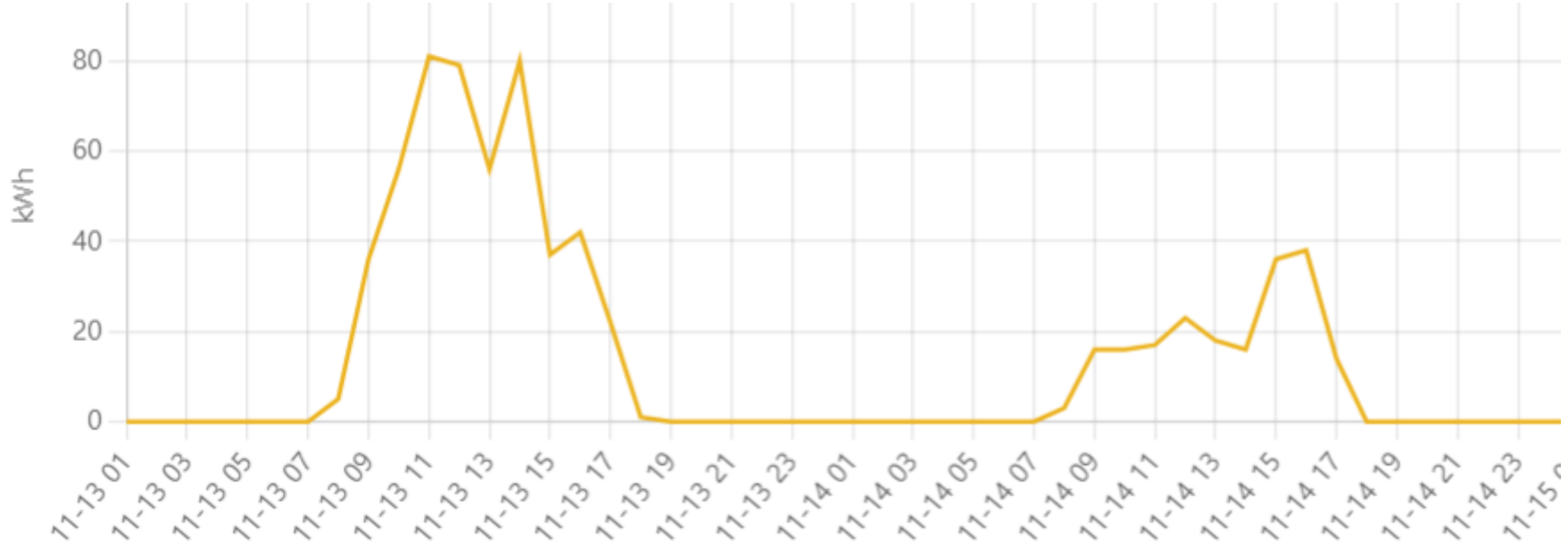


However, during the competition period in mid-November,

the sudden change in weather likely caused drop in performance

Modeling

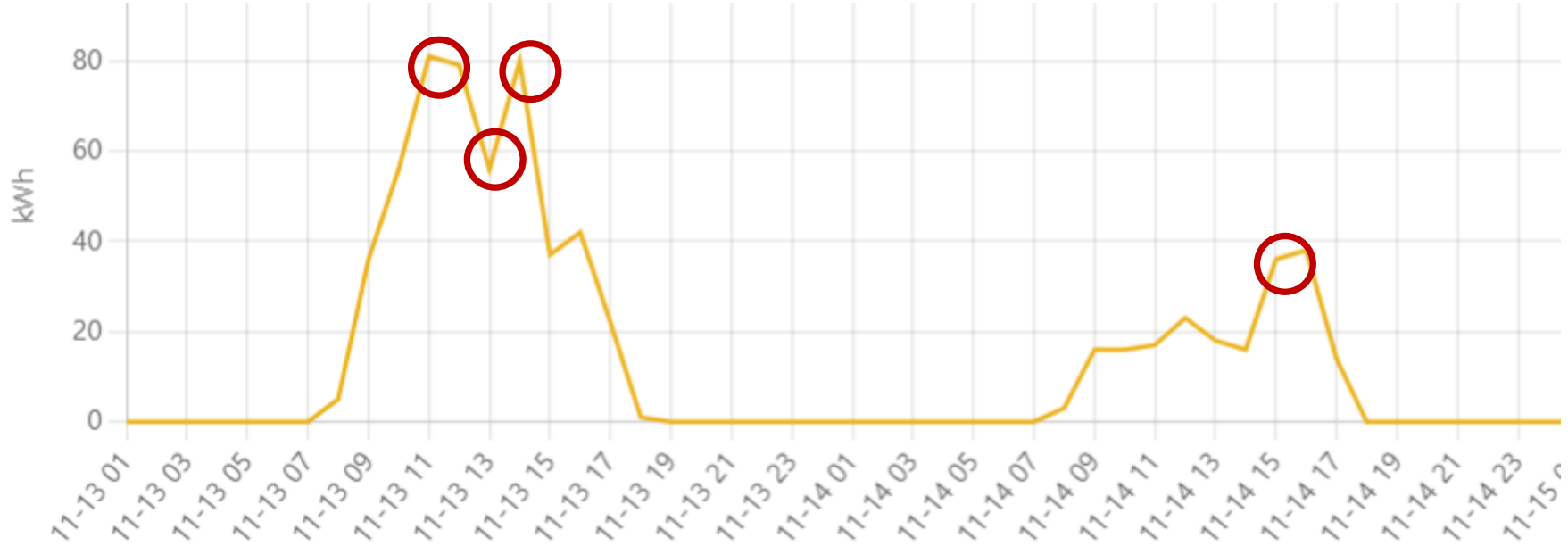
Vector similarity – Derived variable



Decided to add a new variable to reflect the colder weather

Modeling

Vector similarity – Derived variable



Spike points can be considered as outliers,

but since these values will certainly exist in past data

-> calculate the similarity and add it as a variable.

Modeling

Vector similarity – Derived variable

```
# xy_df1 에 유사도 발전량 추가
make_train1 = xy_df1[["m0","m1","m2","m3","m4","elevation","uv_idx","amount"]]
make_train2 = xy_df1[["m0","m1","m2","m3","m4","elevation","uv_idx","humidity","amount","vis","dew_point","ground_press"]]
feature = make_train1.drop(["amount"],axis=1)
cosine_sim = cosine_similarity(feature,feature)
np.fill_diagonal(cosine_sim, 0)
most_similar_idx = cosine_sim.argmax(axis=1)
for i in range(len(xy_df1)):
    idx = most_similar_idx[i]
    xy_df1.loc[i,"amount1"] = xy_df1.at[idx,"amount"]

feature = make_train2.drop(["amount"],axis=1)
cosine_sim = cosine_similarity(feature,feature)
np.fill_diagonal(cosine_sim, 0)
most_similar_idx = cosine_sim.argmax(axis=1)
for i in range(len(xy_df1)):
    idx = most_similar_idx[i]
    xy_df1.loc[i,"amount2"] = xy_df1.at[idx,"amount"]
```

Cosine similarity relates to the angle between two vectors

-> Calculate similarity using variables correlated with power generation

Modeling

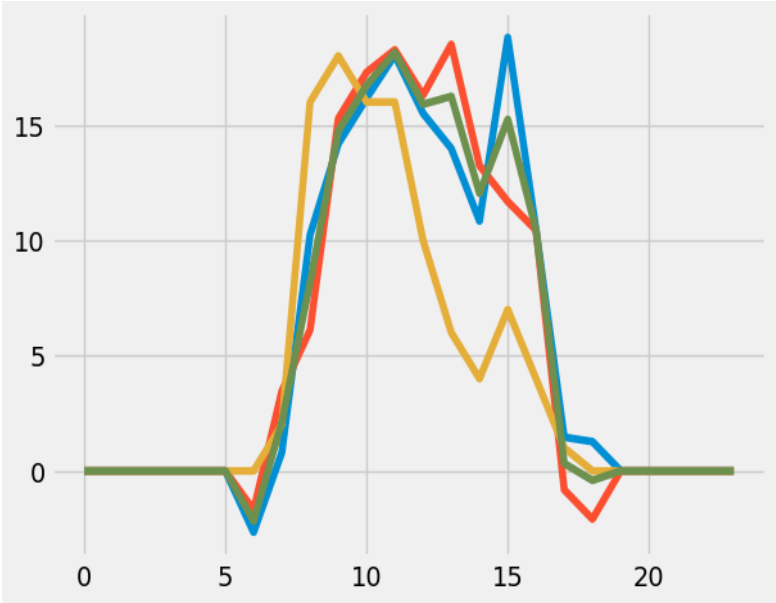
Vector similarity – Derived variable

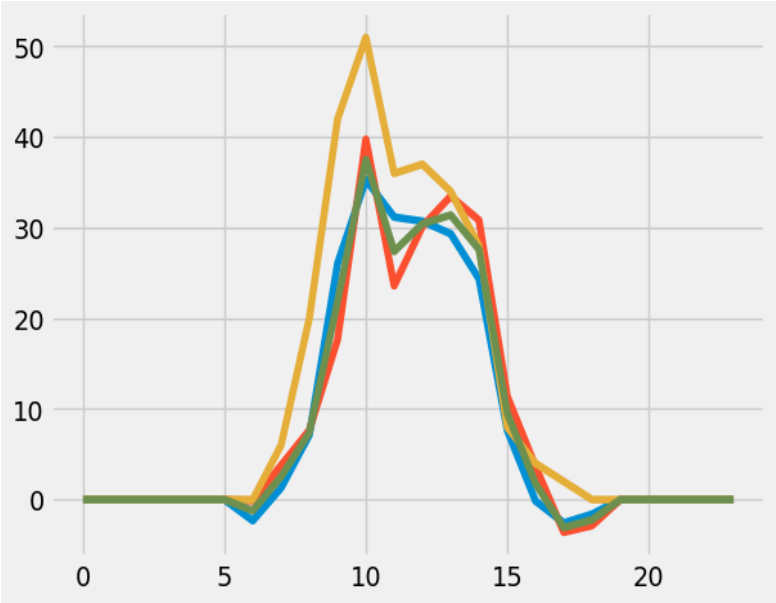
```
# xy_df1 에 유사도 발전량 추가
make_train1 = xy_df1[["m0","m1","m2","m3","m4","elevation","uv_idx","amount"]]
make_train2 = xy_df1[["m0","m1","m2","m3","m4","elevation","uv_idx","humidity","amount","vis","dew_point","ground_press"]]
feature = make_train1.drop(["amount"],axis=1)
cosine_sim = cosine_similarity(feature,feature)
np.fill_diagonal(cosine_sim, 0)
most_similar_idx = cosine_sim.argmax(axis=1)
for i in range(len(xy_df1)):
    idx = most_similar_idx[i]
    xy_df1.loc[i,"amount1"] = xy_df1.at[idx,"amount"]

feature = make_train2.drop(["amount"],axis=1)
cosine_sim = cosine_similarity(feature,feature)
np.fill_diagonal(cosine_sim, 0)
most_similar_idx = cosine_sim.argmax(axis=1)
for i in range(len(xy_df1)):
    idx = most_similar_idx[i]
    xy_df1.loc[i,"amount2"] = xy_df1.at[idx,"amount"]
```

Using only October and November data resulted in generally lower error rates compared to using data from all seasons

Modeling

Vector similarity + regression + lgbm





yellow actual

red R1

blue R2

green avg(R1,R2)

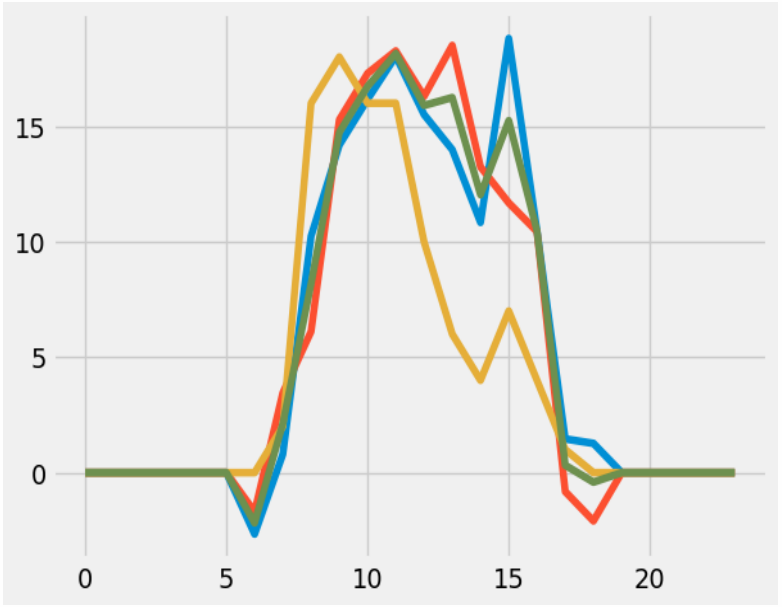
The model seemed to follow the spike patterns

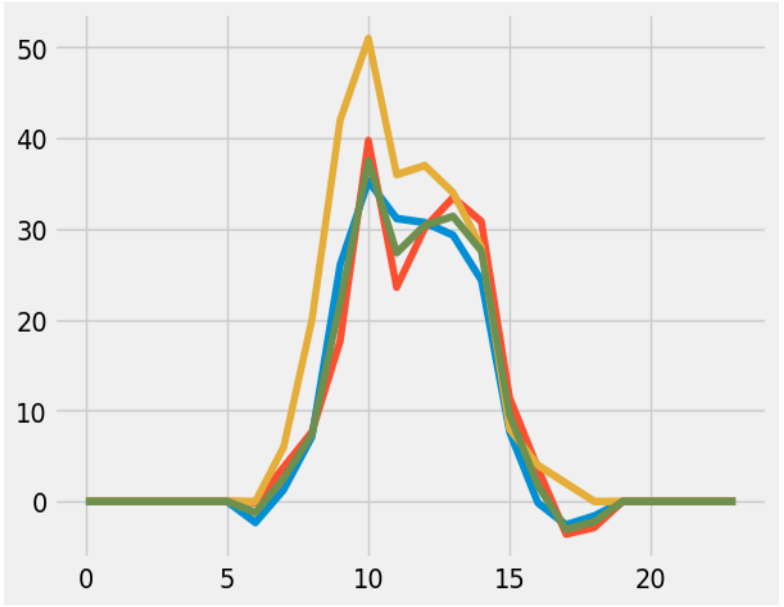
Total incentive were high on days with unusual power generation patterns

Final model

Final model

Vector similarity + regression + lgbm

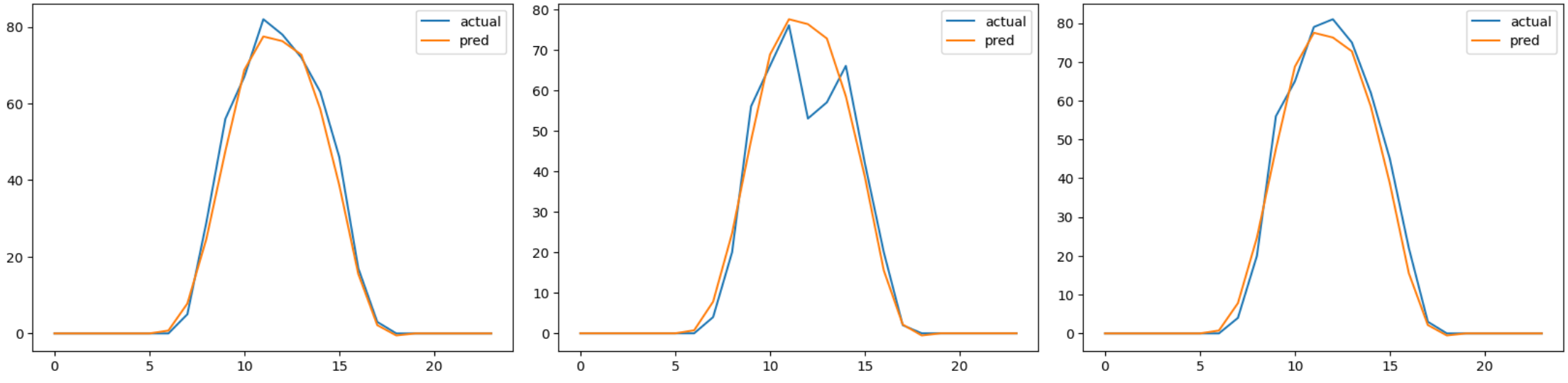




On days with heavy cloud cover until a certain time, total day power generation is very low, in that case this model is used for submission

Final model

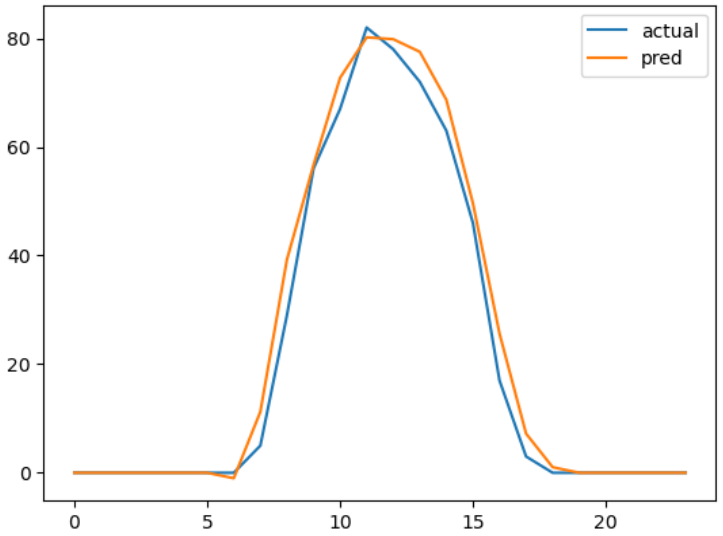
Linear regression

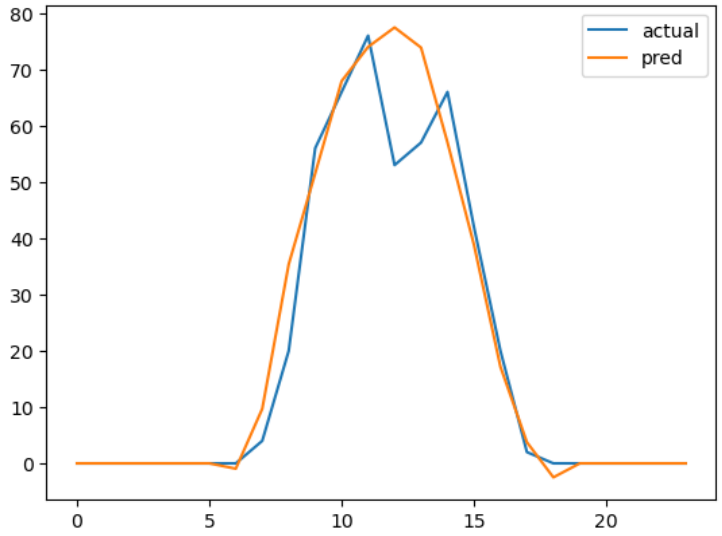


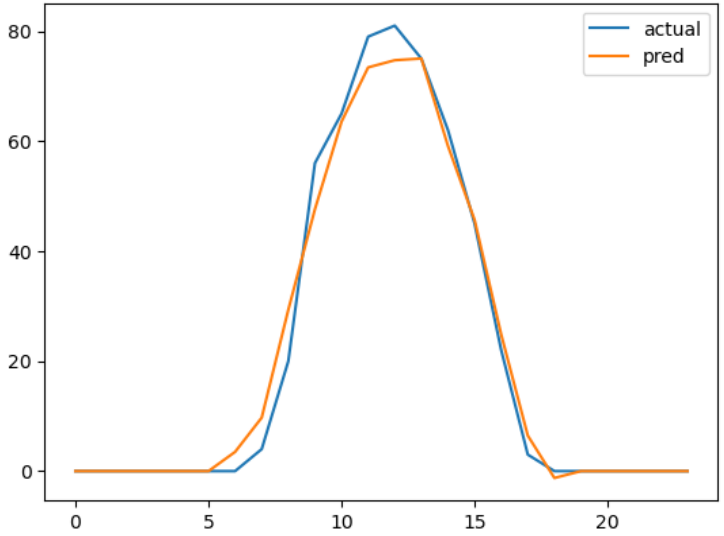
Submit the regression model on days with cloud cover between 90-100 throughout the day

Final model

MLP + LGBM







Submit MLP + LGBM, which generally shows good performance, in typical cases


Thank you
This was team deaplearning~!
I am Queen
울면안대
쫄면안대
밀면안대