Predict Power Consumption of Tetouan City:

주건재

OVERVIEW

1. INTRODUCTION
2. EDA
3. MODEL
4. RESULT

1. INTRODUCTION

1.1 TOPIC

Tetouan City:

국가: 모로코 (지중해 인근)
인구: 약 38만명
종교: 이슬람
기온: 연평균 18.4도 (월평균 13~25도)
강수: 연평균 약75mm (11월~3월 집중)

Research Question

1~3 구역의 전력소모를 예측하여 대비한다.

Goals

1st goal: 실시간 모델 모형을 사용

2nd goal: 1~3구역의 전력소모 예측 모델링

1.2 DATA OVERVIEW

Raw Data

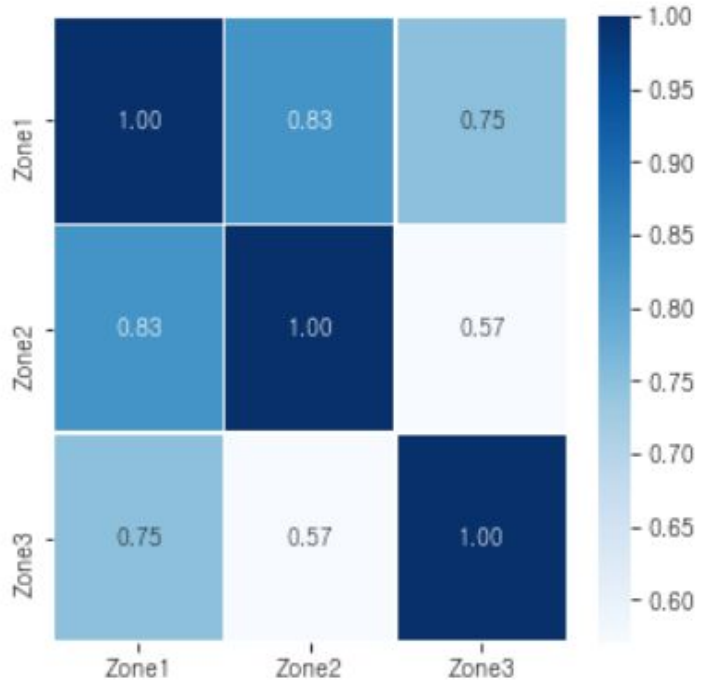
Data Resource: uci repository (https://archive.ics.uci.edu/ml/datasets/Power+consumption+of+Tetouan+city)

	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
0	1/1/2017 0:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.87538	20240.96386
1	1/1/2017 0:10	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.07599	20131.08434
2	1/1/2017 0:20	6.313	74.5	0.080	0.062	0.100	29128.10127	19006.68693	19668.43373

Data Description

샘플수 52417개
(1년)

변수명	데이터 타입	사용여부
Date Time (each 10mins)	시계열	독립변수
Temperature	연속형	독립변수
Humidity	연속형	독립변수
Wind Speed	연속형	독립변수
General Diffuse flows	연속형	독립변수
Diffuse flows	연속형	독립변수
power consumption zone1	연속형	종속변수
power consumption zone2	연속형	종속변수
power consumption zone3	연속형	종속변수



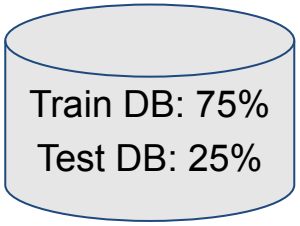
<종속변수 피어슨 상관관계>

1.3 Analysis Overview

Data Understanding

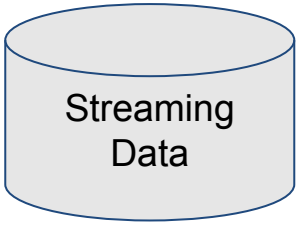
Research Goal:
전력소모 예측

Offline Learning



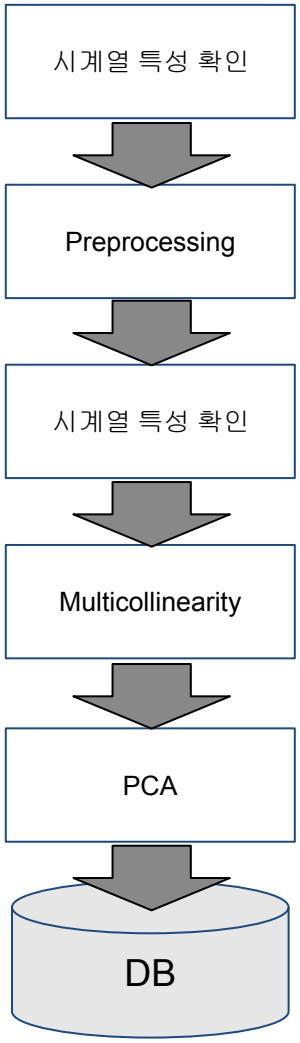
특징:
- shape: (52416,9)
- 시계열 data

Online Learning



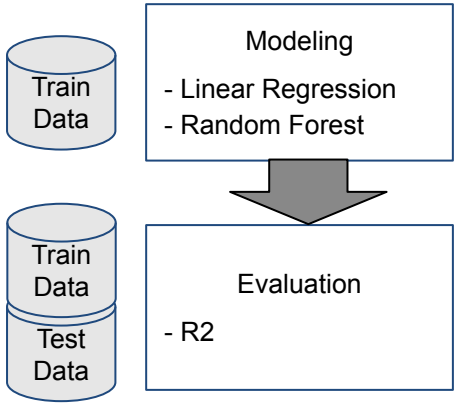
특징:
- shape: (52416,9)
- 시계열 data
- Streaming data

Preprocessing & EDA

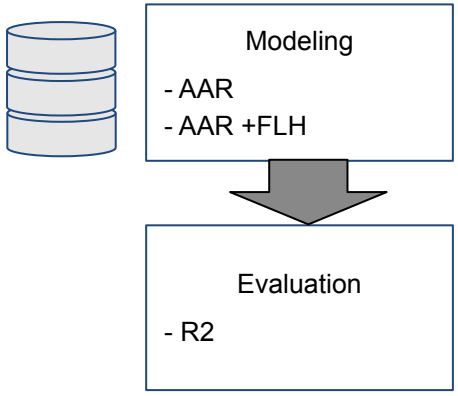


Modeling & Evaluation

Offline Learning



Online Learning



2. EDA

2.1 시계열

ACF (Autocorrelation Function)

- y_t와 y_(t+k) 와의 상관관계를 측정한것
- k 는 lag값
- 1에 가까울 수록 상관관계가 큼

$$ACF(k) = \frac{cov(y_t, y_{t+k})}{var(y_t)} = \frac{\sum_{t=1}^{N-k}(y_t - \overline{y})(y_{t+k} - \overline{y})}{\sum_{t=1}^{N}(y_t - \overline{y})^2} \frac{N}{N-k}$$

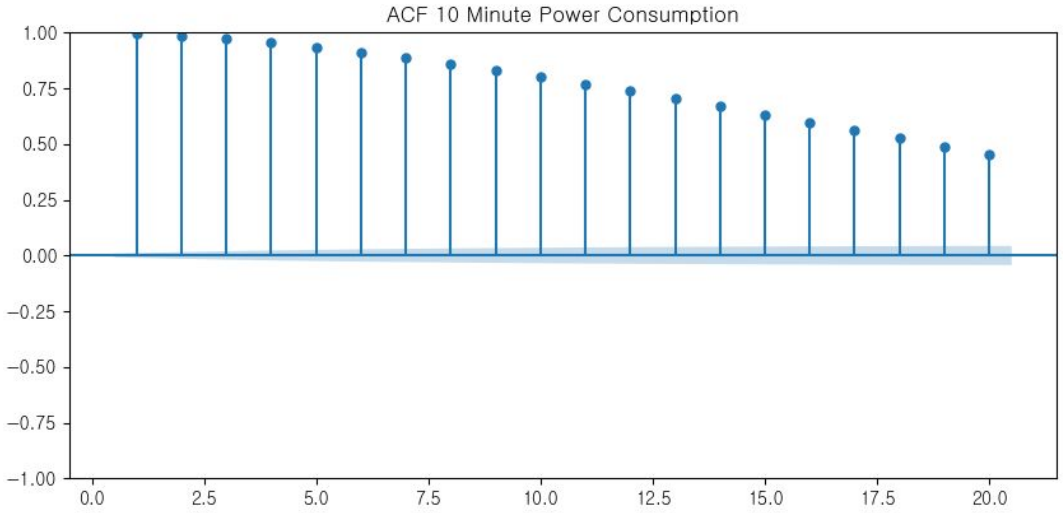
PACF (Partial Autocorrelation Function)

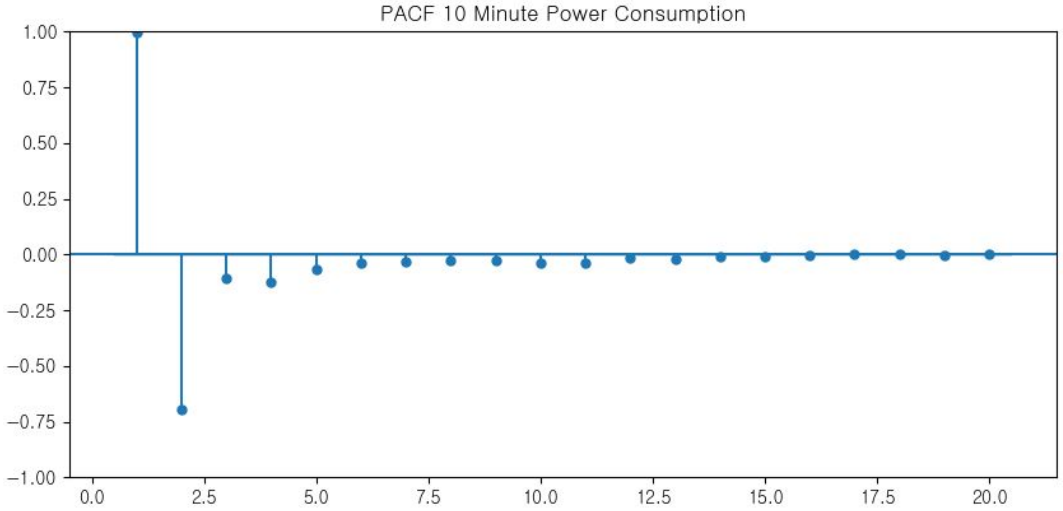
- 온전히 y_t와 y_(t-k) 만의 상관관계를 측정한것
- ACF 그래프와 더불어 사용할 시계열 모형을 특정하기 위해 주로 사용

$$PACF(k) = Corr(e_t, e_{t-k})$$

$$e_t = Y_t - (\beta_1 Y_{t-1} + \cdots + \beta_1 Y_{t-(k-1)})$$

2.1 시계열


ACF 10 Minute Power Consumption
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


PACF 10 Minute Power Consumption
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

2.1 시계열

ACFs	PACFs	Model
Decay to zero with exponential pattern	Cuts off after lag p	AR(p)
Cuts off after lag q	Decay to zero with exponential pattern	MA(q)
Decay to zero with exponential pattern	Decay to zero with exponential pattern	ARMA(p, q)

Daily Based Aggregation

- min, max, mean value of the day
- sum of power consumption

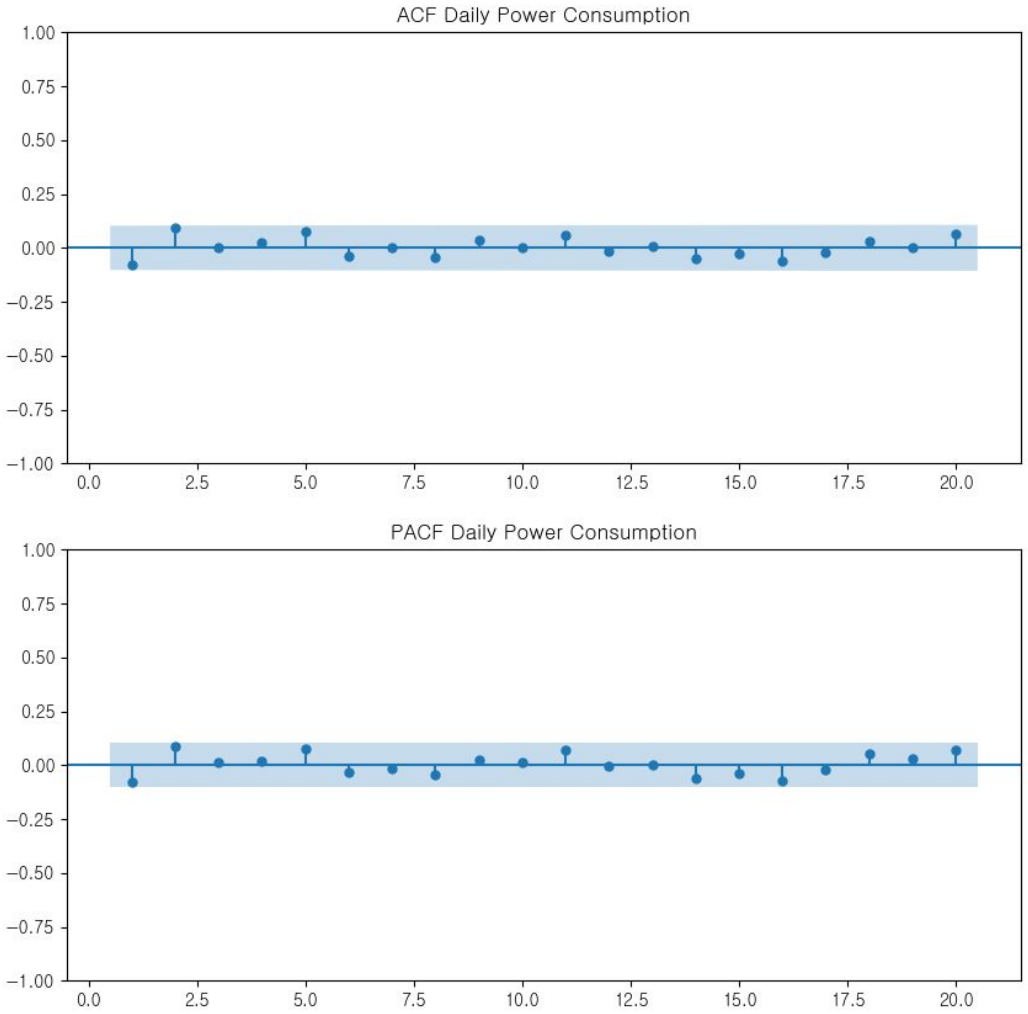
2.2 Data Aggregation

Aggregated Data

shape (364,17)

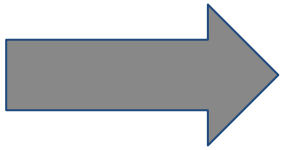
date	maxTemp	minTemp	meanTemp	maxHum	...	meanDiff	powerConsumption
3/19/2017	14.59	9.74	11.993889	90.80	...	37.473132	9.078
8/10/2017	35.89	25.29	29.543889	66.56	...	89.199465	13.045
1/12/2017	17.61	9.06	13.895556	83.90	...	70.149000	9.997

2.2 Data Aggregation



2.3 Multicollinearity

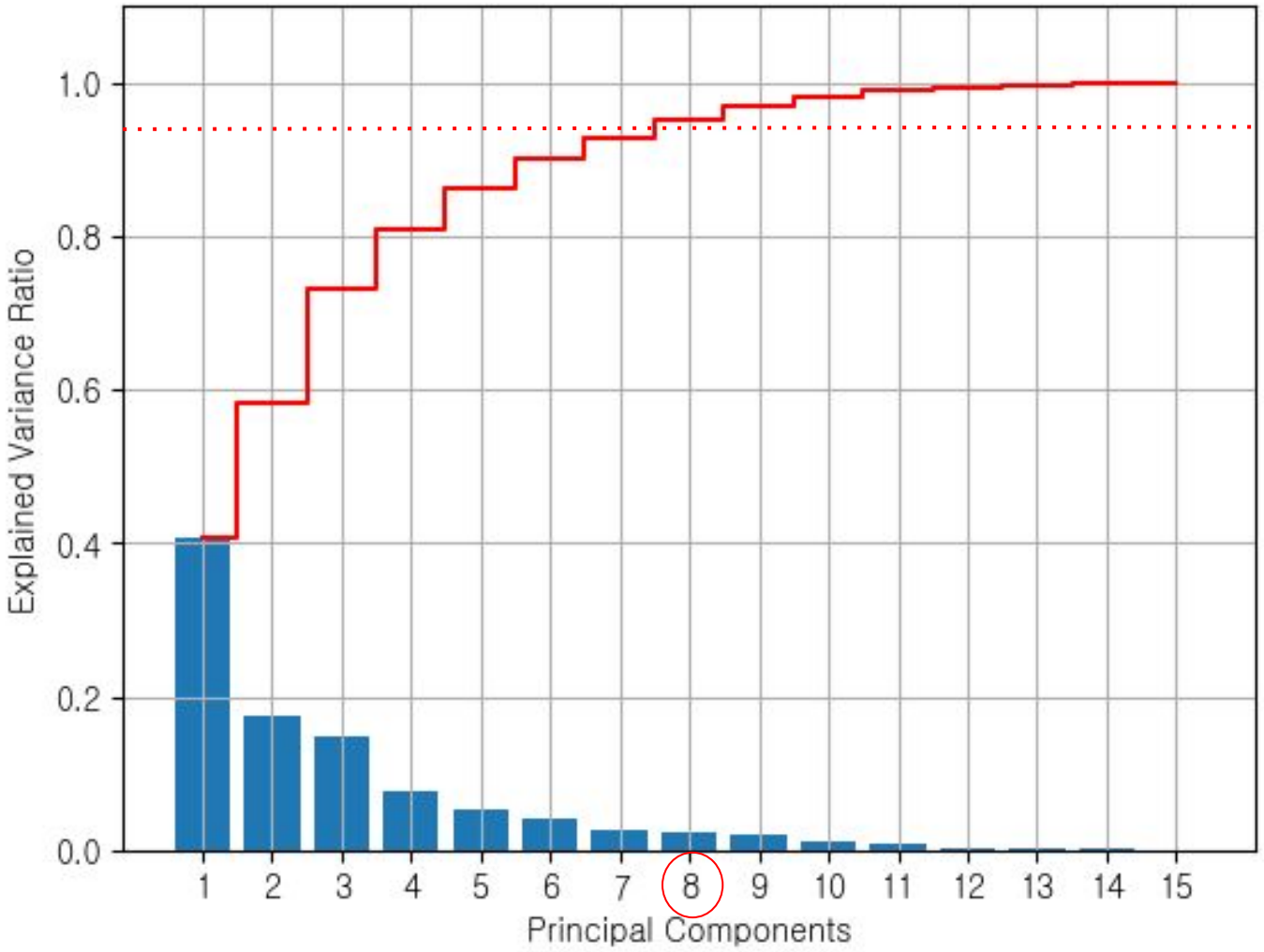
	Attribute	VIF Scores
0	maxTemp	872.75
1	minTemp	434.90
2	meanTemp	2178.60
3	maxHum	560.19
4	minHum	162.14
5	meanHum	814.74
6	maxWind	11.49
7	mimWind	7.99
8	meanWind	23.23
9	maxGenDiff	50.57
10	minGenDiff	11.89
11	meanGenDiff	24.57
12	manDiff	24.35
13	minDiff	19.29
14	meanDiff	17.74



한 개씩 반복제거

	Attribute	VIF Scores
0	minHum	5.01
1	maxWind	3.82
2	mimWind	2.95
3	minGenDiff	6.88
4	meanGenDiff	7.80
5	meanDiff	5.82

2.4 PCA



variance = 95%

3. MODEL

3. Model

Online Learning:

- 데이터를 샘플 한개씩 스트림하게 훈련하는 모델
- 예측 대상의 x데이터까지 훈련하여 예측

ex) 배달가능여부, 추천시스템

AAR(Aggregating Algorithm for Regression):

It is a modified ridge regression (RR) considering online context instead of batch analysis. For RR, the optimal solution is well known as:

$$\mathbf{w}_* = (\sum_{j=1}^{t} \mathbf{x}_j \mathbf{x}_j^T + \gamma \mathbb{I})^{-1} \sum_{j=1}^{t} y_j \mathbf{x}_j$$

Let's say

$$\mathbf{A} = \sum_{j=1}^{t} \mathbf{x}_j \mathbf{x}_j^T + \gamma \mathbb{I}$$

$$\mathbf{b} = \sum_{j=1}^{t} y_j \mathbf{x}_j$$

For a new data point $\mathbf{x}_{m+1}$, RR predicts $\hat{y}_{m+1}$ as

$$\hat{y}_{t+1} = \mathbf{w}_*^T \mathbf{x}_{t+1} = \mathbf{b}^T \mathbf{A}^{-1} \mathbf{x}_{t+1}$$

You can keep updating $\mathbf{A}$ and $\mathbf{b}$ then predicting $\hat{y}_{t+1}$ as described below:

3. Model

Online Learning:

- 데이터를 샘플 한개씩 스트림하게 훈련하는 모델
- 예측 대상의 x데이터까지 훈련하여 예측

ex) 배달가능여부, 추천시스템

Ridge 와 AAR(Aggregating Algorithm for Regression)의 차이:

Algorithm 1 Ridge Regression

Input:

$\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$: m input vectors

$\{y_1, \ldots, y_m\}$: m targets

Output:

$\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)\}$: model predictions

1: Initialize $\mathbf{A} = \gamma\mathbb{I}$ and $\mathbf{b} = 0$
2: **for** $t = 1$ to m **do**
3: read new $\mathbf{x}_t$
4: output prediction $f(\mathbf{x}_t) = \mathbf{b}^\intercal \mathbf{A}^{-1}\mathbf{x}_t$
5: $\mathbf{A} = \mathbf{A} + \mathbf{x}_t\mathbf{x}_t^\intercal$
6: Read new y_t
7: $\mathbf{b} = \mathbf{b} + y_t\mathbf{x}_t$
8: **end for**

<표8> Ridge Regression Pseudo Code

Algorithm 2 *The aggregating algorithm for regression*

Input:

$\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$: m input vectors

$\{y_1, \ldots, y_m\}$: m targets

Output:

$\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)\}$: model predictions

1: Initialize $\mathbf{A} = \gamma\mathbb{I}$ and $\mathbf{b} = 0$
2: **for** $t = 1$ to m **do**
3: read new $\mathbf{x}_t$
4: $\mathbf{A} = \mathbf{A} + \mathbf{x}_t\mathbf{x}_t^\intercal$
5: output prediction $f(\mathbf{x}_t) = \mathbf{b}^\intercal \mathbf{A}^{-1}\mathbf{x}_t$
6: Read new y_t
7: $\mathbf{b} = \mathbf{b} + y_t\mathbf{x}_t$
8: **end for**

<표9> ARR Pseudo Code

3. Model

Ridge Regression:

train/ test set을 나누어 train set을학습 후 train과 test에 대한 예측

Train

Test

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19
1	0.59	-0.36	-1.70	-0.84	-0.04
2	1.33	0.68	-1.22	0.30	-0.87
3	2.15	1.83	-0.69	1.57	-1.79
4	0.03	-1.03	-1.91	-1.58	0.50
5	0.09	-0.92	-1.84	-1.46	0.42
6	0.82	0.10	-1.37	-0.33	-0.40
7	0.46	-0.35	-1.54	-0.83	-0.04
8	0.39	-0.42	-1.54	-0.91	0.02
9	1.31	0.86	-0.96	0.51	-1.00
10	0.19	-0.64	-1.58	-1.14	0.19

독립변수 (sensor data)

3. Model

AAR(Aggregating Algorithm for Regression):

데이터를 Streaming으로 하나씩 학습 및 예측을 동시에 진행

train/ test 셋으로 나누지 않음

STEP 1-1

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	

receive x_1
update A
predict y_1

STEP 1-2

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19

receive y_1
update B

STEP 2-1

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19
1	0.59	-0.36	-1.70	-0.84	

receive x_2
update A
predict y_2

3. Model

FLH(Follow the Leading History):

목적: data drift/shift (장비 노후등으로 인한 관측치 변화) 대비

방법: 학습모델(Expert)을 데이터가 들어오는 매번 새로 만들어
학습시작의 시점을 다르게 한 후, 모델들에 가중치를 주고 앙상블하여 사용한다.

프로세스

1) 데이터가 하나 들어올때마다 새로운 모델을 하나 생성한다.
2) 들어온 하나의 데이터에 대해 생성된 모든 모델들을 학습한다.
3) 모델들의 가중치인 v를 업데이트 한다.

Algorithm Follow-the-Leading-History (FLH)

Input: online regression algorithm E (called expert), hyperparameter $\alpha > 0$

$v_1^{(1)} = 1$ ▷ $v_t = (v_t^{(1)}, \ldots, v_t^{(t)})$ is a probability vector for each t

for $t = 1, \ldots, T$ **do**

Get a new instance $E^{(t)}$ of algorithm E

After receiving x_t, update $E^{(j)}$ for $j = 1, \ldots, t$

Set $\hat{y}_t^{(j)} = E^{(j)}(x_t)$ for $j = 1, \ldots, t$ ▷ predictions of the experts

Predict $\hat{y}_t = \sum_{j=1}^{t} v_t^{(j)} \hat{y}_t^{(j)}$ ▷ convex combination of the predictions

After receiving y_t, update $E^{(j)}$ for $j = 1, \ldots, t$

Set $\hat{v}_{t+1}^{(t+1)} = 0$ and update for $i = 1, \ldots, t$

$$\hat{v}_{t+1}^{(i)} = \frac{v_t^{(i)} \exp(-\alpha(y_t - \hat{y}_t^{(i)})^2)}{\sum_{j=1}^{t} v_t^{(j)} \exp(-\alpha(y_t - \hat{y}_t^{(j)})^2)}$$

Set $v_{t+1}^{(t+1)}$ to $1/(t+1)$ and update for $i = 1, \ldots, t$

$$v_{t+1}^{(i)} = \left(1 - \frac{1}{t+1}\right) \hat{v}_{t+1}^{(i)}$$

end for

<그림3> FLH Pseudo Code

3. Model

FLH(Follow the Leading History):

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19
1	0.59	-0.36	-1.70	-0.84	-0.04
2	1.33	0.68	-1.22	0.30	-0.87

receive data x_3

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19
1	0.59	-0.36	-1.70	-0.84	-0.04
2	1.33	0.68	-1.22	0.30	-0.87
3	2.15	1.83	-0.69	1.57	

receive data y_3

	Sensor_1	Sensor_2	Sensor_3	Sensor_4	Measurement
0	1.67	1.08	-1.10	0.75	-1.19
1	0.59	-0.36	-1.70	-0.84	-0.04
2	1.33	0.68	-1.22	0.30	-0.87
3	2.15	1.83	-0.69	1.57	-1.79

3 experts (model)
e_0: trained sample 0~2
e_1: trained sample 1~2
e_2: trained sample 2

create e_3
update A of all experts then predict y_3

4 experts (model)
e_0: trained sample 0~2+x_3 → prediction y_3_e_1
e_1: trained sample 1~2+x_3 → prediction y_3_e_2
e_2: trained sample 2~2+x_3 → prediction y_3_e_3
e_3: trained sample x_3 → prediction y_3_e_4

y_pred_4 = (v0 * y_3_e_0)+...+(v3 * y_3_e_3)

update create v4 and update v1~3
update B of all experts

4 experts (model)
e_0: trained sample 0~3
e_1: trained sample 1~3
e_2: trained sample 2~3
e_3: trained sample 3

4. RESULT

4.1 OLS

OLS Regression Results

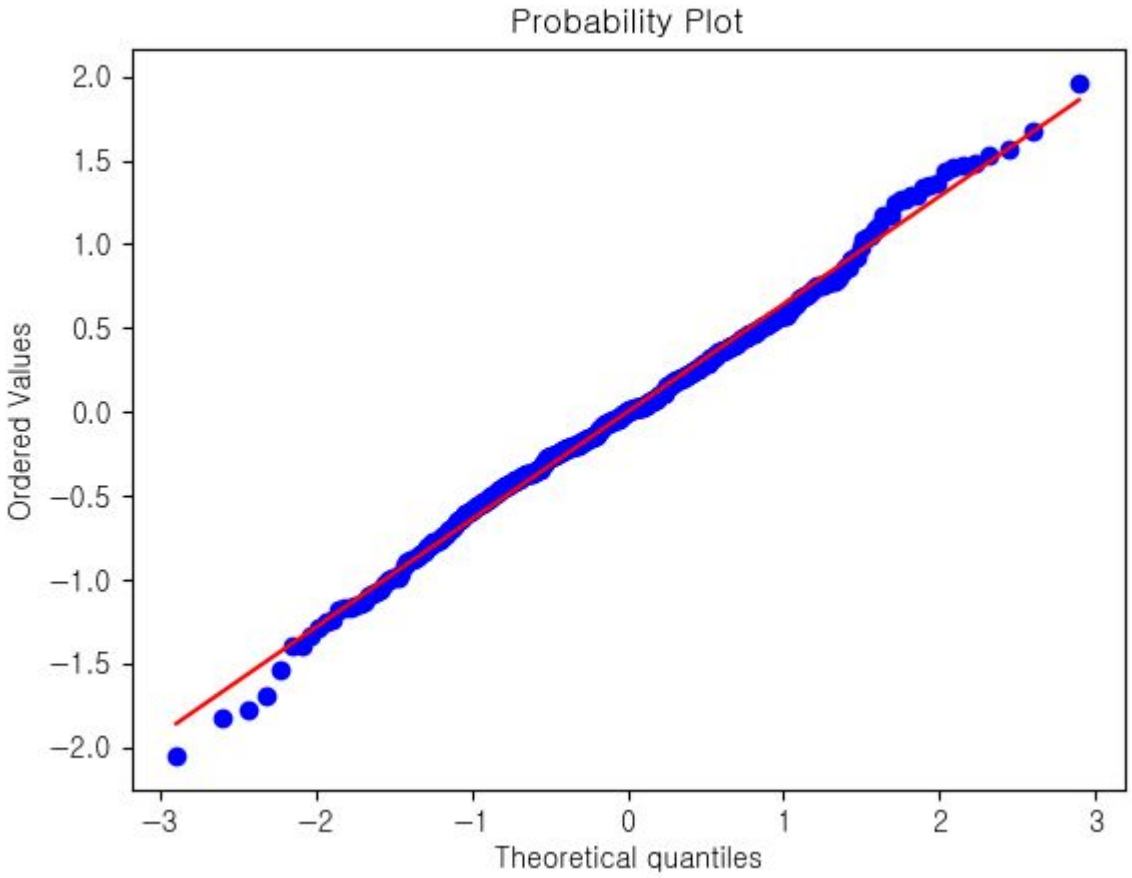
Dep. Variable:	('powerConsumption',)	R-squared (uncentered):	0.594
Model:	OLS	Adj. R-squared (uncentered):	0.585
Method:	Least Squares	F-statistic:	65.07
Date:	Tue, 31 May 2022	Prob (F-statistic):	4.53e-65
Time:	12:17:56	Log-Likelihood:	-352.50
No. Observations:	364	AIC:	721.0
Df Residuals:	356	BIC:	752.2
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
component1	0.2892	0.014	21.159	0.000	0.262	0.316
component2	-0.0217	0.021	-1.042	0.298	-0.063	0.019
component3	-0.1157	0.023	-5.122	0.000	-0.160	-0.071
component4	-0.1384	0.031	-4.428	0.000	-0.200	-0.077
component5	-0.1532	0.038	-3.995	0.000	-0.229	-0.078
component6	0.1195	0.043	2.776	0.006	0.035	0.204
component7	0.0708	0.053	1.332	0.184	-0.034	0.175
component8	-0.0410	0.058	-0.708	0.479	-0.155	0.073

Omnibus:	3.552	Durbin-Watson:	2.182
Prob(Omnibus):	0.169	Jarque-Bera (JB):	4.030

4.1 Linear Regression

Component 2, 7, 8 제거 후 QQ-plot



4.2 Models

Result

(Offline learning)	Train Score	Test Score
Ride Regression	0.590	0.587
Random Forest	0.903	0.709
(Online learning)	Score	
AAR	0.532	
AAR + FLH	0.551	

Research Result

- **Further Study**
  - Online Random Forest 모델 구현
  - mini batch offline learning 이후 online learning으로 전환

Q&A