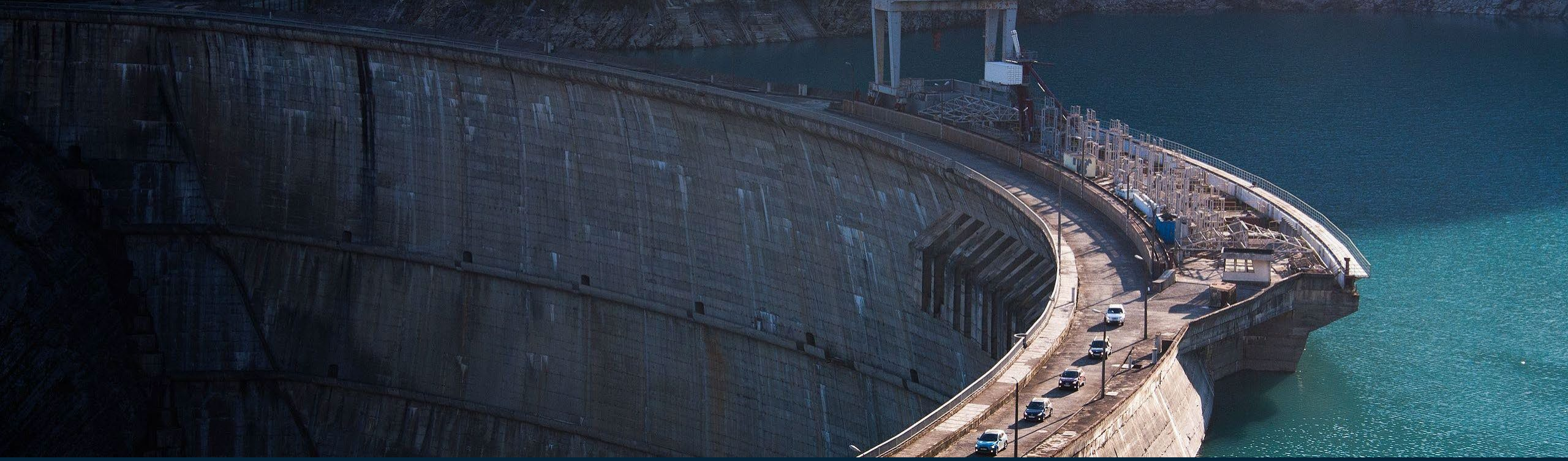
2021 빅콘테스트 :: 홍수ZERO 부문 :: 실험 설계 및 결과 도출

<홍수빅타>
팀장 : 오창준 (statco19@gmail.com)
팀원 1 : 권다인 (creamsoffle@yonsei.ac.kr)
팀원 2 : 김상회 (why750@yonsei.ac.kr)
팀원 3 : 최현준 (hyunjoonc8@naver.com)



Contents

1. 데이터 설명 및 EDA

1) 데이터 설명
2) EDA

데이터 설명 및 EDA

1) 주어진 데이터 설명 (EDA/hongsuEDA.ipynb)

홍수ZERO	댐 유입 수량 예측을 통한 최적의 수량 예측 모형 도출 ▪ 환경 빅데이터 플랫폼에서 제공하는 댐 수위 데이터를 활용하여 댐에 유입되는 수량을 예측하고, 최적의 댐 유입 수량 예측 모형 제시

그림1. 프로젝트 설명

01_제공 데이터와 02_평가 데이터로 구성

각 데이터는 홍수사상번호, 시간 데이터(연/월/일/시간), 유역 평균 강수, 강우 데이터(A, B, C, D), 수위 데이터(D, E), + 예측 column에 해당되는 댐 유입량으로 구성

	홍수사상번호	연	월	일	시간	유입량	데이터 집단 1_ 유역평균강수	데이터 집단 1_ 강우(A 지역)	데이터 집단 1_ 강우(B 지역)	데이터 집단 1_ 강우(C 지역)	...	데이터 집단 5_ 강우(D 지역)	데이터 집단 5_ 수위(E 지역)	데이터 집단 5_ 수위(D 지역)	데이터 집단 6_ 유역평균강수	데이터 집단 6_ 강우(A 지역)	데이터 집단 6_ 강우(B 지역)	데이터 집단 6_ 강우(C 지역)	데이터 집단 6_ 강우(D 지역)	데이터 집단 6_ 수위(E 지역)	데이터 집단 6_ 수위(D 지역)
0	1	2006	7	10	8	189.100000	6.4	7	7	7	...	8	2.54	122.660	6.4	7	7	8	8	2.54	122.610
1	1	2006	7	10	9	216.951962	6.3	7	8	7	...	10	2.53	122.648	7.3	7	8	10	10	2.53	122.600
2	1	2006	7	10	10	251.424419	6.4	7	9	7	...	11	2.53	122.636	8.2	7	9	10	11	2.53	122.590
3	1	2006	7	10	11	302.812199	7.3	7	10	7	...	14	2.53	122.620	11.3	9	10	15	14	2.53	122.585
4	1	2006	7	10	12	384.783406	8.2	7	12	8	...	16	2.53	122.604	14.4	12	12	18	16	2.53	122.575

그림2. 데이터 head

유입량과 시간 데이터를 제외한 7개의 변수는 6개의 데이터 집단으로 분할되어 총 48개의 column을 갖는다.

평가 데이터: 홍수사상번호 26의 시간별 유입량 데이터를 예측

데이터 설명 및 EDA

2) EDA – 데이터 plot (EDA/hongsuEDA.ipynb)

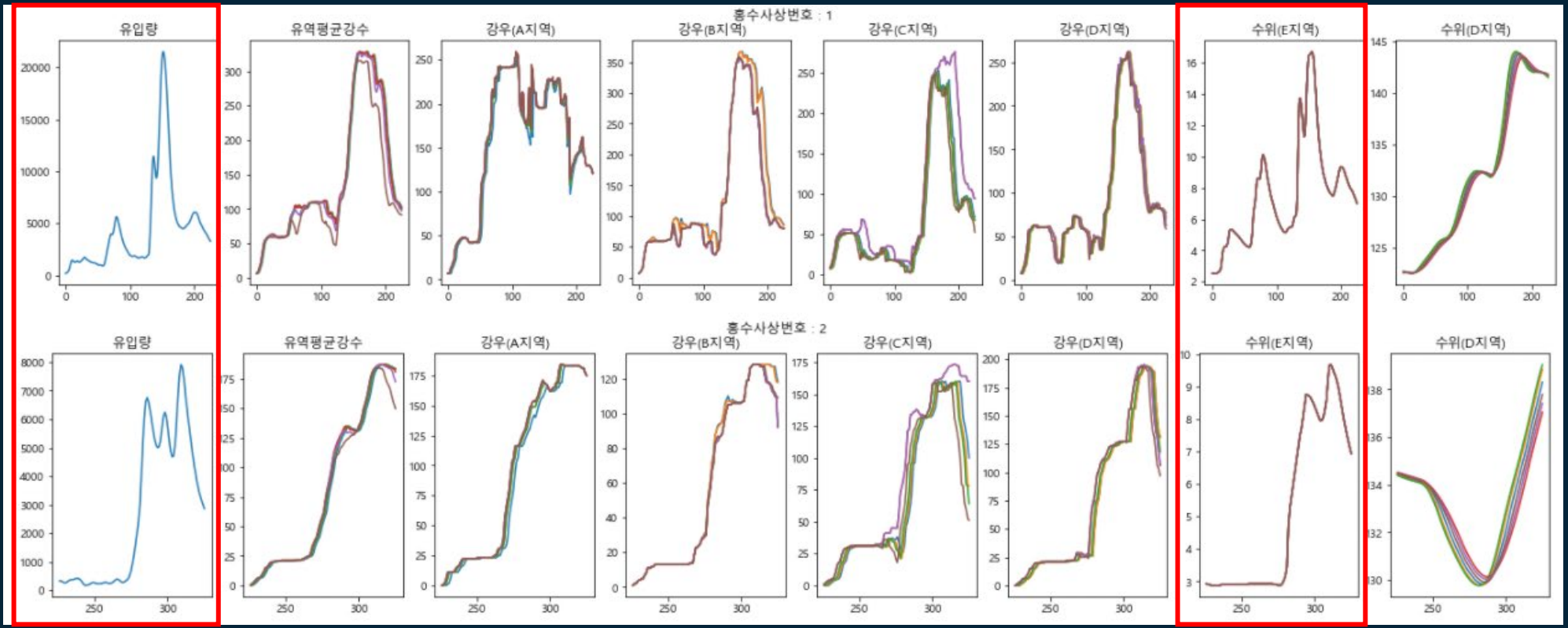


그림3. 홍수사상번호 1과 2의 데이터 plot

홍수사상번호 별로 데이터를 plot해 보았을 때 위와 같이 나왔으며, 다음과 같은 정보를 알 수 있다.

- 데이터집단 별 차이는 거의 존재하지 않는다.
- 수위(E지역)의 움직임이 유입량의 움직임과 매우 비슷하다.

데이터 설명 및 EDA

2) EDA – 데이터 shift (EDA/data shift & x^2.ipynb)

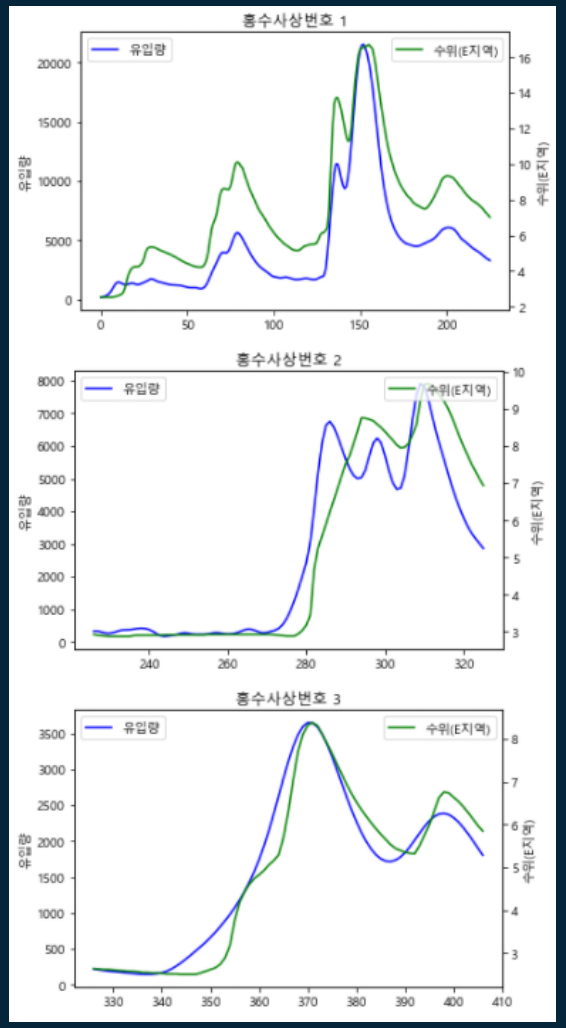


그림4. 유입량과 수위(E지역)

유입량과 수위(E지역)만 분리하여 보았을 때(그림4) 움직임이 매우 비슷한 것을 볼 수 있었다. 또한 유입량이 수위(E지역)의 데이터를 약간 선행하는 것처럼 보이기 때문에 shift를 진행하고 heatmap을 살펴 보았다.

- 대부분의 변수에서 shitft를 진행하였을 때 유입량과의 상관계수가 증가하는 것을 확인할 수 있었다.
- 대부분의 홍수사상번호에서 수위(E지역)의 상관계수가 매우 높게 나오지만 21번과 23번은 낮게 나온다.
- 5번은 수위(D지역)의 값이 모두 동일하게 나오고, 9번은 수위(D지역)의 상관계수가 매우 다르다.

shift 미적용

shift(1) 적용

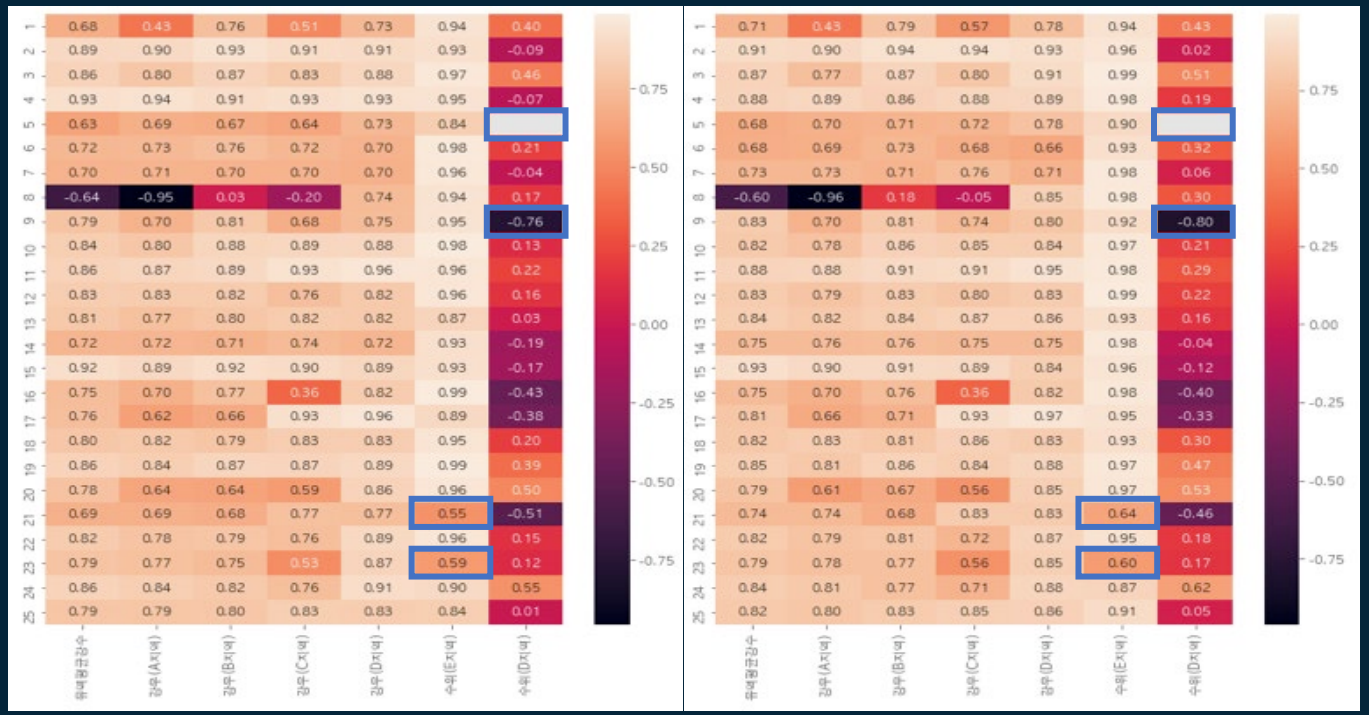


그림5. 유입량과 변수 상관계수

데이터 설명 및 EDA

2) EDA – 특이값

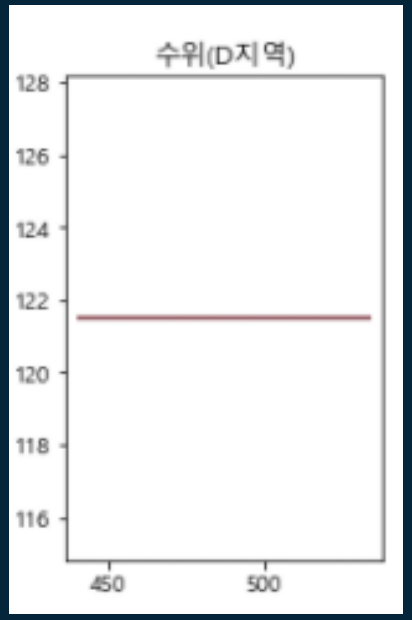


그림6. 5번 수위(D지역)

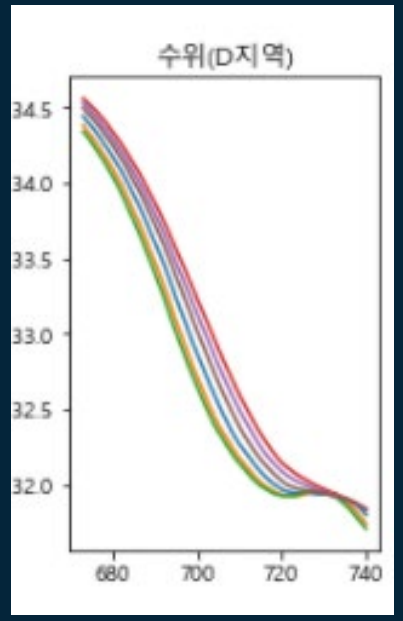


그림7. 9번 수위(D지역)

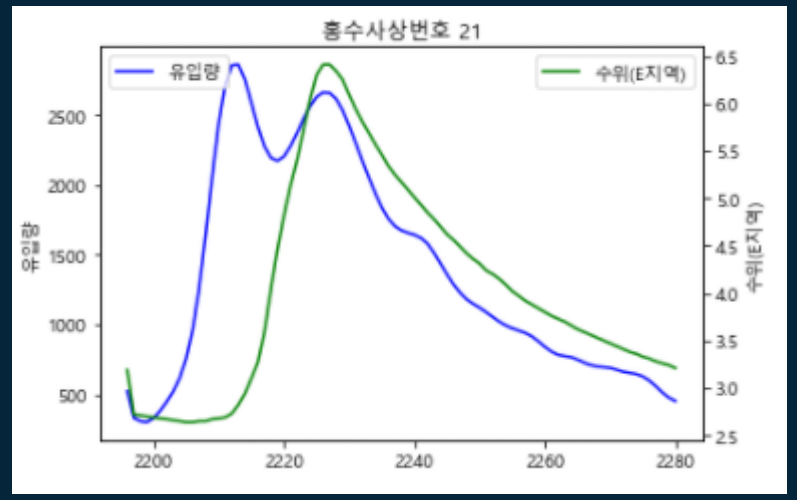


그림8. 21번 유입량과 수위(E지역)

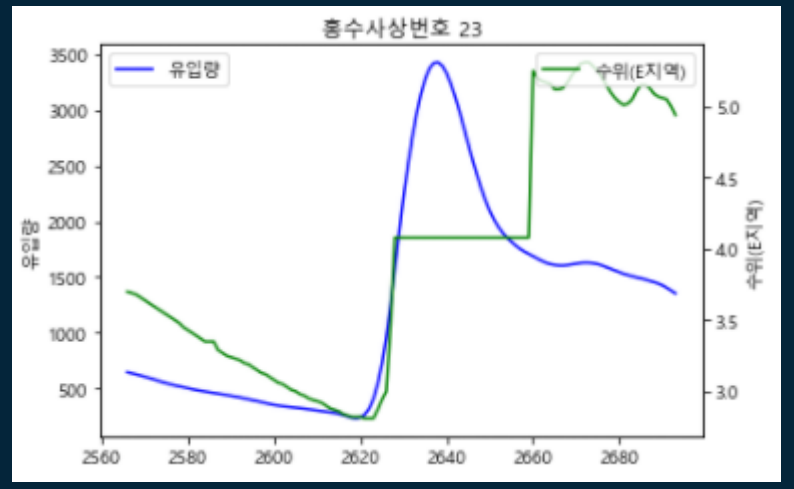


그림9. 23번 유입량과 수위(E지역)

데이터 설명 및 EDA

2) EDA – X^2 변환 (EDA/data shift & x^2.ipynb)

$X' = X^2$ 변환 미적용

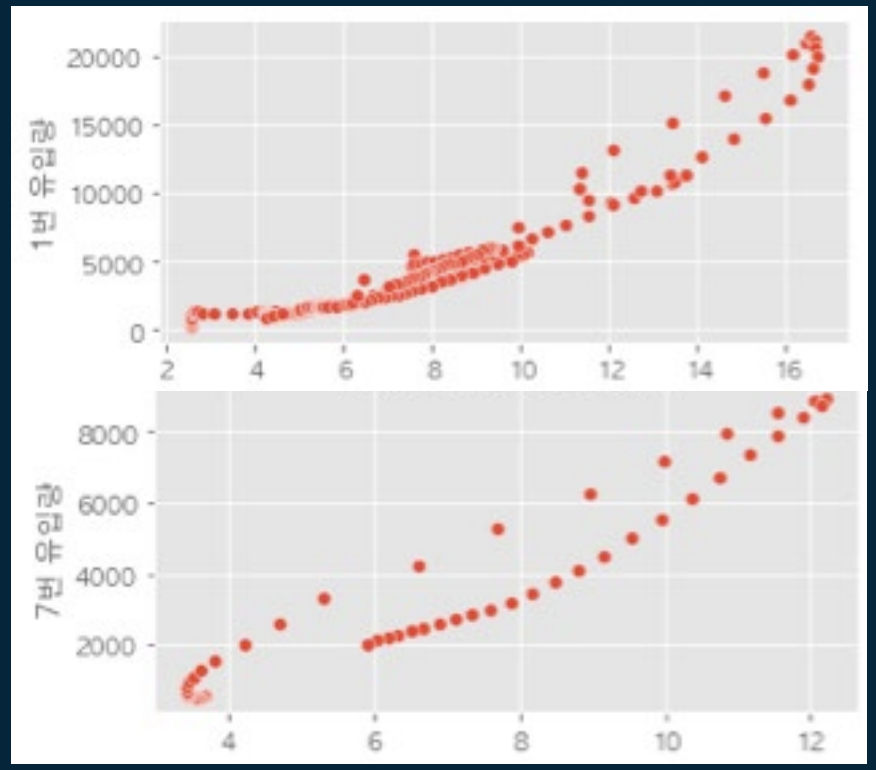


그림10. 유입량과 수위(E 지역) scatter plot

$X' = X^2$ 변환 적용

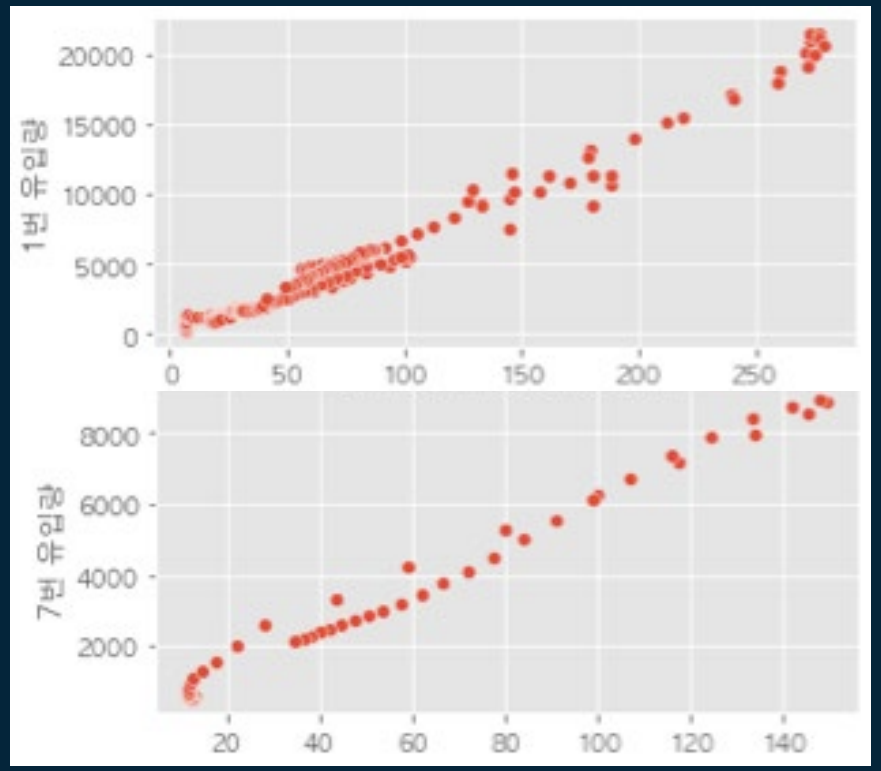


그림11. 유입량^2과 수위(E 지역) scatter plot

2. 실험 설계

1) 모델 소개
2) Evaluation Metrics 소개
3) 시계열 클러스터링
4) 래깅 배깅
5) 로그 변환
6) 제곱 값 추가
7) 최종 실험 설계

실험 설계

1) 모델 소개 – LR, DT, XGB, SGD Regressor

Linear Regression

- 예상변수와 예측변수 사이 선형 관계를 다룸.
- 현재 독립변수가 2개 이상이므로 다중선형회귀를 사용한다.
- 잔차 제곱의 합이 최소화 되는 회귀식 도출.

Decision Tree Model

- 분류와 회귀 모두 가능하며 1과 0으로 이분법적으로 나누어 노드를 이어 나감.
- 모든 노드를 지나야 하기 때문에 오버피팅 위험이 있어 노드 분할, 혹은 노드 당 데이터 최소화가 필요하다.

XGB Regressor

- Decision Tree를 기반으로 한 Ensemble 기법
- 비슷한 잔차값끼리 군집화 해주기 위해 한 변수를 선택한 새로운 분류 기준값을 사용해 샘플을 분리한다.

SGD Regressor

- Non convex한 함수의 극값 수렴을 확인
- 랜덤하게 추출한 일부 데이터를 사용해 파라미터를 업데이트해서 속도가 빠르다.
- 하지만 학습 중간 과정에서 결과의 진폭이 크고 불안정하다.

실험 설계

1) 모델 소개 – Kernel Ridge, DNN

Kernel Ridge Regression

- 데이터 포인트 중 일부를 train 나머지를 test로 사용하는 ML 모델
- Input 변수 X를 커널 함수로 활용하여 Mapping한 변수를 활용하여 파라미터를 추정

DNN

- 입력 변수들 간의 비선형 조합이 가능
  >> 가설 공간의 확장
- Feature Extraction을 자동으로 수행
- 하지만 가중치 수치 해석이 난해하며 데이터가 적으면 Overfitting될 위험이 있다.

실험 설계

2) Evaluation Metrics 소개

(1) $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$

표준편차 역할로 크기 의존적 에러이다.
단점: 작은 값에 대해서는 에러 값이 작아 과소예측의 위험이 있다.

(2) $RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(\widehat{y_i}+1) - \log(y_i+1))^2}$

크기 의존적, 상대적 에러이다.
과소예측에 페널티를 두기 때문에 예측이 작게 나오면 피해가 더 큰 현재 상황에 적합하다.

(3) $MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$

크기 의존적 에러의 단점을 보완.
단점: 실제값이 1보다 작거나 0이라면 무한대에 가까운 값이 나온다 -> 해당사항이 아니다.

(4) $R^2 = 1 - \frac{RSS}{TSS}$

추정한 모델의 주어진 자료에 대한 적합한 정도를 측정하는 척도로 종속변인, 독립변인의 상관계수가 높을수록 1에 가까워진다.

실험 설계

2) Evaluation Metrics 소개

(1) $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$

표준편차 역할로 크기 의존적 에러이다.
단점: 작은 값에 대해서는 에러 값이 작아
과소예측의 위험이 있다.

(2) $RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(\widehat{y_i}+1) - \log(y_i+1))^2}$

크기 의존적, 상대적 에러이다.
과소예측에 페널티를 두기 때문에 예측이 작게
나오면 피해가 더 큰 현재 상황에 적합하다.

(a)

```
1 y_true = [1000]
2 y_pred = [600]
3 RMSE = np.sqrt(mean_squared_error(y_true, y_pred))
4 print("RMSE :", RMSE)
5
6 RMSLE = np.sqrt(mean_squared_log_error(y_true, y_pred))
7 print("RMSLE :", RMSLE)
```

```
RMSE : 400.0
RMSLE : 0.5101598447800129
```

(b)

```
1 y_true = [1000]
2 y_pred = [1400]
3 RMSE = np.sqrt(mean_squared_error(y_true, y_pred))
4 print("RMSE :", RMSE)
5
6 RMSLE = np.sqrt(mean_squared_log_error(y_true, y_pred))
7 print("RMSLE :", RMSLE)
```

```
RMSE : 400.0
RMSLE : 0.3361867670217862
```

절대적인 오차인 RMSE은 동일하지만 y_true > y_pred인 (a)에서 RMSLE의 오차가 더 크게 나온다

실험 설계

3) 시계열 클러스터링

EDA 결과 유입량과 E 지역의 수위 데이터 간 관계성 파악 가능

>> 수위 데이터를 클러스터링 한 결과를 input 데이터에 포함

수위(E지역)을 홍수사상번호별 묶음으로 만들고, 클러스터 진행

>> 최적의 군집 개수 찾기 + 결과 one-hot encoding으로 반영

시계열 클러스터링 방법

- Clustering: tslearn.clustering 모듈 이용
- Metric: “soft DTW”

  기존에 사용되는 Euclidean Distance Matching이 아닌 Dynamic Time Warping Matching (DTW) 이용

  >> 시계열 데이터의 개수/길이가 다르더라도 패턴에 따른 유사성을 측정하여 군집화

  >> 실제로 홍수상번호 별로 길이는 30여개에서 260여개까지 큰 폭으로 차이가 있어 Euclidean 적용 불가

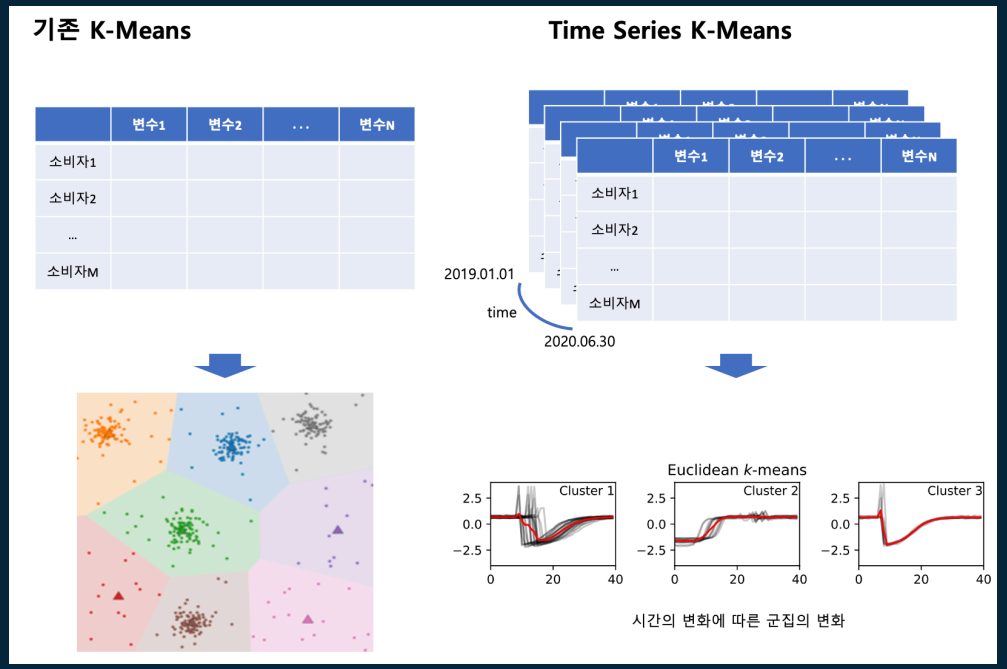


그림12. 시계열 클러스터링 예시

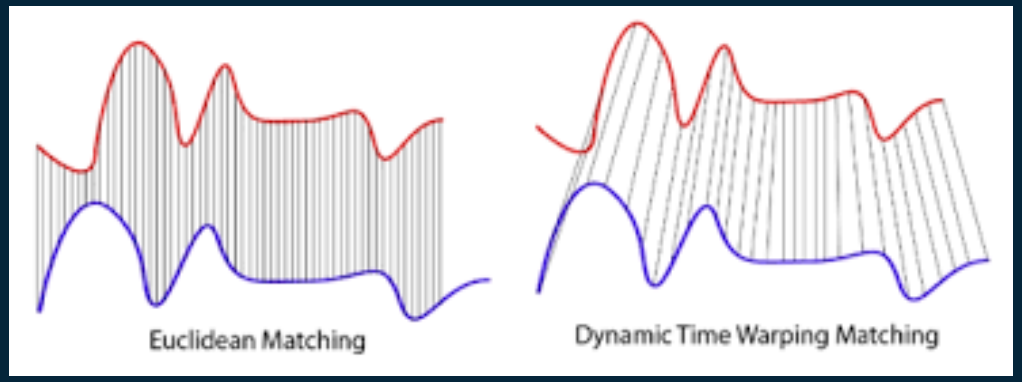


그림13. Euclidean과 DTW 비교

실험 설계

3) 시계열 클러스터링 (clustering/ 시계열 E지역 clustering.ipynb)

- 최적 K 값 선정 지표 - `silhouette_score` 이용

  >> -1에서 -1까지의 수치를 갖는다.

  +1에 가까움: 데이터가 클러스터 안에 잘 속해있고 다른 클러스터와는 멀리 떨어져 있다는 뜻

  0에 가까움: 클러스터의 경계에 데이터가 위치해 있다는 뜻

  -1에 가까움: 샘플이 잘못된 클러스터에 할당되었다는 뜻

>> K = 3, 4, 5, 6의 구간이 적절한 실루엣 점수를 가짐

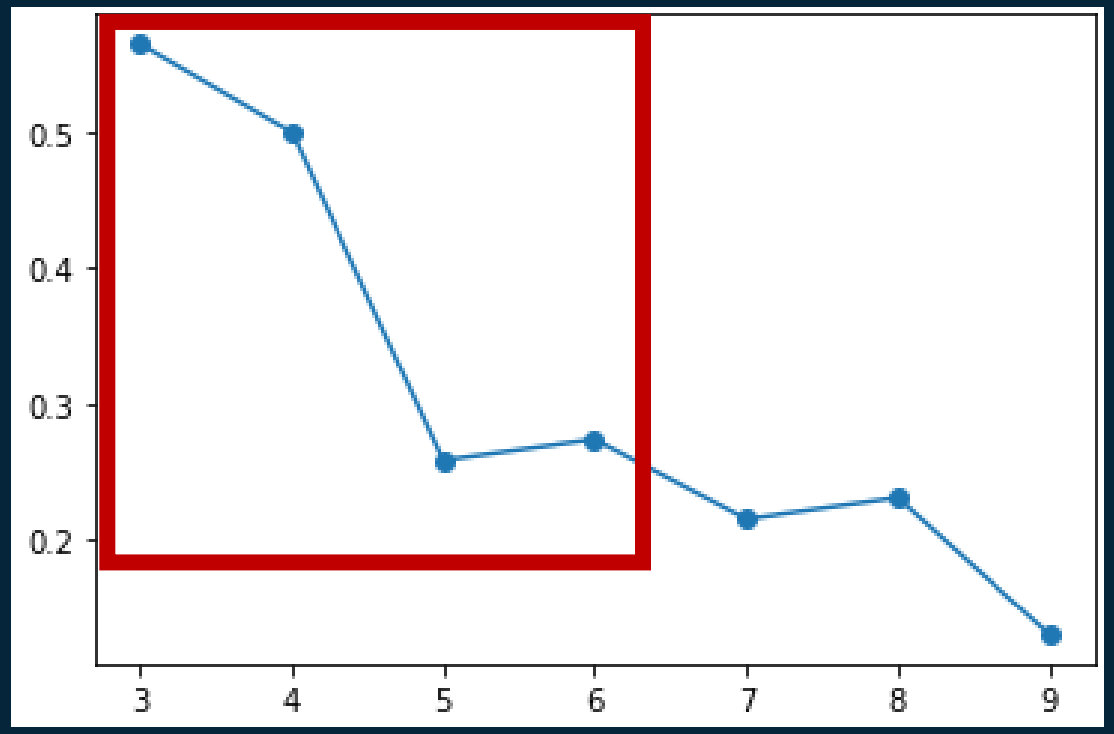


그림14. 시계열 클러스터링 결과

실험 설계

3) 시계열 클러스터링 (clustering/ 시계열 E지역 clustering.ipynb)

K= 3, 4 일 때의 cluster 결과

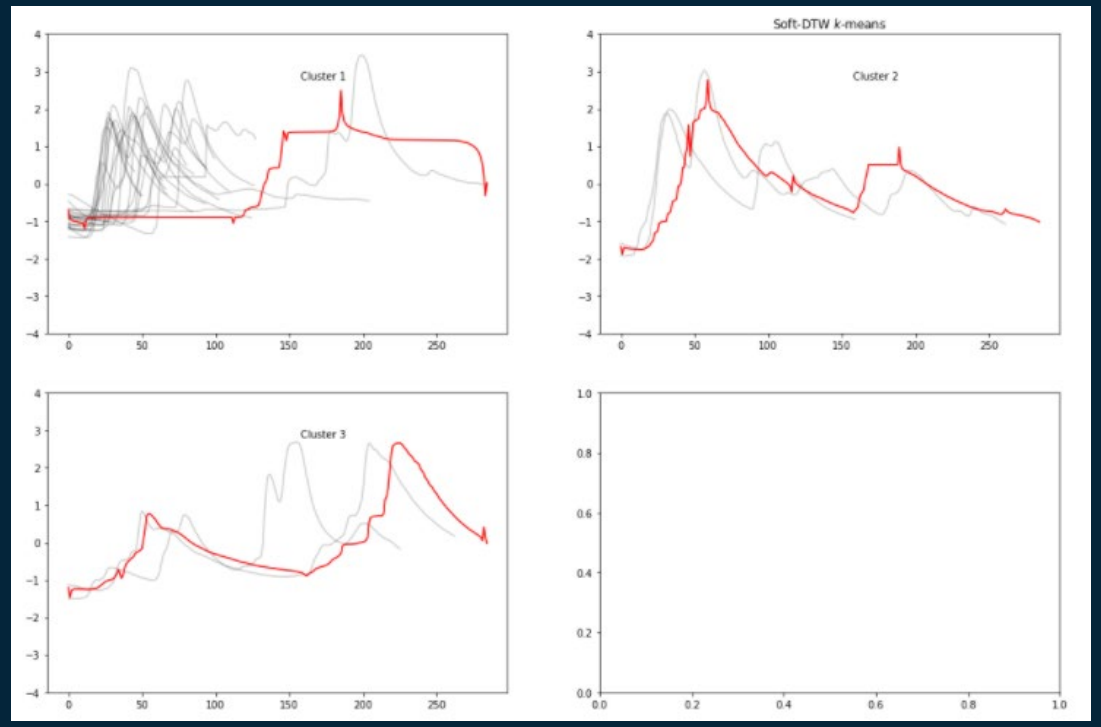


그림15. k=3 일 때 군집 결과

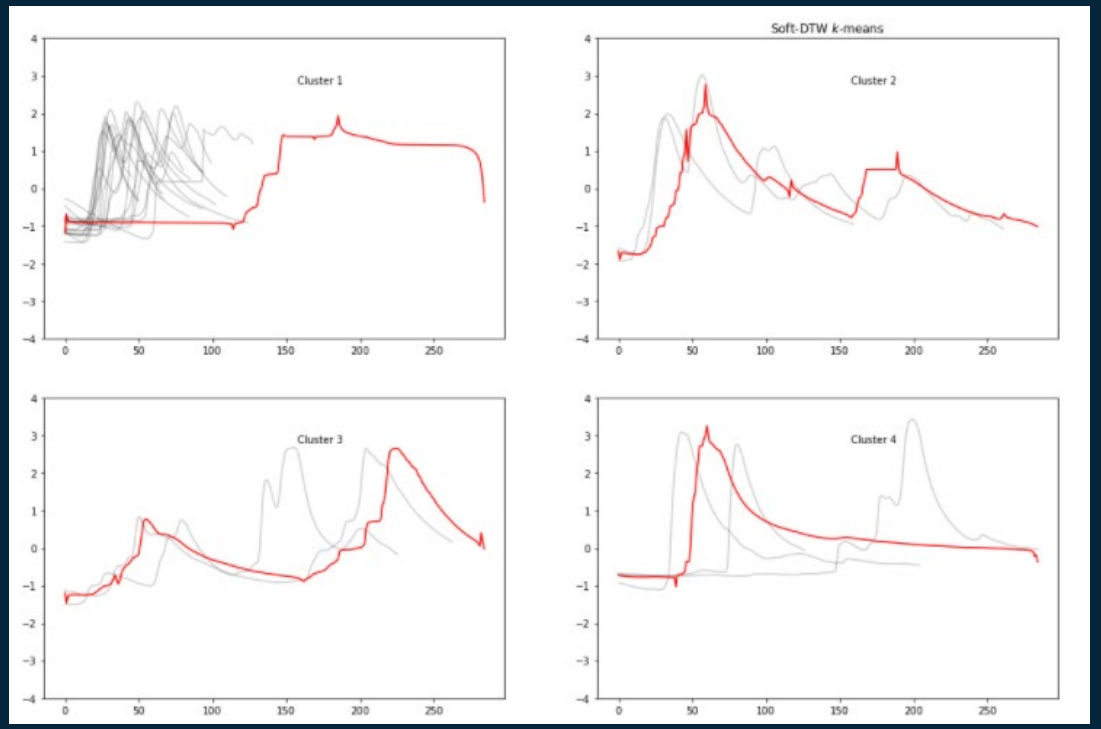


그림16. k=4 일 때 군집 결과

실험 설계

3) 시계열 클러스터링 (clustering/ 시계열 E지역 clustering.ipynb)

K= 5, 6 일 때의 cluster 결과

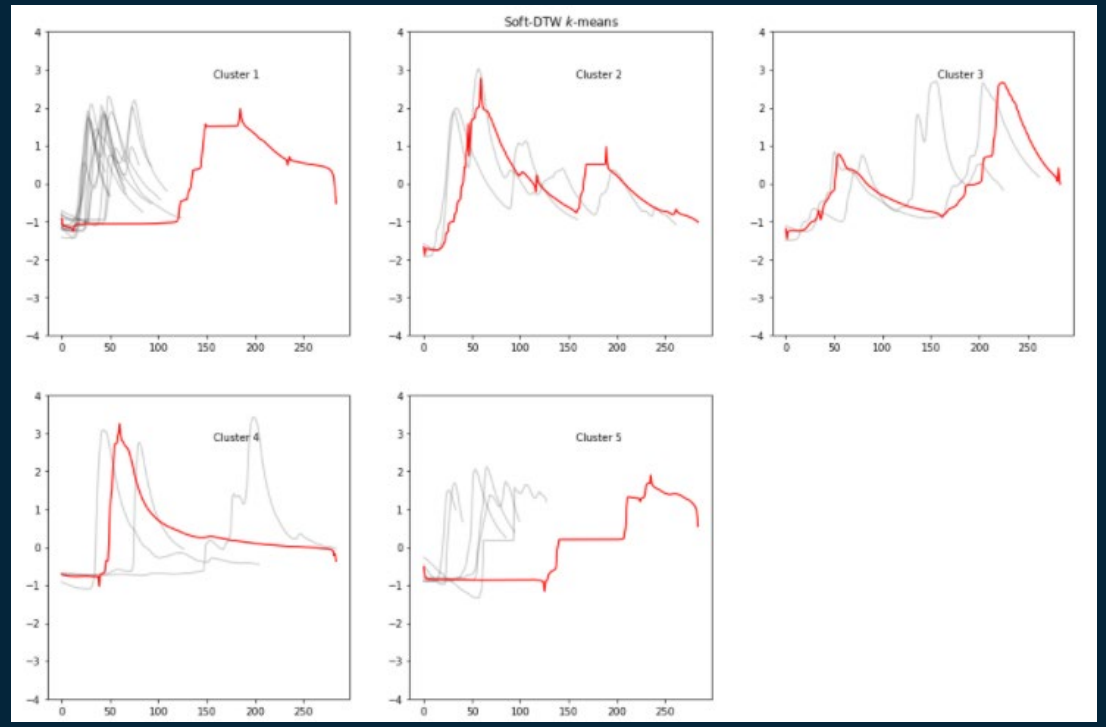


그림17. k=5 일 때 군집 결과

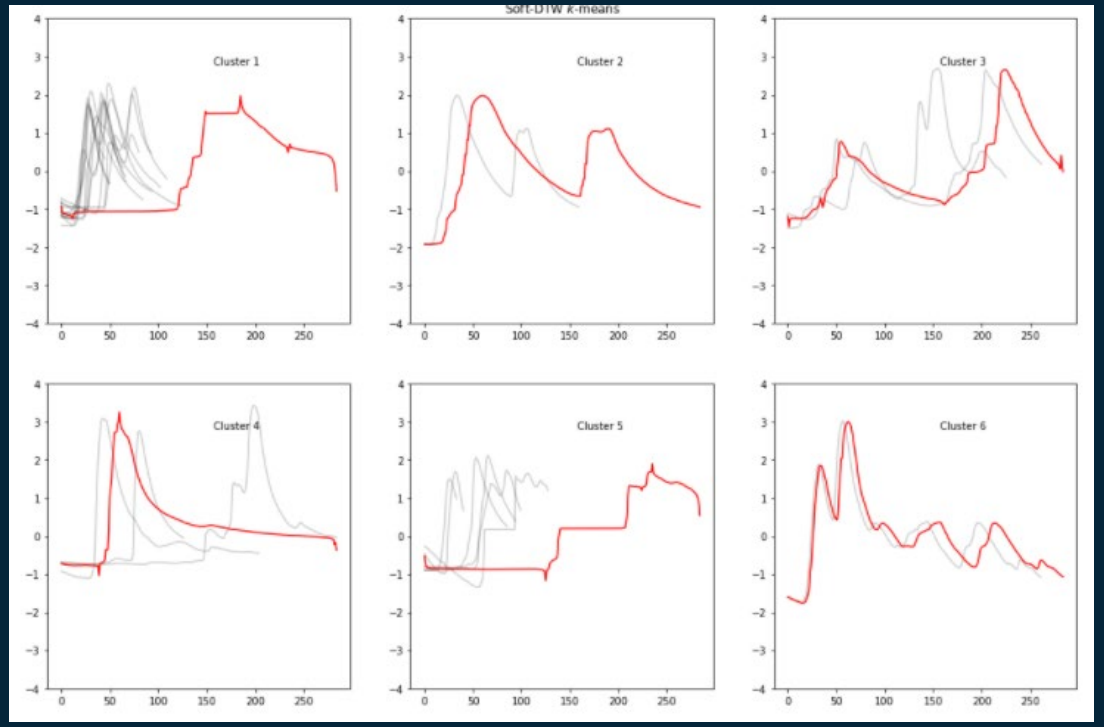


그림18. k=6 일 때 군집 결과

실험 설계

3) 시계열 클러스터링 (clustering/ 시계열 E지역 clustering.ipynb)

- Label 결과 input data에 알맞게 더미화

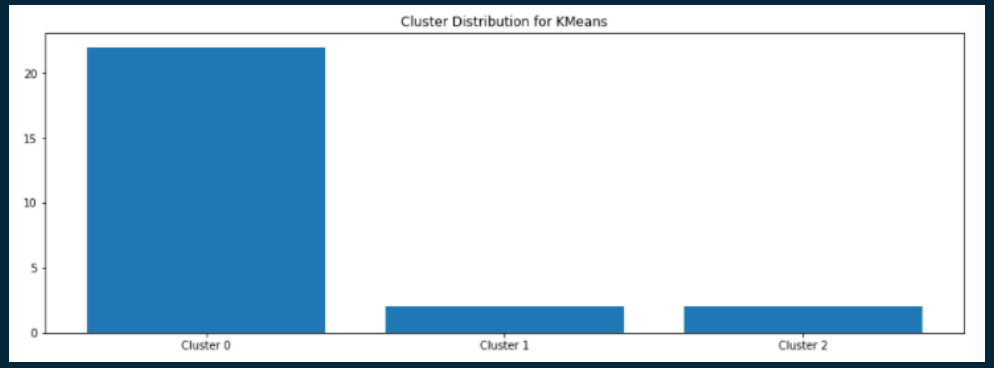


그림18. k=3 일 때 군집 별 데이터 개수

```
k3_column = pd.DataFrame(train['홍수사상번호'], columns = ['홍수사상번호'])
k3_column["군집 결과"] = np.nan
for i in range(len(k3_column)):
    tmp = k3_column['홍수사상번호'][i]
    tmp = tmp - 1
    k3_column['군집 결과'][i] = labels3[tmp]
k3_column_final = pd.get_dummies(k3_column['군집 결과'])
k3_column_final
#k3_column_final 이용
```

	0.0	1.0	2.0
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
...	...	...	...
3046	0	1	0
3047	0	1	0
3048	0	1	0
3049	0	1	0
3050	0	1	0

3051 rows × 3 columns

그림19. 군집화 결과 one-hot encoding 코드

실험 설계

4) 데이터 시프트(shift)

본 실험의 모델들은 시퀀스를 인식할 수 없는 모델들이다. 하지만 홍수 데이터의 경우 홍수사상번호에 따라 이어지는 시퀀스 데이터이기 때문에 앞과 뒤의 데이터를 시프트 하여 전해주면 모델이 시퀀스 데이터를 인식할 수 있을 것이라 판단하였다.

홍수사상번호 별로 pandas 라이브러리의 shift를 사용하여 직전과 직후의 데이터 값을 붙여주었다. 이때 홍수사상번호의 제일 앞과 제일 뒤의 데이터는 nan 값이 되기 때문에 첫번째 유입량 예측은 두번째 유입량 예측 값으로 대체하고, 마지막 유입량 예측은 마지막에서 두번째 유입량 예측 값으로 대체하였다.

```
name = 't-1 t+1'

data_t = data_raw.copy()

for col in data_t.columns.difference([y_col] + PK_col):
    data_t[f'{col}_shift 1'] = data_t[col].shift()
    data_t[f'{col}_shift1 -1'] = data_t[col].shift(-1)

shift_col = data_t.filter(regex='shift').columns
data_t['홍수사상번호_shift 1'] = data_t['홍수사상번호'].shift()
data_t['홍수사상번호_shift -1'] = data_t['홍수사상번호'].shift(-1)

data_t.loc[(data_t['홍수사상번호'] != data_t['홍수사상번호_shift 1']), shift_col] = np.nan
data_t.loc[(data_t['홍수사상번호'] != data_t['홍수사상번호_shift -1']), shift_col] = np.nan

data_t = data_t.dropna()
```

그림20. 데이터 shift 코드

실험 설계

5) 로그 변환

현재 홍수 데이터의 경우 대부분이 왼쪽에 몰려 있는 편향된 데이터이다. 하지만 머신러닝은 데이터가 정규 분포를 따를 때 더 학습이 잘 되기 때문에 데이터를 로그 변환 시켜 줌으로써 정규 분포에 가깝게 만들어준다.

이때 0 값은 로그 변환이 안되기 때문에 각 칼럼 별 min 값으로 빼고 0.01을 더한 후 로그 변환을 진행하였다.
원래 0이었던 데이터는 _zero 칼럼을 만들어 1을 표시하여, 해당 값이 0 값이었던 것을 인식할 수 있도록 하였다.

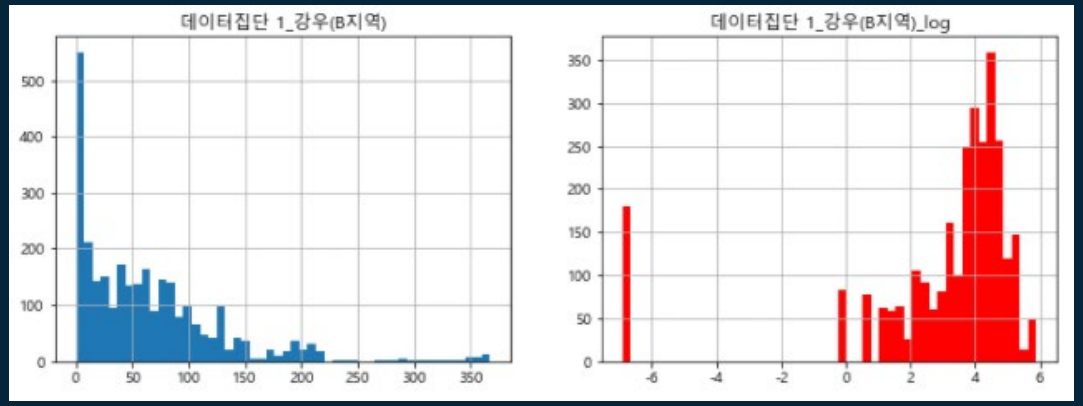


그림21. 강우(B지역) 히스토그램 원본 (좌) 로그 변환 후 (우)

```
name = 'log'
data_log = data_raw.copy()

log_col = data_log.columns.difference(list(data_log.filter(regex="수위\(D지역\)").columns) + [y_col] + PK_col)
for col in log_col:
    data_log[col + '_zero'] = (data_log[col] == 0).astype(int)
    data_log[col] = data_log[col].apply(lambda x : np.log(x - data_log[col].min() + 0.01))
```

그림22. 로그 변환 코드

실험 설계

6) 제곱값 추가

EDA 결과 수위(E지역)의 값과 유입량은 매우 비슷한 방향으로 움직이는 것을 확인할 수 있었으며, 또한 scatter plot으로 확인하였을 때 convex한 관계를 보였다.

이때 수위(E지역) 데이터를 제곱한다면 좀 더 선형 적인 관계를 얻을 수 있었다. 따라서 수위(E지역)의 제곱한 값을 칼럼으로 추가해 주었다.

```
name = 'x2'
data_x2 = data_raw.copy()

x2_col = data_x2.filter(regex="수위\(E지역\)").columns
for col in x2_col:
    data_x2[col + '_x2'] = data_x2[col] ** 2
```

그림24. x^2 칼럼 추가 코드

$X' = X^2$ 변환 적용

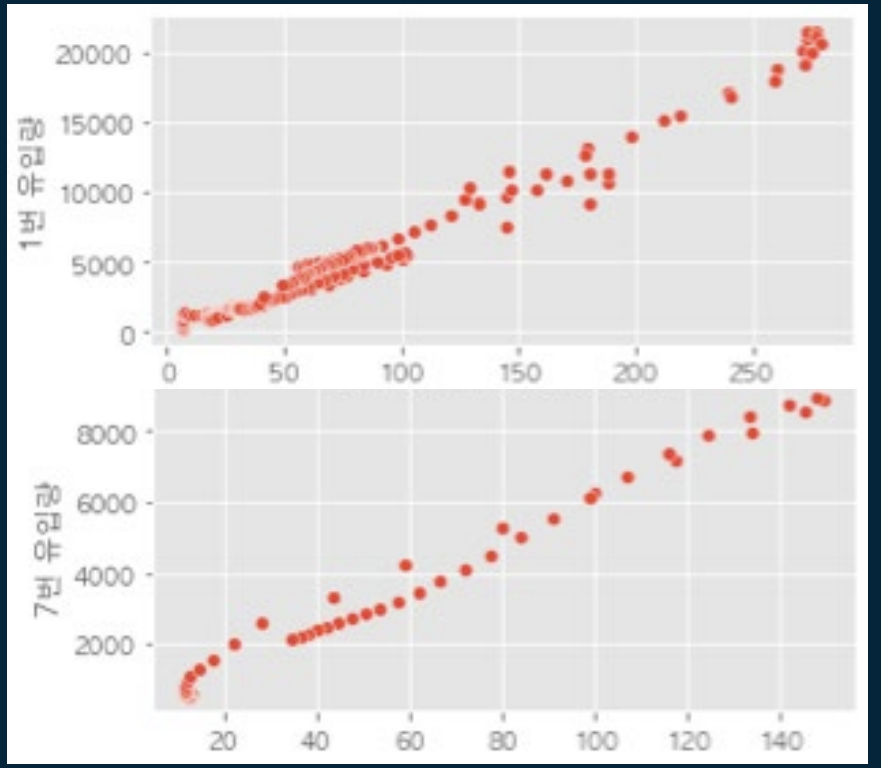


그림23. 유입량^2과 수위(E 지역) scatter plot

실험 설계

7) 최종 실험 설계 (models/1.Machine Learning.ipynb & models/2.Deep Learning.ipynb)

- 모델
  1) Machine Learning : LR (Linear Regression), DT (Decision Tree), SGDRegressor, KernelRidge
  2) Deep Learning : DNN (Deep Neural Network)

- 평가 척도 : RMSE, RMSLE, R2_score, MAPE

- Scaler : Standard Scaler사용 (DNN의 경우 Robust Scaler 사용)

- 평가 방법 : 모든 홍수사상번호를 돌아가며 target으로 설정하고 나머지 데이터로 학습 후 target에 대한 예측 결과로 평가 한다.
  ex) 1번 홍수사상번호를 제외한 데이터로 모델을 학습 후 1번 홍수사상번호에 대해 예측을 진행한다.

실험을 진행한 방법은 총 15개이며 model code는 다음과 같다.

model code	설명
base	원본 데이터 그대로 넣기
base_2123	홍수사상번호 21번과 23번 데이터 삭제
t-1 t+1	t-1 데이터와 t+1 데이터 추가
waterlevel t-1 t+1	수위 E지역의 t-1 데이터와 t+1 데이터 추가
3 cluster	수위 E지역을 3개로 군집화 한 변수 추가
4 cluster	수위 E지역을 4개로 군집화 한 변수 추가
5 cluster	수위 E지역을 5개로 군집화 한 변수 추가
6 cluster	수위 E지역을 6개로 군집화 한 변수 추가
log	log 변환한 데이터
origin + log	원본 데이터에 log 변환 추가
x2	수위 E지역을 제곱한 변수 추가
5cluster + x2	5개 군집 데이터와 제곱 변수 추가
5cluster + t	5개 군집 데이터와 t-1 t+1 데이터 추가
t + x2	제곱 변수와 t-1 t+1 데이터 추가
5cluster + t + x2	5개 군집 데이터와 제곱 변수, t-1 t+1 데이터 추가

그림25. 모델 코드 표

3. 실험 결과

1) 결과 분석
2) 타겟 예측

실험 결과

1) 결과 분석 (models/3.결과 분석.ipynb)

	LR				DT				SGDRegressor			
	RMSE	RMSLE	R2_score	MAPE	RMSE	RMSLE	R2_score	MAPE	RMSE	RMSLE	R2_score	MAPE
base	884.28	1.83	0.84	45.92	1122.25	0.53	0.74	56.30	933.32	1.93	0.82	51.27
base_2123	906.78	1.91	0.84	47.60	1214.99	0.55	0.71	59.40	917.97	2.00	0.83	50.08
t-1 t+1	842.22	1.81	0.85	45.47	1110.15	0.50	0.74	49.96	891.04	1.93	0.83	49.02
waterlevel t-1 t+1	885.95	1.79	0.84	45.99	1104.88	0.52	0.74	50.88	879.43	1.92	0.84	49.52
3 cluster	884.93	1.70	0.84	45.79	1119.07	0.54	0.74	58.84	890.29	1.97	0.83	50.26
4 cluster	888.88	1.67	0.83	45.42	1118.74	0.53	0.74	57.09	839.14	1.87	0.85	48.50
5 cluster	889.02	1.70	0.83	46.46	1098.28	0.54	0.75	60.02	881.02	1.96	0.84	50.76
6 cluster	889.02	1.70	0.83	46.46	1135.65	0.56	0.73	61.49	915.39	1.85	0.82	49.61
log	1442.59	1.63	0.56	91.81	1120.20	0.52	0.74	55.97	1628.80	1.67	0.44	97.62
origin + log	1442.59	1.63	0.56	91.81	1120.20	0.52	0.74	55.97	1648.95	1.83	0.43	104.89
x2	537.08	1.14	0.94	42.35	1146.99	0.53	0.72	57.54	534.38	1.25	0.94	40.08
5cluster + x2	544.02	1.16	0.94	49.04	1118.74	0.55	0.74	64.20	545.71	1.13	0.94	48.58
5cluster + t	512.87	1.18	0.94	46.62	1113.21	0.52	0.74	51.13	509.85	1.18	0.95	47.20
t + x2	503.98	1.14	0.95	40.05	1116.02	0.53	0.74	52.98	503.75	1.13	0.95	40.78
5cluster + t + x2	512.87	1.18	0.94	46.62	1113.21	0.52	0.74	51.13	504.69	1.10	0.95	51.59

	KernelRidge				XGB				DNN			
	RMSE	RMSLE	R2_score	MAPE	RMSE	RMSLE	R2_score	MAPE	RMSE	RMSLE	R2_score	MAPE
base	1445.31	4.31	0.56	80.92	984.90	0.46	0.80	43.33	509.75	0.45	0.95	24.22
base_2123	1467.43	4.26	0.57	80.61	1032.48	0.53	0.79	47.99	490.85	0.47	0.95	26.60
t-1 t+1	1430.47	4.32	0.57	81.39	973.41	0.46	0.80	42.81	465.12	0.34	0.95	21.17
waterlevel t-1 t+1	1453.24	4.32	0.56	81.18	1028.03	0.63	0.78	50.63	532.76	0.44	0.94	23.27
3 cluster	1429.99	4.28	0.57	80.49	1001.61	0.49	0.79	47.65	524.74	0.63	0.94	26.29
4 cluster	1435.70	4.30	0.57	80.75	1001.97	0.46	0.79	47.01	468.06	0.54	0.95	24.76
5 cluster	1438.71	4.30	0.57	80.87	1004.49	0.52	0.79	47.19	500.41	0.58	0.95	25.04
6 cluster	1438.71	4.30	0.57	80.87	1004.49	0.52	0.79	47.19	500.88	0.41	0.95	23.13
log	2021.06	3.84	0.14	79.40	985.69	0.46	0.80	43.46	613.79	0.70	0.92	43.35
origin + log	2021.06	3.84	0.14	79.40	985.69	0.46	0.80	43.46	494.40	0.41	0.95	29.15
x2	1314.17	4.66	0.64	84.81	984.90	0.46	0.80	43.33	454.84	0.41	0.96	24.39
5cluster + x2	1321.96	4.67	0.63	84.91	1004.49	0.52	0.79	47.19	455.41	0.52	0.96	25.68
5cluster + t	1310.35	4.69	0.64	85.22	991.91	0.48	0.79	45.65	449.92	0.43	0.96	23.70
t + x2	1300.60	4.69	0.64	85.06	973.41	0.46	0.80	42.81	413.58	0.41	0.96	23.08
5cluster + t + x2	1310.35	4.69	0.64	85.22	991.91	0.48	0.79	45.65	457.79	0.58	0.96	27.17

그림26. 최종 결과

① 여러 모델로 다양한 경우에 대해 실험을 진행한 결과 DNN 모델이 압도적으로 좋은 성과를 냈다.

② 다음으로는 LR과 SGD Regressor가 성능이 좋았으며, XGB는 대부분의 경우에서 비슷한 성능을 보였다.

③ 선형 모델에서 클러스터링 추가, t-1, t+1 데이터 추가 그리고 수위(E지역)의 제곱 칼럼 추가가 합쳐 졌을 때 성능이 개선됨을 보였다.

실험 결과

1) 결과 분석 (models/3.결과 분석.ipynb)

	DNN			
	RMSE	RMSLE	R2_score	MAPE
t + x2	413.58	0.41	0.96	23.08
5cluster + t	449.92	0.43	0.96	23.70
x2	454.84	0.41	0.96	24.39
5cluster + x2	455.41	0.52	0.96	25.68
5cluster + t + x2	457.79	0.58	0.96	27.17
t-1 t+1	465.12	0.34	0.95	21.17
4 cluster	468.06	0.54	0.95	24.76
base_2123	490.85	0.47	0.95	26.60
origin + log	494.40	0.41	0.95	29.15
5 cluster	500.41	0.58	0.95	25.04
6 cluster	500.88	0.41	0.95	23.13
base	509.75	0.45	0.95	24.22
3 cluster	524.74	0.63	0.94	26.29
waterlevel t-1 t+1	532.76	0.44	0.94	23.27
log	613.79	0.70	0.92	43.35

그림27. DNN 결과 표

최종 결과 분석에서는 가장 성능이 좋은 DNN 모델을 사용하였다.

RMSE 기준으로 오름차순 하였을 때 t-1, t+1 데이터와 수위 (E지역)의 제곱 데이터를 추가하였을 때가 성능이 가장 높게 나왔다.

RMSLE는 RMSE와 비교했을 때 상대적 에러를 측정하여, Under Estimation에 더 큰 페널티를 주게 된다. RMSLE 기준으로 보았을 때 t-1, t+1 데이터만 추가했을 때 성능이 가장 높게 나왔다.

실험 결과

1) 결과 분석 (models/3.결과 분석.ipynb)

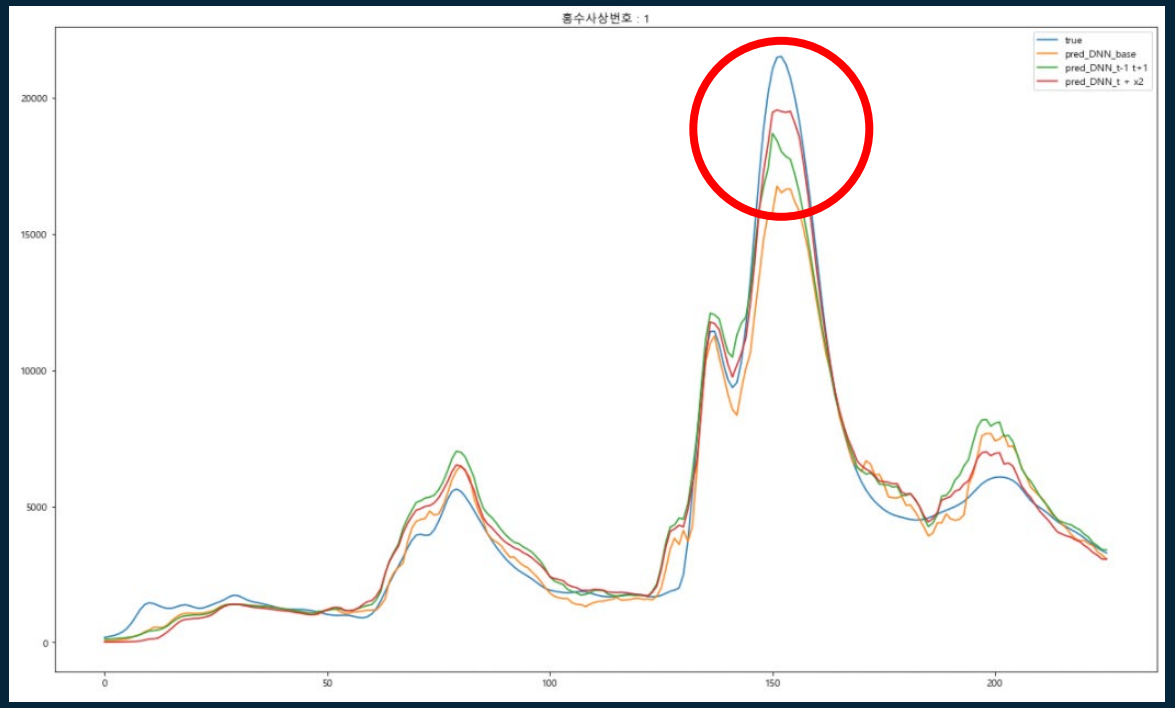


그림28. 홍수사상번호 1번 예측 결과

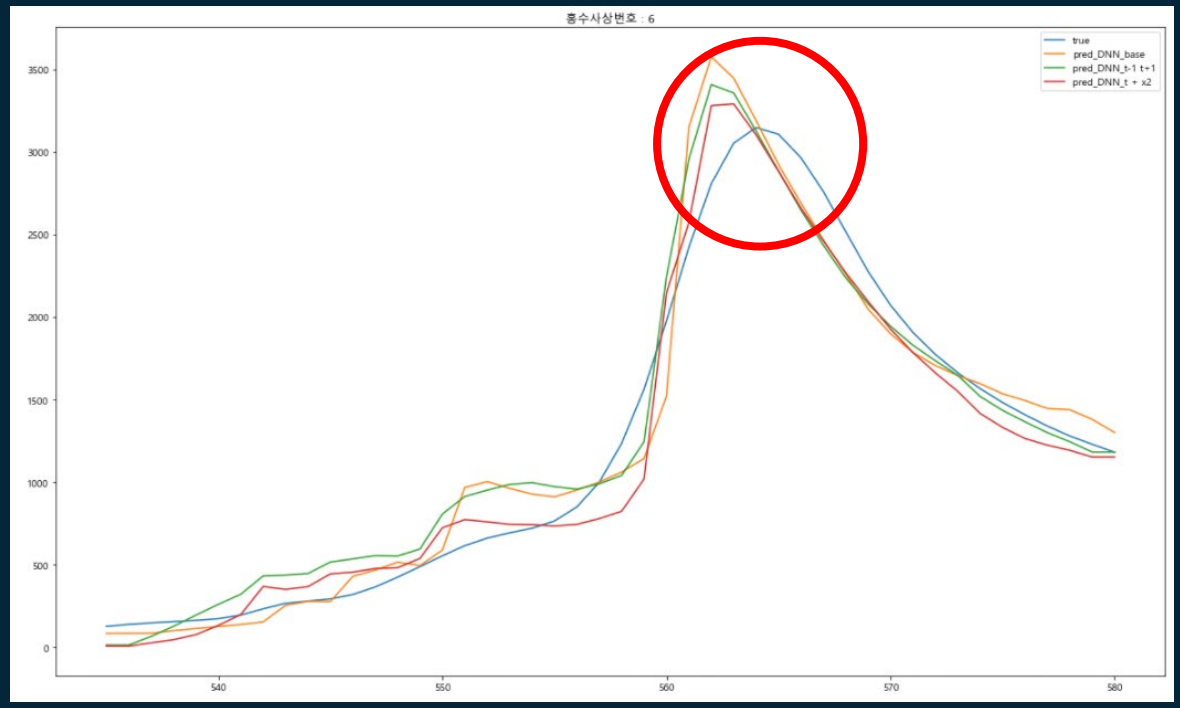


그림29. 홍수사상번호 6번 예측 결과

성능이 좋았던 DNN_t-1 t+1과 DNN_t + x2 모델의 정확도를 확인하기 위해 실제 유입량과 예측 값을 plot 하여 비교해 보았다. 원본 데이터를 그대로 넣는 base 모델의 예측 값도 비교를 위해 추가하였다.

대부분의 홍수사상번호에서 두 모델은 비슷한 성능을 보였지만, 홍수사상번호 1번과 같이 유입량이 크게 증가하는 부분에서 t + x2 모델이 더 잘 예측하는 것을 확인할 수 있었다.

실험 결과

2) 구조 최적화 (models/4.타겟 예측.ipynb)

이전 실험을 통해 선택한 최적 모델에 대해서 DNN 구조의 최적화를 위해 Validation Set을 선택한 후 Validation Set에 대해서 가장 예측력이 좋은 모델 구조를 탐색하였다.

실험 설계

- Validation Set : 이전 시계열 클러스터링 결과를 통해 26번의 수위(E지역)과 가장 흐름이 비슷한 15번으로 선택
- Structure List : hidden layer / batch norm 여부 / drop out 여부

```
hidden_layers = ([1,],[1,1],[1,1,1],[1,1,1,1],
                 [2,],[2,2],[2,2,2],[2,2,2,2],
                 [0.5,],[0.5, 0.5],[0.5,0.5,0.5],[0.5,0.5,0.5,0.5],
                 [0.2,],[0.2,0.2],[0.2,0.2,0.2])
batch_norms = [0,1]
drop_outs = [0,1]
```

그림30. Structure List

	RMSE
[10000, 0.01, [0.2, 0.2], 1, 1, 0.5]	164.49
[10000, 0.01, [0.2], 1, 0, 0.5]	173.32
[10000, 0.01, [0.5], 1, 0, 0.5]	173.98
[10000, 0.01, [1], 1, 0, 0.5]	177.23

그림31. Validation Set에 대한 RMSE 결과

실험 결과 hidden layer는 [0.2, 0.2]를 사용하고 batch norm과 drop out 모두를 사용할 때 가장 성능이 좋게 나왔다.

실험 결과

3) 타겟 예측 (models/4.타겟 예측.ipynb)

이전 실험 결과에 따른 최적 모델을 사용하여 타겟 홍수사상번호인 26번을 예측하였다.

모델 : DNN / (29, 29, 1) / Drop Out : 0.5 / Batch Norm 사용
전처리 방법 : Robust Scaler + 수위(E지역) 제곱 데이터 + t-1 t+1 데이터 사용
학습 방법 : Loss Function : MSE, Adam Optimizer 사용, Learning Rate : 0.01 / Epoch : 10,000

학습 결과 모델이 예측한 유입량은 다음과 같다.

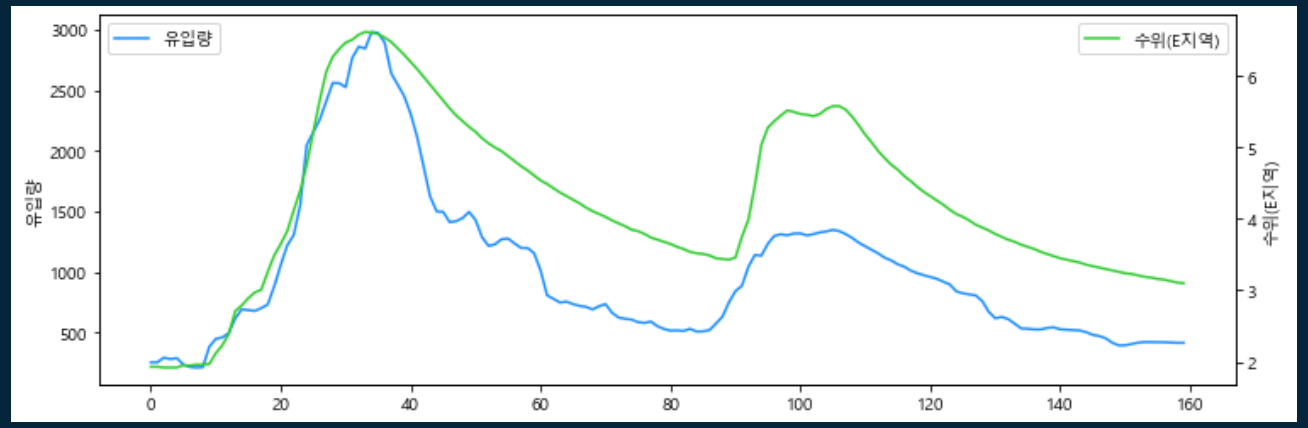


그림32. 홍수사상번호 26번 유입량 예측과 수위(E지역)

	NO	홍수사상번호	연	월	일	시간	유입량
1	1.00	26.00	2018.00	7.00	1.00	6.00	253.74
2	2.00	26.00	2018.00	7.00	1.00	7.00	253.74
3	3.00	26.00	2018.00	7.00	1.00	8.00	292.10
4	4.00	26.00	2018.00	7.00	1.00	9.00	281.16
5	5.00	26.00	2018.00	7.00	1.00	10.00	288.38
...	...	...	...	...	...	...	...
156	156.00	26.00	2018.00	7.00	7.00	17.00	420.27
157	157.00	26.00	2018.00	7.00	7.00	18.00	419.39
158	158.00	26.00	2018.00	7.00	7.00	19.00	417.26
159	159.00	26.00	2018.00	7.00	7.00	20.00	414.63
160	160.00	26.00	2018.00	7.00	7.00	21.00	414.63

그림33. 홍수사상번호 26번 예측치

감사합니다!

홍수빅타