



USER BEHAVIOR PATTERN PREDICT

사용자 행동 패턴 예측
Feature engineering & modeling

김봉석 / 노승찬

CONTENTS



01 프로젝트 소개

02 EDA

03 Feature engineering& modeling

- 01 LSTM
- 02 XGB
- 03 Extra Tree
- 04 Weighted average ensemble

04 Conclusion

About Project

프로젝트 설명



"국내 최대규모 인공지능 competition platform"

SUBJECT

과거의 데이콘 데이터를 활용한, 미래의 사용자 행동 패턴 예측
다변량 시계열 데이터 분석

데이터 소개

지난 2년 동안의 데이콘에 관한 데이터

[학습 데이터] - train.csv

2018.09.09 ~ 2020.12.08 동안 기록된 **한 시간 간격**의 사용자 행동 데이터

DATE	사용자	세션	신규방문자	페이지뷰
2018-09-09	281	266	73	1826
2018-09-10	264	247	51	2092
...				
2020-12-07	2979	2988	753	77443
2020-12-08	3033	2990	772	68857

사용자: 일별 데이콘에 방문한 사람의 수

세션: 일별 사이트에서 모든 사용자가 시작한 개별 세션 수

신규 방문자: 일별 데이콘에 처음으로 방문한 사용자 수

페이지 뷰: 홈페이지에 들어온 사용자가 본 페이지 수

[제출] - submission.csv

2020.11.09 ~ 2021.01.08 모두 0으로 채워진 **일 단위** 데이터

columns : 사용자, 세션, 신규 방문자, 페이지 뷰

[추가 제공 데이터]

- info_user.csv: 유저 관련 정보
- info_competition.csv: 대회 관련 정보
- info_submission.csv: 제출 관련 정보
- info_login.csv: 로그인 관련 정보

+DATE

외부 데이터 추가 사용

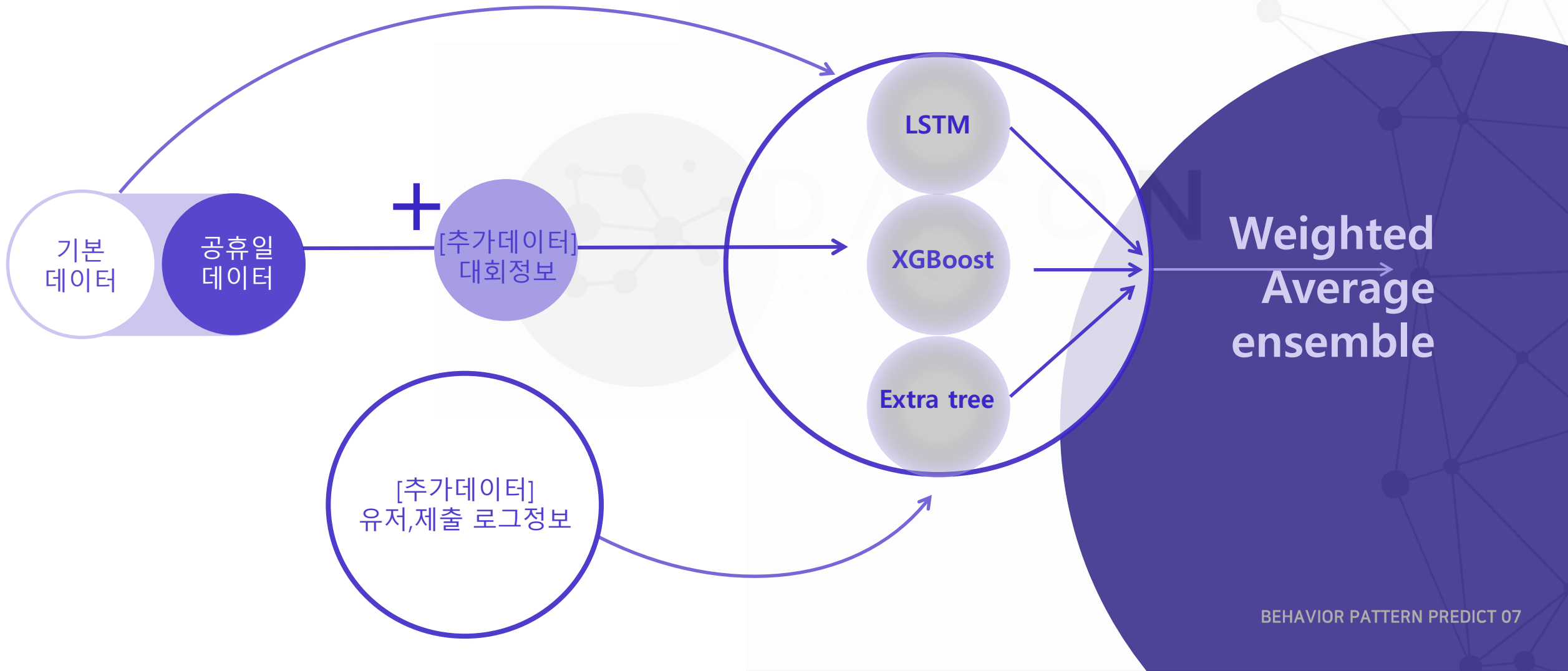
공공데이터 포털에서 '특일 정보' 오픈 API 사용



=> 해당 연도의 모든 국경일, 공휴일의 정보가 포함된 데이터

→ 공휴일, 연도별 휴일(선거일 등), 대체휴일의 정보 활용

PROJECT MAP

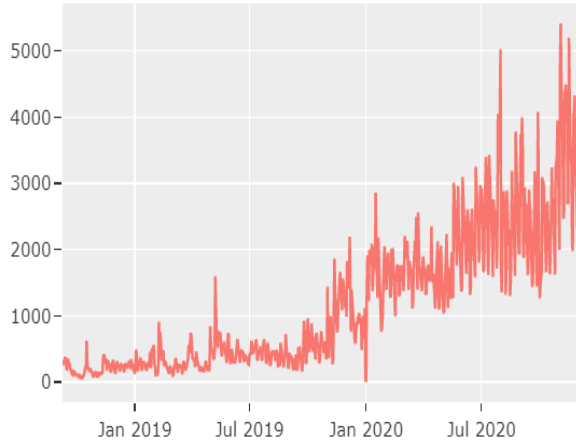


About EDA

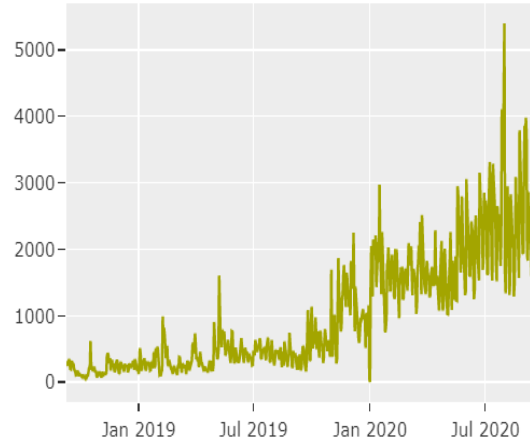


[전체 train 기간 시각화]

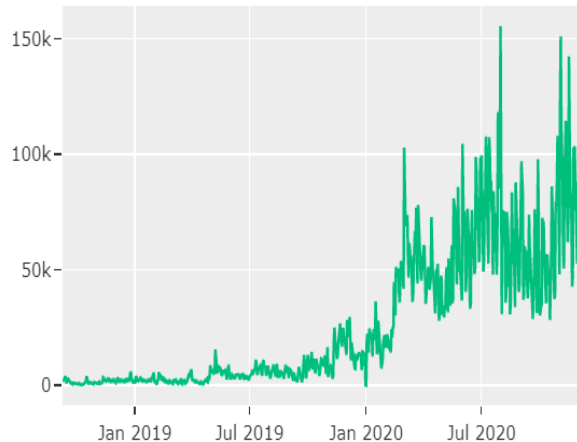
일별 사용자 합계



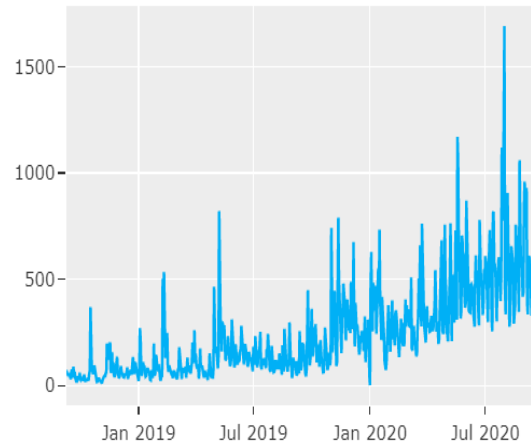
일별 세션 합계



일별 페이지뷰 합계



일별 신규방문자 합계



전체적인 추세 파악을 위한 전체 Train 시각화

2020 ▲

상승 패턴을 보이면서, 18년~19년은 비슷한 흐름
2020년 부터 급격히 증가하는 것을 확인

USER

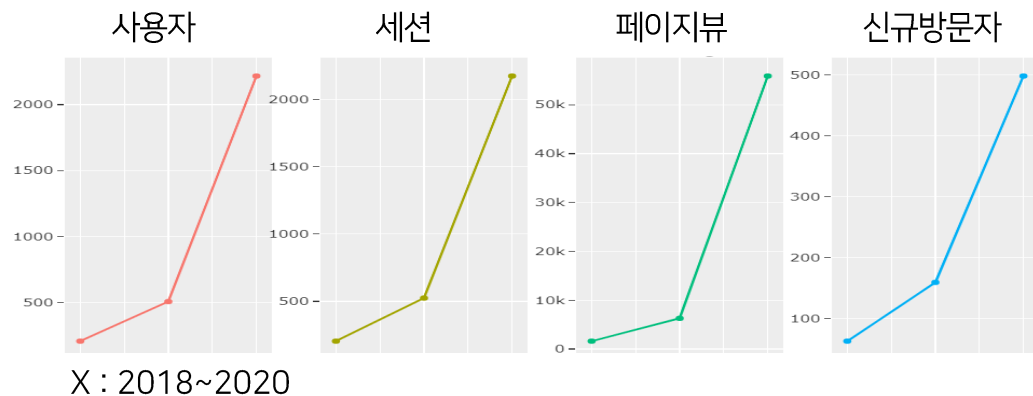
SESSION

사용자와, 세션은 거의 동일한 스케일, 패턴을 보임

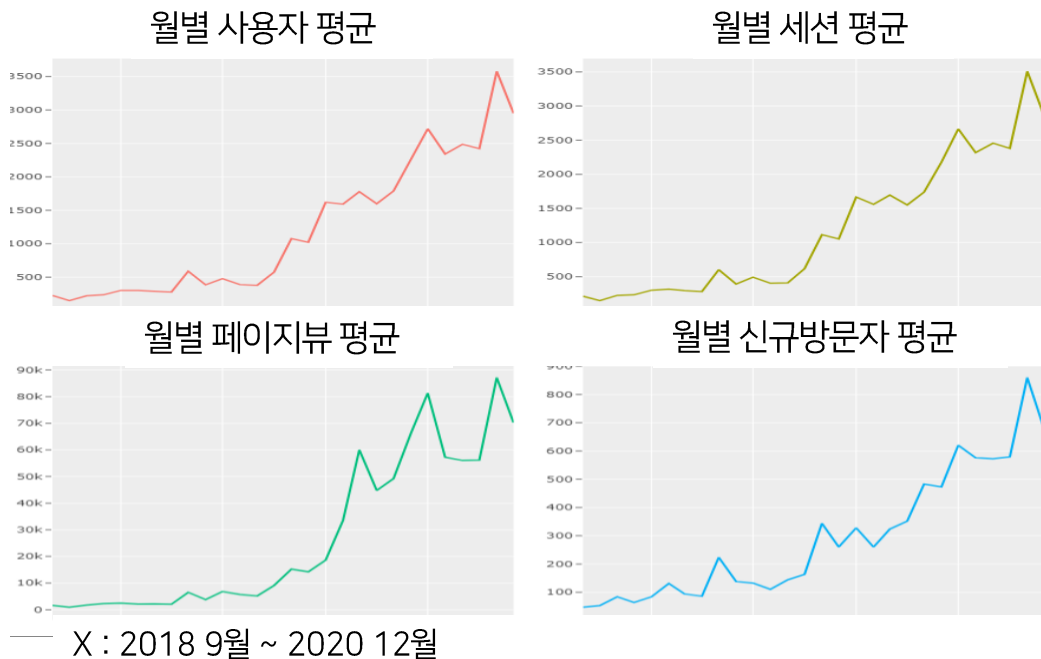
PAGE VIEW

페이지 뷰는 다른 컬럼에 비해 스케일이 매우 큼

[연도별 시각화]



[19년 9월 이후 시각화]



2018년에 비해 2020년 많이 성장

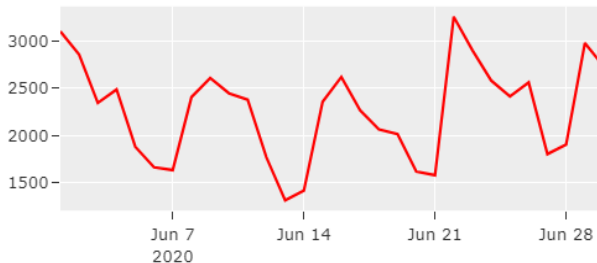
2019 ▲

2019년 전반기 이후 부터 모든 피처의 기울기가 급격히 높아짐
대략적으로 2019년 9월 시점으로 부터 급격한 증가 추세

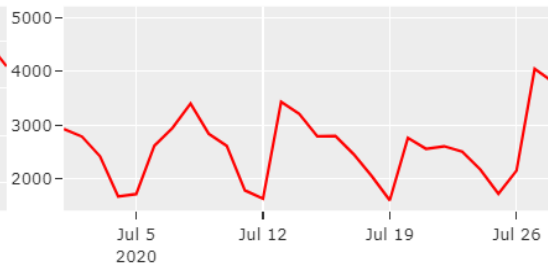
→ 최근 트렌드를 반영하기 위해 **Train data 분리 필요**

[사용자 월별 시각화]

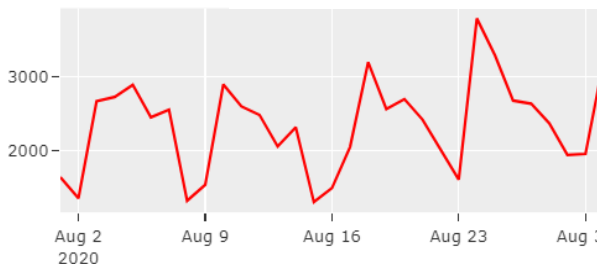
2020년 6월



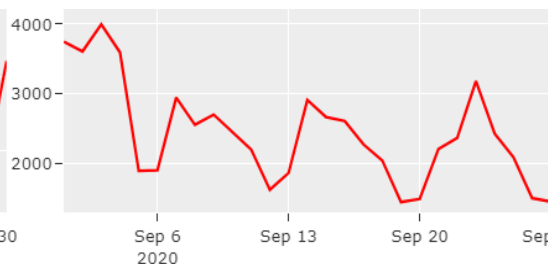
2020년 7월



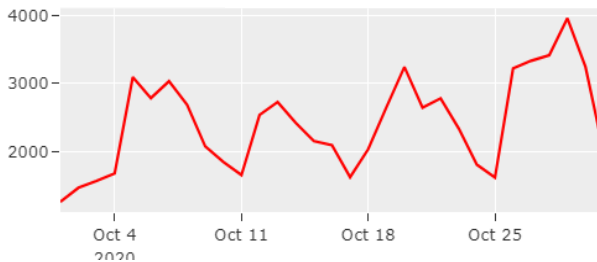
2020년 8월



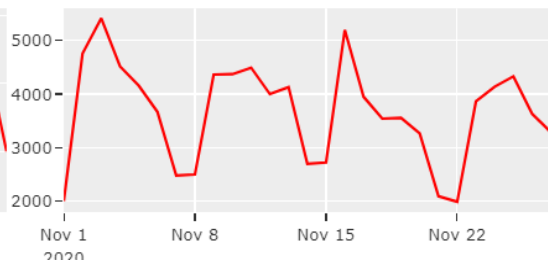
2020년 9월



2020년 10월



2020년 11월



월별 시각화를 통한 특징 파악

주말 ▼

평일 ▲

2020년에는 주말 및 휴일은 하락, 평일은 상승하는 패턴

USER

SESSION

PAGE VIEW

NEW USER

사용자, 세션, 페이지 뷰, 신규방문자는 대체적으로
모두 동일한 패턴으로 하락 상승을 반복

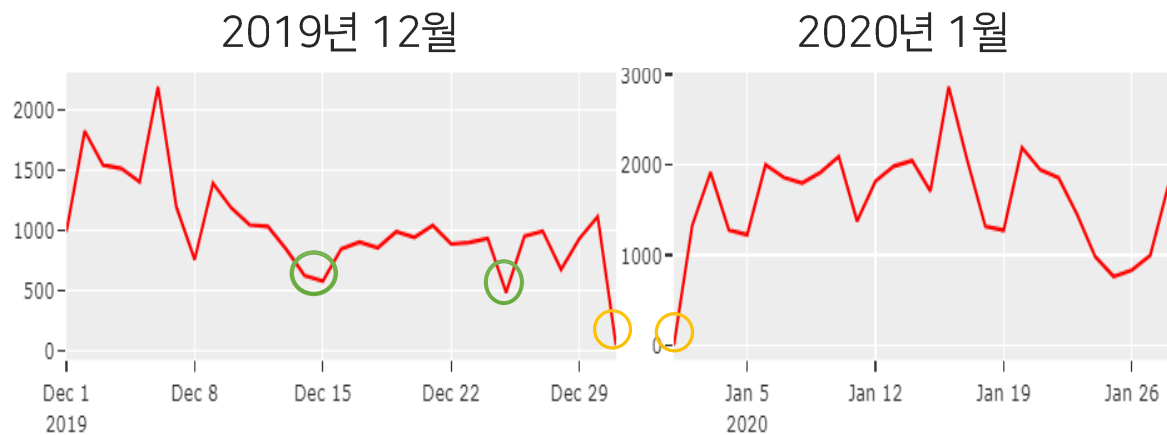
TOTAL

특정 날을 제외하고 한달 이내 상승과 하락의 높이는 비슷

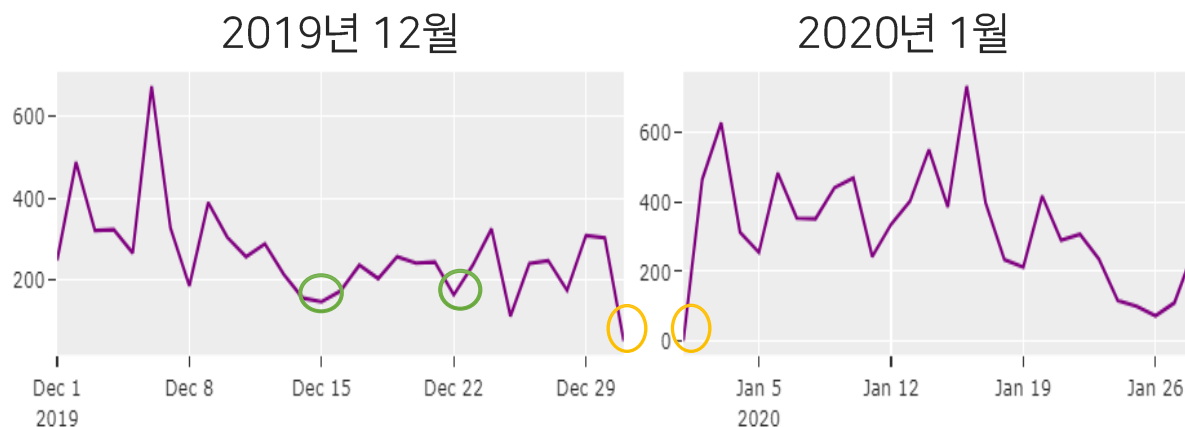
→ 모든 달은 각기 비슷한 패턴을 보임

→ 한달 이내에서 하한과 상한 값은 일정범위 내에서 움직임

[사용자 월별 시각화]



[세션 월별 시각화]



월별 시각화를 통한 특징 파악

비정상

12.31~1.1 급격한 하락 (노란색 부분) ○

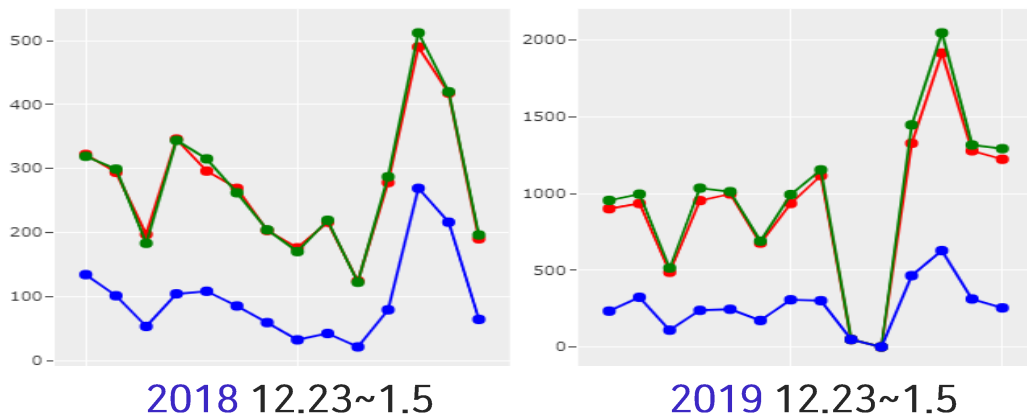
정상

공휴일과 주말이 비슷한 모습을 보임.
크리스마스와 주말 (초록색 부분) ○

각 월은 전체적으로
일정 패턴을 따르고 / 일부는 비정상 패턴을 보임

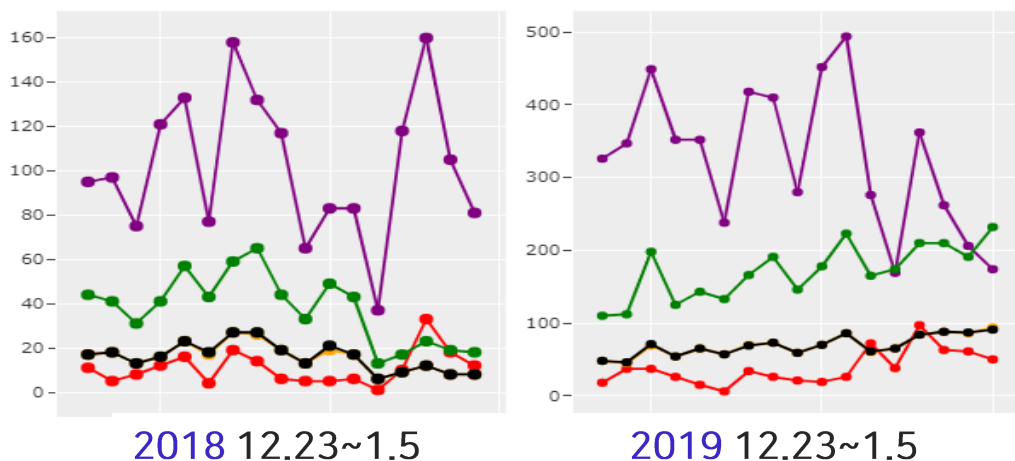
[feature set 연말 연초 분포]

Y : 일별 사용자, 세션, 신규방문자



[user info 연말 연초 분포]

Y: 로그인수, 제출 수 등 user info



연휴기간 영향 파악 연말, 연초

연휴 기간

연휴 기간에도 휴일과 같이 연속적으로 하락

연말/연초

연초 2019년 12월 31일 연말 모습

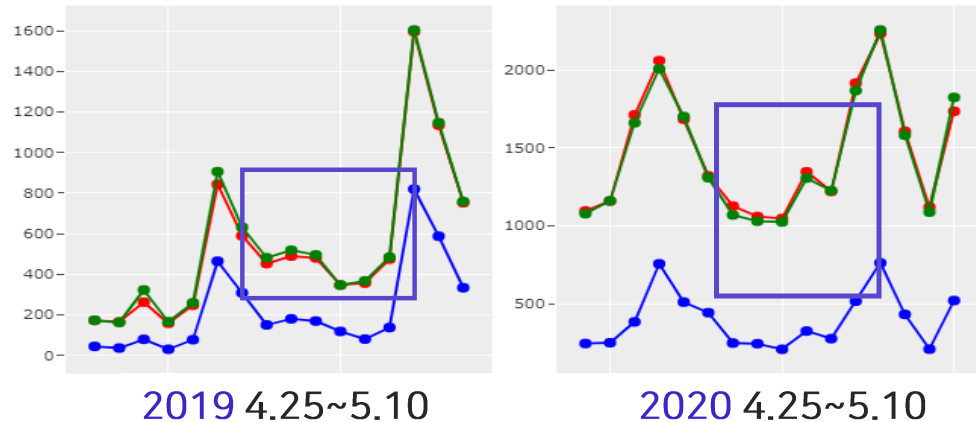
연휴 영향

긴 연휴에 포함되는 날은 공휴일 및 주말의 영향을 받음

로그인 수, 제출 수 또한 연휴 기간에 영향을 받아 하락

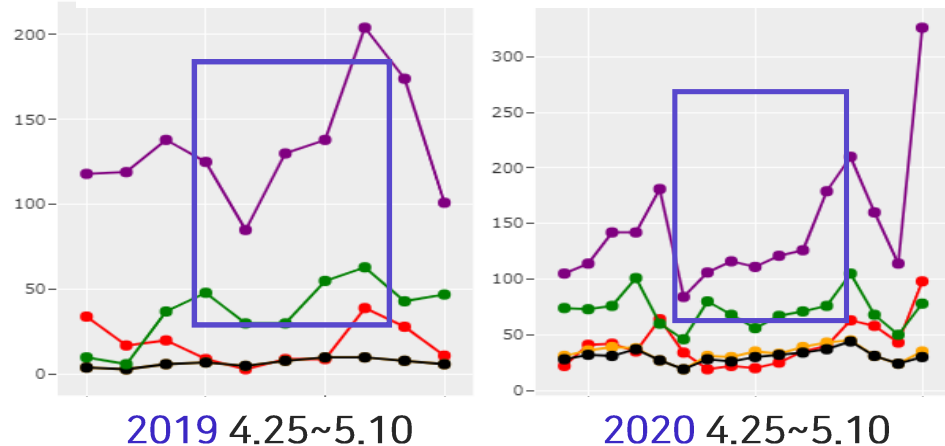
[feature set의 연속적 연휴기간 분포]

Y : 사용자, 세션, 신규방문자 분포



[user info의 연속적 연휴기간 분포]

Y: 로그인수, 제출 수 등 user info



연휴기간 영향 파악 연속적인 연휴

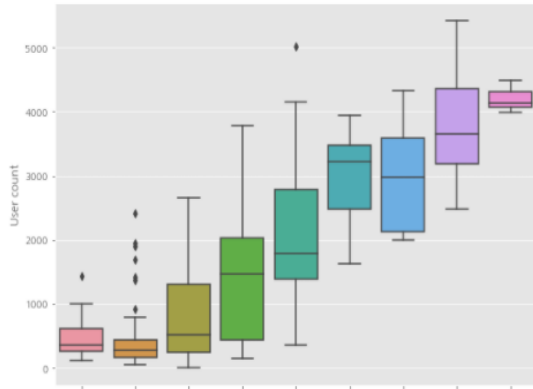
위의 기간들은 황금연휴 기간으로 4/30, 5/1, 5/2, 5/3, 5/4, 5/5로 6일동안 이어지는 연휴기간

예를 들어, 20년의 경우 피쳐 모두 석가탄신일을 기점으로 하락하면서 연휴 기간 내내 하락세

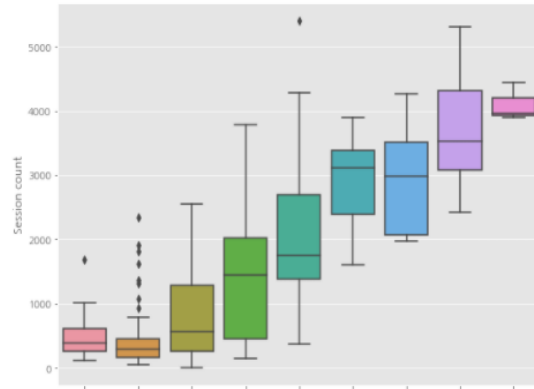
▶ 연휴 기간 사이에 있는 평일도 휴일로 포함시키는 것이 좋다

[대회 개수별 feature set 분포]

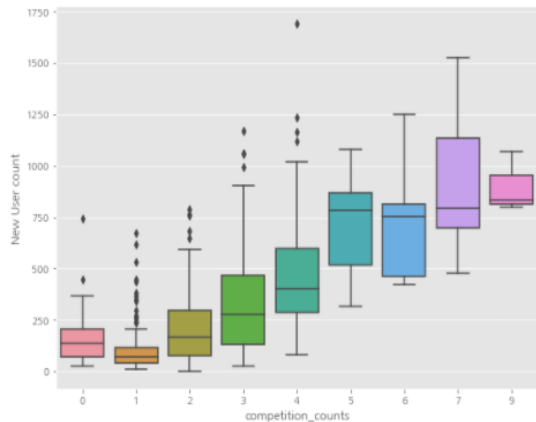
일별 사용자



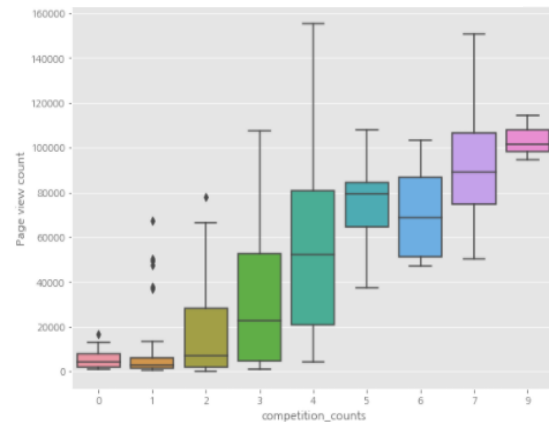
일별 세션



일별 신규사용자



일별 페이지뷰



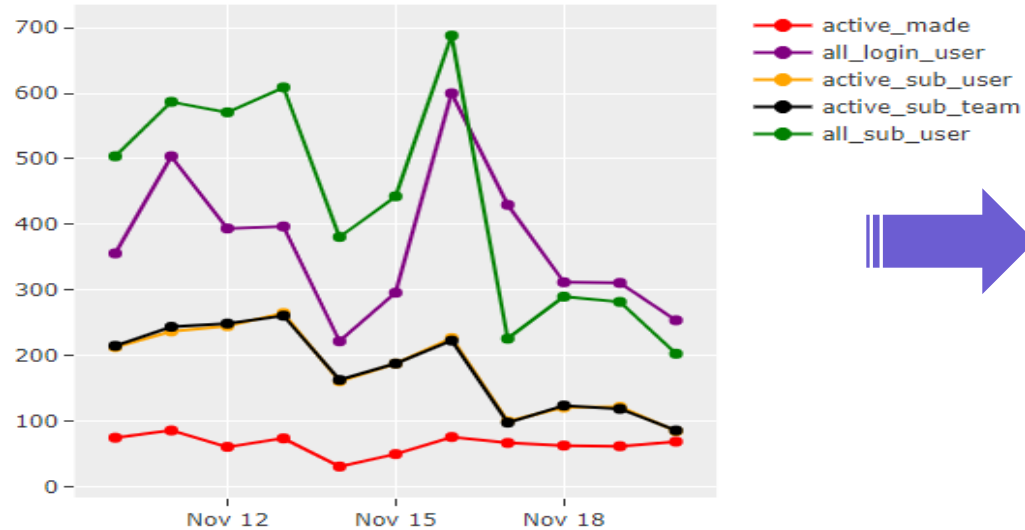
특수 패턴 일별 대회 개수의 영향력

- 일별 진행중인 대회 개수가 다르고 그에 따른 참가자 수 또한 다름
- 진행중인 대회 수에 따른 Y의 분포 차이 확인
- 진행되는 대회 수가 많을 수록 Y가 모두 커짐
- 피처가 해당 일자에 진행 중인 대회 개수에 의해 영향을 받음

▶ 추후 대회 개수에 따른 데이터 분할

[test 기간 확인 – 추가데이터 활용]

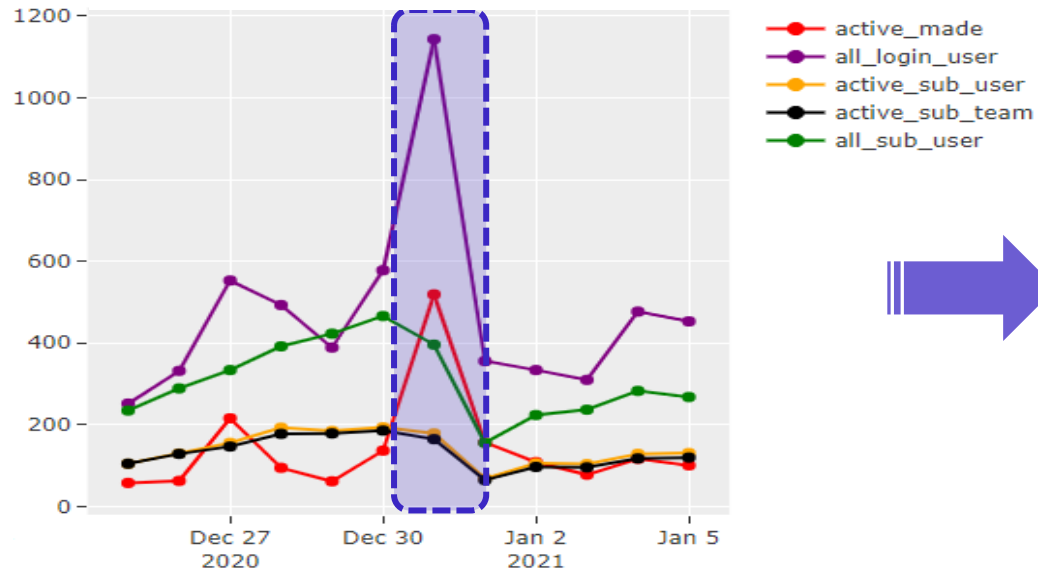
2019.11.10 ~ 2020.11.20



11/16일 1000명 넘는 참가자의 대회가 마감

'아이디 생성자'를 제외하고 나머지가 전날에 비해 크게 증가

2020.12.25 ~ 2021.01.05



12/31일 'NH 투자증권 대회'의 마감으로

'로그인 수', '제출 수' 등이 증가

→ 20년에는 연말 하락세의 패턴을 보이면 안된다고 판단

Feature engineering & modeling

- LSTM
- XGB
- EXTRA TREE
- Weighted Average Ensemble



LSTM

03 Feature engineering & modeling

날짜 관련 변수 생성 → 범주형 인코딩

LSTM에 시간흐름에 따른 특징을 반영하여 하기 위함

date	사용자	세션	신규방문자	페이지뷰	One-hot-Encoding				Binary Encoding		
					dayofweek (요일)	quarter	month	year	dayofyear (연 기준 일)	dayofmonth (일)	weekofyear (연 기준 week)
2018-09-09	281	266	73	1826	6	3	9	2018	252	9	36
2018-09-10	264	247	51	2092	0	3	9	2018	253	10	37
2018-09-11	329	310	58	1998	1	3	9	2018	254	11	37
2018-09-12	300	287	45	2595	2	3	9	2018	255	12	37
2018-09-13	378	344	50	3845	3	3	9	2018	256	13	37

➔ 사용자, 세션, 신규방문자, 페이지뷰를 포함하여 총 53개의 feature를 train데이터로 사용

[공휴일 추가 데이터]

2019-01-01	1월1일
2019-02-04	설날
2019-02-05	설날
2019-02-06	설날
2019-03-01	삼일절
2019-05-05	어린이날
2019-05-06	대체공휴일
2019-05-12	부처님 오신날
2019-06-06	현충일
2019-07-17	제헌절
2019-06-15	광복절
2019-09-12	추석
2019-09-13	추석
2019-09-14	추석
2019-10-03	개천절

페이지뷰에 공휴일 페널티 부여


- 다른 Y들과 다르게 '페이지 뷰'는 공휴일을 높게 예측
- EDA 결과, 공휴일에는 전체적으로 값이 낮아짐
- 사용자 증감에 따라 변동이 심한 특징을 고려

공휴일에 사용자가 급격하게 감소하는 것을 고려하여
공휴일에는 페이지뷰를 10% 감소시킴

표준화

사용자, 세션, 신규방문자, 페이지뷰에 대해 MinMaxScaling

date	사용자	세션	신규방문자	페이지뷰
2018-09-09	281	266	73	1826
2018-09-10	264	247	51	2092
2018-09-11	329	310	58	1998
2018-09-12	300	287	45	2595
2018-09-13	378	344	50	3845



date	사용자	세션	신규방문자	페이지뷰
2018-09-09	0.051689	0.049083	0.042604	0.011735
2018-09-10	0.048551	0.045564	0.029586	0.013446
2018-09-11	0.060550	0.057233	0.033728	0.012842
2018-09-12	0.055197	0.052973	0.026036	0.016682
2018-09-13	0.069596	0.063530	0.028994	0.024724

데이터 셋 구성

Train 데이터 셋

2018.09.09 ~ 2020.12.8

Target = 최종 예측

Test 데이터셋

train데이터의 뒤에서 61일치를 test셋으로 구성

2020.10.09 ~ 2020.12.08

2020.12.09 ~ 2021.01.08

데이터 reshape

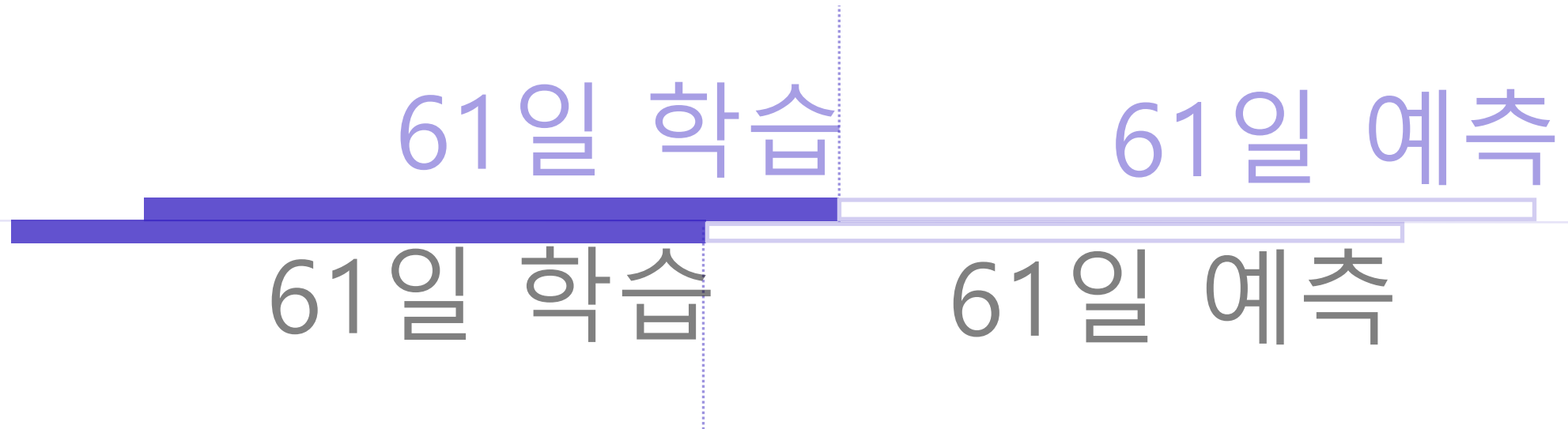
(sample_size, time_step, feature) 크기인 (700, 61, 53) 으로 데이터를 reshape

`time_step = window_size (61)`

`history_size`: 몇 개의 과거 데이터를 학습할 것인지

`target_size`: 얼마나 멀리 있는 예측을 배워야 하는가

H history_size = 61
T target_size = 61



Xgboost

03 Feature engineering & modeling

데이콘 추가데이터 (대회 정보)

A	B	C
period_start	period_end	name
2018-08-14 0:00	2018-09-13 23:59	대출 상점 총 매출 예측 경진대회
2018-09-15 0:00	2018-10-13 23:59	병원 개/폐업 분류 예측 경진대회
2018-10-18 0:00	2018-12-31 23:59	아파트 경매가격 예측 경진대회
2018-11-13 0:00	2019-01-31 23:59	아파트 실거래가 예측
2018-12-25 0:00	2019-01-10 23:59	신용카드 거래 데이터 시각화
2019-02-08 0:00	2019-07-18 23:59	KBO 타자 OPS 예측 경진대회
2019-03-26 0:00	2019-05-20 23:59	KBO 외국인 투수 스카우팅 최적화
2019-05-06 0:00	2019-07-08 23:59	KCB 금융스타일 시각화 경진대회

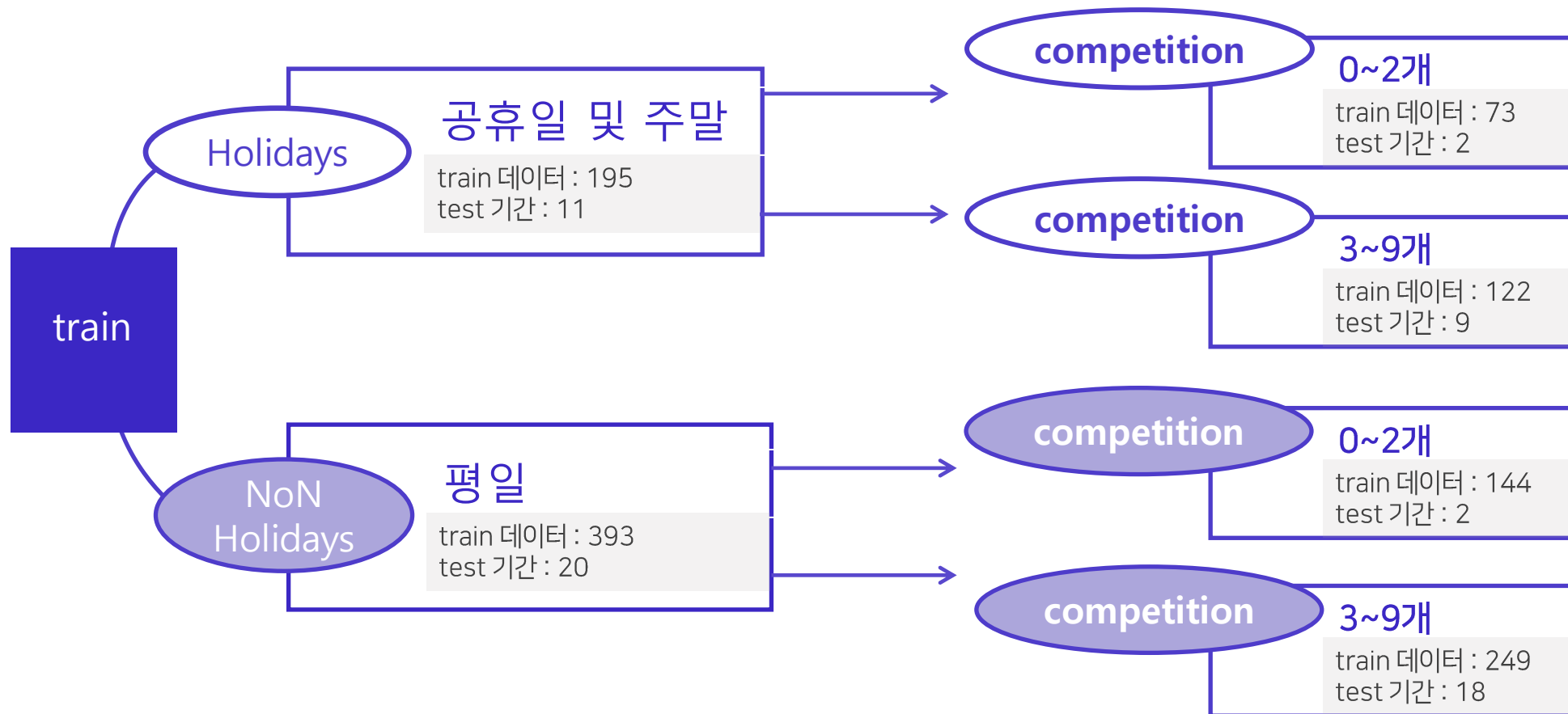
학습데이터에 포함할 기간 선택

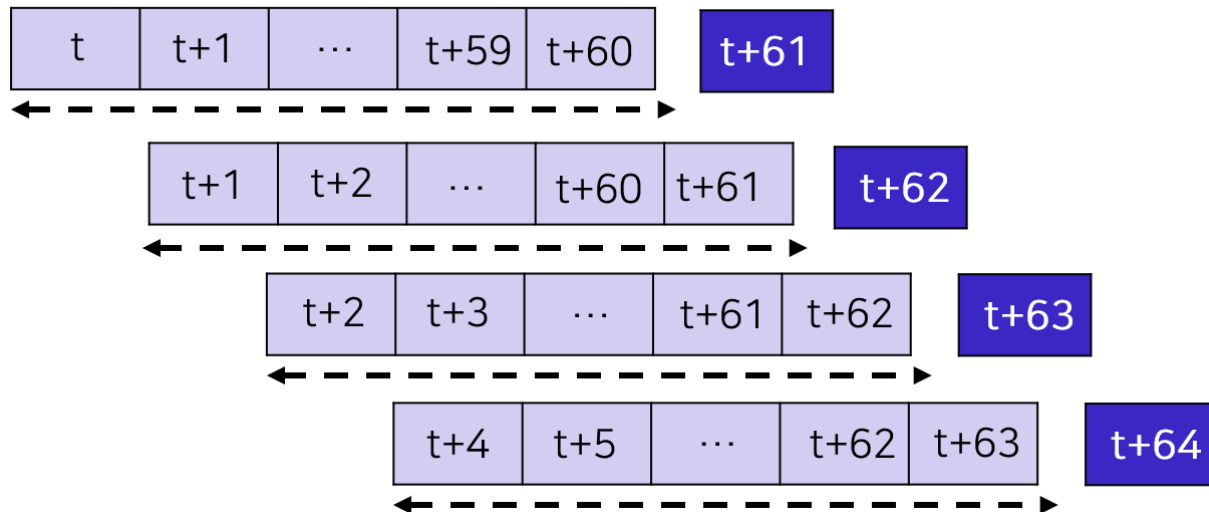
- train 데이터의 기간은 2018.9.9 ~ 2020년 12월 8일
- 데이콘 설립일, 중요 공지사항의 수정 및 체계화 날짜 확인
- 19년도 초반까지 신규 방문자와 제출 수는 대부분 한 자릿수
- 2019년 5월 이전에는 불규칙한 흐름을 보이고 있음
- 설립초기부터 19년도 5월까지의 개최된 대회를 볼 때 영향력 있는 대회가 없음

➔ train데이터 기간을 2019년 5월 1일 이후로 설정

공휴일 및 주말 vs 평일

- 당일 진행중인 대회 개수에 따라 분할
- 조건에 맞는 관측치의 수를 고려하여 학습데이터 분할
- train 데이터의 기간으로 알고리즘 fitting





시점을 기준으로 데이터 예측

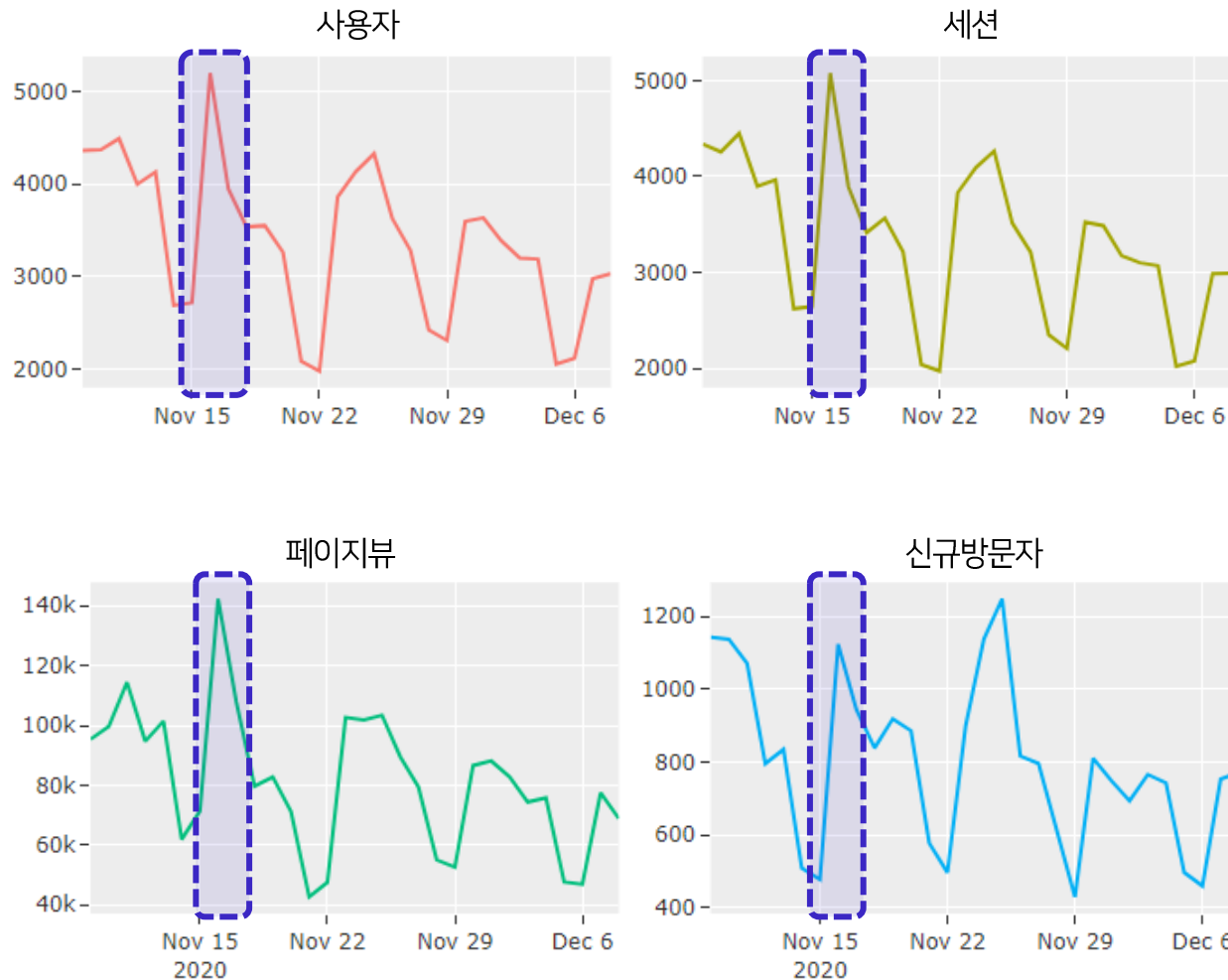
- 예측 값을 새로운 훈련데이터로 사용하여 다음날 예측
- 학습일을 60으로 잡으면 60일을 본 후 61일을 예측
- rmse 값 기준으로 예측 값으로 몇일을 삼을지 결정
- 분할된 데이터와 Y에 따라 알고리즘 객체 16개 생성

Extra Tree Regressor

03 Feature engineering & modeling

xgb와 lstm 한계 극복

=> valid data로 사용한 test 기간 전의 한달(11/8 ~ 12/8)



valid set으로 활용한 11/8~12/8일에서
불규칙한 날짜와 특징 확인

valid 기간 중 가장 높은 11월 16일의 경우
진행되고 있는 대회는 5개에 불과

1000명이 넘는 대회의 팀 병합 마감, 시작, 대
회 마감날의 경우 모든 y가 폭발적으로 증가



위의 정보를 반영할 수 있는
feature set을 만들어 새롭게 학습

새로운 feature set 구성

	아이디 생성	실제 로그인 수	총 로그인 수	실제 제출 수	실제 제출 팀	총 제출 수	총 제출 팀
2020.01.01	38	115	169	65	65	174	174
...
2020.12.08	70	265	299	98	100	223	223

추가 데이터를 활용하여 '총 수'와 '실제 수'로 데이터를 조합해 7개의 변수로 구성

- * 유저 관련 정보
- * 제출 관련 정보
- * 로그인 관련 정보

참가자가 1000명이 넘는 날이 없는 20년도 이전의 관측치는 학습에서 제거

'총 수'와 '실제 수'의 차이

ex) 하루에 A가 3번 제출, B가 2번 제출
→ 총 제출 : 5 / 실제 제출 : 2

EXTRA
TREE

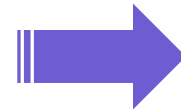
Weighted average ensemble

03 Feature engineering & modeling

알고리즘 별 valid set에 대한 rmse

모델	RMSE
LSTM	2.5787
XGB	2.6159
LSTM+XGB	1.9797
EXTRA	1.6943
LSTM+XGB+EXTRA	1.1625

- LSTM과 XGB를 스택킹 결과보다 EXTRA Tree의 개별 결과값이 더 좋았음
- 주말 및 몇몇 평일에서의 예측은 LSTM+XGB가 조금 더 정답에 가까움
- 특징 있는 날에 있어서 EXTRA Tree가 압도적인 성능을 보여줌

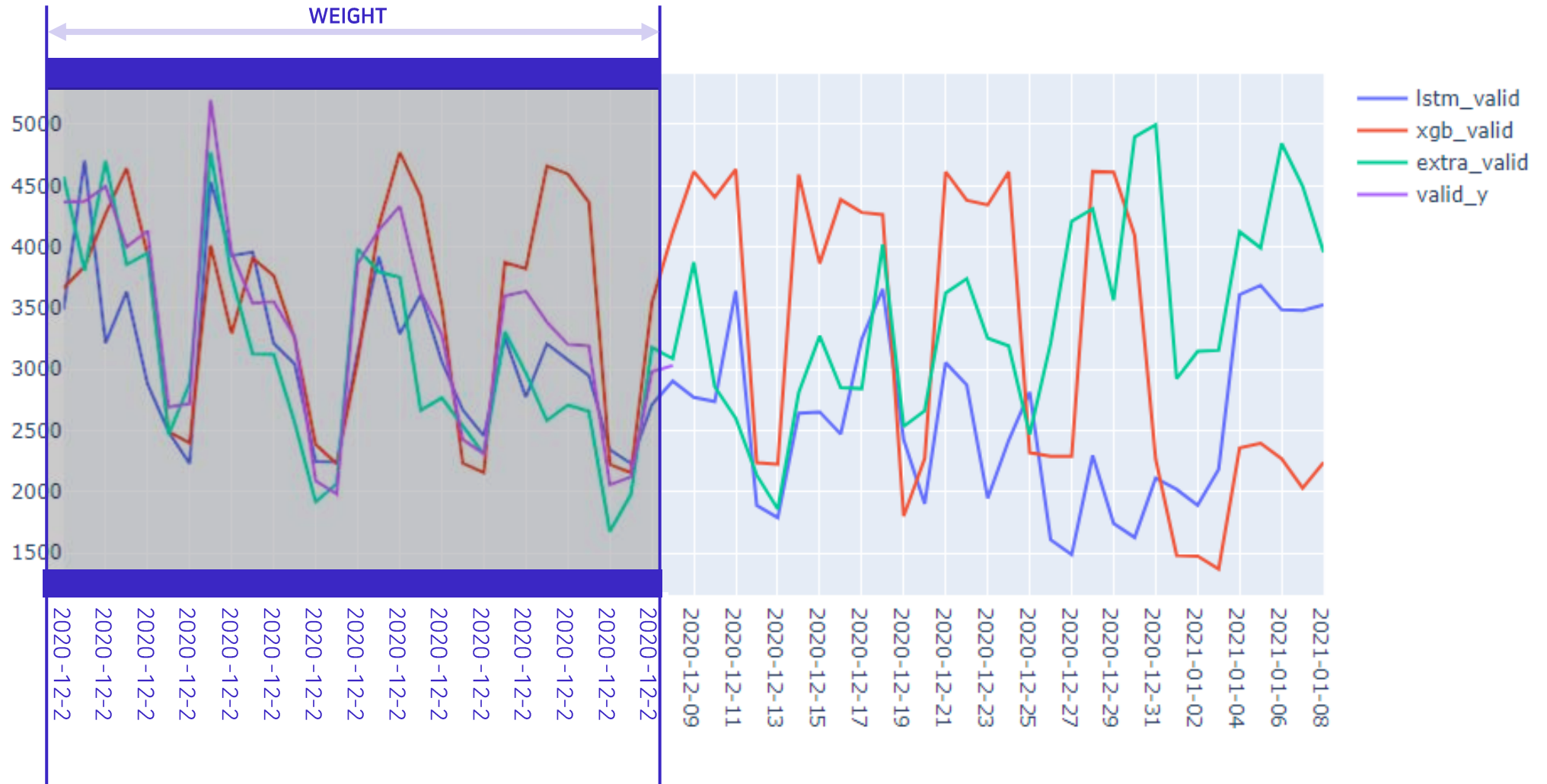


극단적으로 예측하는 EXTRA Tree의 결과를 조금 더 완만하게 만들기 위해 세 알고리즘을 스택킹

서로의 단점을 보완하여 최고 성능을 냄

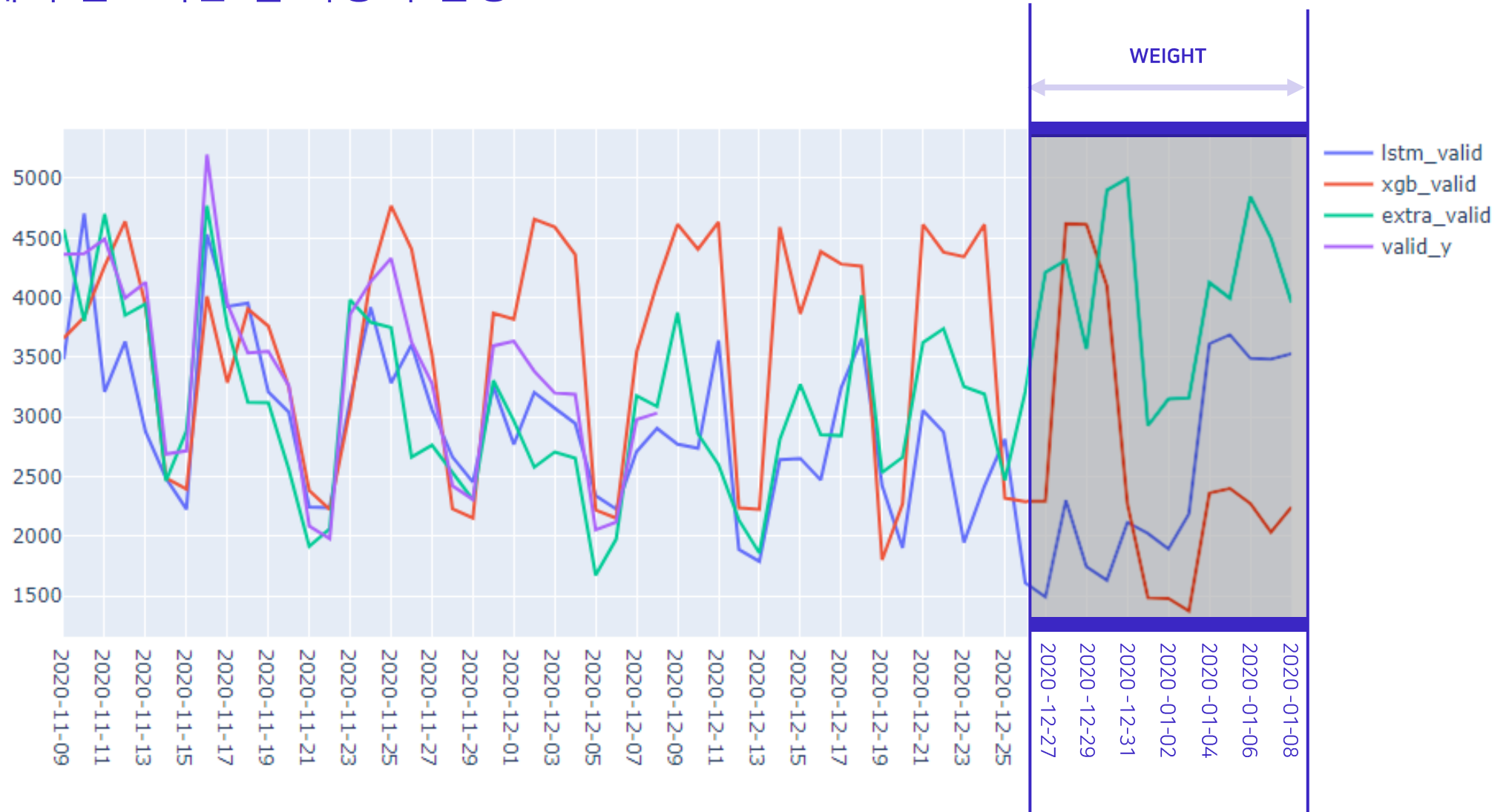
Weighted average ensemble

Stacking에서 알고리즘 별 가중치 설정

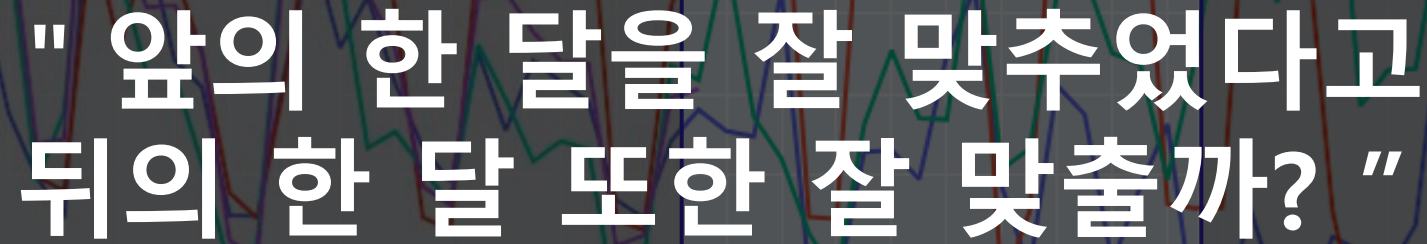


Weighted average ensemble

Stacking에서 알고리즘 별 가중치 설정



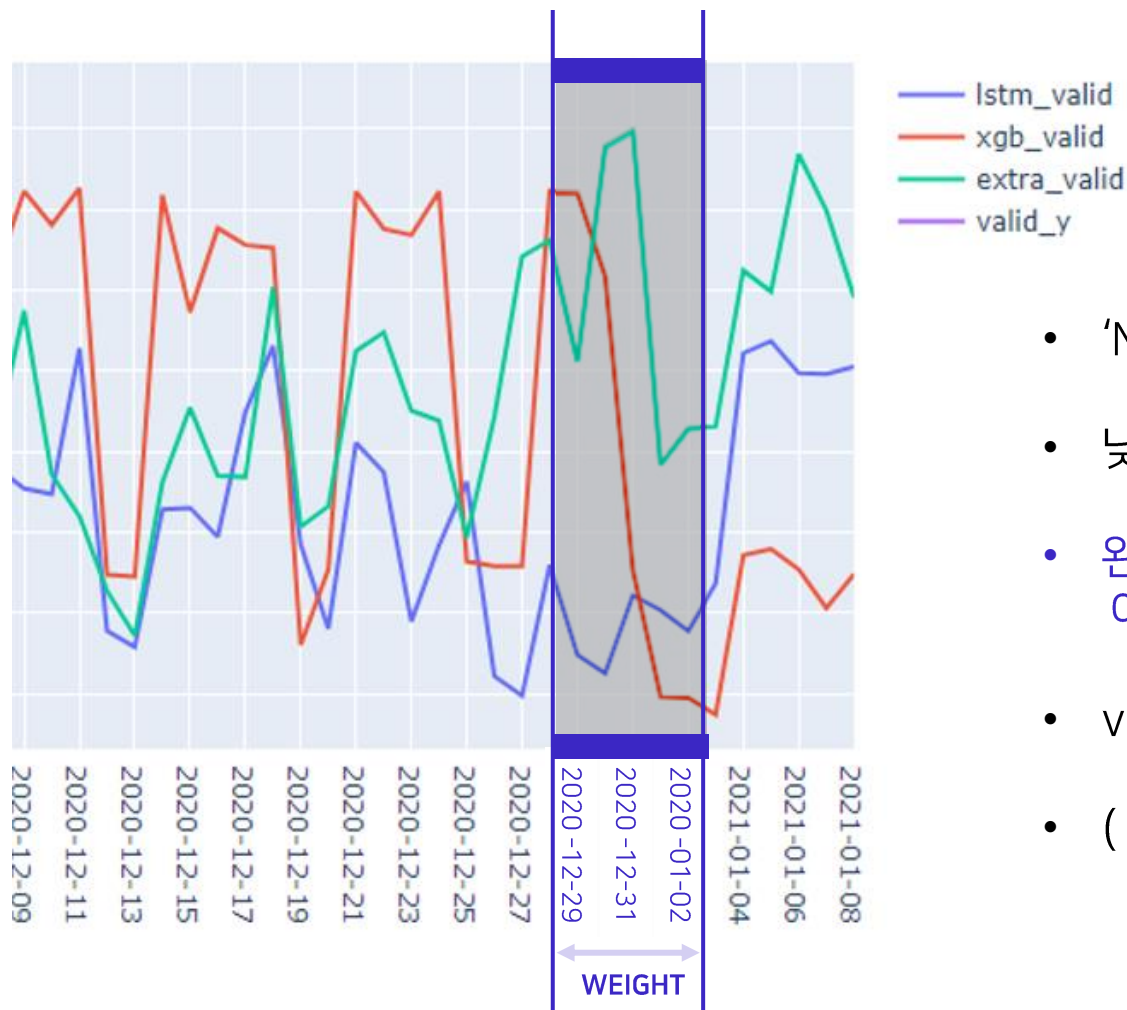
Stacking에서 알고리즘 별 가중치 설정



Stacking에서 알고리즘 별 가중치 설정



알고리즘 별 Valid set에 대한 rmse



- 'NH투자증권' 대회 종료날로 폭발적인 증가가 예상되는 12/31
- 낮게 예측하는 xgboost와 lstm과 달리 특징을 잘 잡아낸 extra tree
- **완만한 특징**을 조금만 부여하고 **이벤트가 존재**하는 날을 확실하게 맞추는 것을 목적으로 함
- valid에서의 특징을 바탕으로 **모델별 가중치 선정**
- $(lstm*0.3 + xgb*0.7) * 0.2 + extra * 0.8$

CONCLUSION



느낀점 및 한계점

EDA와 feature engineering

다양한 시각화를 통해 얻은 다양한 인사이트

→ EDA기반 Feature engineering을 통한 성능 향상

앙상블

각기 다른 feature set으로 구성된 다양한 알고리즘 사용하여 앙상블

→ 성능이 크게 향상

시계열 모델 활용의 한계

2년치 데이터 밖에 주어지지 않아서 ARIMA와 fb prophet 사용 시,
좋은 성능을 내지 못함





감사합니다