데이터 마이닝 텀 프로젝트 최종발표

기상위성 자료를 활용한 여름철 자외선 산출 기술 개발

2022 날씨 빅데이터 콘테스트 수상작 개선

김도훈, 김민지, 안나연, 이다혜, 이예찬, 조재홍, 최지혜

1 [제안 발표] 요약

주제: 기상위성 자료를 이용한 여름철 자외선 산출 기술 개발

0. 주제의 필요성

자외선 측정장비가 설치된 지점은 전국 7곳 뿐 !

관측장비가 없는 지점의
UVI를 알 수 있게 되어
기존 자외선 지수 관측
한계 보완

1. 활용 데이터 변경

~~강수량~~	전운량	중하층운량

15개 지점 분리
결측 多 => 강수량 삭제

2. 결측치 처리방법 개선

MICE
KNN

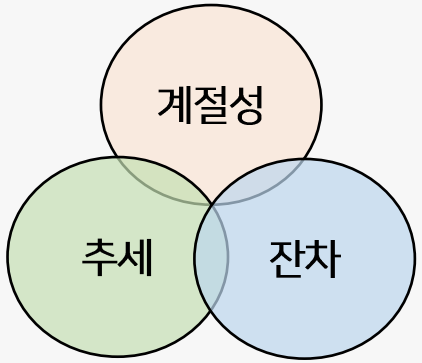
→

시간
보간법

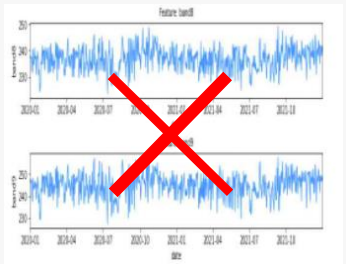
3. 이상치 처리방법 개선

STL
분해

→



4. 변수 축소: PCA



→

PCA

육안으로 확인 후
상관없는 변수 제거

5. 모델 선정

Random Forest	XGBoost	LGBM

시계열데이터에 더 적합한 모델로 변경

↓

LSTM	ARIMA	LGBM

1 [제안 발표] 피드백 반영 / 추가 고려사항

	발표안	변경안
1	강수량 / 위도 경도 / 전운량 / 중하층 운량 지점 분리	강수량 / 위도 경도 / 전운량 / 중하층 운량 / 미세 먼지 / 해발 고도 지점 통합, 변수 2개 더 추가
2	강수량 결측치 90% 이상 강수량 변수 삭제	관측 데이터가 아닌 예보 데이터를 활용해 결측 없는 강수량 변수 확보
3	상관계수가 작은 변수는 제외하고 연속형 변수에 대해서만 PCA 시행	범주형 변수와 연속형 변수를 함께 고려하는 FAMD 시행
4	여름만을 validation set으로 두어 Nested Cross-Validation 이용	validation set을 여름이 아닌 다른 계절까지 추가적으로 고려
5	ARIMA 시계열성을 고려하여 ARIMA 모델 적용 결정	Prophet ARIMA 모델의 적용한계로 Prophet 모델 적용 결정

2 활용 데이터 추가

기상청

기상 위성 데이터

천리안위성 2A

수집데이터 기간
2020.01.01 ~ 2021.12.31

테이블명	내용
YearMonthDayHourMinute	년/월/일/시간/분
STN	지점번호
Lon	경도
Lat	위도
UV	자외선
Band1	파랑 가시밴드
Band2	초록 가시밴드
Band3	빨강 가시밴드
Band4	식생 가시밴드
Band5	눈/얼음 채널
Band6	권운 밴드
Band7	야간안개/ 하층운 밴드
Band8	상층 수증기 밴드

테이블명	내용
Band9	중층 수증기 밴드
Band10	하층 수증기 밴드
Band11	구름상 밴드
Band12	오존 밴드
Band13	대기창 밴드
Band14	깨끗한 대기창 밴드
Band15	오염된 대기창 밴드
Band16	이산화탄소 밴드
SolarZA	태양 천정각
SateZA	위성 천정각
ESR	대기 외 일사량
LandType	지면타입

기상자료개방포털

기상 데이터

기상청의 자외선 산출방식을 바탕으로
수상작이 선정한 변수에
전운량, 중하층운량을 추가 고려

테이블명
기온
풍속
풍향
습도
증기압

테이블명
이슬점온도
해면기압
지면온도
전운량
중하층운량

강수량 데이터

결측치가 많았던 실제 강수량 값 대신
초단기 기상 예보 데이터로 대체

테이블명
강수량

Air korea

미세먼지 데이터

먼지가 자외선에 영향을 줄 수 있으므로
미세먼지/초미세먼지 변수 추가

테이블명	내용
Pm2.5	초미세먼지
Pm10	미세먼지

해발고도 데이터

관측지의 해발고도에 따라
기상 지수가 달라질 수 있으므로
관측지 해발고도 변수 추가

테이블명	내용
Height	해발고도

피쳐 엔지니어링

변수 변환 과정

테이블명	내용		테이블명	내용
YearMonthDay HourMinute	년/월/일/ 시간/분	년/월/일/시간/분을 분리 후, datetype 으로 변환	Datetime	'yyyy-mm-dd hh:mm:00'
SolarZA	태양 천정각	90 – '태양천정각' = '태양고도'	Solaraa	태양고도
풍향	16방위로 표현	360도로 표현된 방위: 원형 변수 연속적인 값으로 표현해주기 위해 코사인 / 사인 변환 진행	풍향_cos	풍향의 cos값
			풍향_sin	풍향의 sin값

최종 변수 및 유형

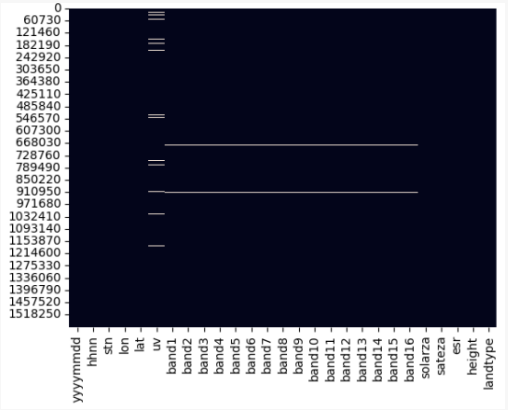
```
a.columns

Index(['Datetime', 'uv', 'band1', 'band2', 'band3', 'band4', 'band5', 'band6',
       'band7', 'band8', 'band9', 'band10', 'band11', 'band12', 'band13',
       'band14', 'band15', 'band16', 'esr', 'pm10', 'pm2.5', '강수량', '기온(°C)',
       '풍속(m/s)', '습도(%)', '중기압(hPa)', '이슬점온도(°C)', '해면기압(hPa)', '전운량(10분위)',
       '중하층운량(10분위)', '지면온도(°C)', 'solaraa', '풍향_cos', '풍향_sin', 'lon', 'lat',
       'sateza', 'height', 'landtype', '해발고도'],
      dtype='object')
```

- 연속형 변수: 범주형 변수 3개를 제외한 36개 변수
- 범주형 변수: 전운량, 중하층운량, landtype, 총 3개 변수

2 [EDA] 결측치 탐색 및 처리

결측치 탐색

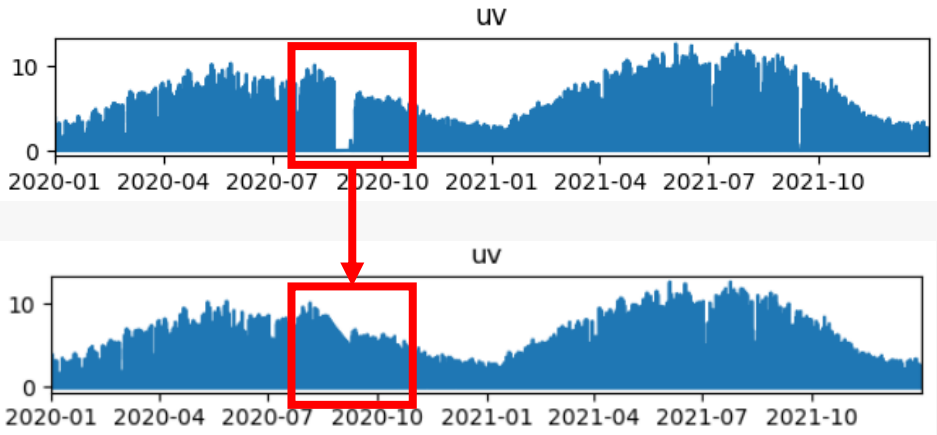
결측치가 여러 변수에 불규칙적인 분포로 존재
했기 때문에 제거보다 대체의 방법을 택함



결측치 테이블	결측치 비율
UV	약 3%
Band1 ~ Band 16	약 1.5%
타 변수	약 0.05% 이하

결측치 처리 과정

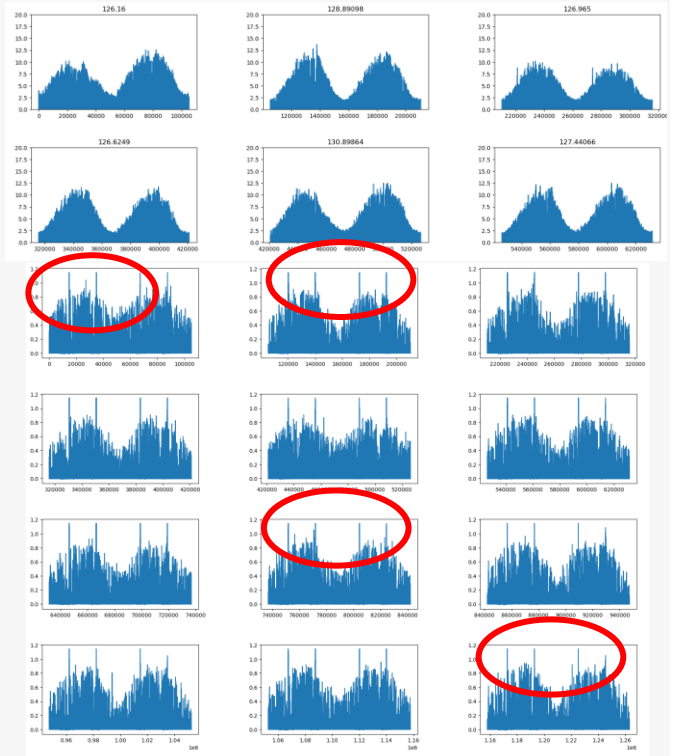
```
df_c.interpolate(method='time', inplace=True)
```



결측치가 일정 구간에 다량으로 발생한 경우,
결측치가 처음과 마지막 값에 존재하는 경우
제대로 보간하지 못하는 상황 발생 !!

처음값과 마지막 값은 가장 가까운 시간대의 값으로 대체

결측치 처리 결과



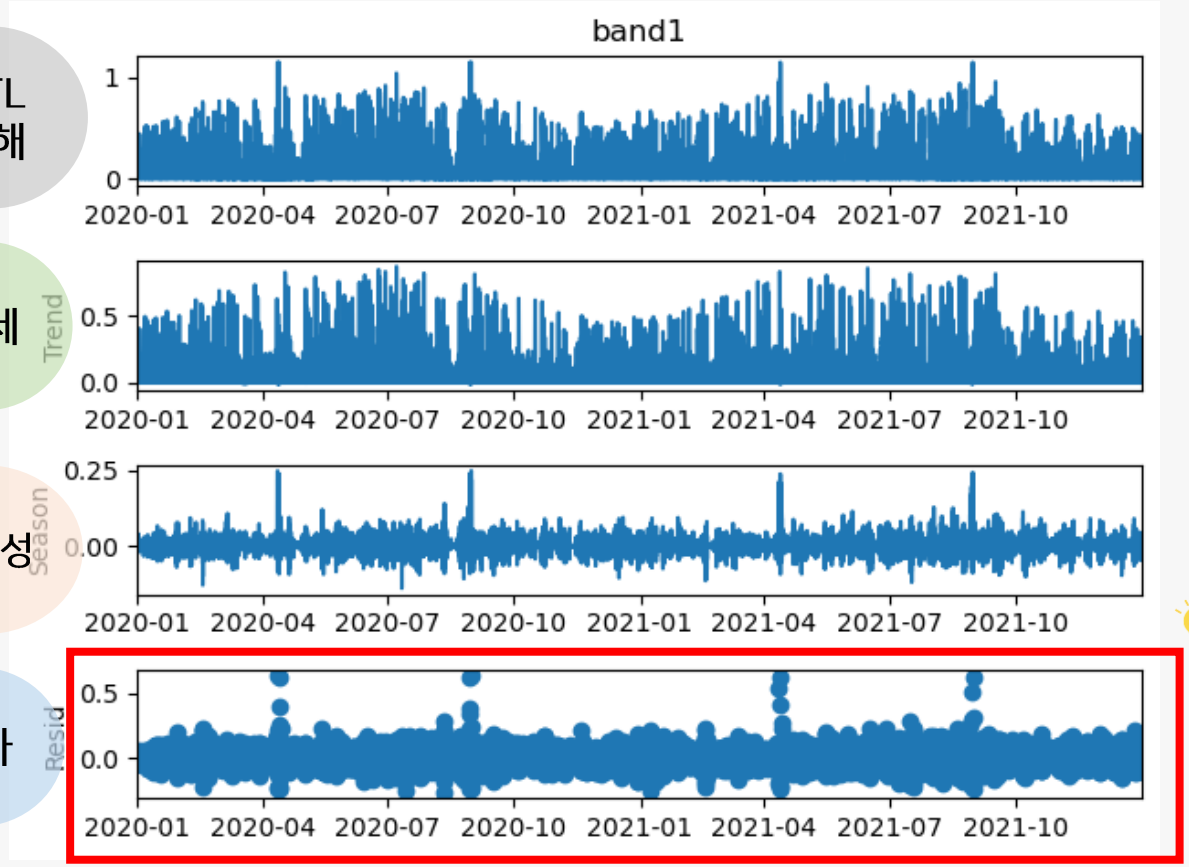
UV와 band 값 등에 튀는 값들이 발견됨
이상치 처리 필요

2 [EDA] 이상치 탐색 및 처리

이상치 탐색

데이터를 추세 / 계절성 / 잔차로 분해 후
이상치 탐색

STL 분해

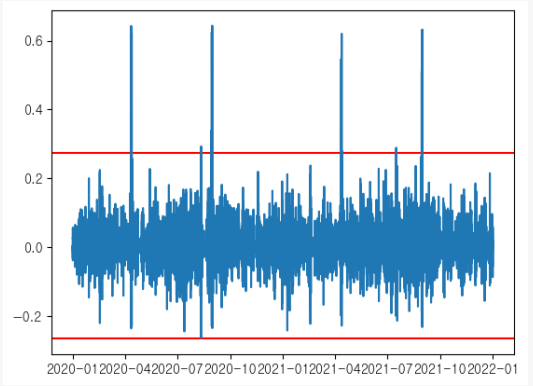


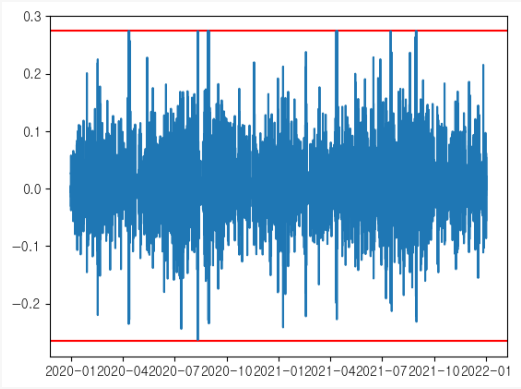
추세

계절성

잔차

이상치 처리





상위 1퍼센트와 하위 1퍼센트의 데이터를 윈저화 하여
극한 값들을 상한값과 하한값으로 대체함

2 [EDA] 변수간 상관관계 파악

변수간 상관관계 파악

```
[ ]  df_corr = df[df.columns[3:]].corr()
```

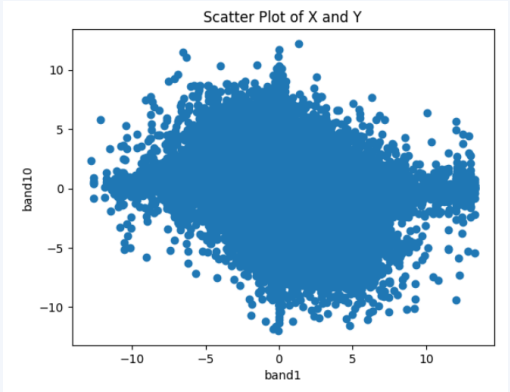
	uv	band1	band2	band3	band4	band5	band6	band7	band8	band9	...	지면온도 (° C)	solaraa	풍향
uv	1.000000	-0.070620	-0.071400	-0.065856	-0.065423	-0.035462	-0.044744	0.017876	0.017438	0.019369	...	0.091056	0.041297	-0.00
band1	-0.070620	1.000000	0.990765	0.853714	0.910830	0.347957	0.591336	-0.126591	-0.147602	-0.172895	...	-0.053388	0.028522	-0.00
band2	-0.071400	0.990765	1.000000	0.872446	0.935128	0.341826	0.605146	-0.123475	-0.145955	-0.171173	...	-0.054007	0.025731	-0.00
band3	-0.065856	0.853714	0.872446	1.000000	0.895231	0.278938	0.551792	-0.106183	-0.126816	-0.148363	...	-0.049015	0.019808	-0.00
band4	-0.065423	0.910830	0.935128	0.895231	1.000000	0.288632	0.648372	-0.091452	-0.131444	-0.154677	...	-0.046268	0.024229	-0.00

정규화

Standard Scaler

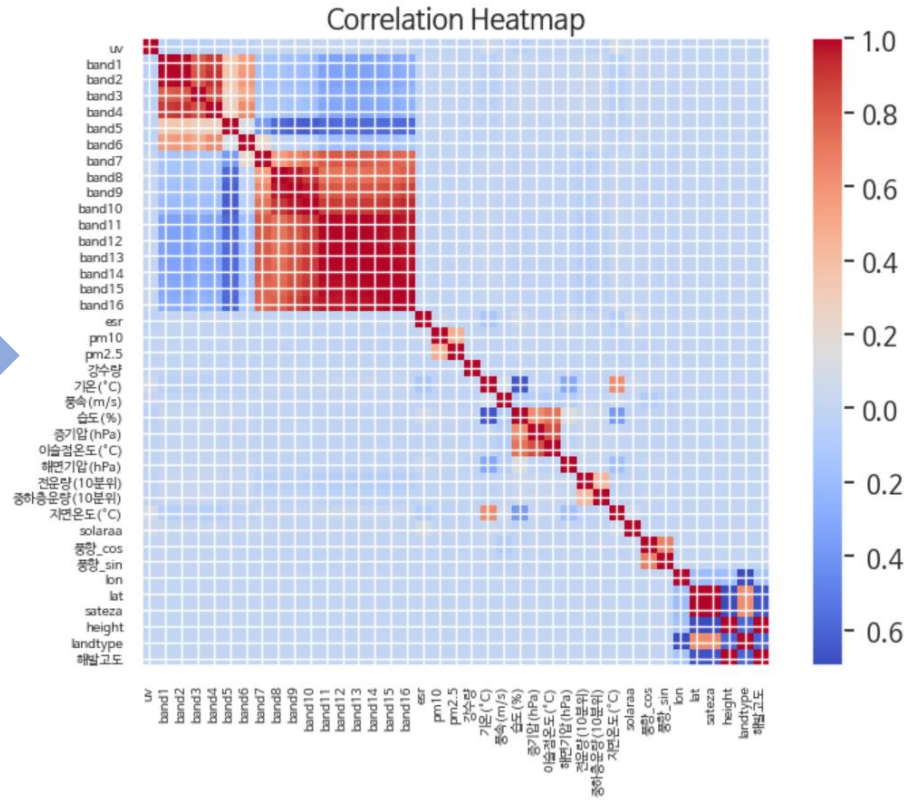
데이터의 skewness를
고려하여 Standard Scaler 적용

선형관계 확인



비선형적 관계인 변수들 多
스피어만 상관계수 활용해 상관계수 히트맵 생성

히트맵 생성



히트맵을 통한 상관관계 확인 시
변수들간 상관관계가 커 차원 축소 필요

2 변수 축소: FAMD

FAMD

연속형 변수와 범주형 변수가 혼합된 데이터에 대한
차원 축소를 위한 효과적인 기법

PCA(주성분 분석) + MCA(다차원 척도법)

💡 다양한 유형의 변수를 동시에 고려하면서 데이터 차원 축소 가능

주성분 개수 선택

```
explained_variance_ratio = pca.explained_variance_ratio_
```

```
주성분 1의 설명된 분산 비율: 31.69%
주성분 2의 설명된 분산 비율: 20.11%
주성분 3의 설명된 분산 비율: 20.10%
주성분 4의 설명된 분산 비율: 19.61%
```

누적 설명 분산 91.51%
주성분 개수 4개로 결정

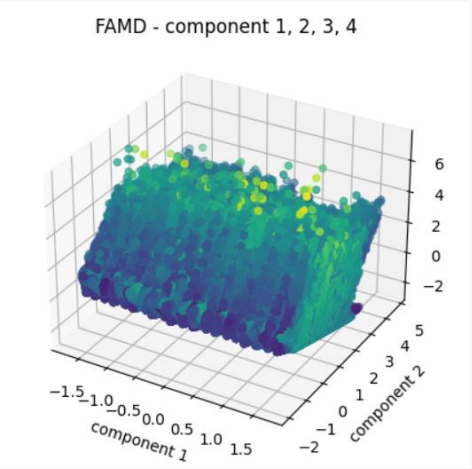
Train / Test

Train에 대한 FAMD의 매개변수를 저장하고 Test에 동일한 FAMD 적용 후 평가

FAMD 결과

```
# 범주형 변수 처리
categorical_cols = ['전운량(10분위)', '중하층운량(10분위)', 'landtype']
label_encoders = {}

for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    df[col] = label_encoders[col].fit_transform(df[col])

# 인덱스를 datetime으로 설정
df['Datetime'] = pd.to_datetime(df['Datetime'])
df.set_index('Datetime', inplace=True)

# 타겟 변수 분리
uv_column = df['uv']
df = df.drop('uv', axis=1)

# FAMD 적용
famd = FactorAnalysis(n_components=4)
transformed_data = famd.fit_transform(df)
```

3D 시각화를 통한 데이터 분포 확인



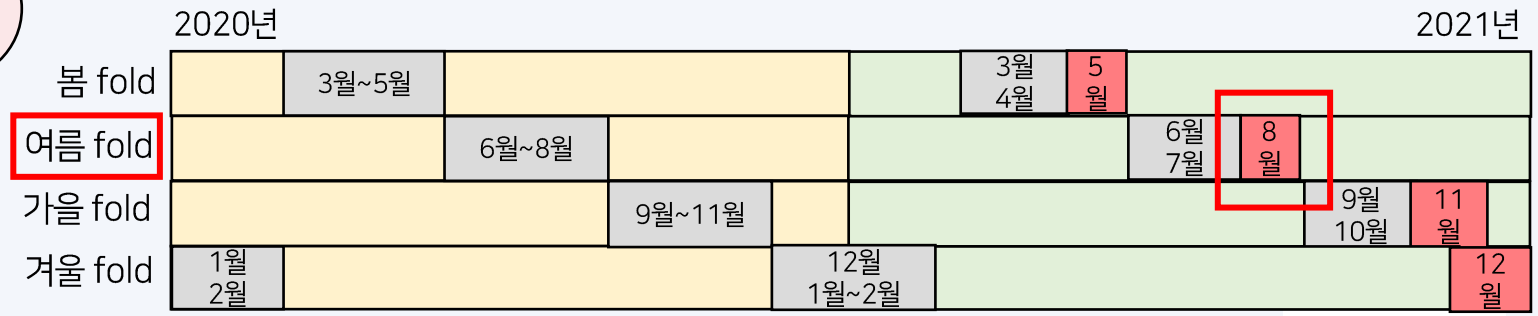
	component1	component2	component3	component4	uv
Datetime					
2020-01-01 00:00:00	1.732049	-0.064357	-0.670131	0.929740	-0.560483
2020-01-01 00:10:00	1.732046	0.047545	-0.724787	0.944402	-0.560483
2020-01-01 00:20:00	1.732043	0.015936	-0.708659	0.938095	-0.560483
2020-01-01 00:30:00	1.732039	0.013079	-0.708355	0.925208	-0.560483
2020-01-01 00:40:00	1.732036	-0.004569	-0.704180	0.929824	-0.560483
...	...	...	...	...	...
2021-12-31 23:10:00	-1.732036	0.059482	-0.759261	1.346864	-0.560483
2021-12-31 23:20:00	-1.732039	0.135414	-0.794549	1.338024	-0.560483
2021-12-31 23:30:00	-1.732043	0.158084	-0.805059	1.322808	-0.560483
2021-12-31 23:40:00	-1.732046	0.115364	-0.785915	1.317779	-0.560483
2021-12-31 23:50:00	-1.732049	0.199901	-0.828197	1.334720	-0.560483

[1052640 rows x 5 columns]

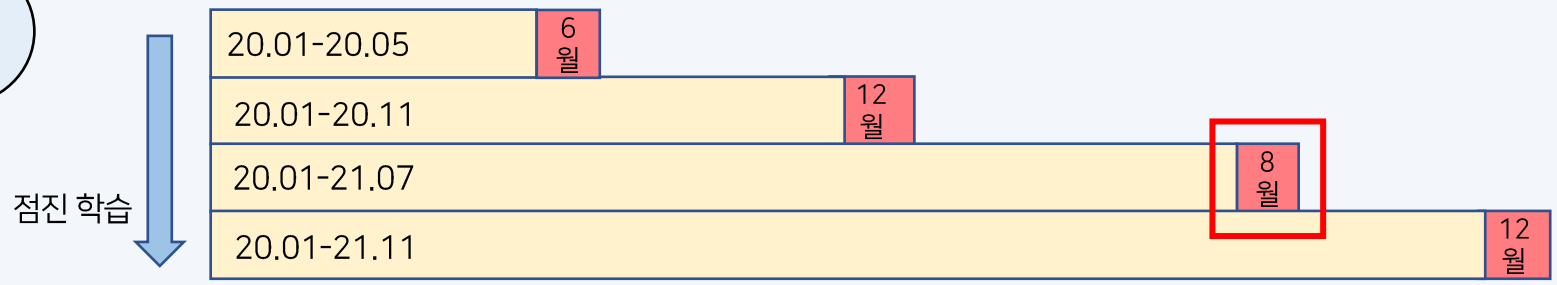
3 학습 방법 및 모델 선정

1. Validation

방안 1



방안 2



최종 데이터가 여름인 점을 고려해, 21년 8월 데이터에 대한 예측 성능을 비교해본 결과 방안2가 뛰어났음

☑ 방안 2

- Expanding Window 방식
  - 계절성 패턴을 더 잘 반영, data leakage X
  - 여러 계절에 다양하게 적용시킬 수 있음

2. 모델 선택

Random Forest	XGBoost	LGBM

↓

LSTM (딥러닝)	Prophet (회귀)	LGBM (트리)

ARIMA 모델은 지점별로 Datetime변수가 겹쳐 고려 불가

- ☑ 설명력보다 정확도에 집중
- ☑ 다양한 모델 고려
- ☑ 시계열 특성을 잘 반영하는 모델 고려

↓

하이퍼 파라미터 튜닝으로 최적 모델 선정

3 모델 별 학습 및 평가: LGBM

LGBM

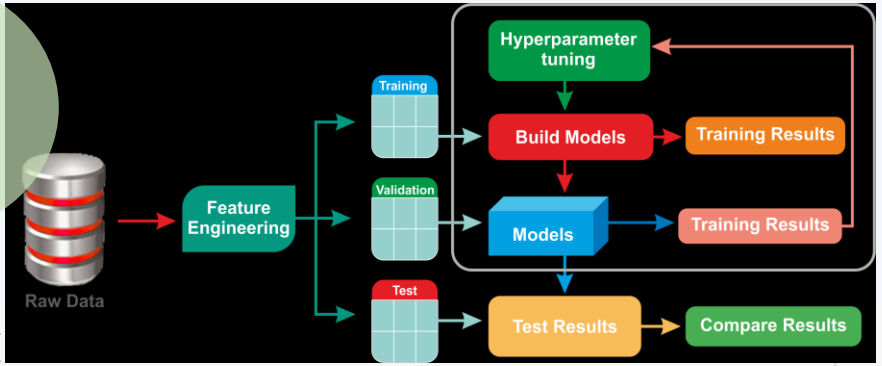
경사 부스팅 알고리즘을 기반으로 한 머신러닝 모델

+ 대용량 데이터셋에 대한 빠른 학습과 예측 가능
+ 시간적 의존성 학습
+ 계절성 및 주기적 패턴 학습

큰 규모와 시계열 기반인 우리 데이터에 적합하다고 판단

파라미터 조정

Random Search




```
# 파라미터 공간
param_space = {
    'num_leaves': np.arange(50, 151, 10),  # 탐색할 num_leaves의 후보 값들
    'learning_rate': [0.01, 0.05, 0.1],  # 탐색할 learning_rate의 후보 값들
    'feature_fraction': np.arange(0.7, 1.1, 0.1),  # 탐색할 feature_fraction의 후보 값들
    'bagging_fraction': np.arange(0.7, 0.9, 0.1),  # 탐색할 bagging_fraction의 후보 값들
    'bagging_freq': np.arange(4, 7)  # 탐색할 bagging_freq의 후보 값들
}
```

모델 성능 평가

RMSE	비정상성 제거 전		비정상성 제거 후
	이상치 처리 전	이상치 처리 후	
Train	0.3574	0.3255	0.4325
Validation	0.5446	0.5032	0.7556
Test	0.7886	0.7064	1.2336

이상치를 처리한 것이 처리하기 전보다 성능이 좋았음
비정상성 제거 전이 비정상성 제거 후보다 성능이 좋았음

```
Best parameters found:  {'num_leaves': 130, 'learning_rate': 0.05, 'feature_fraction': 0.9
, 'bagging_freq': 6, 'bagging_fraction': 0.8999999999999999
```

Num leaves	Learning rate	Feature fraction	Bagging fraction	Bagging freq
130	0.05	0.9	0.9	6

3 모델 별 학습 및 평가 : LSTM

LSTM

RNN의 한 종류로, 시계열 데이터 분석에 사용되는 딥러닝 모델

+ 시계열 데이터의 특징 파악 강점
+ 시간적 의존성 학습
+ 기존 RNN모델들보다 더 장기적인 종속성 학습

긴 시퀀스 학습에 효과적인 모델이므로 적합하다고 판단

파라미터 조정

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
param_space = {
    'units': np.arange(50, 151, 10),  # LSTM 유닛 개수
    'learning_rate': [0.01, 0.05, 0.1],  # 학습률
    'dropout': np.arange(0.1, 0.6, 0.1),  # 드롭아웃 비율
    'activation': ['relu', 'tanh'],  # 활성화 함수
    'optimizer': ['adam', 'rmsprop']  # 옵티마이저
}
```

Units	Learning rate	Dropout	Activation function	Optimizer
LSTM 유닛 개수	학습률	과적합을 방지하기 위한 정규화	활성화 함수	모델 가중치 업데이트

모델 성능 평가

RMSE	비정상성 제거 전		비정상성 제거 후
	이상치 처리 전	이상치 처리 후	
Train	0.5319	0.4503	0.6712
Validation	1.1605	0.9561	1.9074
Test	1.5247	1.0759	1.8416

이상치를 처리한 것이 처리하기 전보다 성능이 좋았음
비정상성 제거 전이 비정상성 제거 후보다 성능이 좋았음

```
Best parameters found:  {'units': 110, 'optimizer': 'adam',
                         'learning_rate': 0.01, 'dropout': 0.5, 'activation': 'relu'}
```

Units	Learning rate	Dropout	Activation function	Optimizer
110	0.01	0.5	relu	Adam

3 모델 별 학습 및 평가 : Prophet

Prophet

가법회귀모델로 시계열 예측을 위한 오픈소스 라이브러리

+ 자동으로 계절성과 트렌드 모델링
+ 이상치 식별 후 적절하게 처리 가능
+ 일부 회귀 모델의 가정을 완화하는 유연성 제공

시계열에 더 가중을 두는 모델이므로 적합하다고 판단

파라미터 조정

```
# 파라미터 공간 정의
param_space = {
    'growth': ['linear', 'logistic'],
    'changepoint_prior_scale': [0.01, 0.1, 1.0],
    'seasonality_mode': ['additive', 'multiplicative'],
    'seasonality_prior_scale': [0.01, 0.1, 1.0, 10]
}
```

Changepoint Prior scale	Seasonality mode	Seasonality Prior scale	Growth
변곡점의 유연성 제어	계절성 모드	계절성 요소 조절	데이터의 패턴

모델 성능 평가

RMSE	비정상성 제거 전		비정상성 제거 후
	이상치 처리 전	이상치 처리 후	
Train	1.3623	1.1323	1.5568
Validation	1.6756	1.4234	1.8865
Test	1.9723	1.5423	2.1245

이상치를 처리한 것이 처리하기 전보다 성능이 좋았음
비정상성 제거 전이 비정상성 제거 후보다 성능이 좋았음

```
Best parameters found: {'changepoint_prior_scale': 0.05, 'seasonality_mode':
'additive', 'seasonality_prior_scale': 10, 'growth': 'multiplicative'}
```

Changepoint Prior scale	Seasonality mode	Seasonality Prior scale	Growth
0.05	multiplicative	10	logistic

3 모델 별 평가 및 결과

수상작과의 성능 비교

수상작

Fold	Fold 1		Fold 2	
Model	Train	Valid	Train	Valid
R.F.	0.088	0.591	0.120	0.644
LightGBM	0.164	0.574	0.346	0.631
XGBoost	0.027	0.595	0.077	0.640

RMSE : 0.604573

수행작

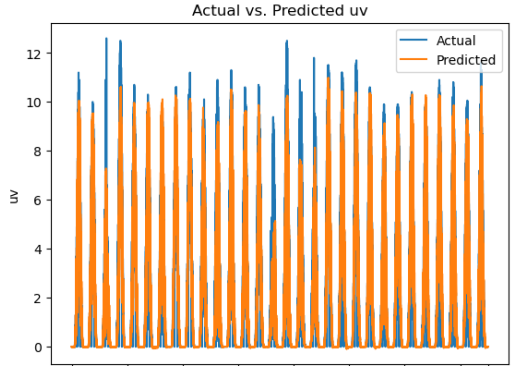
	이상치 처리 전			이상치 처리 후			비정상성 제거 후		
	TRAIN	VAL	TEST	TRAIN	VAL	TEST	TRAIN	VAL	TEST
LGBM	0.3574	0.7162	0.7886	0.3255	0.5032	0.7064	0.4325	0.7556	1.2336
LSTM	0.5319	1.1605	1.5247	0.4503	0.9561	1.0759	0.6712	1.9074	1.8416
Prophet	1.3623	1.6756	1.9723	1.1323	1.4234	1.5423	1.5568	1.8865	2.1245

최종 Train Set / 모델 선정

이상치 처리 전	STL 분해로 이상치 처리 후 데이터	비정상성 제거 후

LGBM	LSTM	Prophet

최종 결과



이상치 처리 후 데이터로
LGBM 모델에 적용한 결과가
예측력이 가장 높음

RMSE : 0.7064

3 결과 분석 & 평가 개선안

성능지표 분석결과

데이터의 비정상성 처리 목적으로
잔차만을 남기고 모델을 학습시켰을 때,
성능이 가장 떨어지는 것을 확인

→ 최종 예측 변수인 'uv지수' 가 계절성이 있는 변수임을 고려했을 때,
계절성을 제거하고 잔차만 남겨 모델을 학습시키는 것은 적절치 x

STL분해로 이상치를 탐색하여 처리 이후
다시 계절성과 추세를 더하여 모델을 학습시켰을 때,
성능이 가장 높게 나온 것을 확인

→ 전처리 과정에서의 다양한 시도로
수상작보다 성능이 떨어짐에도 나름의 개선사항이 존재했음

한계점

ARIMA 모델을 사용하고자 했으나 15개 지점의 관측값을 사용하여 Datetime이 겹쳐 모델 적용 불가
따라서 이 문제를 해결하는 방안이 있었다면 시계열모델도 학습해 결과를 확인해볼 수 있지 않았을까 하는 아쉬움

평가 개선안

모델 성능평가를 기존 대회 규정에 따라 RMSE
바탕으로만 기반하여 진행

→ 해당 프로젝트의 목적을 고려하여, 예측값과 실제값 사이의
양의 상관정도를 고려하여 성능을 평가하는 방안 모색