LH 한국토지주택공사

COMPAS

국토도시 데이터 분석 경진대회

화재발생 예측모델 개발

팀 : Philos

화제발생 예측모델 개발

팀 : Philos

리더보드(랭킹예측순위)

순위	팀명/개인명(닉네임)	중간점수 (Public Score)	최종점수 (Private Score)	과제 제출 횟수
1	Central-Park	0.63407	0.59187	41
2	nchos	0.62434	0.57856	1267
3	philos	0.60678	0.57319	142
4	TEAM-EDA	0.56591	0.53174	176
5	황아저씨	0.55806	0.52174	153

이문규
(경제학과)
ansrb55@gmail.com

신우석
(경영학과)
useok2791@naver.com

이명호
(SW융합공학과)
audgh3710@gmail.com

목차

1. 데이터 전처리 과정

Keyword — 이전사례, EDA

1. 데이터 전처리 과정

```
array(['dt_of_fr', 'fr_yn', 'bldng_us', 'bldng_archtctr', 'bldng_cnt',
       'bldng_ar', 'ttl_ar', 'lnd_ar', 'dt_of_athrztn', 'ttl_grnd_flr',
       'ttl_dwn_flr', 'bldng_us_clssfctn', 'tmprtr', 'prcpttn', 'wnd_spd',
       'wnd_drctn', 'hmdt', 'gas_engry_us_201401', 'ele_engry_us_201401',
       'gas_engry_us_201402', 'ele_engry_us_201402',
       'gas_engry_us_201403', 'ele_engry_us_201403',
       'gas_engry_us_201404', 'ele_engry_us_201404',
       'gas_engry_us_201405', 'ele_engry_us_201405',
       'gas_engry_us_201406', 'ele_engry_us_201406',
       'gas_engry_us_201407', 'ele_engry_us_201407',
       'gas_engry_us_201408', 'ele_engry_us_201408',
       'gas_engry_us_201809', 'ele_engry_us_201809',
       'gas_engry_us_201810', 'ele_engry_us_201810',
       'gas_engry_us_201811', 'ele_engry_us_201811',
       'gas_engry_us_201812', 'ele_engry_us_201812', 'lw_13101010',
       'lw_13101110', 'lw_13101210', 'lw_13101310',
       'lw_13101410', 'lw_13111010', 'lw_13111110', 'lw_13121010',
       'lw_13121011', 'lw_13131010', 'lw_13131110', 'lw_13141010',
       'lw_13141011', 'jmk', 'id', 'rgnl_ar_nm', 'rgnl_ar_nm2',
       'lnd_us_sttn_nm', 'rd_sd_nm', 'emd_nm', 'hm_cnt', 'fr_sttn_dstnc',
       'bldng_ar_prc', 'fr_wthr_fclt_dstnc', 'fr_mn_cnt', 'mlt_us_yn',
       'cctv_dstnc', 'fr_wthr_fclt_in_100m', 'cctv_in_100m',
       'tbc_rtl_str_dstnc', 'sft_emrgnc_bll_dstnc', 'ahsm_dstnc',
       'no_tbc_zn_dstnc', 'bldng_cnt_in_50m', 'trgt_crtr',
       'fr_fghtng_fclt_spcl_css_5_yn', 'fr_fghtng_fclt_spcl_css_6_yn',
       'us_yn', 'dngrs_thng_yn', 'slf_fr_brgd_yn',
       'blk_dngrs_thng_mnfctr_yn', 'cltrl_hrtg_yn'], dtype='<U28')
```

1. 수 많은 칼럼들 중 불필요한 칼럼은 삭제

2. 새로운 파생변수 생성

1. 데이터 전처리 과정

1. 결측치가 1/2 이상인 칼럼 삭제

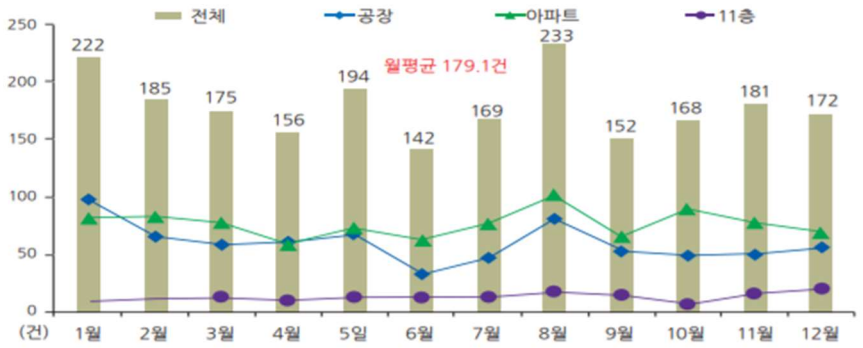
```
dt.loc[:,dt.isnull().sum()>30000].columns
```

```
Index(['prcpttn', 'lw_13101010', 'lw_13101110', 'lw_13101210', 'lw_1310
1211',
       'lw_13101310', 'lw_13101410', 'lw_13111010', 'lw_13111110',
       'lw_13121010', 'lw_13121011', 'lw_13131010', 'lw_13131110',
       'lw_13141010', 'lw_13141011', 'trgt_crtr',
       'fr_fghtng_fclt_spcl_css_5_yn', 'fr_fghtng_fclt_spcl_css_6_yn',
'us_yn',
       'dngrs_thng_yn', 'slf_fr_brgd_yn', 'blk_dngrs_thng_mnfctr_yn',
       'cltrl_hrtg_yn'],
      dtype='object')
```

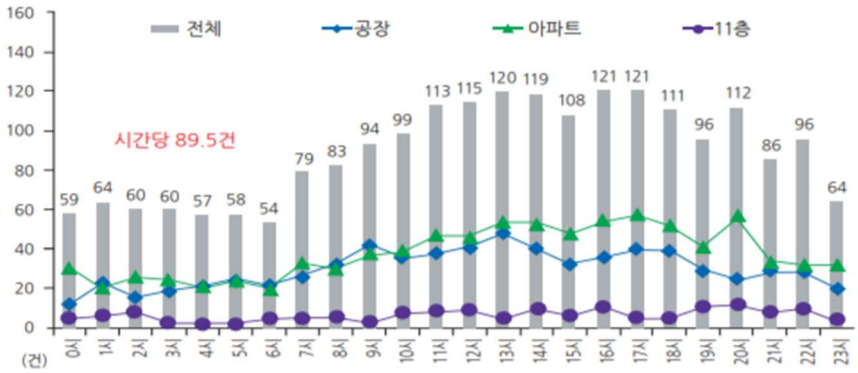
1. 데이터 전처리 과정

2016년 특수건물 화재통계·안전점검 결과분석

(월별)



(시간대별)



출처 : 한국화재보험협회

이전 사례에서 화재발생일시가 큰 의미가 있었음

1. 데이터 전처리 과정

2. 화재발생일시 칼럼화

dt_of_fr	year	month	day	time
	2017	10	20	5:54
	2018/09/30 8:26			
	2016/10/30 14:57			
	2016/06/14 5:23			
	2018/04/22 5:38			
	2018/04/21 15:41			
	2015/09/02 1:35			
	2018/03/03 23:11			
	2018/03/04 8:51			
	2014/12/08 5:23			
	2014/12/10 8:54			
	2014/12/05 7:43			
	2017/07/17 13:15			

1. 데이터 전처리 과정

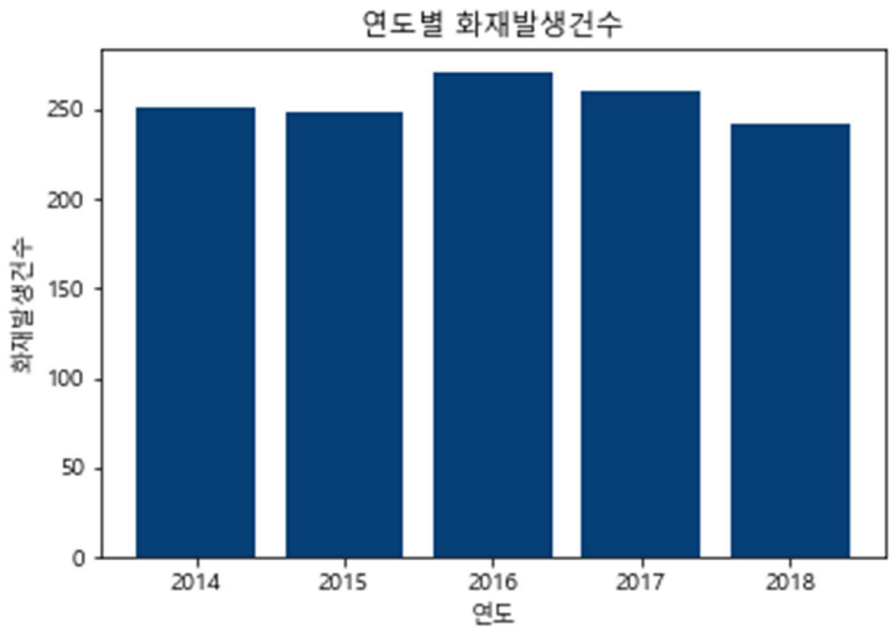
dt_of_fr	year	month	day	time
2017/10/20 5:54				
2018/09/30 8:26				
2016/10/30 14:57				
2016/06/14 5:23				
2018/04/22 5:38				
2018/04/21 15:41				
2015/09/02 1:35				
2018/03/03 23:11				
2018/03/04 8:51				
2014/12/08 5:23				
2014/12/10 8:54				
2014/12/05 7:43				
2017/07/17 13:15				
2016/11/02 12:54				
2017/09/24 11:35				
2016/09/29 23:05				
2015/11/21 9:51				

연도, 월, 일, 시간, 요일으로 분리

1. 데이터 전처리 과정

- 발생일시 변수화

연도



1. 데이터 전처리 과정

- 발생일시 변수화

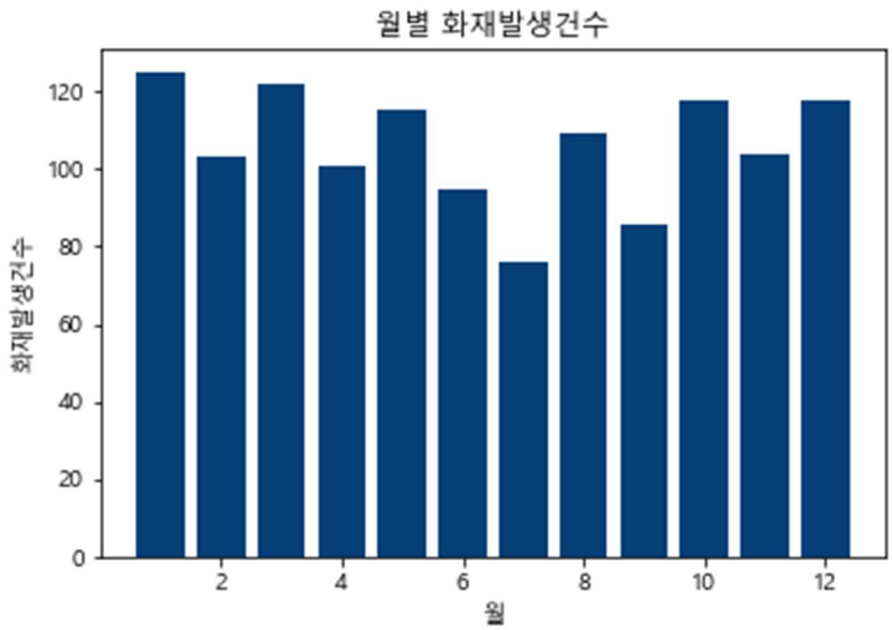
연도

연도에는 의미부여 불가능

1. 데이터 전처리 과정

- 발생일시 변수화

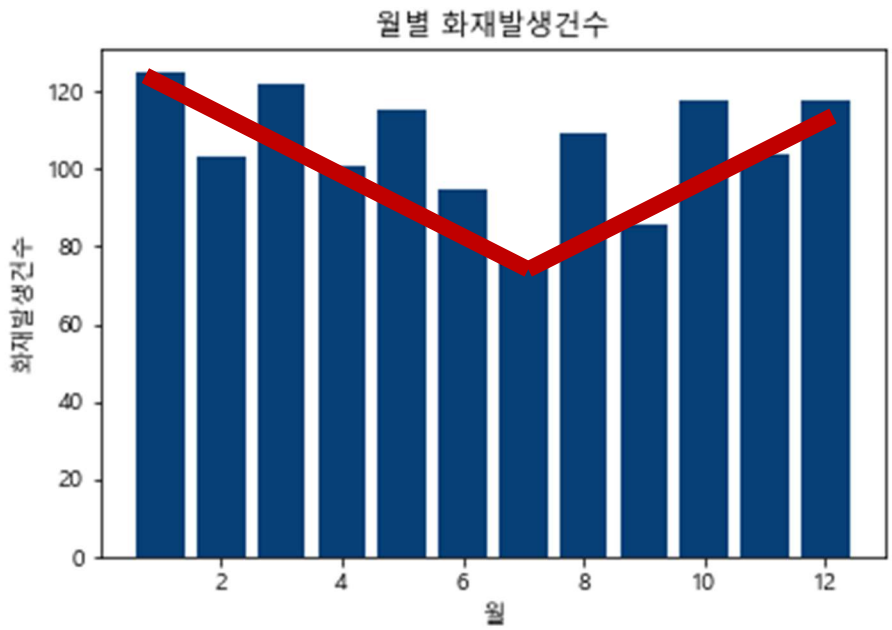
월



1. 데이터 전처리 과정

- 발생일시 변수화

월

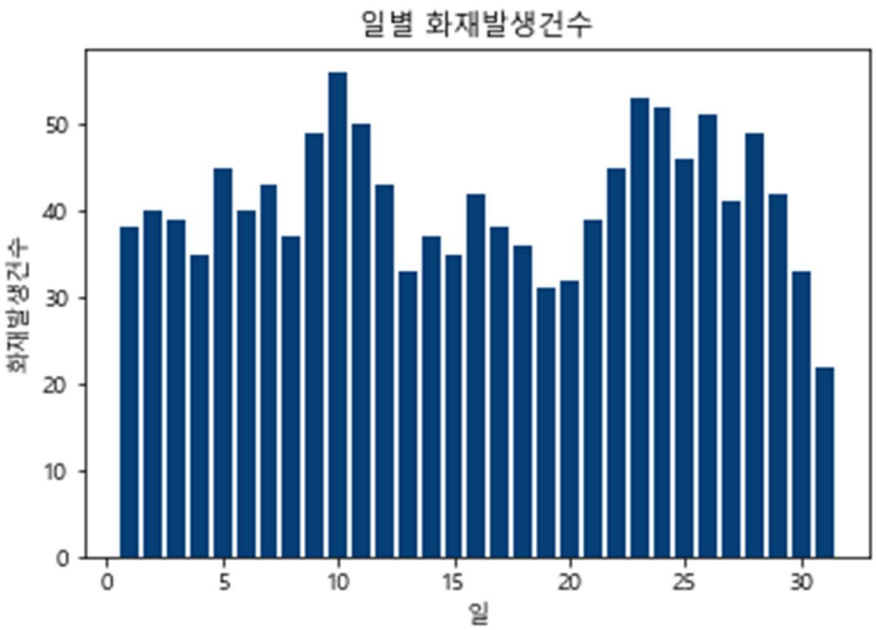


특정 기간에 화재발생이 적음

1. 데이터 전처리 과정

- 발생일시 변수화

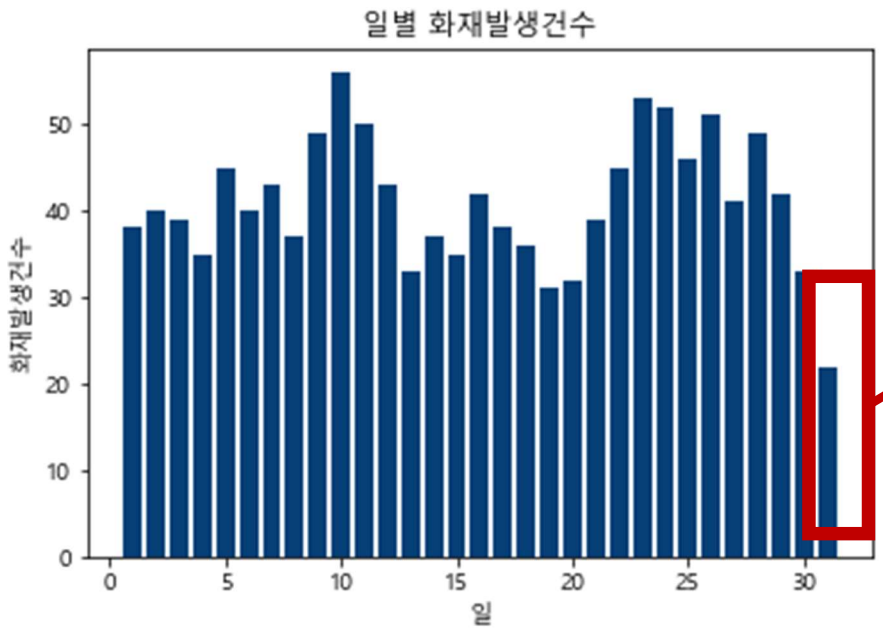
일



1. 데이터 전처리 과정

- 발생일시 변수화

일

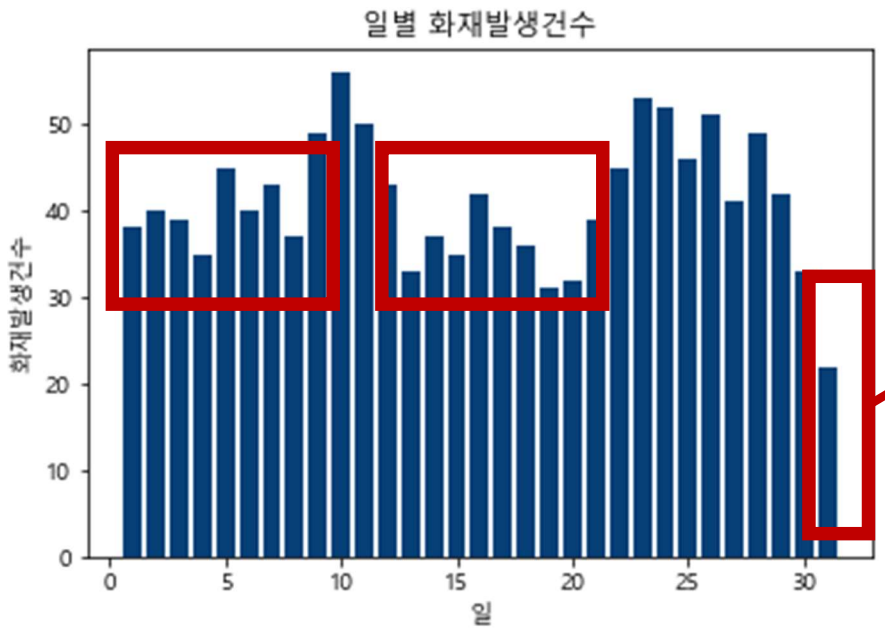


31일의 경우는 존재하지 않는
월이 많기때문에 의미x

1. 데이터 전처리 과정

- 발생일시 변수화

일



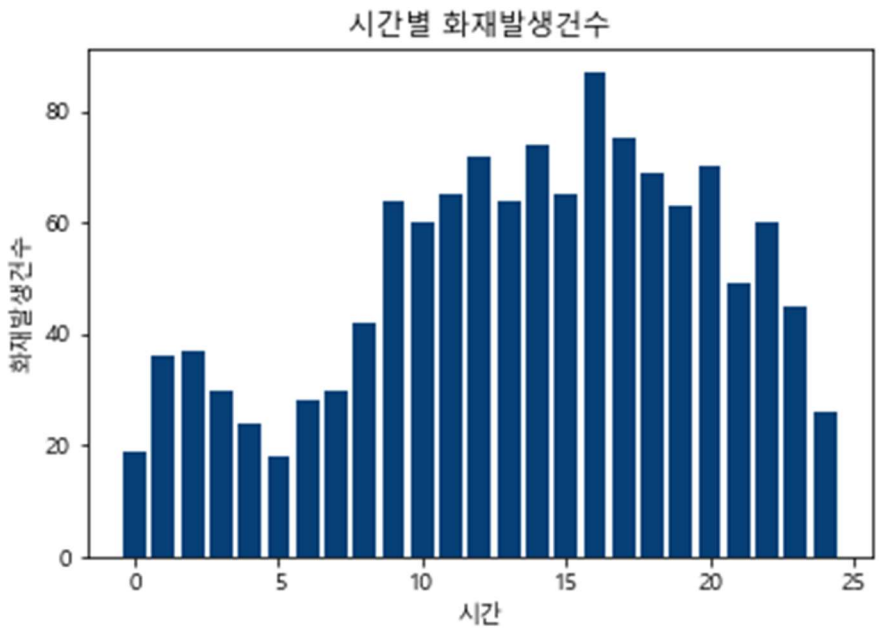
31일의 경우는 존재하지 않는
월이 많기때문에 의미x

월과 마찬가지로 특정 기간에 발생 수가 적음

1. 데이터 전처리 과정

- 발생일시 변수화

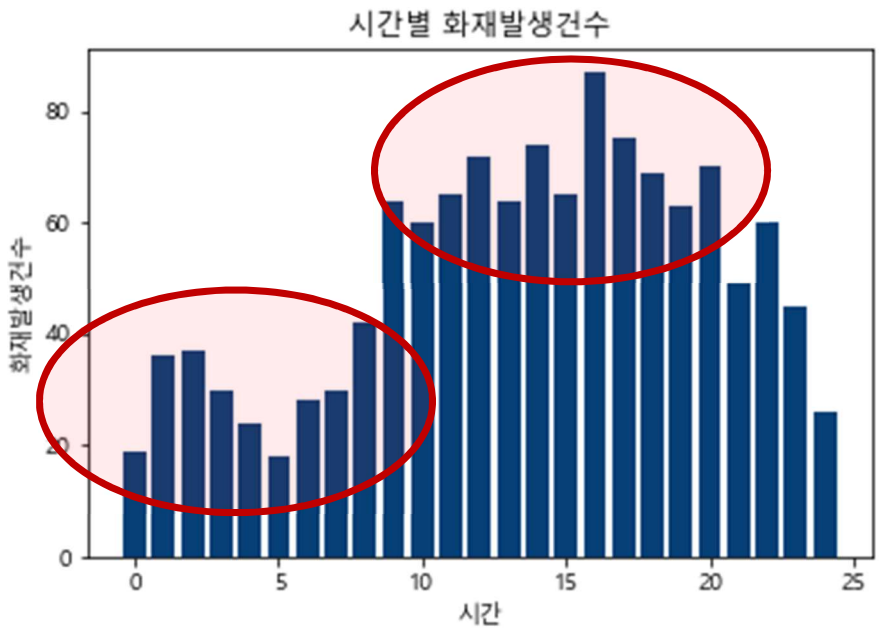
시간



1. 데이터 전처리 과정

- 발생일시 변수화

시간

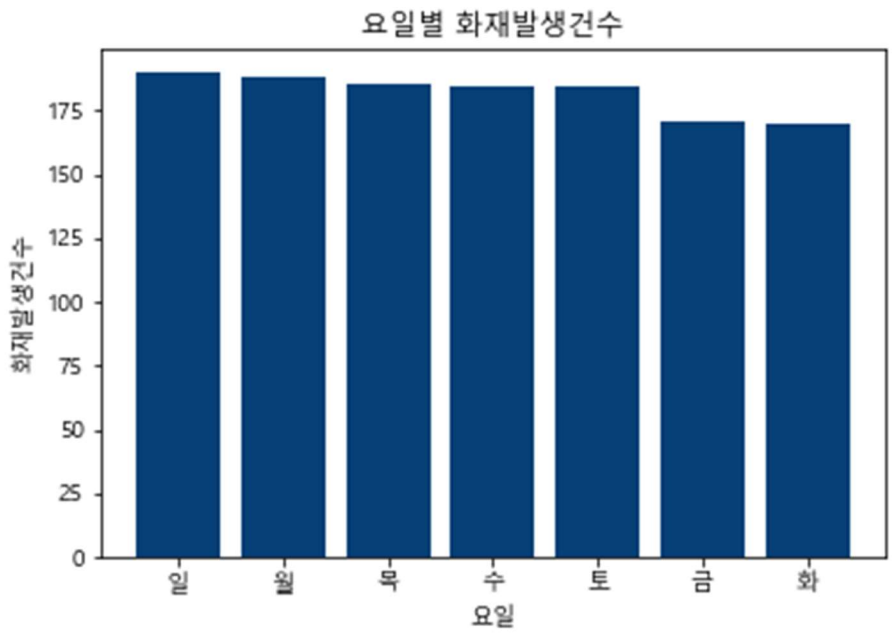


새벽과 오후의 발생건수 차이가 매우 큼

1. 데이터 전처리 과정

- 발생일시 변수화

요일



큰 차이는 없지만 금요일과 화요일에 발생건수가 적음

1. 데이터 전처리 과정

3. 날씨변수 결측치 처리

```
## 결측치 부분
풍향_obj = 날짜랑날씨[날짜랑날씨["풍향"].isnull()].iloc[:,:-4]
풍속_obj = 날짜랑날씨[날짜랑날씨["풍속"].isnull()].iloc[:,:-4]
온도_obj = 날짜랑날씨[날짜랑날씨["온도(c)"].isnull()].iloc[:,:-4]
습도_obj = 날짜랑날씨[날짜랑날씨["습도"].isnull()].iloc[:,:-4]
## 머신러닝으로 처리
model = RandomForestRegressor().fit(풍향_X, 풍향_y)
풍향결측치처리값 = model.predict(풍향_obj)
model = RandomForestRegressor().fit(풍속_X, 풍속_y)
풍속결측치처리값 = model.predict(풍속_obj)
model = RandomForestRegressor().fit(온도_X, 온도_y)
온도결측치처리값 = model.predict(온도_obj)
model = RandomForestRegressor().fit(습도_X, 습도_y)
습도결측치처리값 = model.predict(습도_obj)
```

발생일시를 사용하여 결측치가 아닌 시점으로 학습(fit)
시킨 뒤, 결측 시점 값을 예측(predict)하여 처리.

1. 데이터 전처리 과정

4. 범주형 데이터 결측치 처리

건물용도	건물구조
단독주택	블록구조
단독주택	철근콘크리트구조
공장	일반철골구조
단독주택	벽돌구조
단독주택	철근콘크리트구조
제1종근린생활시설	철근콘크리트구조
제2종근린생활시설	철근콘크리트구조
단독주택	기타조적구조
단독주택	철근콘크리트구조
단독주택	철근콘크리트구조
공장	일반철골구조

1. 데이터 전처리 과정

4. 범주형 데이터 결측치 처리

```
dt[dt["bldng_us"].isnull()]["fr_yn"].value_counts()
```

```
N    27647
Y       30
Name: fr_yn, dtype: int64
```

결측치의 화재 발생유무 비율

```
dt[dt["bldng_us"].isnull()]["fr_yn"].value_counts()/dt[dt["bldng_us"].isnull()]["fr_yn"].value_counts().sum()
```

```
N    0.998916
Y    0.001084
Name: fr_yn, dtype: float64
```

결측치인 값들은 화재 발생 빈도가 낮은 경향을 보임

1. 데이터 전처리 과정

4. 범주형 데이터 결측치 처리

```
dt[dt["bldng_archtctr"].isnull()]["fr_yn"].value_counts()
```

```
N    27644
Y       21
Name: fr_yn, dtype: int64
```

결측치의 화재 발생유무 비율

```
dt[dt["bldng_archtctr"].isnull()]["fr_yn"].value_counts()/dt[dt["bldng_archtctr"].isnull()]["fr_yn"].value_counts().sum()
```

```
N    0.999241
Y    0.000759
Name: fr_yn, dtype: float64
```

결측치인 값들은 화재 발생 빈도가 낮은 경향을 보임

1. 데이터 전처리 과정

4. 범주형 데이터 결측치 처리

건물용도	건물구조
단독주택	블록구조
결측	결측
단독주택	철근콘크리트구조
공장	일반철골구조
단독주택	벽돌구조
단독주택	철근콘크리트구조
제1종근린생활시설	철근콘크리트구조
제2종근린생활시설	철근콘크리트구조
결측	결측
단독주택	기타조적구조
단독주택	철근콘크리트구조

결측치도 의미가 있다고 판단하여 범주형 데이터들의 결측치는
"결측"으로 처리하여 One-Hot-Encoding 하였음

1. 데이터 전처리 과정

5. 이외의 수치형 데이터 전처리(1)

```
dt["ttl_grnd_flr"].isnull().sum()
```

```
10210
```

지하 층수의 총 합의 결측치 수

```
dt[dt["ttl_grnd_flr"].isnull()]["fr_yn"].value_counts()
```

```
N    9775
Y     435
Name: fr_yn, dtype: int64
```

```
dt[dt["ttl_grnd_flr"].notnull()]["fr_yn"].value_counts()
```

```
N    41767
Y     7222
Name: fr_yn, dtype: int64
```

지상·지하 층수의 총 합과 같은 데이터의
결측치의 경우 결측 유무 칼럼을 생성

5. 이외의 수치형 데이터 전처리(2)

```
print("결측치 수 :",dt["dt_of_athrztn"].isnull().sum())
print("결측치인 값들의 화재발생 여부 : ")
dt[dt["dt_of_athrztn"].isnull()]["fr_yn"].value_counts()
```

```
결측치 수 : 27581
결측치인 값들의 화재발생 여부 :

N    27581
Name: fr_yn, dtype: int64
```

Train 에서 건축승인연도의 경우
결측치인 값들은 모두 화재발생이 N

2. 머신러닝 과정

Keyword —

Over-fitting, Join, Probability

2. 머신러닝 과정

1. Over-Fitting (과대적합)

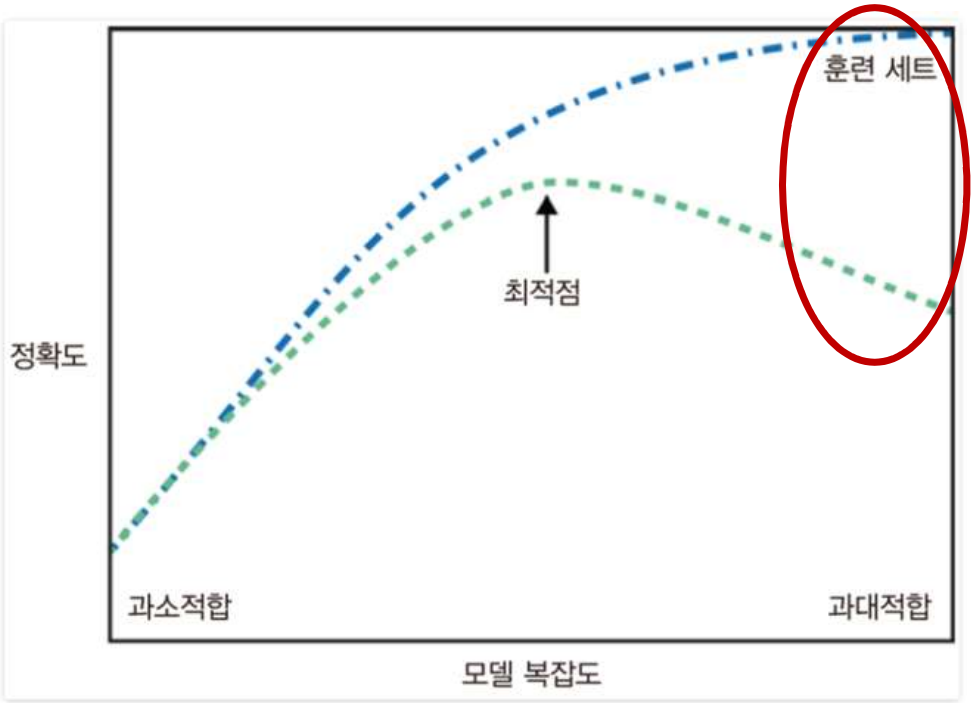


그림 2-1 모델 복잡도에 따른 훈련과 테스트 정확도의 변화

2. 머신러닝 과정

ex) 목표 : 공 분류

탁구공

축구공

사과

2. 머신러닝 과정

ex) 목표 : 공 분류

1. 첫 번째 모델

1. 음식이 아니다.
2. 육각형과 오각형으로 이루어져 있다.

2. 머신러닝 과정

ex) 목표 : 공 분류

1. 첫 번째 모델

1. 음식이 아니다.
2. 육각형과 오각형으로 이루어져 있다.

2. 머신러닝 과정

ex) 목표 : 공 분류

2. 두 번째 모델

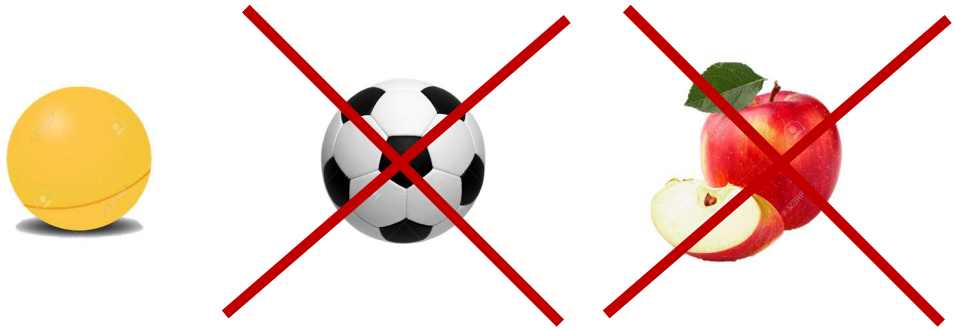
1. 플라스틱으로 만들어졌다.
2. 속이 비어 있다.

2. 머신러닝 과정

ex) 목표 : 공 분류

2. 두 번째 모델

1. 플라스틱으로 만들어졌다.
2. 속이 비어 있다.

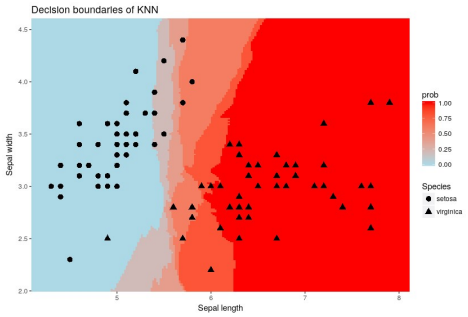


2. 머신러닝 과정

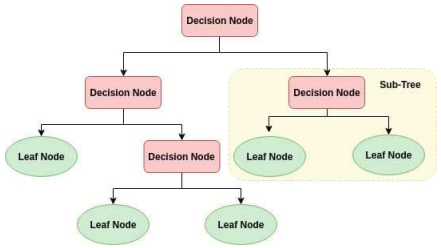
- 각 모델 구성 방법

- **서로 다른 알고리즘 및 하이퍼파라미터 사용**
  - (KNN, LogisticRegression, Tree, Bagging, Boosting …)

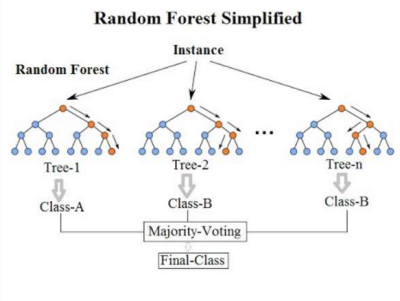
- **서로 다른 학습 데이터**
  - Train만으로 학습한 모델, Val만으로 학습한 모델, Train+Val로 학습한 모델을 각각 사용



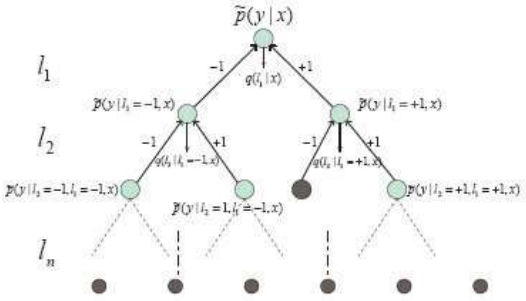
KNN



Tree



Bagging



Boosting

2. 머신러닝 과정

2. Join

1. 첫 번째 모델 

2. 두 번째 모델 

2. 머신러닝 과정

2. Join

1. 첫 번째 모델

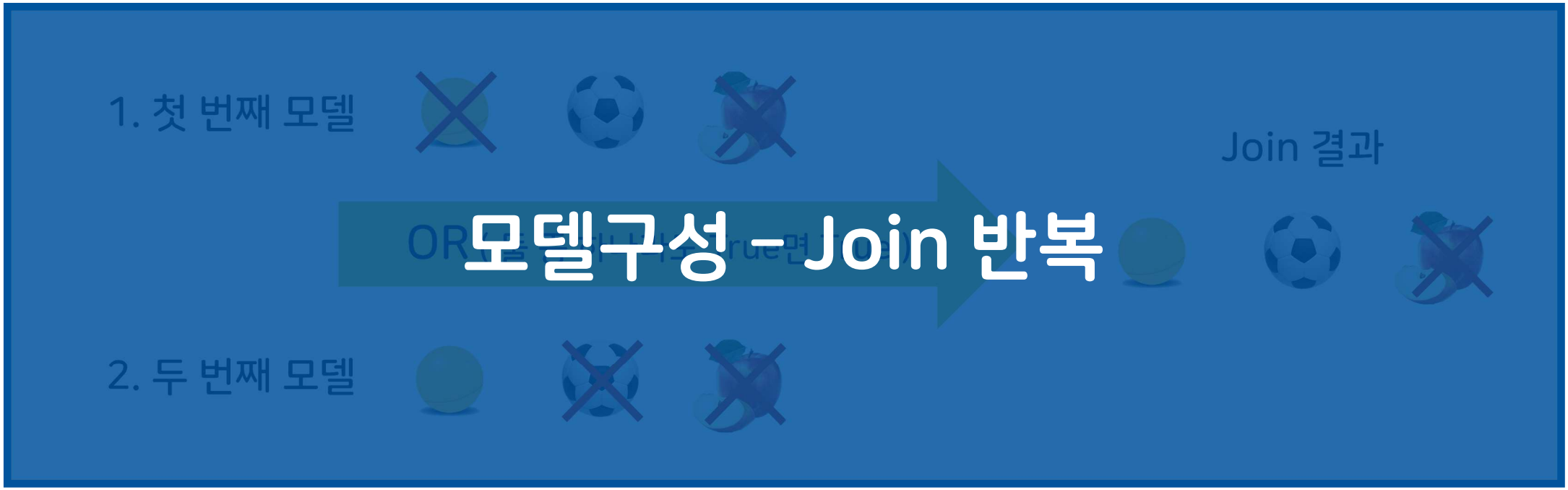
2. 두 번째 모델

OR (둘 중 하나라도 True면 True)

Join 결과

2. 머신러닝 과정

2. Join



3. Probability

	setosa 확률	versicolor 확률	virginica 확률
0	0.878030	0.121959	0.000011
1	0.797058	0.202911	0.000030
2	0.851998	0.147976	0.000026
3	0.823406	0.176536	0.000058
4	0.896035	0.103954	0.000011
...	...	...	...
145	0.001165	0.232330	0.766505

분류모델에서 예측 시 분류 연산에
사용된 확률 값을 이용

3. Probability

```
model.predict_proba()
```

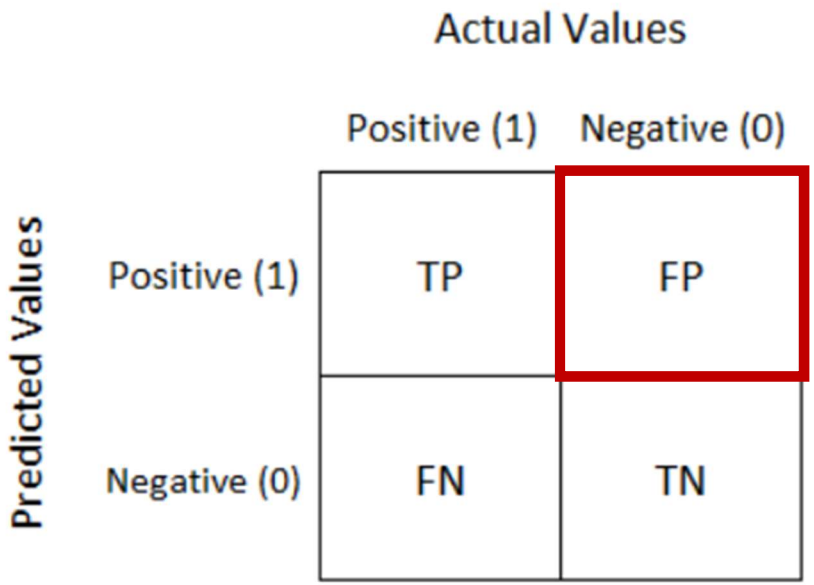
```
Signature: model.predict_proba(X)
Docstring:
Probability estimates.

The returned estimates for all classes are ordered by the
label of classes.

For a multi_class problem, if multi_class is set to be "multinomial"
the softmax function is used to find the predicted probability of
each class.
```

predict_proba 를 이용

3. Probability



확실한 화재예측 대상으로 분류하여
화재예방에 대한 기회비용을 낮추기 위함.

3. Probability

```
proba_high = []
for i in range(len(proba)):
    if proba[1][i] > 0.63:
        proba_high.append(1)
    else:
        proba_high.append(0)

첫번째모델예측값 = pd.DataFrame(proba_high)
```

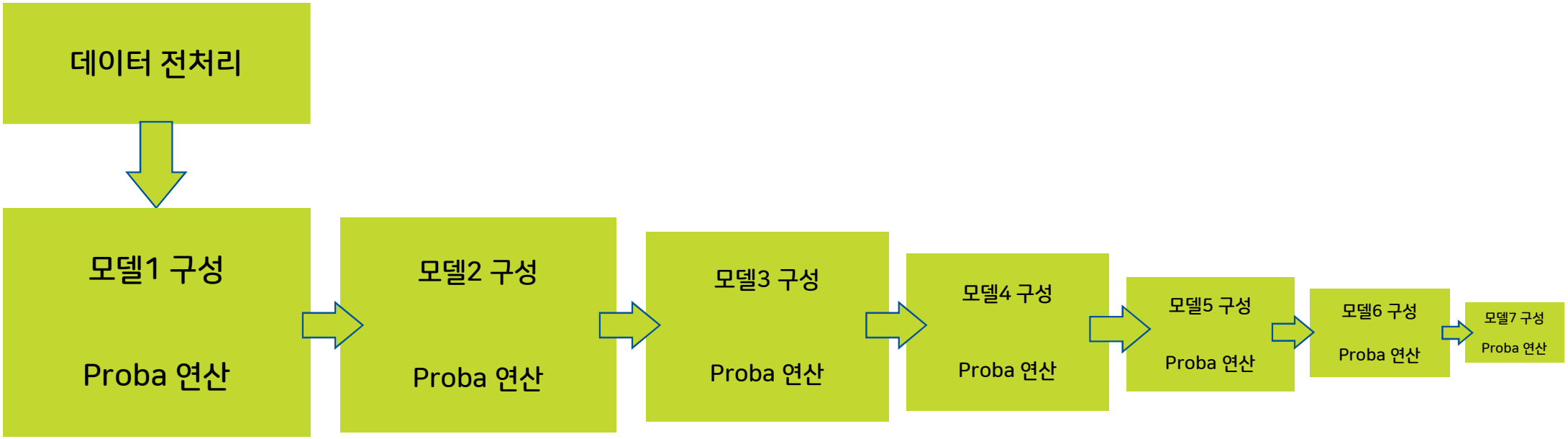
모델이 '화재발생'으로 예측한 값 중 확률이 높은 값들을
진짜 화재발생으로 간주하여 예측 값 구성

3. 결론

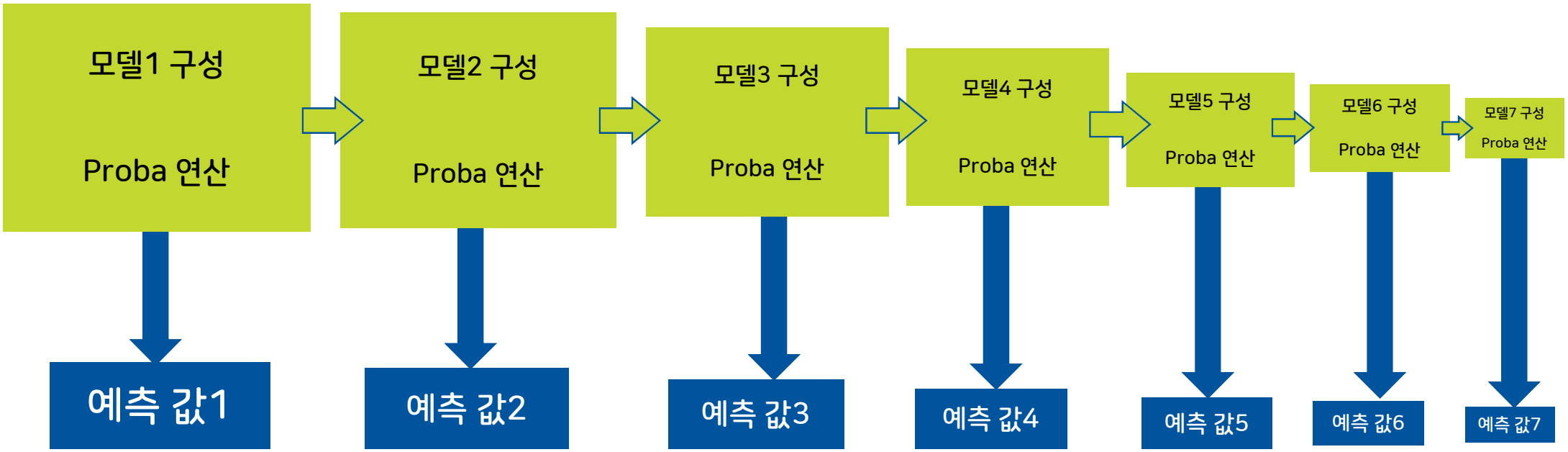
3. 결론

데이터 전처리

3. 결론


데이터 전처리
모델1 구성
Proba 연산
모델2 구성
Proba 연산
모델3 구성
Proba 연산
모델4 구성
Proba 연산
모델5 구성
Proba 연산
모델6 구성
Proba 연산
모델7 구성
Proba 연산

3. 결론


모델1 구성
Proba 연산
모델2 구성
Proba 연산
모델3 구성
Proba 연산
모델4 구성
Proba 연산
모델5 구성
Proba 연산
모델6 구성
Proba 연산
모델7 구성
Proba 연산
예측 값1
예측 값2
예측 값3
예측 값4
예측 값5
예측 값6
예측 값7

3. 결론

예측 값1 or 예측 값2 or 예측 값3 or 예측 값4 or 예측 값5 or 예측 값6 or 예측 값7

JOIN

최종 Submission

3. 결론

단일 모델 사용 시 F1 Score : 약 55%
3개의 모델을 Join한 F1 Score : 약 57%

.

.

.

.

.

13개의 모델을 Join한 F1 Score : 60%

구현 시스템 제원(소프트웨어)

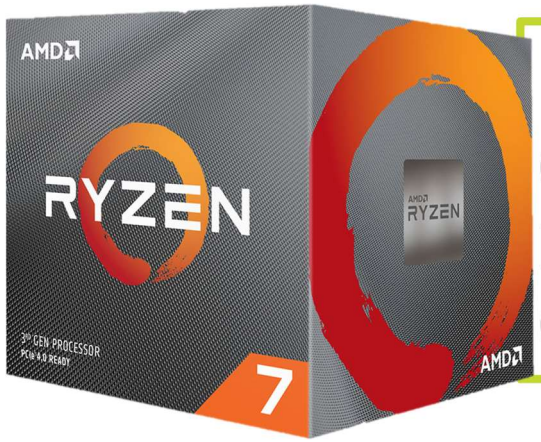
Excel office 365

python 3.7

- Pandas
- Numpy
- matplotlib
- seaborn
- shap
- sklearn
- catboost
- lightGBM
- XGBoost
- warnings

Anaconda3
jupyternotebook

구현 시스템 제원(하드웨어)



랩탑

CPU : Ryzen7 2700U 4 core
RAM : DDR4 16GB
GPU : Radeon VEGA 10

감사합니다.