에이콘아카데미(강남) _ 최종 프로젝트

Acorn Academy Final Project
Gguldangi프로젝트 (주식 및 금융 데이터 분석)

TEAM 1팀 (Gguldangi)

목차

01 프로젝트 개요

주제 및 프로젝트 주제 구체화

- **주제 – 금융 데이터를 분석**
- **주제 구체화 – 거래량이 많은 서울 노원구 단지 커뮤티니 및 가격 예측**

| 문제접근

- 2017.01 ~ 2022.08 **기간 내 서울 노원구 아파트 매매 가격, 거래량을 데이터로**
- **선정하여 8개의 모델로 테스트, 가장 성능 좋은 모델을 선정하여 향후 집값 예측**

| 프로젝트 결과

02 프로젝트 팀 구성 및 역할 – 데이터 분석

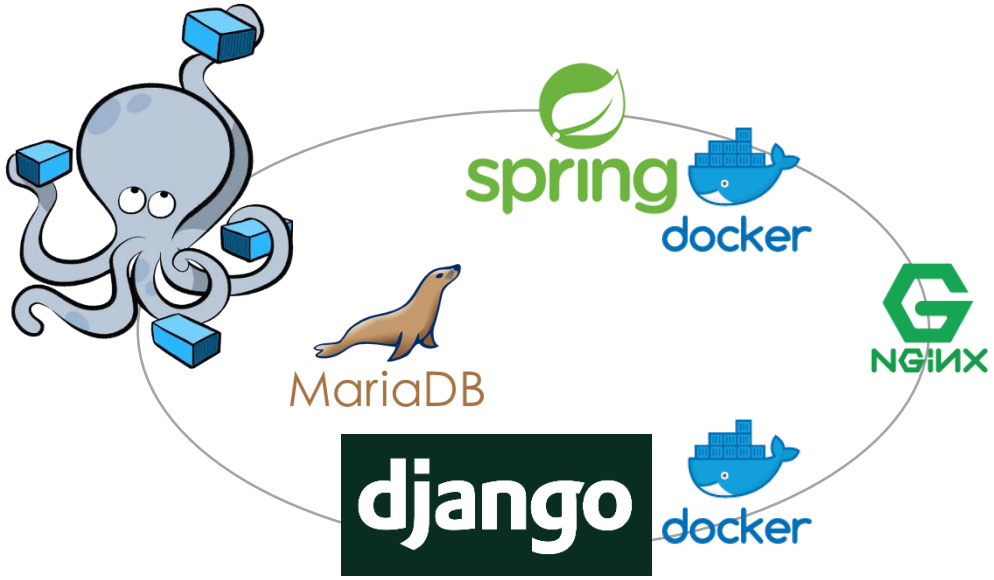
훈련생	역할	담당 업무
*강현**	*팀장*	▶ 데이터 수집, Prophet으로 시계열 예측
*김혁**	*팀원*(Analysis)	▶ ppt 제작, 데이터 수집, 데이터 전처리 및 데이터 정규화 RMSLE값 분석 후 모델선정 사용 및 분석
*노태**	*팀원*(Analysis)	▶ 데이터 수집, 데이터 전처리 및 데이터 정규화 RMSLE값 분석 후 모델선정 사용 및 분석
*안정**	*팀원*(Analysis)	▶ 데이터 수집, 데이터 전처리 및 데이터 정규화 RMSLE값 분석 후 모델선정 사용 및 분석

02 프로젝트 팀 구성 및 역할 - 백엔드

훈련생	역할	담당 업무
*강현**	*팀장*	▶ 쟝고 어드민 사용자 구현 프론트
*전해**	*팀원(Web)*	▶ 아파트 단지 정보, 단지톡
*최광**	*팀원(Web)*	▶ 아파트 실거래 구현 북마크

03 프로젝트 환경

종류	환경
운영체제	Windows 11 / Ubuntu 20.04
백엔드	Django, Spring
데이터 분석	Python, Colab Jupyter
협업툴	Git, Github, Discord



03 프로젝트 환경

1. 각 Local 환경에서 개발 후 Remote Repository의 해당 기능 Branch에 올림

2. 해당 기능 구현이 완료되면 PR(Pull Request) 작성

3. GIthub Actions 수행
   test
   formatting
4. 코드 리뷰 및 확인

5. Merge 조건에 부합하면 Main(Master) Branch에 병합

6. Github Actions 수행
   - test
   - build
   - push docker images to Dockerhub
   - connect to IaaS(GCP or EC2) using ssh
   - execute
7. Deploy

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (DATA 수집)

- 서울 아파트 매매 실거래 데이터.csv - 서울시 아파트(201701 ~ 202208) 부동산 매매 총 거래량

(출처 :열린데이터광장 http://data.seoul.go.kr/dataList/OA-15818/S/1/datasetView.do)

- 노원구5년거래량.csv – 서울 아파트 매매 실거래 데이터.csv 에서 노원구 아파트 거래량만 정제

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (DATA 수집)

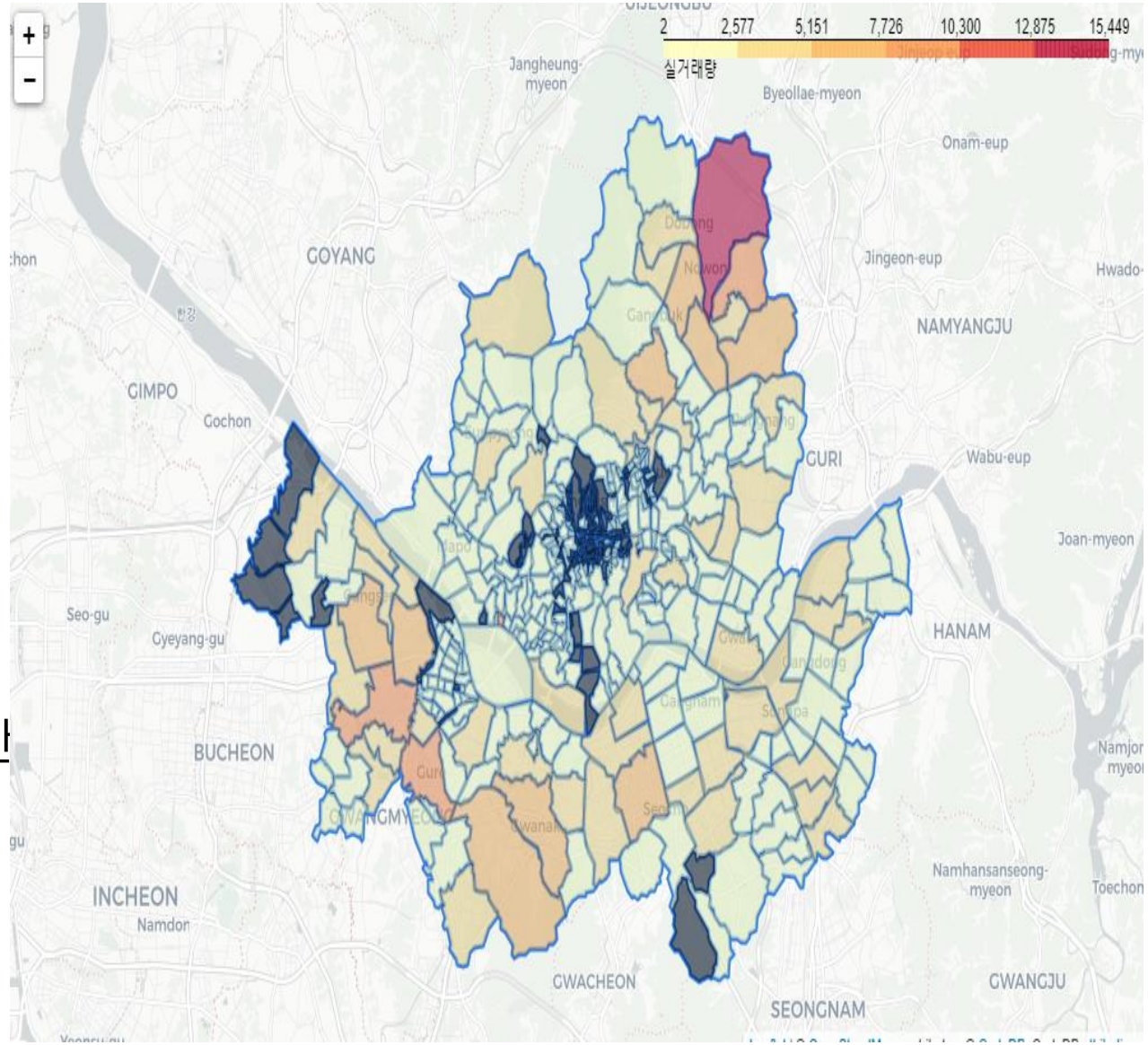
노원구 설정 이유

1. 구를 특정하기 위해
   서울 전체 거래 데이터를 시각화

2. 노원구가 가장 많은 거래량을 보임

3. 아파트가 없이 빌라 등만 존재하는 구도 있어
   5년 동안 아파트 거래가 없는 곳도 존재

4. 구를 특정한다면 거래량이 가장 높은
   노원구를 기점으로 확장하는 것이 좋다고 판단



04 프로젝트 수행 과정 및 결과 – 데이터 분석 (DATA 수집)

서울시5년거래량.csv 의 노원구 데이터만 추출, 노원구5년거래량.csv 저장

Microsoft Excel - 노원구5년거래량.csv:1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1		지역코드	도로명	법정동	지번	아파트	건축년도	층	전용면적	년	월	일	거래금액	도로명건물	도로명건물	도로명시군	도로명일련	도로명지상	도
2	0	11350	초안산로1	월계동	556	주공2	1992	3	84.815	2017	1	2	32000	18	0	11350	1	0	4
3	1	11350	초안산로2	월계동	946	우남푸르미	2006	9	59.92	2017	1	2	29000	91	0	11350	1	0	4
4	2	11350	석계로18길	월계동	380-1	삼능스페이	2006	4	84.81	2017	1	2	34000	23	0	11350	1	0	4
5	3	11350	월계로53길	월계동	940	동원베네스	2005	4	84.627	2017	1	2	35700	21	0	11350	1	0	4
6	4	11350	석계로	월계동	929	현대	2000	11	84.98	2017	1	2	41000	49	0	11350	1	0	3
7	5	11350	석계로	월계동	929	현대	2000	22	59.95	2017	1	2	30500	49	0	11350	1	0	3
8	6	11350	석계로13길	월계동	927	한일2	2000	4	114.816	2017	1	2	42700	6	10	11350	1	0	4
9	7	11350	마들로	월계동	18	한진한화그	2002	17	84.97	2017	1	3	42300	31	0	11350	1	0	3
10	8	11350	마들로	월계동	13	미성	1986	5	50.14	2017	1	4	28800	59	0	11350	1	0	3
11	9	11350	마들로	월계동	13	미륭	1986	6	51.48	2017	1	5	29100	111	0	11350	1	0	3
12	10	11350	덕릉로60길	월계동	923	초안1	1998	5	59.85	2017	1	5	24300	185	0	11350	1	0	4
13	11	11350	초안산로1	월계동	556	주공2	1992	10	44.52	2017	1	6	18750	18	0	11350	1	0	4
14	12	11350	마들로	월계동	18	한진한화그	2002	18	59.94	2017	1	7	37800	31	0	11350	1	0	3
15	13	11350	월계로45가	월계동	780	청백3	1998	2	49.77	2017	1	7	21600	89	0	11350	1	0	4
16	14	11350	초안산로1	월계동	556	주공2	1992	2	38.64	2017	1	7	16950	18	0	11350	1	0	4
17	15	11350	마들로	월계동	13	미성	1986	6	50.14	2017	1	7	28500	59	0	11350	1	0	3
18	16	11350	초안산로1	월계동	556	주공2	1992	8	38.64	2017	1	9	17400	18	0	11350	1	0	4
19	17	11350	초안산로1	월계동	556	주공2	1992	4	44.52	2017	1	10	19000	18	0	11350	1	0	4
20	18	11350	우이천로2	월계동	939	월계흥화브	2004	4	70.94	2017	1	10	31500	14	0	11350	1	0	4
21	19	11350	마들로	월계동	13	삼호3	1986	1	59.22	2017	1	10	32000	111	0	11350	1	0	3
22	20	11350	월계로45가	월계동	781	청백4	1998	14	39.84	2017	1	10	17300	94	0	11350	1	0	4
23	21	11350	우이천로2	월계동	939	월계흥화브	2004	6	84.91	2017	1	11	35000	14	0	11350	1	0	4
24	22	11350	광운로2나	월계동	436	동신	1983	4	70.81	2017	1	12	27000	30	0	11350	1	0	4
25	23	11350	월계로55길	월계동	320-11	사슴3	1995	10	39.6	2017	1	13	19800	15	0	11350	1	0	4
26	24	11350	광운로	월계동	925	대동	1997	9	59.76	2017	1	14	27000	46	0	11350	1	0	3
27	25	11350	석계로15길	월계동	926	한일1	1999	7	84.942	2017	1	14	37000	25	0	11350	1	0	4
28	26	11350	마들로	월계동	12	삼호4	1987	9	59.49	2017	1	14	31000	127	0	11350	1	0	3
29	27	11350	마들로	월계동	13	미륭	1986	14	51.48	2017	1	14	28500	111	0	11350	1	0	3
30	28	11350	마들로	월계동	13	삼호3	1986	12	59.22	2017	1	15	33800	111	0	11350	1	0	3

노원구5년거래량

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

8개의 모델을 사용해서 RMSLE를 비교하고, 가장 낮은 RMSLE가 나온 모델을 선정

- RMSLE (Root Mean Squared Log Error)
- 회귀 평가를 위한 지표는 실제 값과 회귀 에측값의 차이를 기반으로합니다.
- 이때 RMSLE가 작다는건 예측값과 실제값의 차이가 없다는 뜻으로 성능이 좋다는걸 알 수 있습니다.

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(p_i+1) - \log(a_i+1))^2}$$

$$p = Predicted,\ a = Actual$$

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

RSMLE를 비교할 8개 모델 선정

- Linear Regression
- Ridge Regression
- Lasso Regression
- Elasticnet Regression
- Decision Tree
- RandomForest Regressor
- XGBoost Regressor
- LightGBM Regressor

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

- 데이터 전처리 – 사용하지 않는 칼럼제거, 노원구 아파트 거래량 top20선정 그외 other로 정의
  Top20 선정 기준 -> df['아파트'].value_count()[:60} 메소드 사용하여 60개 추출

```
df=df.drop('거래유형',axis=1)
df=df.drop('중개사소재지',axis=1)
df=df.drop('해제사유발생일',axis=1)
df=df.drop('해제여부',axis=1)
df=df.drop('도로명',axis=1)
df=df.drop('일련번호',axis=1)
df=df.drop('지번',axis=1)
df=df.drop('도로명건물본번호코드',axis=1)
df=df.drop('도로명건물부번호코드',axis=1)
df=df.drop('도로명시군구코드',axis=1)
df=df.drop('도로명일련번호코드',axis=1)
df=df.drop('도로명지상지하코드',axis=1)
df=df.drop('도로명코드',axis=1)
df=df.drop('법정동본번코드',axis=1)
df=df.drop('법정동부번코드',axis=1)
df=df.drop('법정동시군구코드',axis=1)
df=df.drop('법정동읍면동코드',axis=1)
df=df.drop('법정동지번코드',axis=1)
```

```
[ ] df['apt_counts'] = 0
    df.groupby('아파트')['apt_counts'].count()
    df = pd.merge(df, df.groupby('아파트')['apt_counts'].count(), on='아파트', how='left').drop('apt_counts_x', axis=1).rename(columns={'apt_counts_y':'apt_counts'})
```

```
[ ] # 노원구 아파트 데이터 위주로 네임작성
    apt_names=['주공','중계','시영','사슴','한신','삼익','미룡','염광','극동','롯데','현대',
               '상계','태강','벽산','한진','장미','보람','은빛','청솔','풍림','삼호']
```

```
[ ] df['transformed'] = False
```

```
# `apt_names_list`의 키워드에 아파트명이 포함되면 해당 키워드로 아파트명을 통일함
# 그리고 `transformed` 컬럼값을 True로 변경
for a in tqdm(apt_names):
    df.loc[df['아파트'].str.contains(a), '아파트'] = a
    df.loc[df['아파트'].str.contains(a), 'transformed'] = True

# 아파트 이름이 변경되지 않았을 경우(`transformed=False` 일 경우) 아파트명을 'others'로 변경
for a in tqdm(apt_names):
    df.loc[~df['transformed'], '아파트'] = 'others'
print(df['아파트'].value_counts())
```

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

아파트 벨류값, 동 벨류값 Xgboost와 LightGBM 사용을 위해 라벨링 후 int 자료형 변환

아파트 라벨링

```
for i, a in enumerate(list(apt_price.index)):
    df.loc[df['아파트'] == a, '아파트'] = i # 라벨 인코딩
apt_price = df.groupby('아파트')['거래금액'].agg('mean').sort_values(ascending=False)
print('변환후₩n', apt_price[:20])
```

```
변환후
 아파트
0     69014.687500
1     61384.427284
2     53810.741990
3     51117.391698
4     50894.237288
5     49942.435424
6     49802.899949
7     48543.924497
8     47669.158249
9     45060.111111
10    45010.266272
11    42072.413793
12    41838.505747
13    41185.472579
14    40490.429429
15    39072.831367
16    37824.399736
17    37349.097778
18    33483.879310
19    33070.512745
Name: 거래금액, dtype: float64
```

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

아파트 벨류값, 동 벨류값 Xgboost와 LightGBM 사용을 위해 라벨링 후 int 자료형 변환

동 라벨링

```
# 가격기준으로 동을 정렬한 리스트를 바탕으로 dong에 대해 라벨 인코딩 진행 - 477 it.
for i, d in tqdm(enumerate(list(dong_price.index)), total=len(dong_price)):
    df.loc[df['법정동'] == d, '법정동'] = i
# print(train_df.head())
print(df.describe())
```

```
100%|██████████| 5/5 [00:00<00:00, 195.92it/s]         Unnamed: 0     지역코드         건축년도             층         전용면적  ₩
count  39620.000000  39620.0  39620.000000  39620.000000  39620.000000
mean   19809.500000  11350.0   1993.854366      8.079278     64.281776
std    11437.453169      0.0      6.560588      4.878513     22.832953
min        0.000000  11350.0   1976.000000      1.000000     12.060000
25%     9904.750000  11350.0   1988.000000      4.000000     49.500000
50%    19809.500000  11350.0   1992.000000      8.000000     59.390000
75%    29714.250000  11350.0   1999.000000     12.000000     84.600000
max    39619.000000  11350.0   2022.000000     36.000000    180.340000

                 년             월             일          거래금액    apt_counts
count  39620.000000  39620.000000  39620.000000   39620.000000  39620.000000
mean    2018.758001      5.330262     16.081045   45082.096870    399.644573
std        1.397460      3.218722      8.718482   19344.133549    307.326794
min     2017.000000      0.000000      1.000000    6750.000000      1.000000
25%     2017.000000      3.000000      9.000000   31000.000000    148.000000
50%     2019.000000      5.000000     16.000000   40900.000000    293.000000
75%     2020.000000      8.000000     24.000000   55000.000000    625.000000
```

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

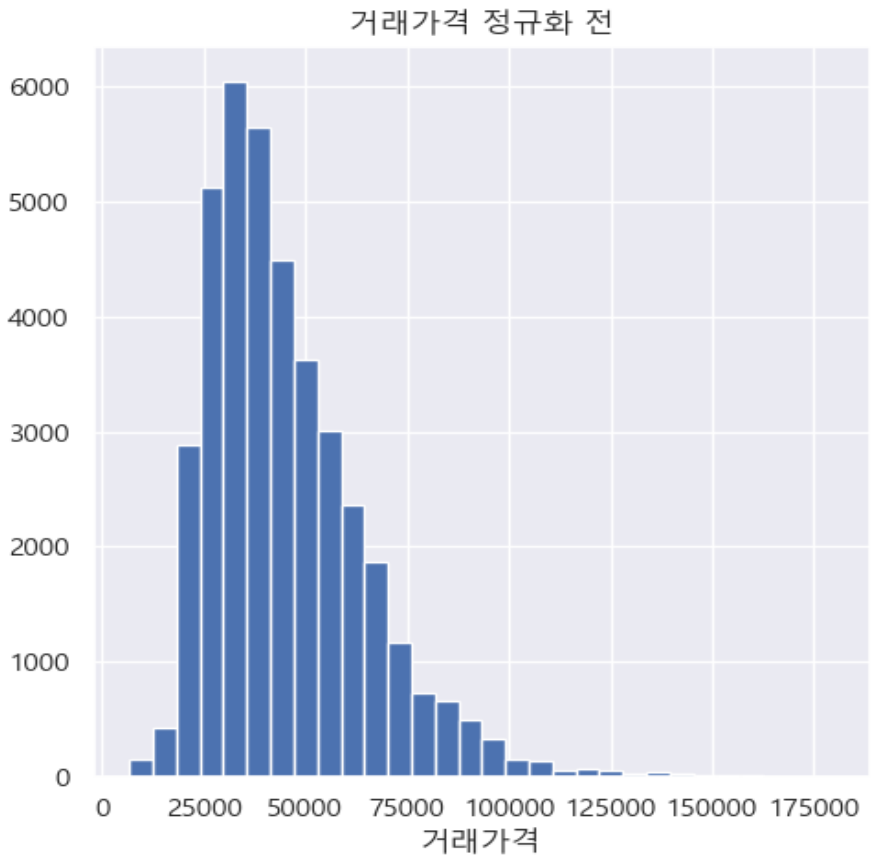
- 아파트 벨류값, 동 벨류값 Xgboost와 LightGBM 사용을 위해 라벨링 후 int 자료형 변환

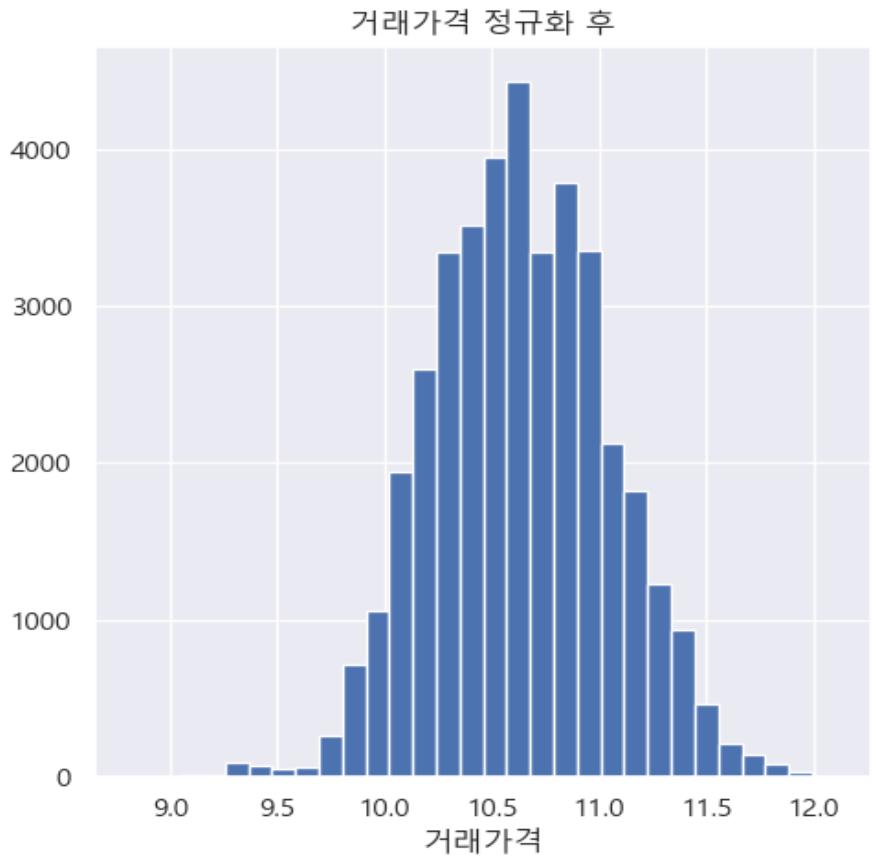
  자료형 변환

```
[ ]  # 형변환
     df['법정동'] = df['법정동'].astype('int64')
     df['아파트'] = df['아파트'].astype('int64')
```

04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

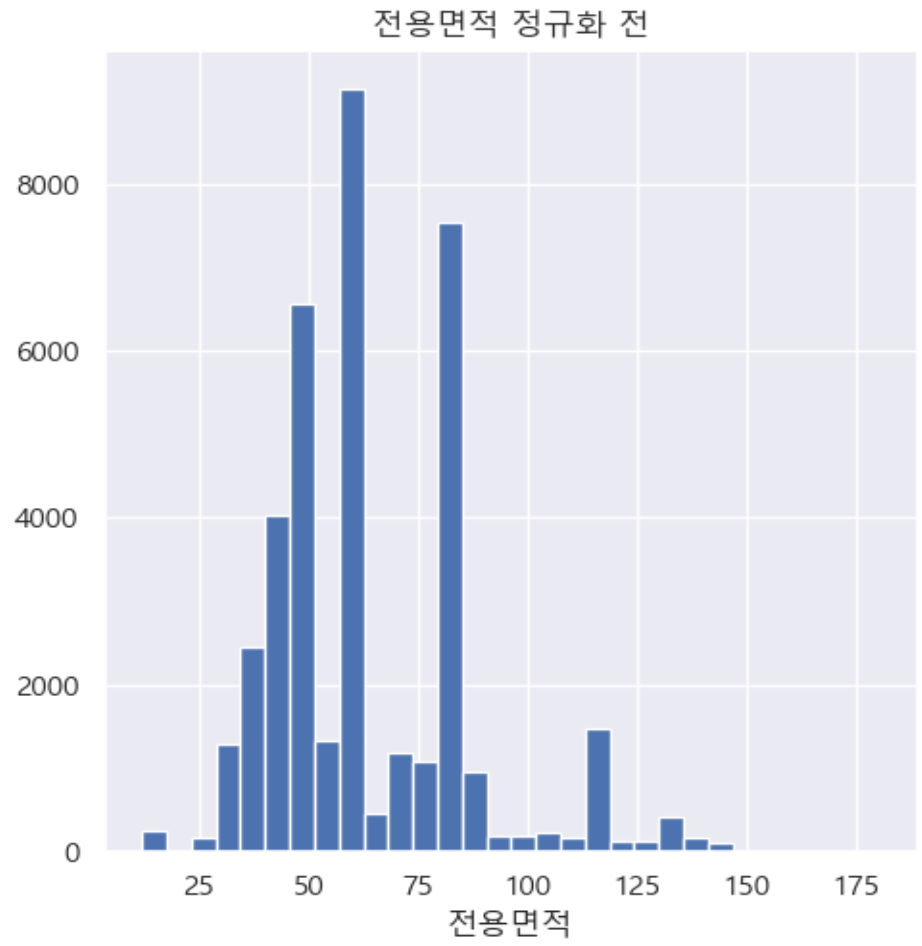
- 거래금액이 왼쪽으로 치우쳐 있어, 정규화작업

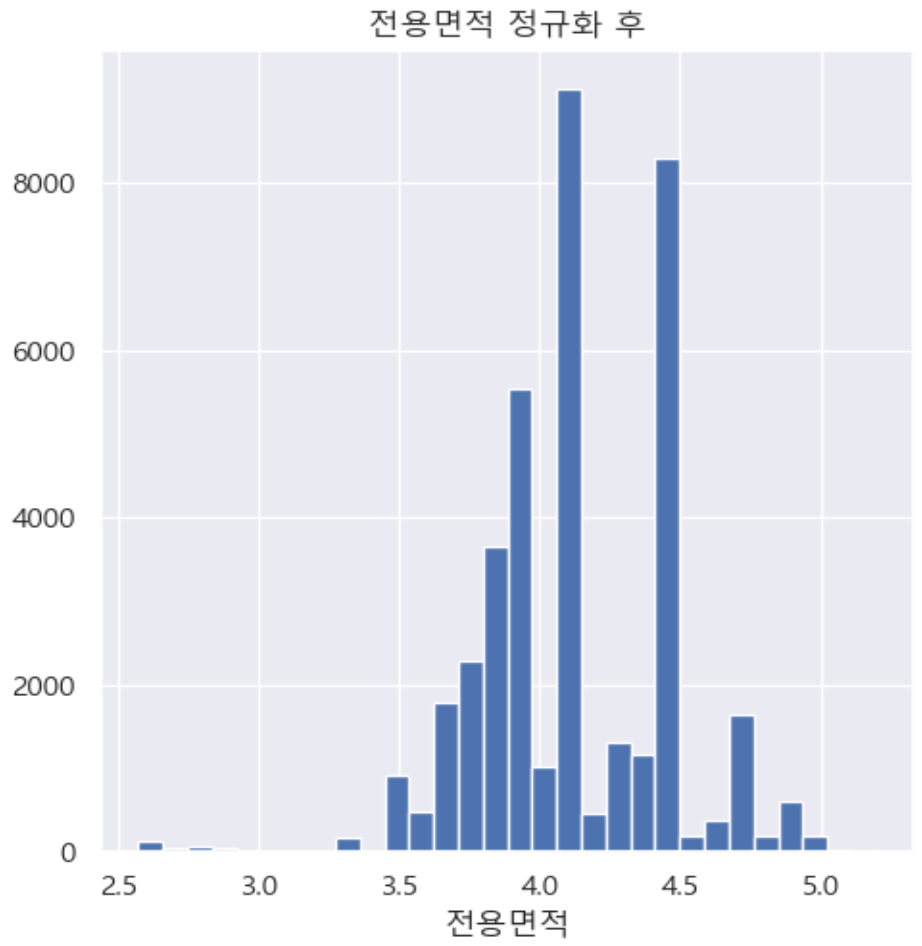




04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

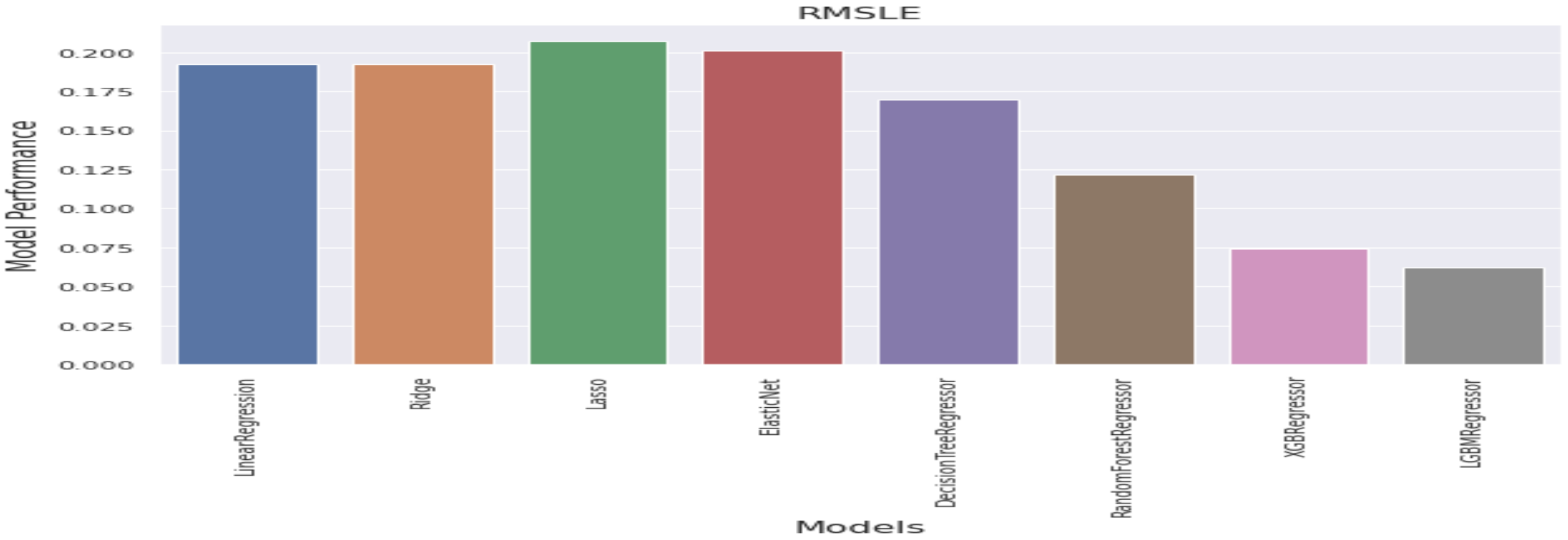
전용면적도 정규화 작업





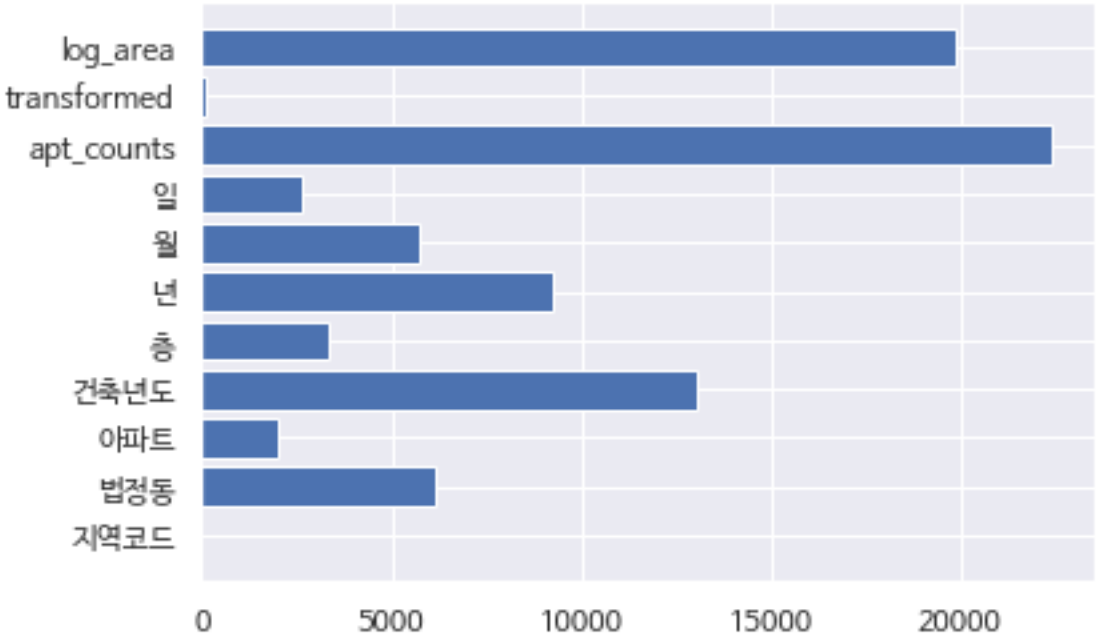
04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

- 8개 모델로 데이터 분석, RMSLE 분석 결과
  제일 수치가 적게나온 LightGBM Regressor 사용



04 프로젝트 수행 과정 및 결과 – 데이터 분석 (Model)

거래금액에 영향을 주는 값 시각화 및 예측



Microsoft Excel - 노원구아파트모델예측결과 보기.csv

	A	B	C	D	E
1	아파트	예측가격			
2	17	35355.03			
3	3	87572.09			
4	3	76395.11			
5	3	113694.1			
6	3	50078.84			
7	14	24878.36			
8	14	36318.82			
9	17	40705.32			
10	3	36394.02			
11	19	33077			
12	3	46985.77			
13	14	62122.94			
14	3	68833.18			
15	9	57641.21			
16	6	60536.57			
17	3	21857.12			
18	14	54060.96			
19	3	71709.39			
20	14	49967.24			
21	7	31206.79			
22	3	34129.19			
23	14	40355.87			
24	3	37581.74			
25	3	29787.08			
26	3	60322.89			
27	18	37004.58			
28	16	27385.42			
29	14	48615.66			
30	6	45919.57			

04 프로젝트 수행 과정 및 결과 –

```
target = target.groupby("transaction_at").agg("mean")
```

0.9s

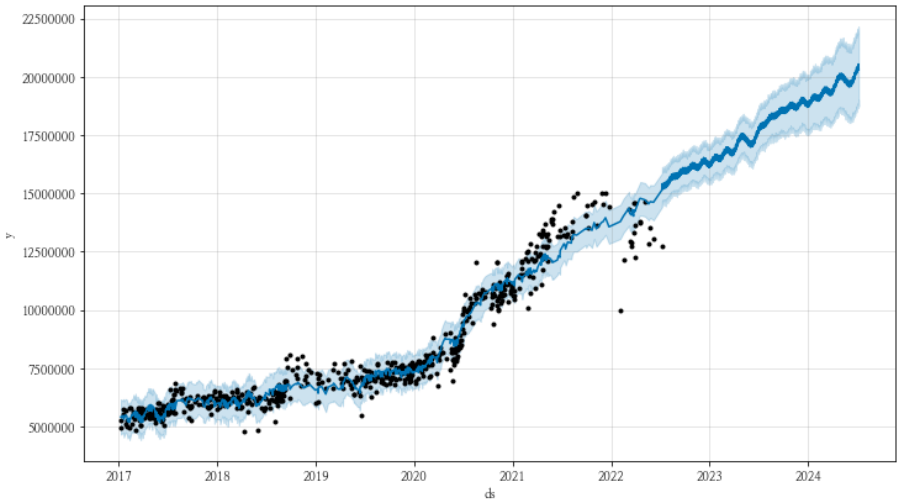
```
target = target.sort_values(by=["transaction_at"])

target["y"] = target["거래금액"] / target["전용면적"] # 1 평당 3.3057 m^2
target["y"] = round(target["y"] * 10_000)
target["ds"] = target.index

target
```

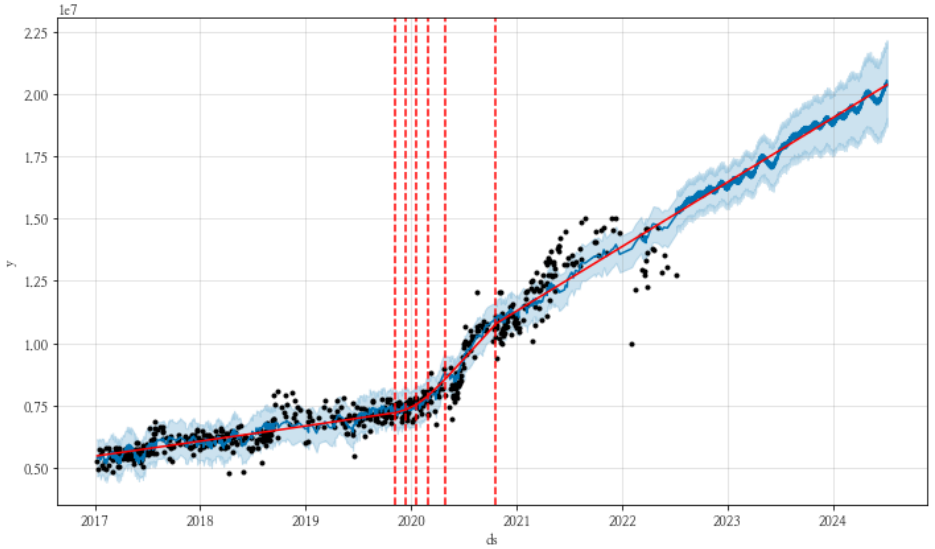
0.1s

transaction_at	층	전용면적	거래금액	해제여부	y	ds
2017-01-09	10	59	31200	0	5247225	2017-01-09
2017-01-11	1	46	23000	0	4953694	2017-01-11
2017-01-18	11	50	28300	0	5717172	2017-01-18
2017-01-20	6	50	27500	0	5516550	2017-01-20
2017-01-25	10	50	28250	0	5707071	2017-01-25
...	...	...	...	...	...	...
2022-05-06	5	40	58500	0	14657980	2022-05-06
2022-05-20	5	50	63500	0	12828283	2022-05-20
2022-05-26	8	50	67000	0	13535354	2022-05-26
2022-06-07	1	40	52000	0	13029316	2022-06-07
2022-07-08	3	50	63000	0	12727273	2022-07-08

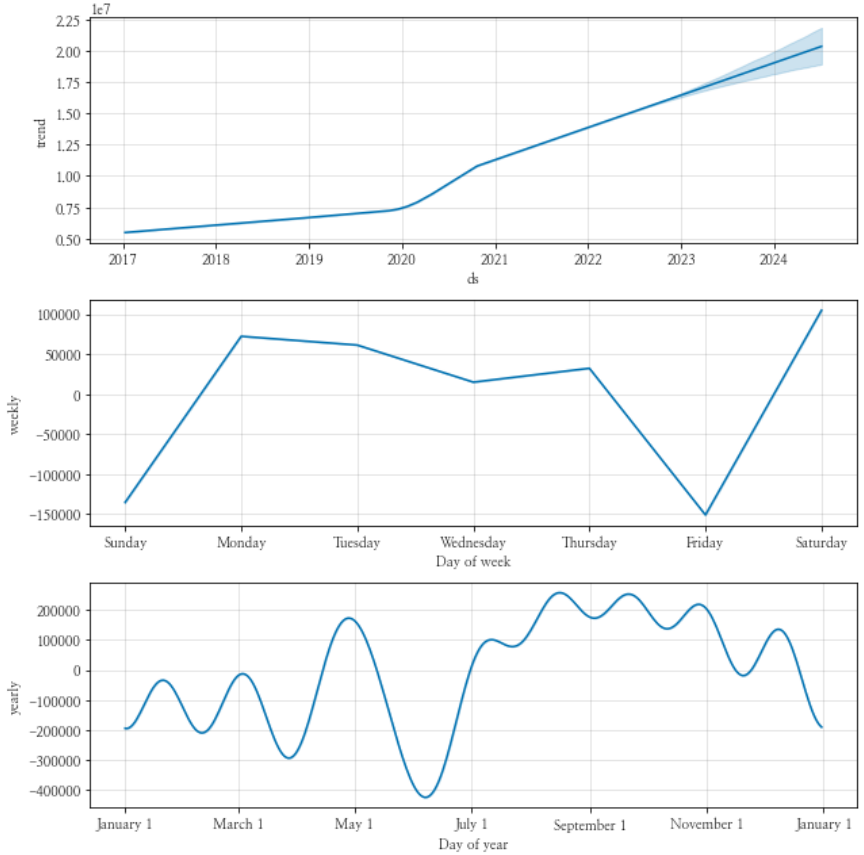
```
future = model.make_future_dataframe(periods=365)
pred = model.predict(future)
```

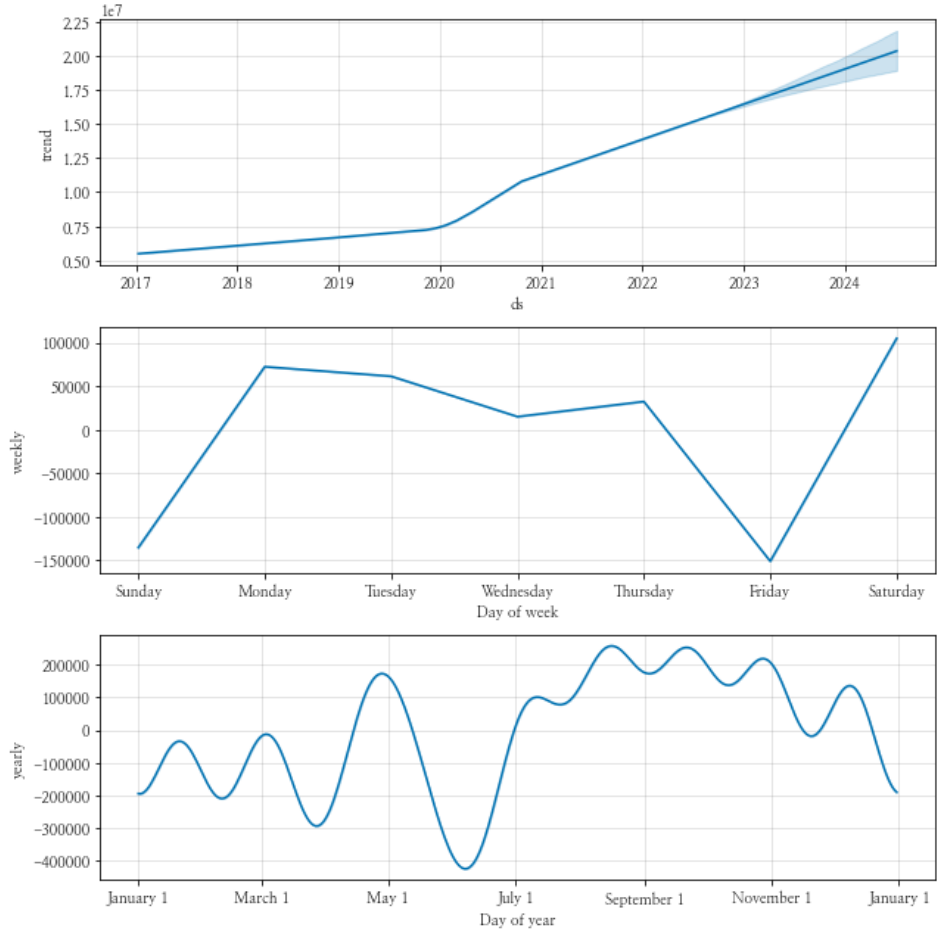
04 프로젝트 수행 과정 및 결과 –





04 프로젝트 수행 과정 및 결과 –





04 프로젝트 수행 과정 및 결과 –

05 자체 평가 의견

데이터 분석

김혁* – 데이터분석 협업을 통해 잘 몰랐던 데이터분석을 공부하게 될 수 있는 계기가 되어서 좋았고, 2~3년뒤 데이터 엔지니어가 꿈인 저에게 좋은 경험이 되었다고 생각합니다.

노태* - 데이터 전처리, 정규화 및 여러 모델을 비교해 모델을 선정하는 것 까지 좋은 경험이었다.

안정* - raw data를 직접 찾으며 가공하고 모델을 만드는 것이 쉽지는 않았지만 궁금한 주제를 직접 선정하고 원하는 결과를 이끌어내는 과정이 의미깊었습니다. 주제에 대해 더 새로운 관점에서 바라볼 수 있는 시각을 길러서 다양한 도전을 해보고 싶다는 생각을 했습니다.

05 자체 평가 의견

강현* – 쟝고와 스프링 둘 같이 써볼 수 있어 좋은 경험이었지만 아쉬운 것이 많이 남는 것 같다..

전해* - API의 데이터 처리가 복잡했지만 도전해 볼 수 있어서 재밌었다.

최광* – 데이터 처리만 좀 더 수월했다면하는 아쉬움이 남는 프로젝트였다.

감사합니다.