2조 스터디 그룹

2조
진성준 김준호 김하연 송현지 엄효범 정다슬

목차

1. 데이터 전처리 및 EDA
   a. 보간법 이용
      i. 의미 없는 변수 제거
      ii. 결측치 100만 이상 변수 제거
   b. 이상치 처리
      i. IQR 사용

2. 변수 선택
   a. 베이스라인
      i. 강남/강북
      ii. 신축/구축
   b. 사용 피처
      i. 아파트 노후도
      ii. 학군 점수
      iii. 역세권 점수
      iv. 한국은행 기준금리

3. 모델 선택
   a. 베이스라인
      i. RandomForest
   b. 사용 모델
      i. LightGBM
   -> 사용 변수 채택
   c. 좋은 예측과 나쁜 예측에 대한 보완

4. 추론 및 결과 분석

01

팀원 소개

팀원 소개

진성준 (usniz300@gmail.com)	김준호 (demyank88@gmail.com)	김하연 (leah8735@gmail.com)
송현지 (hyunjeesong95@gmail.com)	엄효범 (bronzemedal40@gmail.com)	정다슬 (anyone158911@gmail.com)

02

대회 소개

대회 개요

House Price Prediction | 아파트 실거래가 예측

서울시 아파트 실거래가 매매 데이터를 기반으로 아파트 가격을 예측하는 대회

다양한 부동산 관련 의사결정을 돕고자 하는 부동산

실거래가를 예측하는 모델을 개발

시세를 예측하여 적정 가격을 판단하는 기준지표로 삼기 위함

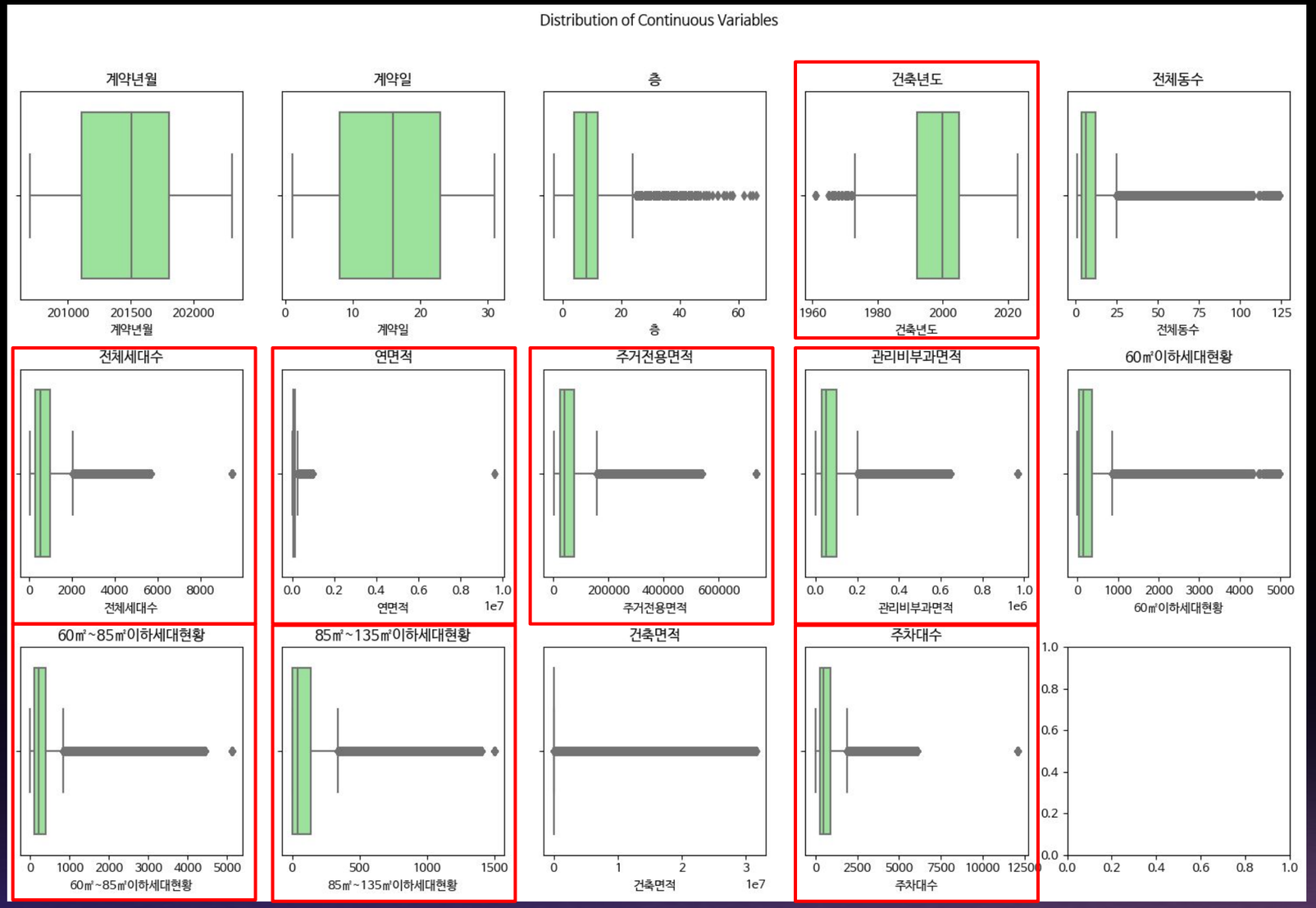
제공 데이터

- 아파트 실거래가 데이터[국토교통부 제공]
- 지하철역 [서울시 제공]
- 버스정류장 [서울시 제공]
- 평가 데이터

03

Data Description

데이터 전처리 - 이상치 처리



외부 데이터 활용

- 학군 변수
  - 서울시 고등학교별 수능 국어수학 영역 백분위 평균 (출처:https://cafe.naver.com/we2you/6167)
  - kakao rest api 좌표에서 가까운 고등학교 상위 3개 학교
  - 3개 학교의 거리 평균, 백분위 평균

- 한국은행 기준금리 (출처:https://www.bok.or.kr/portal/singl/baseRate/list.do?dataSeCd=01&menuNo=200643)

Feature Engineering

- 위경도 좌표를 3차원으로 변환
  - 지구가 평면이 아닌 입체기 때문에 sin, cos 함수를 활용해 3차원 좌표로 변형해주는 것이 좋음 (출처:http://www.gisdeveloper.co.kr/?p=11030)

```
coslat = concat_select['X'].apply(lambda x : math.cos(x))
coslong  = concat_select['Y'].apply(lambda x : math.cos(x))
sinlat = concat_select['X'].apply(lambda x : math.sin(x))
sinlong = concat_select['Y'].apply(lambda x : math.sin(x))

concat_select['x3'] = coslat*coslong
concat_select['y3'] = coslat*sinlong
concat_select['z3'] = sinlat
```

고등학교 거리	학군점수
301.0	55.826667

Feature Engineering

Cyclical Feature

단지승인일
2022-11-17 13:00:29.0
2022-11-17 13:00:29.0
2022-11-17 13:00:29.0
2022-11-17 13:00:29.0
2022-11-17 13:00:29.0

단지신청일
2022-11-17 10:19:06.0
2022-11-17 10:19:06.0
2022-11-17 10:19:06.0
2022-11-17 10:19:06.0
2022-11-17 10:19:06.0

k-사용검사일-사용승인일
1987-11-21 00:00:00.0
1987-11-21 00:00:00.0
1987-11-21 00:00:00.0
1987-11-21 00:00:00.0
1987-11-21 00:00:00.0

k-등록일자	k-수정일자
2022-11-09 20:10:43.0	2023-09-23 17:21:41.0
2022-11-09 20:10:43.0	2023-09-23 17:21:41.0
2022-11-09 20:10:43.0	2023-09-23 17:21:41.0
2022-11-09 20:10:43.0	2023-09-23 17:21:41.0
2022-11-09 20:10:43.0	2023-09-23 17:21:41.0

Feature Engineering

Cyclical Feature

```
lst = [
    ('k-사용검사일-사용승인일','사용승인일'),
    ('k-등록일자','등록일자'),
    ('k-수정일자','수정일자'),
    ('단지승인일','단지승인일'),
]
for x1,x2 in lst:
    df_train.loc[df_train[x1].isna(),x1] = ""
    df_train[f'{x2} 년'] = df_train[x1].map(lambda o: int(str(o)[:4]) if o!="" else None)
    df_train[f'{x2} 월'] = df_train[x1].map(lambda o: int(str(o)[5:7]) if o!="" else None)
    df_train[f'{x2} 일'] = df_train[x1].map(lambda o: int(str(o)[8:10]) if o!="" else o)
    df_train[f'{x2} 월 sin'] = df_train.apply(lambda o: None if o[x1]=="" else np.sin((2*int(o[x1][5:7]))*np.pi)/12,axis=1)
    df_train[f'{x2} 월 cos'] = df_train.apply(lambda o: None if o[x1]=="" else np.cos((2*int(o[x1][5:7]))*np.pi)/12,axis=1)
    df_train[f'{x2} 일 sin'] = df_train.apply(lambda o: None if o[f'{x2} 일']=="" else np.sin(2*int(o[f'{x2} 일'])*np.pi)/days_cyclical_denominator(
    df_train[f'{x2} 일 cos'] = df_train.apply(lambda o: None if o[f'{x2} 일']=="" else np.cos(2*int(o[f'{x2} 일'])*np.pi)/days_cyclical_denominator(

    df_train = df_train.drop(columns=[x1, f'{x2} 월',f'{x2} 일'])
```

Feature Engineering

Categorical Feature

```
df_train['k-복도유형'].unique()
```

✓ 0.0s

```
array(['계단식', '혼합식', '복도식', '타워형', '기타', nan], dtype=object)
```

Feature Engineering

Categorical Feature

```
df_train['k-복도유형'].unique()
```

✓ 0.0s

```
array(['계단식', '혼합식', '복도식', '타워형', '기타', nan], dtype=object)
```

pd.get_dummies 사용

```
df_train.loc[df_train['k-복도유형'].isna(),'k-복도유형'] = "알수없음"
df_tmp = pd.get_dummies(df_train['k-복도유형'],dtype=int)
df_tmp.columns = list(map(lambda o: f"복도유형 {o}",df_tmp.columns.tolist()))
df_train = pd.concat([df_train,df_tmp],axis=1)

df_train = df_train.drop(columns='k-복도유형')
```

Feature Engineering

Categorical Feature

One-Hot Encoding

복도유형 계단식	복도유형 기타	복도유형 복도식	복도유형 알 수없음	복도유형 타워형	복도유형 혼합식
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0

Feature Engineering

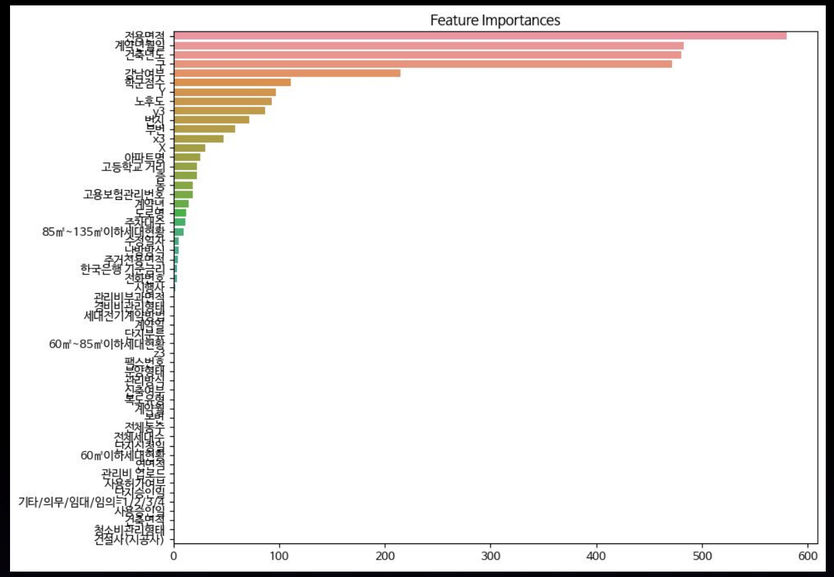
- 지하철 역 데이터
  - 가장 가까운 지하철역까지의 거리
  - 해당 지하철역에서 탈 수 있는 호선의 개수

```
concat_select['지하철역까지의 거리'] = 0.0
concat_select['지하철 호선 개수'] = 0

for idx, row in concat_select.iterrows():
    property_location = (row['Y'], row['X'])

    filtered_subway_df = subway_df[
        (subway_df['위도'] >= property_location[0] - 0.1) & (subway_df['위도'] <= property_location[0] + 0.1) &
        (subway_df['경도'] >= property_location[1] - 0.1) & (subway_df['경도'] <= property_location[1] + 0.1)
    ]

    min_distance = float('inf')
    closest_station_info = None

    for _, station_row in filtered_subway_df.iterrows():
        station_location = (station_row['위도'], station_row['경도'])
        distance = geodesic(property_location, station_location).km

        if distance < min_distance:
            min_distance = distance
            closest_station_info = {
                '역사명': station_row['역사명'],
                '호선': station_row['호선']
```

지하철역까지의 거리	지하철 호선 개수
1.127738	1

Feature Select

Weight	Feature
351502590.3407 ± 1073374.4225	계약년월일
243288934.7088 ± 5824161.7460	전용면적
91154964.4396 ± 1129580.8286	구
80239207.4328 ± 1487766.4897	강남여부
59488603.6229 ± 832421.4046	건축년도
12476889.7434 ± 142011.5959	Y
9175481.9676 ± 53662.9667	노후도
8212196.6383 ± 425344.0989	x3
5340003.8051 ± 84509.4711	학군점수
4633711.7633 ± 20701.7851	y3
3403871.0925 ± 54306.6529	부번
1890649.5703 ± 92211.1470	아파트명
1757389.6602 ± 66946.9088	X
1322124.9022 ± 21378.3206	고등학교 거리
1244737.8145 ± 26944.8203	층
989914.6792 ± 147530.4007	번지
946205.0576 ± 55461.7013	도로명
631273.7605 ± 38962.8210	85㎡~135㎡이하세대현황
419047.3808 ± 71131.1926	주거전용면적
392356.8352 ± 24382.5430	동
... 35 more ...	



04

Modeling

Model Select

```python
from sklearn.ensemble import RandomForestRegressor

# Random Forest 모델 생성 및 학습
model_rf = RandomForestRegressor()
model_rf.fit(x, y)

# 테스트 데이터로 예측
pred_rf = model_rf.predict(test)
```

```python
from catboost import CatBoostRegressor

# CatBoost 모델 생성 및 학습
model_catboost = CatBoostRegressor()
model_catboost.fit(x, y)

# 테스트 데이터로 예측
pred_catboost = model_catboost.predict(test)
```

93284.1274	LGBM Regressor
93917.4369	Random Forest
96494.8808	CatBoost

05

결과

최종 순위 및 평가지표 결과

최종 제출

32	Finished		92827.5670	상세 보기	2024-01-25 11:57			
27	Finished		88064.8558	상세 보기	2024-01-20 19:41			

06

그룹 스터디 진행 소감

그룹 스터디 진행 소감

진성준: 첫 프로젝트는 아니지만 대회 중 채워지는 게시판과 단기간의 집중으로 이제서야 프로젝트가 어떻게 참여하는지 알게 됨

김준호: 개인 사정으로 적극적으로 참여하지 못한 점이 아쉬움.

김하연: house price predict를 서울시 데이터로 진행하니 재미있었습니다.

송현지: 어려웠지만 좋은 기회였습니다. 다음 대회에서는 더 열심히 해보겠습니다.

엄효범: 좋은 스코어를 내면서 순위를 올려나가는게 즐거웠습니다

정다슬: 도메인 지식이 부족하여 어려움이 있었지만 대회 프로젝트 경험을 쌓아 다음번에는 더 좋은 점수를 내보도록 하겠습니다

Q&A

감사합니다.