

# 인공지능 경매 추천 프로그램

- 당신을 위한 최적의 부동산 투자대안-

과정명: 머신러닝을 활용한 빅데이터 분석과정 A

팀명: APTi (3팀)

구성원: 이유\*(팀장), 박진\*, 송창\*



# 목차

1. 프로그램 개발배경
2. 프로그램 시연
3. 프로그램 프로세스
4. 데이터 수집 및 전처리
5. 탐색적 데이터 분석 결과
6. 프로그램 알고리즘
7. 예측 모형 훈련결과
8. 제한점 및 추후 발전방향

# 1. 프로그램 개발배경

- 수도권 부동산 가격의 급등으로 인해 '내 집 마련'의 꿈을 실현하는 것은 점점 더 어려워지고, 청약은 경쟁이 심해 하늘의 별 따기가 되어간다.
- 이에 사람들은 상대적으로 낮은 가격을 기대할 수 있고 경쟁이 덜 한 부동산 경매에 눈을 돌리고 있다.
- 하지만 경매의 경우, 일반 매매거래보다 과정이 생소하고, 고려해야할 사항이 많다. 또한 낙찰을 받을 수 있는 가격을 예상하여 제시해야 한다.
- 따라서, 경매에 익숙하지 않더라도 개인의 성향에 맞는 매물을 가능성 있는 낙찰가와 함께 추천하고, 더 나아가 이후 부동산 매각 시 예상 차익도 함께 제시하여 내 집 마련과 더불어 성공적인 부동산투자를 돕고자 한다.

# 1. 프로그램 시연

- A 씨는 서울에 위치한 아파트를 취득하고자 한다. 최근 부동산 경매에 관심이 생겼지만, 경매에 익숙하지 못한 A 씨는 경매가 어떻게 이루어지는지, 또 원하는 자산을 얼마에 구할 수 있을지 도무지 알 길이 없다. 따라서, A 씨는 부동산투자자문 회사인 호우유진 리얼에스테이트 LLP의 사이트에 방문하여 적합한 매물을 추천 받고자 한다.
- A 씨의 예산은 10억이며, 투자성향은 적극적인 투자자형이다. (5년 후쯤 매매계획 有)

자본금을 입력하세요:10000000000  
번호나 이름을 입력하십시오.  
1.완벽주의 투자자  
2.신중하 투자자  
3.적극적인 투자자  
4.모험심 많은 투자자  
5.공격적인 투자자  
투자타입을 설정하세요:3

# 1. 프로그램 설명 및 시연

예상 매매날짜(연,월)를 YYYYMM 형태로 기입하세요.202609

자본금을 입력하세요:10000000000

번호나 이름을 입력하시오.

- 1.안전지향형 투자자
- 2.신중한 투자자
- 3.적극적인 투자자
- 4.모험적인 투자자
- 5.공격적인 투자자

투자타입을 설정하세요:3

서울특별시 관악구 남현동 1134 예상낙찰가:4.005억 ,예상차익:1.313억, 투자가능도:적정, 예상경쟁률:치열

서울특별시 중랑구 면목동 1527 예상낙찰가:5.135억 ,예상차익:3.328억, 투자가능도:적정, 예상경쟁률:치열

서울특별시 서대문구 홍제동 247-19 예상낙찰가:5.903억 ,예상차익:4.917억, 투자가능도:적정, 예상경쟁률:치열

서울특별시 도봉구 쌍문동 711 예상낙찰가:3.979억 ,예상차익:1.245억, 투자가능도:적정, 예상경쟁률:치열

적정가격 이외의 고수익 매물을 확인하시겠습니까?1

확인을 원하시면 1번을 누르세요.

뒤로 가시려면 아무키나 누르세요.

종료를 원하시면 2번을 누르세요.

확인된 매물:793 개

최저금리로 5분대출 전화한통일면 OK

070-0000-0000 언제나 열려있습니다.

가족처럼 모시겠습니다.

# 1. 프로그램 설명 및 시연

- 5년 후가 아닌, 낙찰 받은 후 바로 매매한다고 가정했을 때의 예측 결과

---

```
addr_name      서울특별시 노원구 상계동 685  
brand_name     보람아파트1단지  
Name: 1004, dtype: object 예측: 예상낙찰가:3.409억 , 예상차익:0.638억, 투자가능도:적정, 예상경쟁률:cool
```

---

```
addr_name      서울특별시 노원구 공릉로 213  
brand_name     동신아파트  
Name: 1006, dtype: object 예측: 예상낙찰가:5.062억 , 예상차익:3.967억, 투자가능도:적정, 예상경쟁률:cool
```

---

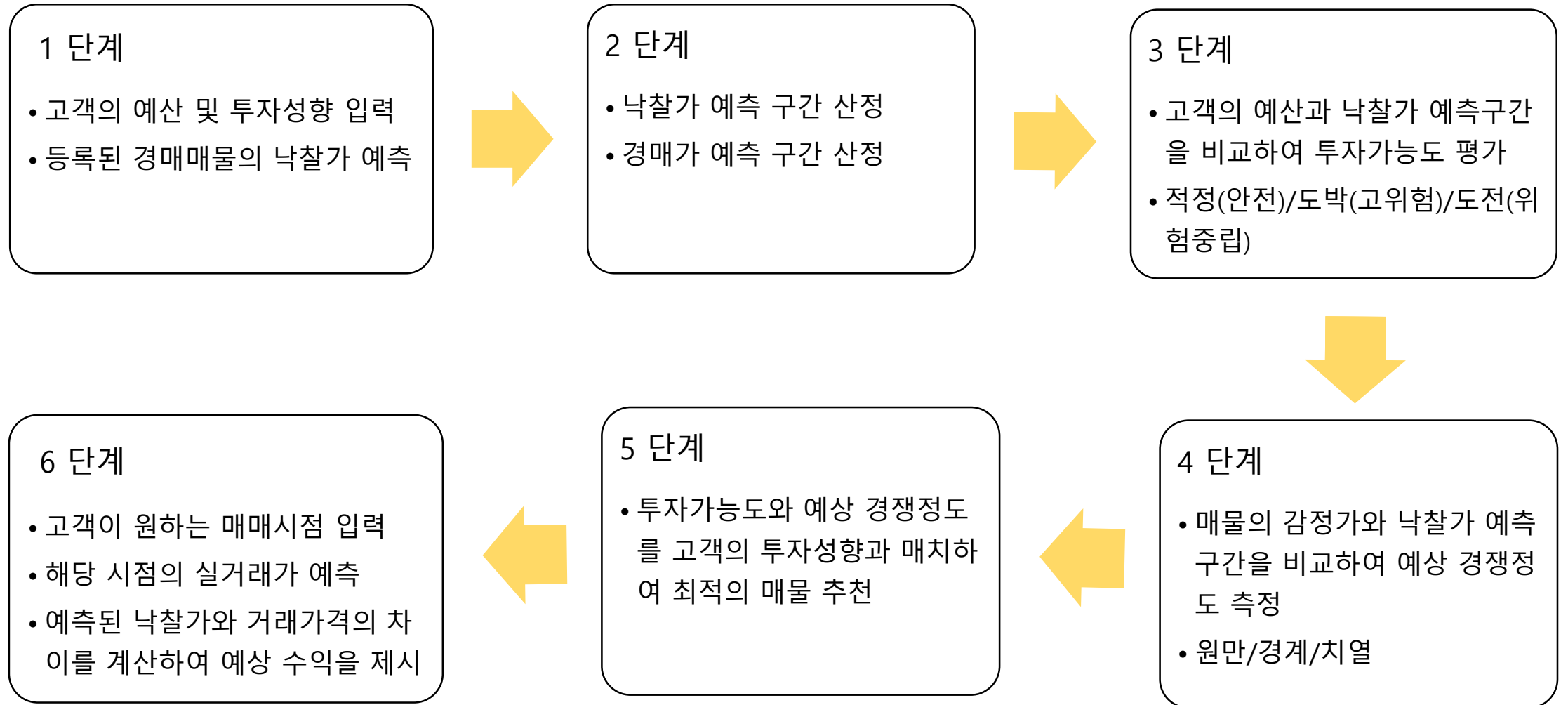
```
addr_name      서울특별시 금천구 시흥동 109-1  
brand_name     무지개아파트  
Name: 1012, dtype: object 예측: 예상낙찰가:4.004억 , 예상차익:0.964억, 투자가능도:적정, 예상경쟁률:cool
```

---

```
addr_name      서울특별시 양천구 신정로 260  
brand_name     신트리4단지아파트  
Name: 1013, dtype: object 예측: 예상낙찰가:2.931억 , 예상차익:0.369억, 투자가능도:적정, 예상경쟁률:cool
```

---

## 2. 프로그램 프로세스



## 2. 프로그램 프로세스

- 현재 등록된 경매 매물 중 A 씨에게 적합한 매물 추천하기

1. 고객의 예산 및 투자성향 입력

투자성향				
안전지향	신중한	적극적인	모험적인	공격적인
1	2	3	4	5

2. 머신러닝 모델을 통해 현재 경매 등록된 매물의 낙찰가 및 거래가격 예측
  - 예측된 가격들을 근거로 고객의 예산수준에 적합한지, 그리고 입찰 경쟁의 수준을 파악
  - 위 두 조건을 조합하여 고객의 투자성향에 맞는 경매매물을 추천

경쟁정도 \ 예산수준	cool	경계	치열
적정	1	2	3
도전	2	3	4
도박	3	4	5



# 3. 데이터 수집 및 전처리

- 수집한 데이터 및 출처

- 1) 아파트 경매 자료

- 기간: 2018 ~ 2021년 (물건등록일/경매기일 기준)
    - 대상: 서울 소재 아파트
    - 출처: 굿프렌드경매(<https://www.gfauction.info>)

- 2) 아파트 실거래가 자료

- 기간: 2018 ~ 2021년 (매매계약일 기준) + 2015 ~ 2017년 자료 추가
    - 대상: 서울 소재 아파트
    - 출처: 국토교통부 실거래가 공개시스템([molit.go.kr](http://molit.go.kr))



# 3. 데이터 수집 및 전처리

## 1) 경매 데이터 수집 및 전처리

- 굿프렌드경매 사이트에 공시된 경매 매물 정보를 크롤링을 통해 수집
- 데이터 컬럼
  - 관할법원, 사건번호
  - 번지명 및 도로명 주소
  - 해당 매물의 동, 호, 층
  - 건물 및 대상토지면적
  - 감정가, 최저가, 낙찰가 및 감정가대비 낙찰가 및 최저가 비율
  - 경매 진행상황, 낙찰일 (또는 예정 경매기일)
  - 특수조건
- 매각 완료된 매물 중 낙찰가가 없는 경우, 최저가로 대체

## 2) 실거래가 데이터 수집 및 전처리

- 국토교통부 공시 실거래가 자료
- 데이터 컬럼
  - 거래 연월일
  - 면적
  - 거래가격
  - 주소(관할 시, 구, 동 또는 도로명)
  - 건축년도

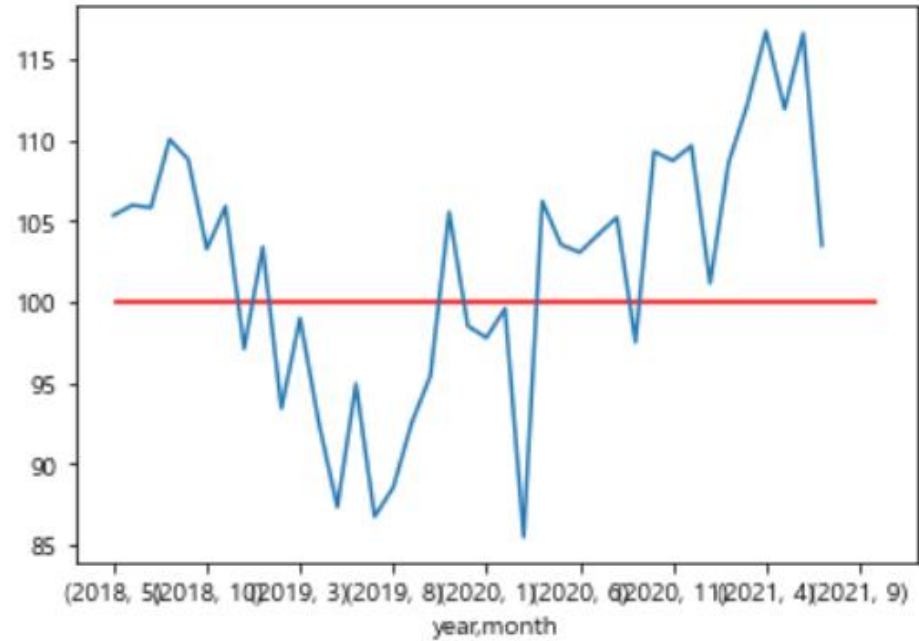
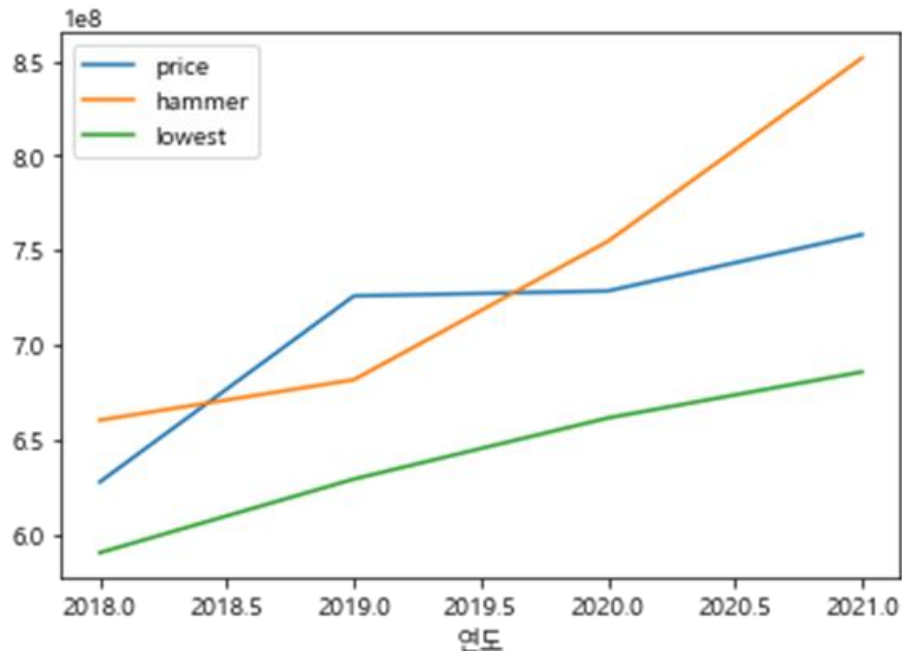
## 3) 공통 데이터 추가 수집

- 아파트 브랜드명을 변수에 포함하고자 데이터 셋의 주소를 이용하여 카카오 API, 네이버주소 API로부터 브랜드명 및 해당 아파트의 위도 및 경도 정보 수집

# 4. 탐색적 데이터 분석 결과

## 1) 경매 데이터 EDA 결과

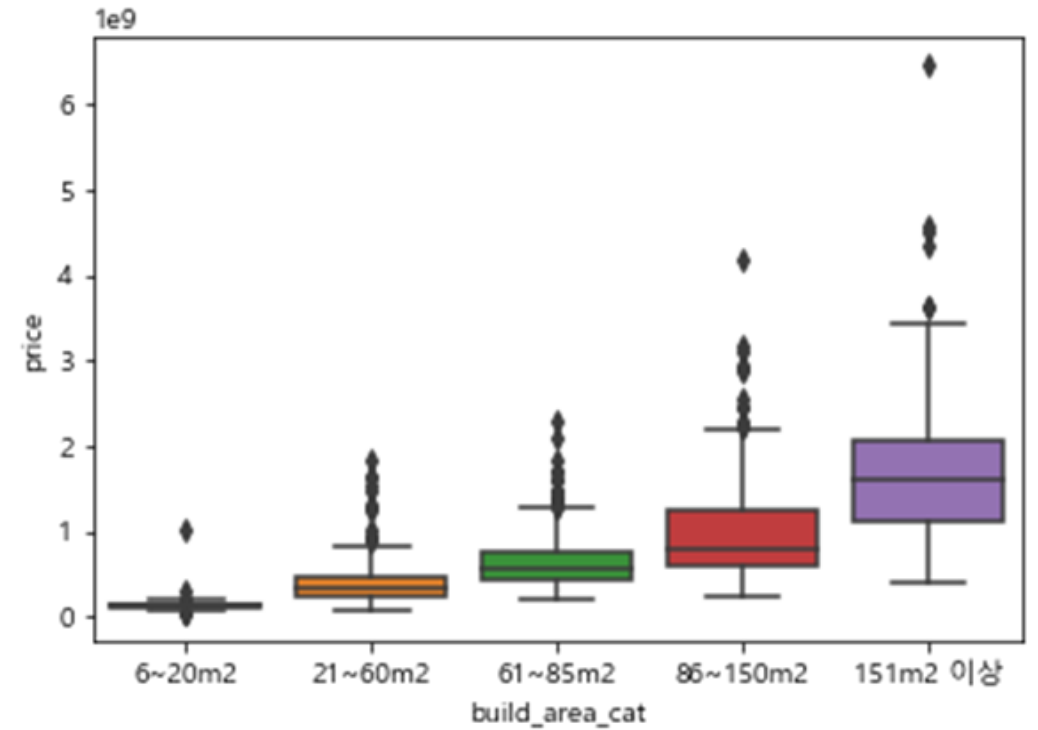
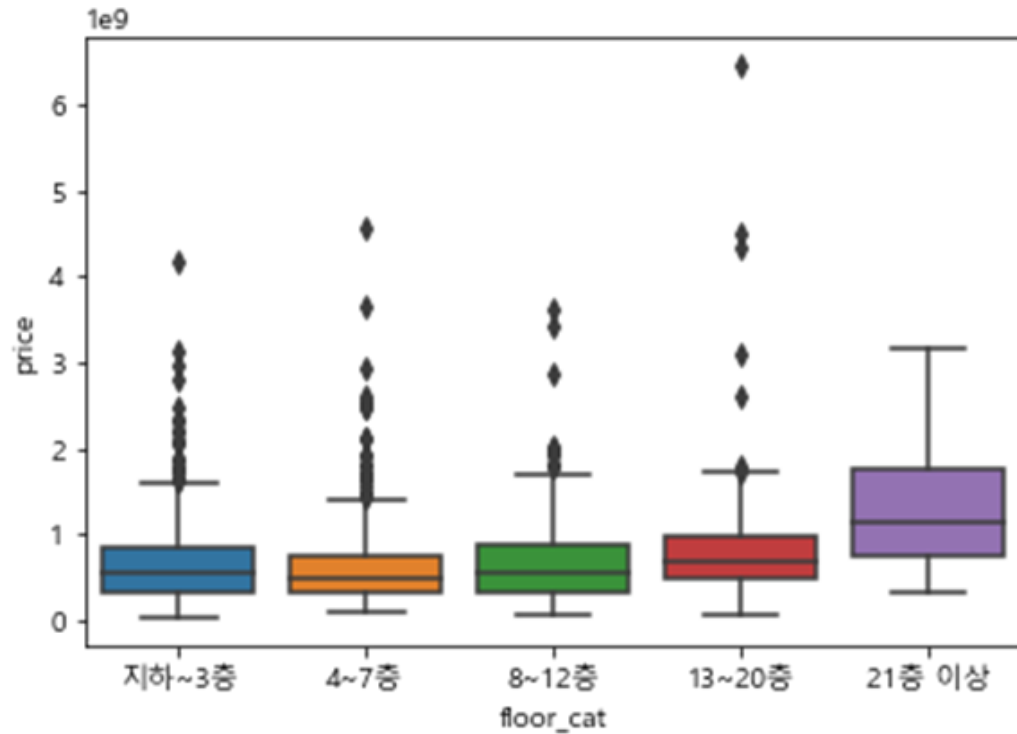
- 총 1095개 데이터
  - 진행 중: 28개
  - 매각 완료: 1067개



# 4. 탐색적 데이터 분석 결과

## 1) 경매 데이터 EDA 결과

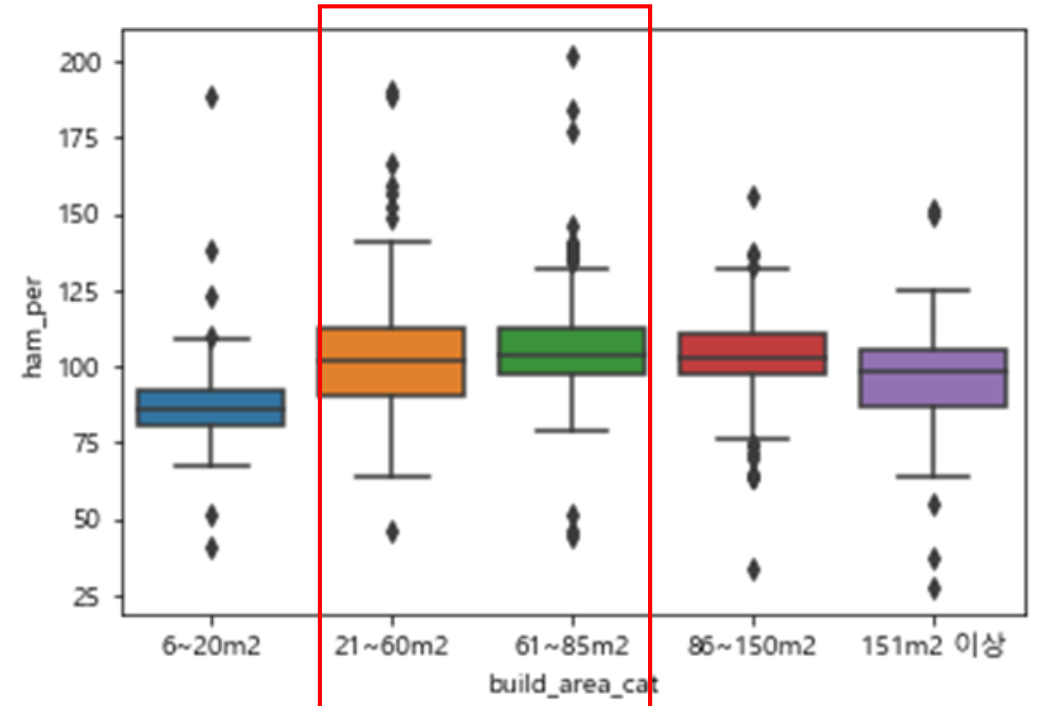
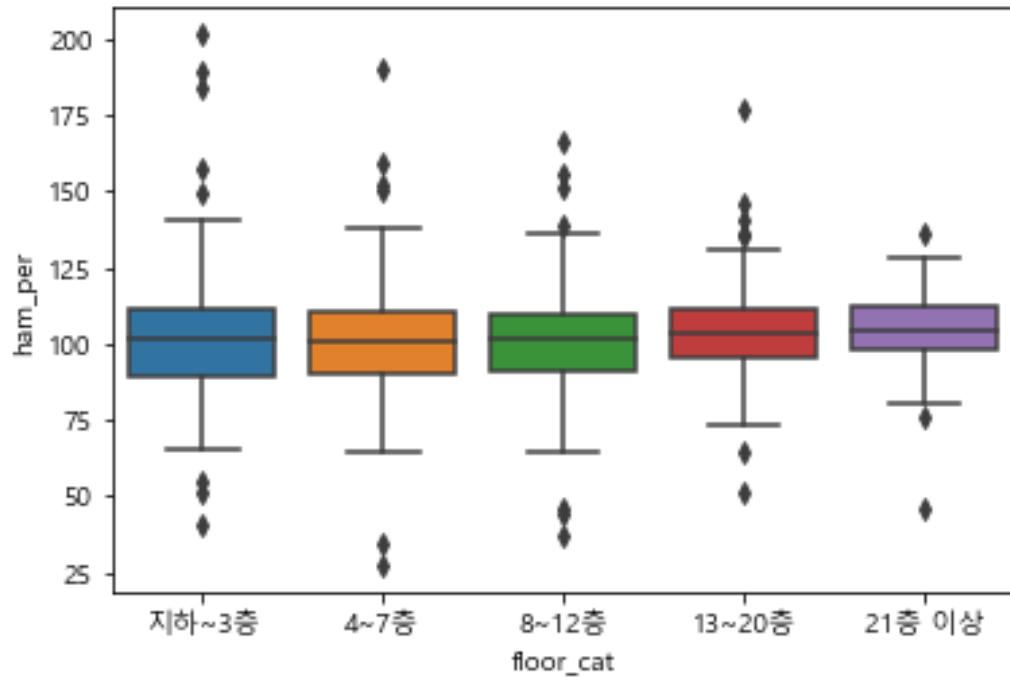
- 층수 및 면적을 구간화하여 분석 (감정가)



# 4. 탐색적 데이터 분석 결과

## 1) 경매 데이터 EDA 결과

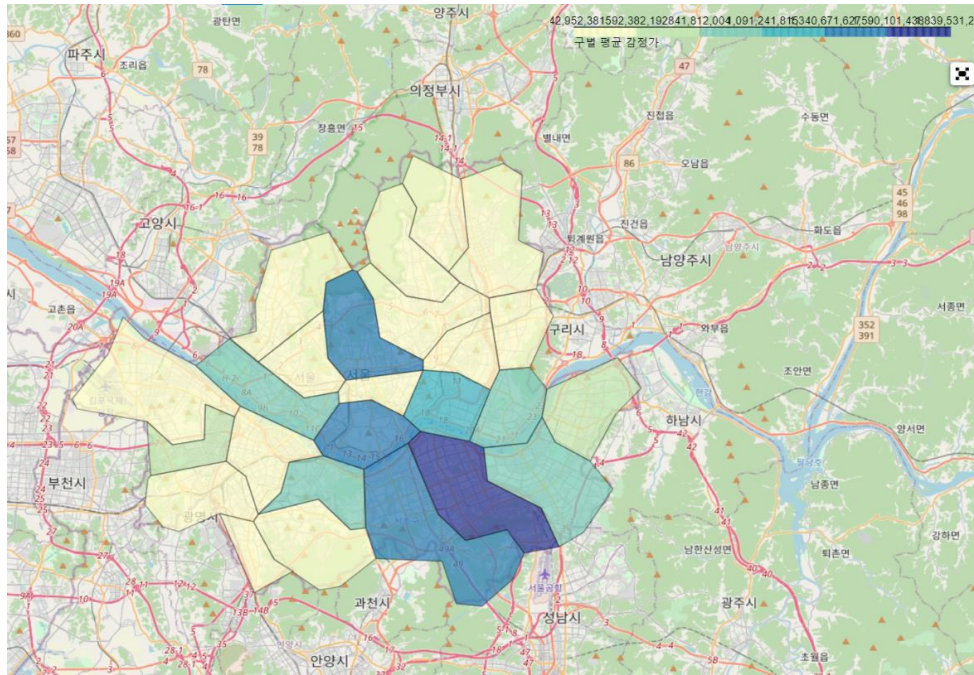
- 층수 및 면적을 구간화하여 분석 (감정가 대비 낙찰가 비율)



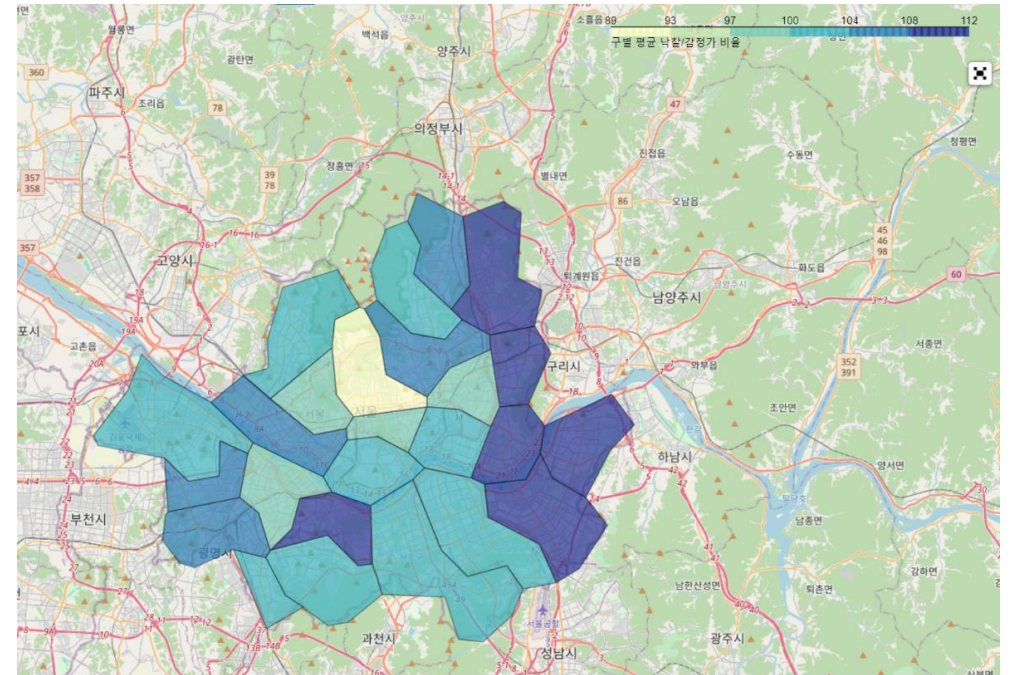
# 4. 탐색적 데이터 분석 결과

## 1) 경매 데이터 EDA 결과

- 구별 분석



감정가



감정가 대비 낙찰가 비율

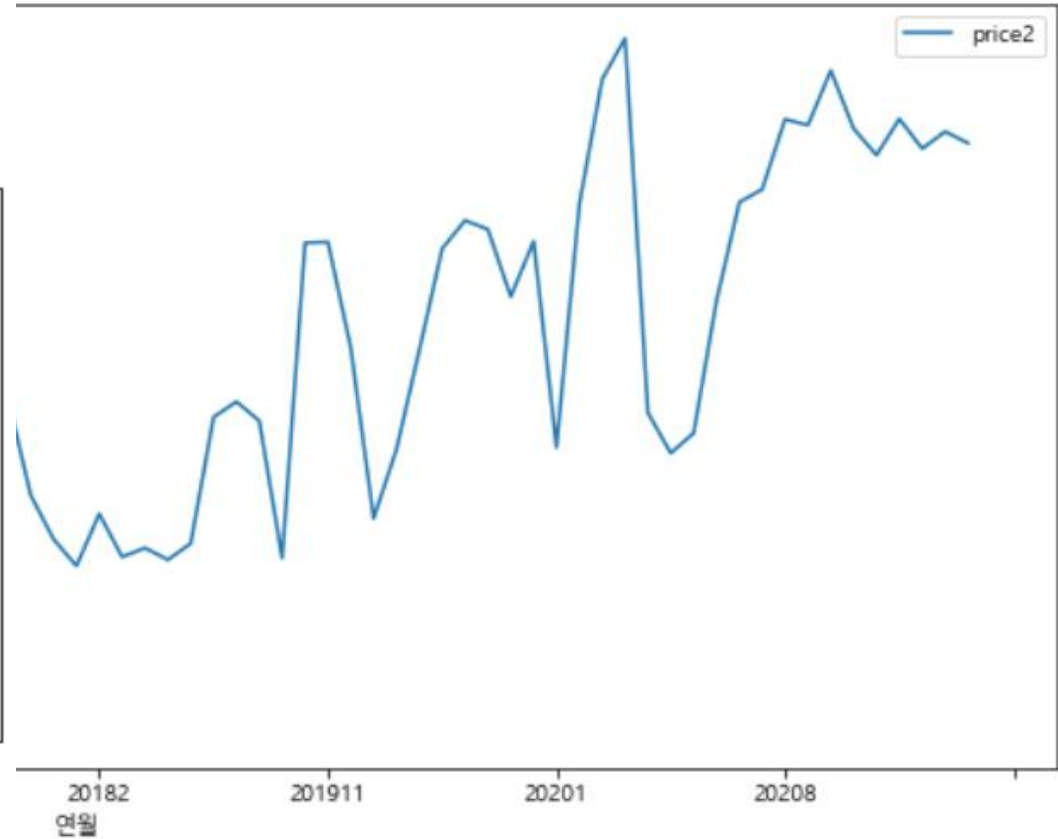
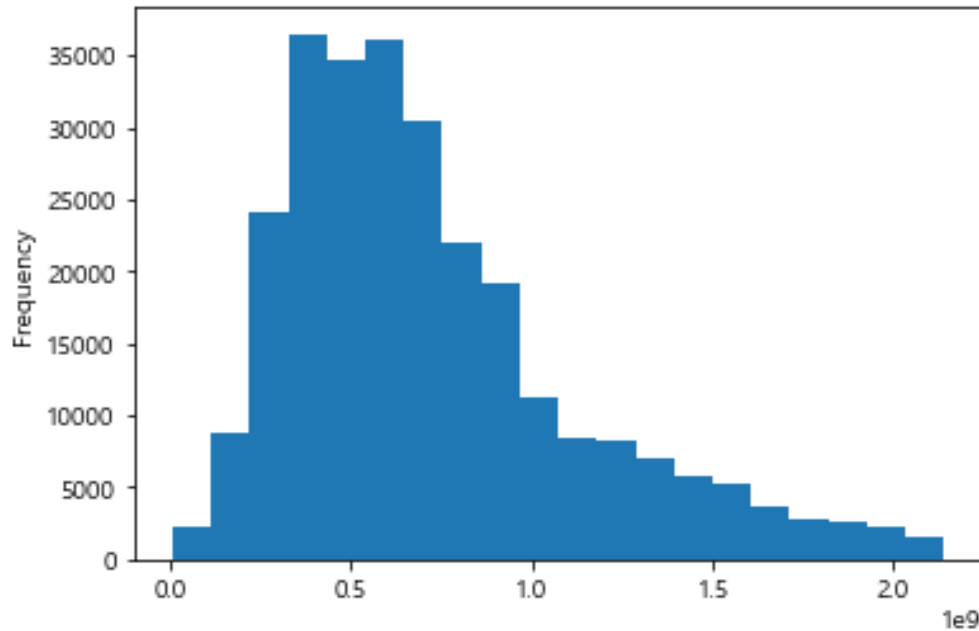
# 4. 탐색적 데이터 분석 결과

## 1) 실거래가 데이터 EDA 결과

- 총 616623개 (2018 ~ 2021년간 데이터: 총 281003개)

거래액 총액: 197.0 조원  
대한민국 1년 예산 : 512조

<AxesSubplot:ylabel='Frequency'>



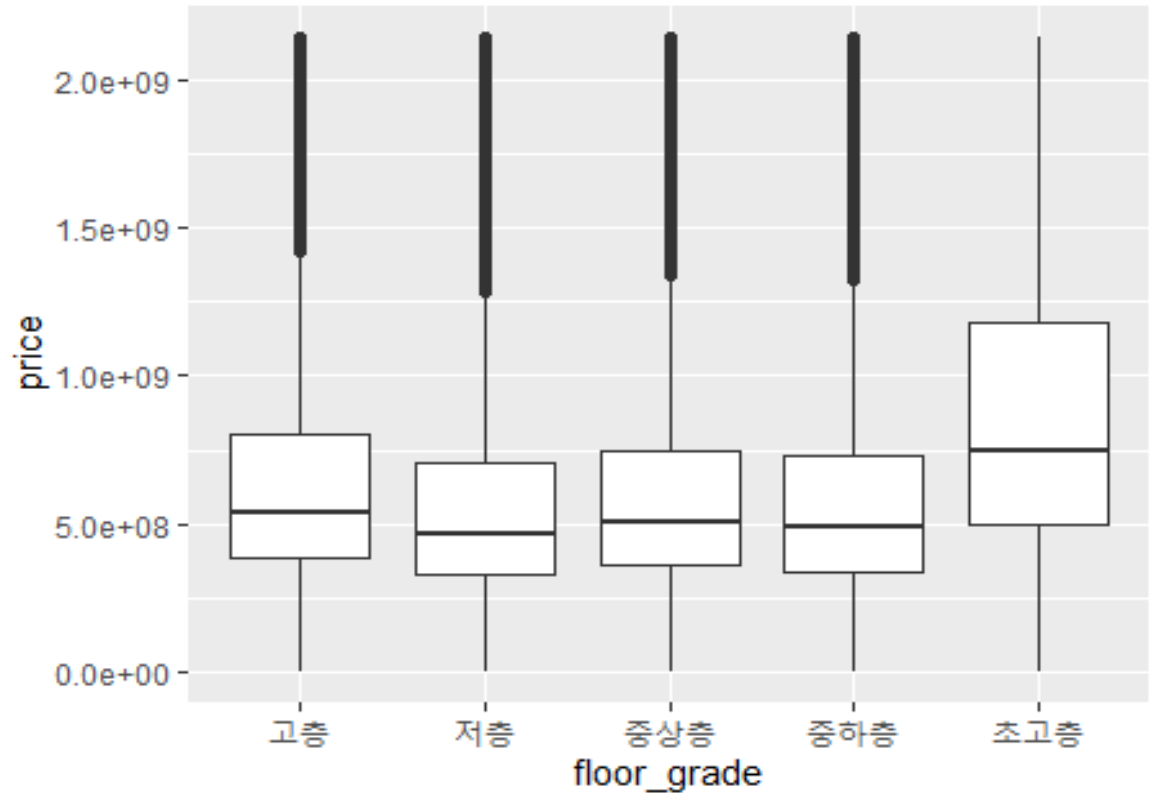
# 4. 탐색적 데이터 분석 결과

## 1) 실거래가 데이터 EDA 결과

- 층수 별

```
case_when(floor <= 3 ~ "저층",  
          floor <= 7 ~ "중하층",  
          floor <= 12 ~ "중상층",  
          floor <= 20 ~ "고층",  
          floor > 20 ~ "초고층"))
```

floor_grade	n	price
고층	128185	638064379.
저층	119811	572571171.
중상층	167584	607247079.
중하층	164416	584120798.
초고층	26006	868038306.





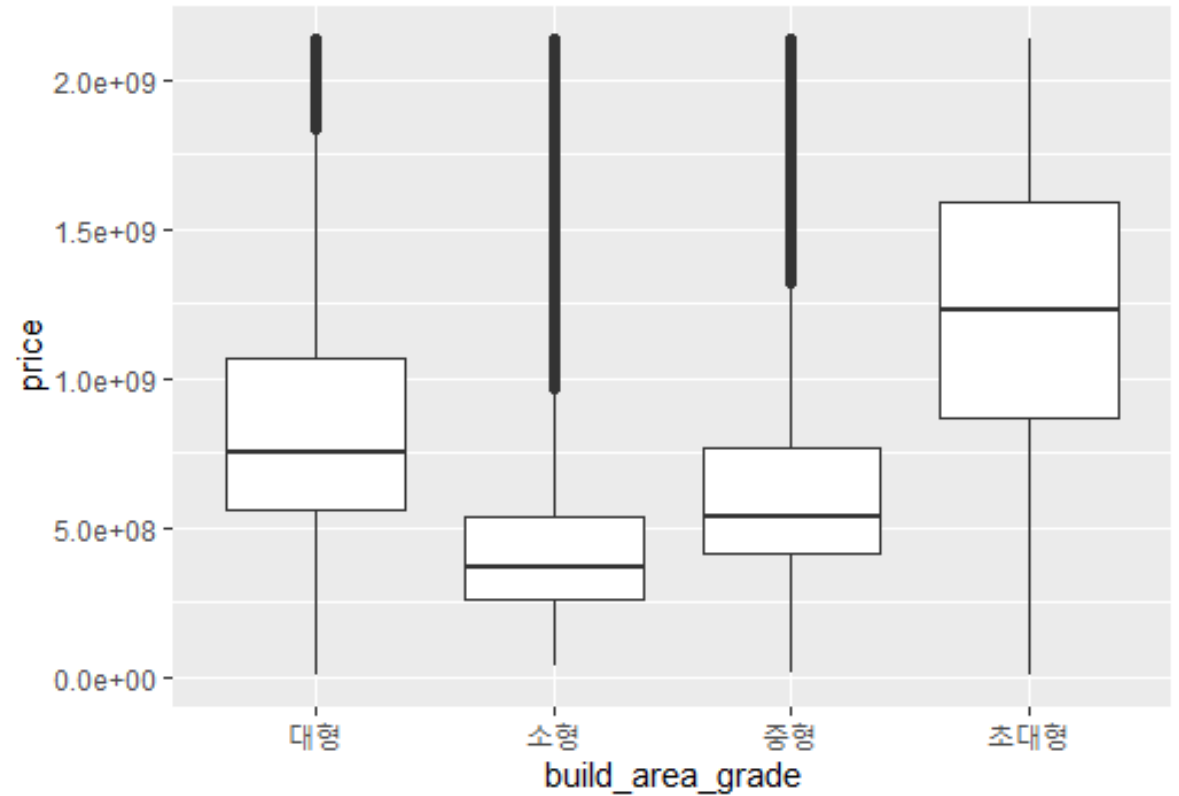
# 4. 탐색적 데이터 분석 결과

## 1) 실거래가 데이터 EDA 결과

- 면적 구간별

```
case_when(build_area<=60 ~ "소형",  
          build_area<=85 ~ "중형",  
          build_area<=135 ~ "대형",  
          build_area>135 ~ "초대형"))
```

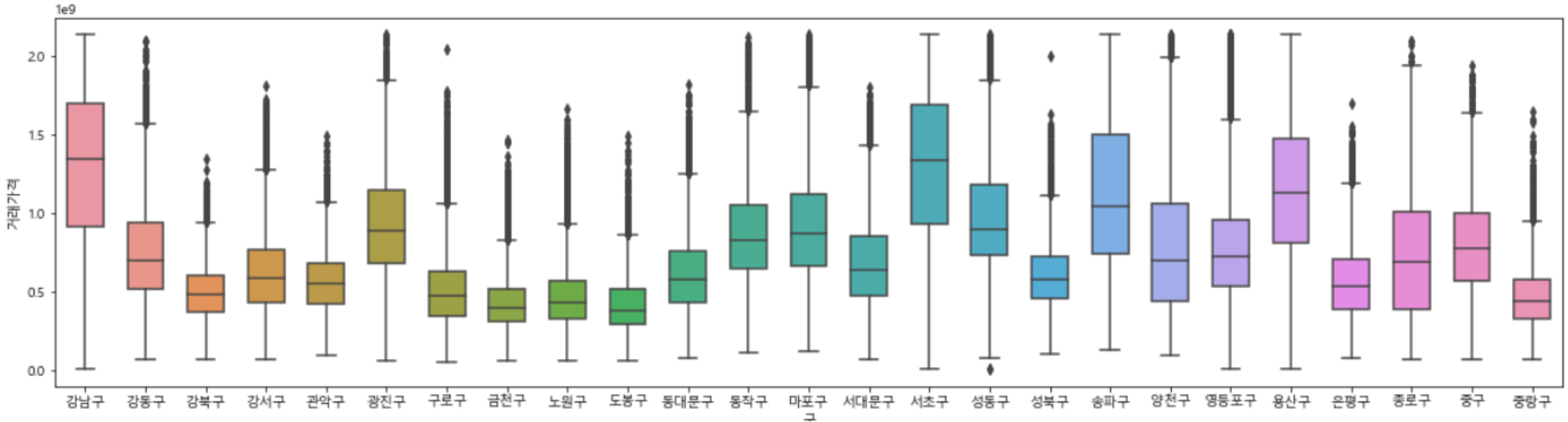
build_area_grade	n	price
<chr>	<int>	<dbl>
대형	87634	854349009.
소형	234621	437815626.
중형	262996	637423763.
초대형	20751	1230680437.



# 4. 탐색적 데이터 분석 결과

## 1) 실거래가 데이터 EDA 결과

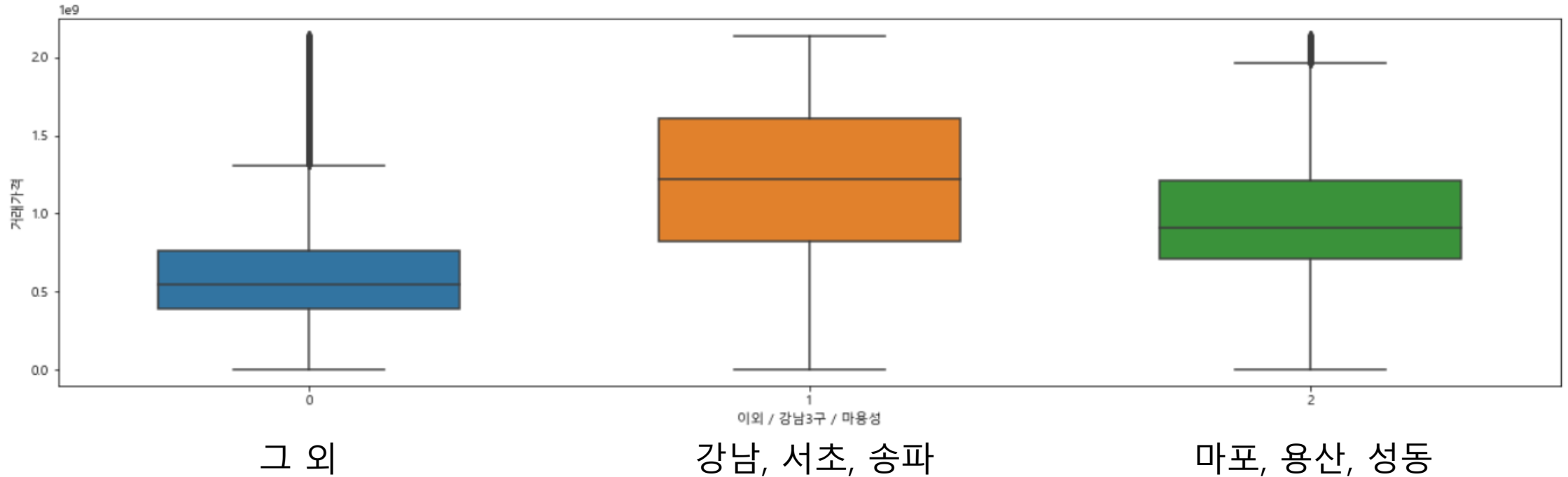
- 구별 분석



# 4. 탐색적 데이터 분석 결과

## 1) 실거래가 데이터 EDA 결과

- 구별 분석



# 5. 프로그램 알고리즘

- 경매 낙찰가격 예측 모형 및 알고리즘

- 낙찰가격 예측모형 매개변수

$$\begin{aligned}
 Hammer_{i,t} = & a_i + \beta_1 Gu_{i,t} + \beta_2 Dong_{i,t} + \beta_3 Lowest_{i,t} + \beta_4 Price_{i,t} + \beta_5 Low\_Per_{i,t} + \beta_6 Floor_{i,t} \\
 & + \beta_7 Build\_Area_{i,t} + \beta_8 Year\_Month_{i,t} + \beta_9 Case\_Year_{i,t} + \beta_{10} Comment_{i,t} + \varepsilon_{i,t}
 \end{aligned}$$

<i>Variables</i>	<i>Definition</i>
<i>Hammer</i>	낙찰가
<i>Gu, Dong</i>	행정구역(구, 동)
<i>Lowest</i>	최저가
<i>Price</i>	감정가
<i>Low_Per</i>	감정가대비 최저가(최저가/감정가)
<i>Floor</i>	층 (일정 구간에 따라 나뉜 변수)
<i>Build_Area</i>	면적(일정 구간에 따라 나뉜 변수)
<i>Year_month</i>	낙찰년월
<i>Case_Year</i>	경매등록년
<i>Comment</i>	매각조건 여부(매각조건이 존재하면 1, 없으면 0)

# 5. 프로그램 알고리즘

- 경매 낙찰가격 예측 모형 및 알고리즘
  - 낙찰가 예측 알고리즘 파라미터 설정

<i>Parameters</i>	<i>Settings</i>
알고리즘	ANN(인공신경망)
Layer 수	1개
데이터 입출력 개수	X: (1067, 10), y: (1067, 1)
활성화 함수	Linear
비용 함수	MSE(Mean Squared Error)
최적화 알고리즘	GradientDescent
학습률	0.0004
Epoch 수	500000번
Batch 크기	1067개

# 5. 프로그램 알고리즘

- 경매 낙찰가격 예측 모형 및 알고리즘

- 낙찰가 예측 알고리즘 파라미터 설정

```
x = tf.placeholder(tf.float32, shape=[None, 10], name='x')
y = tf.placeholder(tf.float32, shape=[None, 1], name='y')

w = tf.Variable(tf.random_normal([10, 1]), dtype=tf.float32, name='w')
b = tf.Variable(tf.random_normal([1]), dtype=tf.float32, name='b')

hypothesis = tf.matmul(x, w) + b

cost = tf.reduce_mean(tf.square(hypothesis - y), name='cost')
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.0004, name='optimizer')
train = optimizer.minimize(cost)

sess = tf.Session()
sess.run(tf.global_variables_initializer())

#coord = tf.train.Coordinator()
#threads = tf.train.start_queue_runners(sess=sess, coord=coord)

feed_train = {x:x_train, y:y_train}
feed_test = {x:x_test, y:y_test}
for step in range(500001):
    c2, h2, _ = sess.run([cost, hypothesis, train],
                        feed_dict=feed_train)

    if step%1000 == 0:
        print(step, 'cost:', c2, '\npredict:', h2)
```

→ 완성된 모델 저장 후 프로그램에 사용

# 5. 프로그램 알고리즘

- 실거래가 예측 모형 및 알고리즘
  - 실거래가 예측모형 매개변수

$$Price_{i,t} = a_i + \beta_1 Gu_{i,t} + \beta_2 Dong_{i,t} + \beta_3 Floor_{i,t} + \beta_4 Build\_Area_{i,t} + \beta_5 Year\_Month_{i,t} + \varepsilon_{i,t}$$

<i>Variables</i>	<i>Definition</i>
<i>Price</i>	실거래가
<i>Gu, Dong</i>	행정구역(구, 동)
<i>Floor</i>	층
<i>Build_Area</i>	면적
<i>Year_month</i>	계약년월



# 5. 프로그램 알고리즘

- 실거래가 예측 모형 및 알고리즘
  - 실거래가 예측 알고리즘 파라미터 설정

<i>Parameters</i>	<i>Settings</i>
알고리즘	CNN(합성곱신경망)
Layer 수	10개
데이터 입출력 개수	X: (598045, 5), y: (598045, 1)
활성화 함수	Linear, ReLU
비용 함수	MAE(Mean Absolute Error)
최적화 알고리즘	RMSprop
학습률	0.001
Epoch 수	1000번
Batch 크기	5000개



## 5. 프로그램 알고리즘

- 실거래가 예측 모형 및 알고리즘
  - 실거래가 예측 알고리즘 파라미터 설정

```
model = Sequential()  
model.add(Dense(64, input_shape=(5,), activation="linear"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(64, activation="linear"))  
model.add(Dense(1))  
  
model.compile(loss="mae", optimizer=RMSprop(lr=0.001), metrics=["mae", "mse"])  
result = model.fit(X_train, y_train, epochs=1000, batch_size=5000)
```

→ 완성된 모델 저장 후 프로그램에 사용

상세 코드: [http://github.com/JinsolPark/final\\_project](http://github.com/JinsolPark/final_project)

# 6. 예측 모형 훈련결과

- 예측 모형 훈련 및 테스트 결과

	낙찰가 예측모형 결과	실거래가 예측모형 결과
훈련 결과(cost)	0.0117	0.0701
테스트 결과(cost)	0.0071	0.0816
<p>테스트 데이터 예측값 vs 실제값 산포도</p>		

## 6. 예측 모형 훈련결과

- 예측가격과 실제 경매 낙찰 비교

1 낙찰가: nan	예상가: 2899217367	
2 낙찰가: nan	예상가: 485910654	
3 낙찰가: nan	예상가: -4601985	
4 낙찰가: nan	예상가: -1253783	
5 낙찰가: nan	예상가: 580216944	
6 낙찰가: nan	예상가: 508834362	
7 낙찰가: nan	예상가: 928044855	
8 낙찰가: nan	예상가: 1102539420	
9 낙찰가: nan	예상가: 681609630	
10 낙찰가: nan	예상가: 638148069	
11 낙찰가: nan	예상가: 498262107	
12 낙찰가: nan	예상가: 3162310838	실제 낙찰가
13 낙찰가: nan	예상가: 400481820	
14 낙찰가: nan	예상가: 513549149	
15 낙찰가: nan	예상가: 590298771	450004500
16 낙찰가: nan	예상가: 397912800	
17 낙찰가: nan	예상가: 276026546	120000818
18 낙찰가: nan	예상가: 147453740	
19 낙찰가: nan	예상가: 161539033	124880000
20 낙찰가: nan	예상가: 420953929	
21 낙찰가: nan	예상가: 18045872	418888880
22 낙찰가: nan	예상가: 306307554	
23 낙찰가: nan	예상가: 85470989	
24 낙찰가: nan	예상가: 238371416	
25 낙찰가: nan	예상가: 1232440352	
26 낙찰가: nan	예상가: 204840019	
27 낙찰가: nan	예상가: 270100593	
28 낙찰가: nan	예상가: 113117933	

# 7. 제한점 및 추후 발전방향

- 제한점

- 경매 데이터와 실거래가 데이터의 근본적인 데이터 차이에서 오는 다수의 결측치 발생
- 가격예측에 경제상황을 나타내는 변수들은 반영하지 않음
- 이상치 컨트롤을 하지 않음

- 추후 발전방향

- 제한점 개선
- 이미지 분류 알고리즘을 활용한 아파트 분류
- 경매 유찰 횟수 예측을 통한 더 정확한 낙찰가 예측
- 하둡, 스파크를 활용한 데이터베이스 구축 및 분석 자동화
- 앱으로 만들어 보급가능 → 광고, 구독, 대출 연계 수입 발생 가능성

아이디어 불펌 금지 Copyright © 송창호,이유성,박진솔

감사합니다.

투자문의: 이유\*

