

Upstage AI Lab

Regression 대회
: [1조] 서울 아파트 매매가 예측 모델 만들기

24.04.17

www.fastcampus.co.kr

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

목차

01. 팀원 소개 및 작업 방식 안내
02. feature engineering
03. 모델 학습
04. 결과 및 인사이트 공유
05. 프로젝트 회고

01

팀원 소개

인간을 잘 이해하는 AI개발자
저는 정인웅 입니다.



Interested in

서비스 개발(앱, 웹)
생성형 ai, EDA

Introduction

- 심리학(임상 및 상담) 석사,
- 소프트웨어 공학 학사

건강하고 성장하는 삶을 지향합니다.

Role

중요 피처 선택, 데이터 모델 파라미터 최적화,
앙상블
팀장 역할, 운동의 중요성 설파

In Upstage AI Lab

서비스를 개발할 수 있는 기본기 익히길
어떤 진로를 결정하든 기본기가 탄탄해지길

인간을 잘 이해하는 AI개발자
저는 정인웅(조정석)입니다.



Interested in

서비스 개발(앱, 웹)
생성형 ai, EDA

Introduction

- 심리학(임상 및 상담) 석사,
- 소프트웨어 공학 학사

건강하고 성장하는 삶을 지향합니다.

Role

중요 피처 선택, 데이터 모델 파라미터 최적화,
앙상블
팀장 역할, 운동의 중요성 설파

In Upstage AI Lab

서비스를 개발할 수 있는 기본기 익히길
어떤 진로를 결정하든 기본기가 탄탄해지길

일단 뭐라도 해보자!
저는 진수훈 입니다.



Interested in

- CV

Introduction

- 컴퓨터 공학 학사

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

In Upstage AI Lab

- 완성도 있는 프로젝트 만들기!
- 테크 블로그 꾸준히 운영하기!

일단 뭐라도 해보자!
저는 진수훈(임시완) 입니다.



Interested in

- CV

Introduction

- 컴퓨터 공학 학사

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

In Upstage AI Lab

- 완성도 있는 프로젝트 만들기!
- 테크 블로그 꾸준히 운영하기!

80%의 힘으로 10배로 노력하자!
저는 이범희 입니다.



Interested in

- NLP를 활용한 Knowledge Tracking

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

Introduction

- 전공: 노어노문학 & 국어교육학

In Upstage AI Lab

- AI 관련 기본기 숙달
- AI 분야에서 함께 발전해갈 동료 확보

80%의 힘으로 10배로 노력하자!
저는 이범희 (손석구)입니다.



Interested in

- NLP를 활용한 Knowledge Tracking

Introduction

- 전공: 노어노문학 & 국어교육학

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

In Upstage AI Lab

- AI 관련 기본기 숙달
- AI 분야에서 함께 발전해갈 동료 확보

일단 뭐라도 해보자!
저는 안수민 입니다.



Interested in

- 갖기 위해 공부 중

Introduction

- 디지털미디어

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

In Upstage AI Lab

해당 분야에서 쪽 공부할 수 있을지 가능성 확인

일단 뭐라도 해보자!
저는 안수민(나무늘보) 입니다.



Interested in

- 갖기 위해 공부 중

Introduction

- 디지털미디어

Role

- 팀원들과 함께 모든 프로젝트에 성실히 참여.

In Upstage AI Lab

해당 분야에서 꼭 공부할 수 있을지 가능성 확인

Upstage AI Lab

프로젝트 진행 방법

: Regression 대회

프로젝트 진행 장소	오프라인 강의장
스크럼 진행 횟수 및 일정	모여서 진행하며 그때그때 상황 공유
프로젝트 진행 방법	<div>각자 관심사에 따라 자연스럽게 역할을 나눠 진행. 정인웅: 주요 feature select, 모델 학습, 파라미터 튜닝, 앙상블 이범희: 전반적인 과정 조율, 아이디어 제시, 데이터 split, feature 생성 안수민: problem_shooting, 코드 효율성 증진, 역세권 feature 생성 진수훈: 주요 feature select, 모델 학습, 데이터 split, 과적합 방지, 각종 아이디어 구현 및 수정</div>

02

Feature Engineering

Final Feature selection

: 모델 학습을 위한 최종 데이터셋입니다. (총 26개의 feature)

1. 본번 (object)
2. 부번 (object)
3. 아파트명 (object)
4. 전용면적 (float64)
5. 계약년월 (int64)
6. 계약일 (object)
7. 층 (object)
8. 건축년도 (int64)
9. 도로명 (object)
10. 거래유형 (object)
11. 중개사소재지 (object)
12. 구 (object)
13. 동 (object)
14. 계약월 (object)
15. 이전_1번째_거래가격 ~ 이전_10번째_거래가격 (float64)
16. 역세권 (object)
17. 구_역세권_랭크 (object)

Feature selection

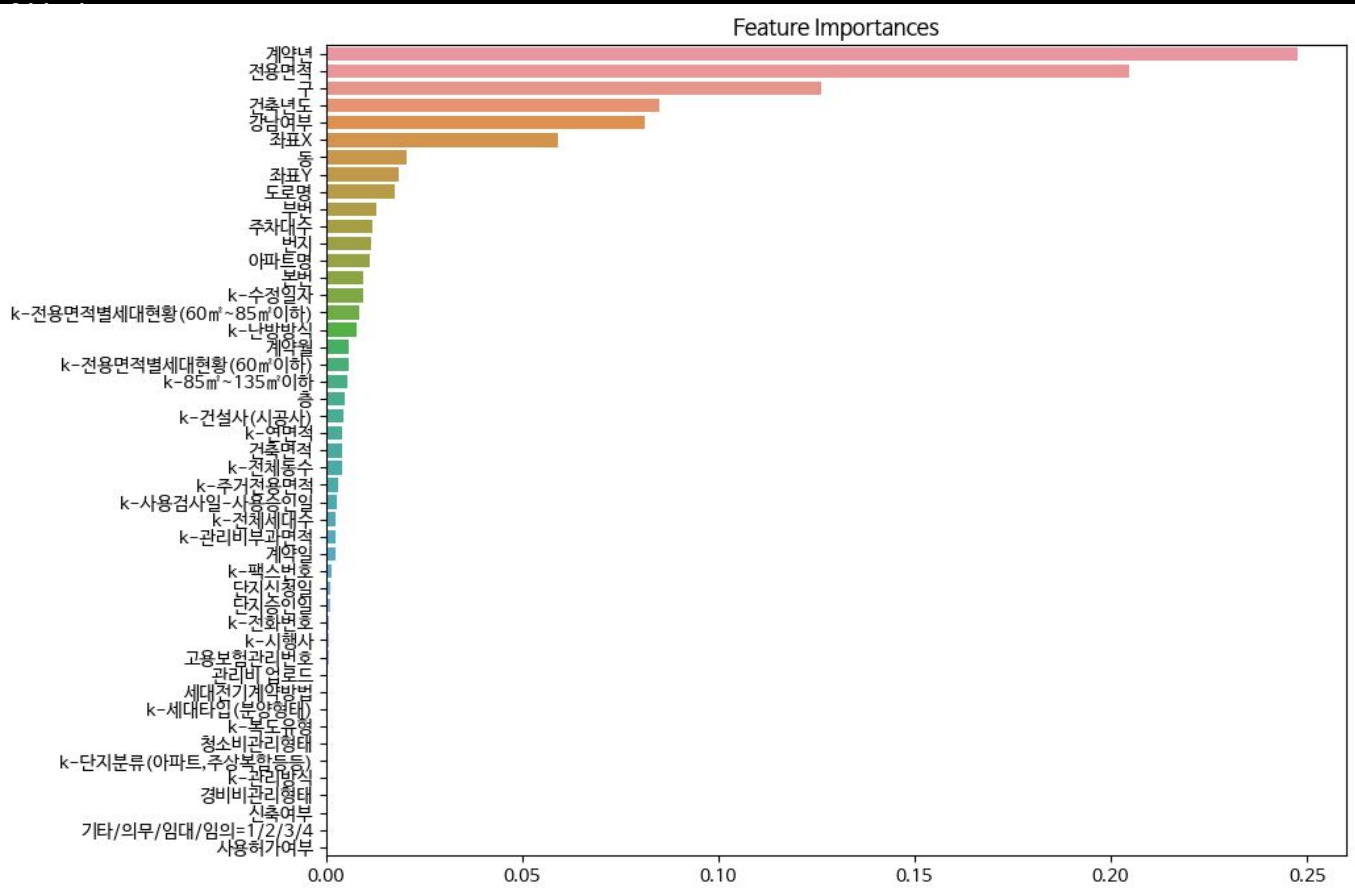
: train.csv, test.csv

- 매물 위치 관련 칼럼(시군구, 번지, 본번, 부번, 아파트명, 층, 도로명, 좌표X, 좌표Y, 중개사소재지 등)
- 시간 관련 칼럼(계약년월, 계약일, 건축년도, 해제사유발생일, 등기신청일자, 단지승인일, 단지신청일)
- 면적 관련 칼럼(전용면적, 건축면적 등)
- K-칼럼(약 20개 정도)
- 기타 관련 정보(거래유형, 고용보험관리번호, 경비비관리형태, 세대전기계약방법 등)

Feature selection

: train.csv, test.csv

- 베이스라인 코드를 따라 쪽 진행 후, feature importance 확인



Feature selection

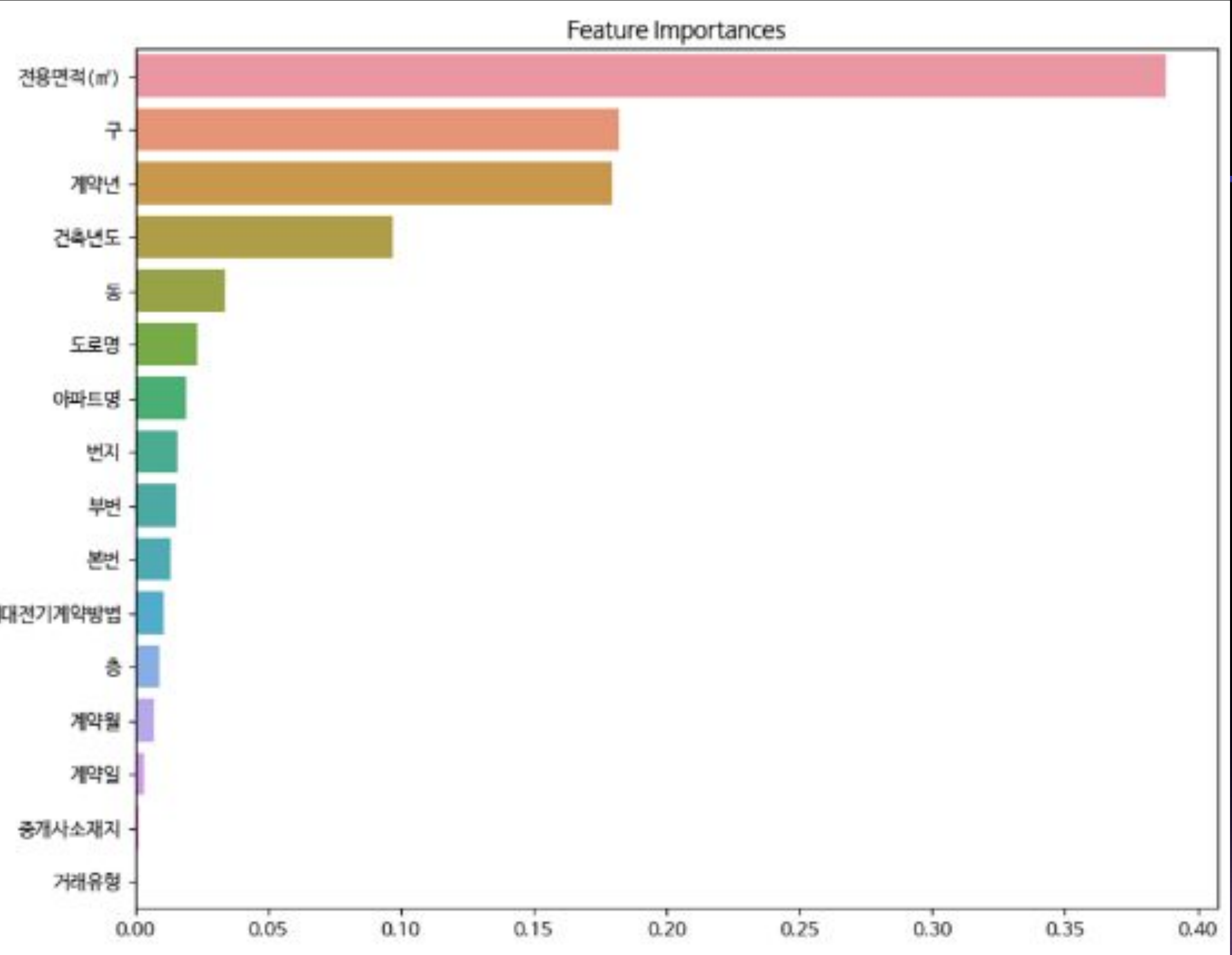
: train.csv, test.csv

- 이번까지 남기고 나머지는 삭제
 - Feature importance 순위 상 이번 이후로는 의미가 없다고 판단
 - But, ‘총’, ‘중개사소재지’, ‘거래유형’ 등은 직관에 따라 유의미하다고 판단하여 남겨두었음. (실제 validation rmse도 더 좋았음.)
 - K-관련 칼럼은 전부 삭제 (결측치가 너무 많음. 80만 개 이상의 결측치를 학습에 유효할 정도로 채우는 것이 불가능하다고 판단함.)
- > 게다가 K-관련 칼럼은 자료 제출자에 따라 같은 매물임에도 다른 정보가 들어가 있는 경우가 허다하고, 자료 조사 참여율도 저조한 지표로서 예측에 부정적인 영향을 미침.

Feature score

: train.csv, test.csv

- 이번까지



Feature selection

: train.csv, test.csv

- 주요 feature만 남기고, caboost (optuna 적용) 모델 학습시의 public 점수:

16818점



catboost_imp...model



16818.2369
15722.3523

2024.07.16 18:56

Feature creation

: 이전_n번째_거래가격

- 다음 target값을 예측하는 데에 가장 강력한 변수는 이전 target값이다.
 - 아파트 매매가는 일종의 흐름을 지니기 때문에, 흐름 예측을 위해 이전 target값을 모델이 인지하는 것은 굉장히 중요하다.
 - 이러한 feature를 lag feature라고 하며, 보통 시계열 데이터에서 필수적으로 활용된다.
- > 이전_n번째_거래가격을 새로운 feature로 추가! (동 번지 기준으로 필터링 후, 계약년월이 max인 행의 target값 배정)
(가장 성능 개선이 많이 된 feature임)

Feature creation


: 이전_n번째_거래가격

- 다음 target값을 예측하는 데에 가장 강력한 변수는 이전

catboost_imp...model		16818.2369 15722.3523	2024.07.16 18:56
----------------------	-------------------------------------------------------------------------------------	--------------------------	------------------

예측을 위해 이전 target값을 모델이 인지하는 것은 굉장히 중요하다.



catboost(최근거래가격 적용)		15278.5829 12747.9266	2024.07.17 01:30
---------------------	---------------------------------------------------------------------------------------	--------------------------	------------------

-> 이전_n번째_거래가격을 새로운 feature로 추가!
(가장 성능 개선이 많이 된 feature임)

Feature creation

: subway_feature.csv

●역세권 feature 생성을 위해 사용

1. 위도, 경도 값을 train.csv에 있는 좌표 X, Y와 비교하여 하버사인 계산 -> 거리 지표를 넣어주는 방법
(연속형 데이터)
2. 위도, 경도 값을 train.csv에 있는 좌표 X, Y와 비교하여 하버사인 계산 -> 500m 내에 있는 매물은 역세권, 아닌 매물은 비역세권으로 칼럼 생성
(범주형 데이터)

Feature creation

: subway_feature.csv

●역세권 feature 생성을 위해 사용

1. 위도, 경도 값을 train.csv에 있는 좌표 X, Y와 비교하여 하버사인 계산 -> 거리 지표를 넣어주는 방법

(연속형 데이터)

2. 위도, 경도 값을 train.csv에 있는 좌표 X, Y와 비교하여 하버사인 계산 -> 500m 내에 있는 매물은 역세권, 아닌 매물은 역세권이 아닌 것으로 칼럼 생성

(범주형 데이터)

Feature creation

: subway_feature.csv

●역세권 feature 생성을 위해 사용

집을 구매할 때, 대략 집에서 500m 내에 있을 때 역세권이라 판단할 것이라 예상.

또한 100m나 500m나 인간이 느끼기엔 같은 역세권처럼 느낄 것이라 생각해 연속형 데이터가 아닌 범주형 데이터로 생성하기로 결정.

Feature creation

: subway_feature.csv

●역세권 feature 생성 과정

1. Naver Geocoding API를 사용해 train 데이터에서 NAN인 X,Y좌표 보완
2. Naver Reverse Geocoding API를 사용해 subway_feature에 행정구 추가
3. 매물의 행정구에 속한 역들과 거리 계산 후, 500m 이내에 해당하는 역이 하나라도 있으면 역세권으로 판단, '역세권' 피쳐 추가(0 또는 1)
4. 역세권 여부보다도 지역구가 target에 더 큰 영향을 미치므로, 행정구별 target값 평균으로 랭크를 산정, '구_역세권_랭크' 피쳐 추가(1위~50위)
→ 구별로 집값 차이가 천차만별이다보니, 역세권 여부도 구별로 구분하며 학습하도록 유도하기 위해 생성

Upstage AI Lab

Feature creation

: subway_feature.csv

	구	역세권	구_역세권_랭크
0	강남구	강남구 비역세권	3.0
1	강남구	강남구 역세권	5.0
2	강동구	강동구 비역세권	28.0
3	강동구	강동구 역세권	24.0
4	강북구	강북구 비역세권	42.0
5	강북구	강북구 역세권	41.0
6	강서구	강서구 비역세권	37.0
7	강서구	강서구 역세권	25.0
8	관악구	관악구 비역세권	40.0
9	관악구	관악구 역세권	44.0
10	광진구	광진구 비역세권	17.0
11	광진구	광진구 역세권	10.0
12	구로구	구로구 비역세권	38.0
13	구로구	구로구 역세권	43.0
14	금천구	금천구 비역세권	48.0
15	금천구	금천구 역세권	35.0
16	노원구	노원구 비역세권	45.0
17	노원구	노원구 역세권	49.0
18	도봉구	도봉구 비역세권	47.0
19	도봉구	도봉구 역세권	50.0

Feature Engineering

: 다중공선성 제거

	Variable	VIF
0	const	0.000000
1	번지	47.077803
2	본번	47.016145
3	부번	1.028451
4	아파트명	1.083086
5	전용면적 (m²)	1.374914
6	계약년월	16164.610422
7	계약일	1.000065
8	층	1.069114
9	건축년도	1.134615
10	도로명	1.099725
11	거래유형	2.962108
12	중개사소재지	2.929444
13	구	1961.536895
14	동	110.517604
15	계약년	16168.306665
16	계약월	1.010368
17	주소	110.098522
18	이전_1번째_거래가격	4.867922
19	이전_2번째_거래가격	4.642271
20	역세권	1963.157883
21	구_역세권_랭크	1.749345

	Variable	VIF
0	const	283594.627858
1	본번	1.009649
2	부번	1.027398
3	아파트명	1.078589
4	전용면적 (m²)	1.373602
5	계약년월	1.139801
6	계약일	1.000041
7	층	1.068298
8	건축년도	1.125038
9	도로명	1.090673
10	거래유형	2.961152
11	중개사소재지	2.927741
12	구	1.106938
13	동	1.156522
14	계약월	1.001745
15	이전_1번째_거래가격	4.850852
16	이전_2번째_거래가격	4.636339
17	역세권	1.045042
18	구_역세권_랭크	1.745636

VIF(팽창인수)

-번지는 본번, 부번과 계약년월은 계약년, 계약월과 주소는 동과 다중 공선성이 높음.

-rmse를 기준으로 번지, 주소, 계약년을 제거하였고, 역세권에는 구가 포함되어 구가 포함되지 않도록 수정.

-그결과 모두 VIF 10이하로 전처리가 깔끔하게 진행됨.

03

모델 학습

베이스라인 그대로 돌렸을 때 모델별 성능

1. *RandomForest*: 2위(public[22,800] & validation[6,900])
2. *XGboost*: 4위(public[48,600] & validation[7,200])
3. *Lightgbm*: 3위(public[47,000] & validation[12,000])
4. *Catboost*: 1위(public[22,000] & validation[6,800])

Completion (상기 케이스)

<input type="checkbox"/>	랜덤포레스트(베이스라인)	수훈	22853.7836 21418.9031	2024.07.12 19:02
<input type="checkbox"/>	basecode1_2019	에단	48609.1720 35100.6685	2024.07.12 16:14
<input type="checkbox"/>	lightgbm_bas..._2019	에단	47163.9224 34264.1324	2024.07.12 16:00
<input type="checkbox"/>	아파트, 주상복합.model	에단	21936.5744 20101.2728	2024.07.15 18:09

커심.

베이스라인 그대로 돌렸을 때 모델별 성능

1. *RandomForest*: 2위(public[22,800] & validation[6,900])
2. *XGboost*: 4위(public[48,600] & validation[7,200])
3. *Lightgbm*: 3위(public[47,000] & validation[12,000])
4. *Catboost*: 1위(public[22,000] & validation[6,800])

시도했던 데이터 split & validation 방법

1. *Hold out split (train 8 : test 2, random_state = 2023)*
2. *TimeSeriesSplit (split_num = 5)*
3. *K-Fold validation (Fold_num = 5)*

시도했던 데이터 split & validation 방법

1. *Hold out split (train 8 : test 2, random_state = 2023)*
2. *TimeSeriesSplit (split_num = 5)*
3. *K-Fold validation (Fold_num = 5)*

모델 학습

: 데이터 split & validation

1.
2.
3.

<div>Hold-out-split catboost(최근거...지 적용)</div>	<div>15341.9782 12639.7484</div>	<div>2024.07.17 12:23</div>
<div>랜덤포레스트(10일이전 거래일, 학습,테스트(시계열 분할)) Time_series_split 랜덤포레스트(10일이전... 분할))</div>	<div>117277.3703 112458.2846</div>	<div>2024.07.17 16:39</div>
<div>K-Fold Validation catboost(k-f...째 가격)</div>	<div>16922.7274 11000.9939</div>	<div>2024.07.17 18:21</div>

Optuna 하이퍼 파라미터 튜닝

*optuna*를 통해 *hyperparameter*를 검증하고, *best parameter*를 검출함

```
# optuna를 사용하여 하이퍼 파라미터 튜닝
study = optuna.create_study(direction='minimize', pruner=optuna.pruners.MedianPruner())
study.optimize(objective, n_trials=100, n_jobs=4) # 병렬 처리 사용

# 최적 파라미터로 모델 재학습
best_params = study.best_params
best_model = xgb.train(best_params, dtrain_full, num_boost_round=1000, evals=[(dvalid, 'eval')], early_stopping_rounds=50, verbose_eval=False)
```


모델 학습

: 하이퍼파라미터 튜닝 (optuna)

```
# CatBoost HyperParameter
params = {
    'iterations': 2000,
    'learning_rate': 0.2619829692634429,
    'depth': 10,
    'l2_leaf_reg': 0.06339510160096859,
    'bootstrap_type': 'MVS',
    'subsample': 0.6742258586023032,
    'random_strength': 6.088205676048116,
    'min_data_in_leaf': 16,
    'loss_function': 'RMSE', # 회귀 문제라면 RMSE를 사용
    'eval_metric': 'RMSE'
}
```


최종 프로세스

1. 앞에서 말한 *Feature*(최근거래가격[10개], 역세권 여부 등)이 추가된 데이터
2. *K-fold Validation*(*split_num*=5)
3. *Catboost hyperparameter tuning*(*optuna*)
4. *Model fit*

최종 프로세스

	kfold_catboo...격_역세권		16999.7092 10798.1379	2024.07.19 18:20
-------------------------------------------------------------------------------------	----------------------	---------------------------------------------------------------------------------------	--------------------------	------------------

- 2. K-fold validation(split_train=0)
- 3. Catboost hyperparameter tuning(optuna)
- 4. Model fit

05

결과 및 인사이트 공유

프로젝트 인사이트 공유

: Regression 대회

1. Lag Feature는 정말 중요하다.
2. Feature가 많다고 좋은 것이 아니다. *Simple is the best!!*
(Noise없는 Feature가 많아야 좋은 것)
3. Model Robustness가 정말 중요하다.
(다중공선성 제거, K-Fold Validation, model ensemble 모두 private 점수 개선에 유의미한 결과)
4. 결측치가 많은 데이터는 그냥 drop하자.
(완벽하게 보간할 수 있다면 사용)
5. Optuna는 정말 성능이 괜찮다.

05

프로젝트 회고

프로젝트 진행 느낀점

: Regression 대회

Point 1

팀 버려!?? 우리팀을 믿자!

이유 : 프로젝트 진행 중, 다양한 디스커션이 올라와 있었는데 우리팀의 작업 흐름에 맞게 진행해야 할 프로세스를 후순위로 두고 새로운 논리들을 적용하였다. 그러나 좋은 프로세스임에도 불구하고 우리팀에는 맞지 않는 프로세스를 적용하여 성능이 좋지 못한 프로세스를 적용하여 귀중한 시간을 활용하지 못하여 아쉬웠다.

Point 2

돌다리 두드리고 건너듯이 데이터셋 확인을 하자

이유 : 매번 피쳐 생성하는 코드를 실행하기 번거로워서 csv파일로 팀원들간 공유했는데, 저장한 csv 파일을 읽을 때 **dtype**이 변환되는 현상을 겪었습니다. 그 상황을 알지 못 해 평가 점수가 다르게 나오는 상황 때문에 매우 당황했는데, **pandas**에서 종종 일어나는 현상이었습니다. 만약 **info** 등으로 데이터프레임을 틸틈이 확인했다면 해당 상황에서 원인을 금방 파악할 수 있었을 것 같습니다.

Point 3

경험을 잘 살리자

이유 : **Enefit Kaggle** 대회에서 우승자가 사용했던 아이디어인 이전날의 에너지 가격(**lag target**)을 추가해 성능을 크게 올린것을 기억하였다. 또한 인간은 보통 집값, 차값을 거래할때 이전 가격을 고려하여 매물을 매매한다고 생각을 하였다. 그래서 동-번지를 기준으로 이전 거래가격을 **feature**로 생성하여 성능을 크게 높일 수 있었다.

프로젝트 아쉬운 점 및 질문들

: Regression 대회

1. 본 데이터셋은 시계열 데이터 셋이라고 보기에 무리가 있지 않나? 시계열 데이터 셋의 기준은 무엇인가?

→ Time Series Split을 했을 때 성능 저하를 경험.

→ 본 데이터는 같은 매물에 대해 지속적으로 정보를 제공하지 않음. (A라는 매물에 대해 매년, 매월, 매일 매매가가 적혀 있지 않음)

→ 시계열 데이터라고 보기 어려운 데이터라는 결론에 도달

2. 모델 앙상블

→ 모델의 일반화를 위해 필요한 과정이었으나 시간 부족 문제로 진행하지 못함..

3. 어떤 모델이 더 좋은 모델인가?

→ Public 점수와 Private 점수의 차이가 적은 모델이 좋은 것인가, 단순히 Private 점수만 좋은 모델이 좋은 것인가?

Life-Changing Education

감사합니다.
