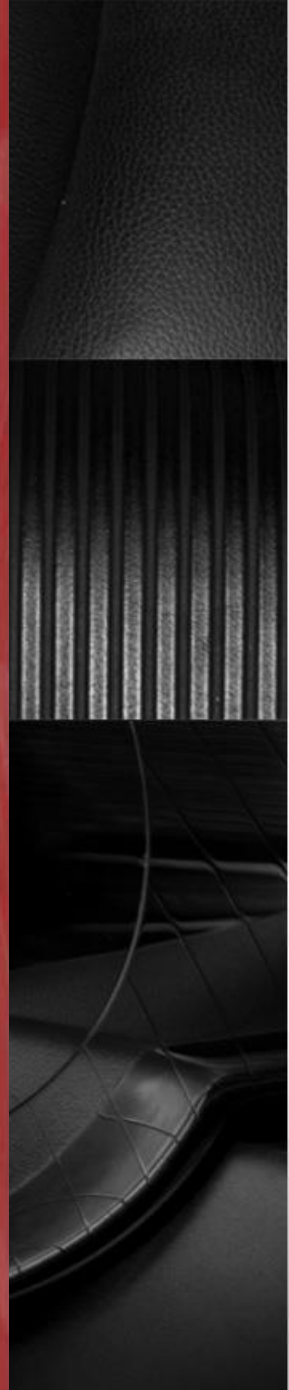


Kaggle - zillow

집값 예측의 오차 맞추기

양종열





목적

- 가격을 예측하는 것이 아니라 예측가격의 오차를 예측하는 것!
- 질문
 - 어떤 특징들이 예측을 어렵게 하는가?
- 쉽게 알 수 있는 것
 - 가격에 영향을 주지 않는 변수는 오차에도 영향을 주지 않을 것이다
 - 가격에 영향을 주는 변수가 오차에도 영향을 줄 것이다
- 가설
 - 아마도 가격에 높은 영향을 주는 변수 중 값이 없거나 값이 잘못될 가능성이 큰 것이 오차를 크게 할 것이다
 - Missing-value라도 그 값이 쉽게 예측 가능하다면 오차에 큰 영향이 없을 것이다



logerrerr

- 그 전에..
- logerror의 level이 적절한가?
- L1 distance(absolute subtract)로 변환하는건 어떨까?



logerrr

- L1 distance로 변환하기
 - L1 distance로 변환하기 위해선 원래의 가격을 알아야 한다.
문제를 풀기 위해 더 어려운 문제를 풀어야 하는 상황...

logerrr

- L1 distance로 변환하기

- 원래의 가격을 가정하면 어떤 상황이 펼쳐질까?
- \$100,000일 경우와 \$1,000,000일 경우
- Logerror: 0.0276

- 원래 가격이 \$100,000일 경우 \$102,798로 예측

$$\log(102,798) - \log(100,000) = 0.0276$$

$$102,798 - 100,000 = 2,798$$

- 원래 가격이 \$1,000,000일 경우 \$1,027,984로 예측

$$\log(1,027,984) - \log(1,000,000) = 0.0276$$

$$1,027,984 - 1,000,000 = 27,984$$

- 원래 가정했던 가격에 비례해 오차가 커진다.

logerrr

- L1 distance로 변환하기
 - 가격의 상대 오차로 하면?
 - Logerror: 0.0276
 - 원래 가격이 \$100,000일 경우 \$102,798로 예측
 $(102,798 - 100,000) / 100,000 = 0.02798$
 - 원래 가격이 \$1,000,000일 경우 \$1,027,984로 예측
 $(1,027,984 - 1,000,000) / 1,000,000 = 0.02798$

logerrr

- 결과

logerror	L1(100,000)	L1(1,000,000)
0.0276	0.02798	0.02798
-0.1684	-0.15498	-0.15498
-0.004	-0.00399	-0.00399
0.0218	0.02204	0.02204
-0.005	-0.00499	-0.00499
-0.2705	-0.23700	-0.23700
0.044	0.04498	0.04498
0.1638	0.17798	0.17798
-0.003	-0.00300	-0.00300
0.0843	0.08796	0.08796
0.3825	0.46594	0.46594

logerrerr


- 결과

logerror	L1(100,000)	L1(1,000,000)
0.0276	0.02798	0.02798
-0.1684	-0.15498	-0.15498
-0.004	-0.00399	-0.00399
0.0218	0.02204	0.02204
-0.005	-0.00499	-0.00499
-0.2705	-0.23700	-0.23700
0.044	0.04498	0.04498
0.1638	0.17798	0.17798
-0.003	-0.00300	-0.00300
0.0843	0.08796	0.08796
0.3825	0.46594	0.46594

Log error와 L1 distance가 거의 같다..

- Log의 차이와 상대오차값은 원래 비슷한가?

$$\ln\left(\frac{v_2}{v_1}\right) \approx \frac{v_2 - v_1}{v_1} ?$$

- 
- 무엇을 예측하는 것이 쉬울까?
 - 예측가의 차이
 - 예측가의 상대오차
 - 예측가의 로그차
 - 즉, 예측하려는 주택의 가격이 올라가면 그에 대한 오차는 비례해서 커지는가?
 - 모름. 세경우 중 실험이 가능한 두가지 경우에 대해 실험



실험

- Library: CatBoost

- GradientBoost 알고리즘을 사용한 라이브러리
- XGBoost와 LightGBM보다 성능이 좋은가? 모름
- Kaggle에 올라온 커널중 LB가 가장 높아서 선정



- Feature Type:

- missing data가 많은 것은 제외
- 나머지 categorical과 real 데이터를 지정하여 입력

- Missing value:


- Categorical 데이터는 새로운 값으로 지정
- Real 데이터는? 마찬가지로 새로운 값으로 지정하면 되지 않을까?

Baseline




See--

Concise catboost starter ensemble (PLB: 0.06435)

last run 3 days ago · Python notebook · 1992 views
using data from [Zillow Prize: Zillow's Home Value Prediction \(Zestimate\)](#) ·  Public

39
voters



[Notebook](#) [Code](#) [Data \(1\)](#) [Output \(6\)](#) [Comments \(10\)](#) [Log](#) [Versions \(6\)](#) [Forks \(106\)](#) [Fork Notebook](#)

Tags finance gradient boosting ensembling housing

Notebook

In [1]:

```
import pandas as pd
import numpy as np
from catboost import CatBoostRegressor
from tqdm import tqdm
```

Missing Value..

- -999로 입력 (이미 잘 처리되어있음)

1.c) Fill missing values

In [9]:

```
# some out of range int is a good choice
train_df.fillna(-999, inplace=True)
test_df.fillna(-999, inplace=True)
```



























기승전 tuning

```
submission_major = 1
for error_type in ['le', 're']:
    for nens in [5, 10]:
        for itr in [200, 400, 600]:
            for lr in [0.03, 0.02, 0.04]:
                for dep in [6, 5, 7]:
                    for llr in [3, 2, 4]:
```

Log error better than Relative error
-> Evaluation이 log error이기 때문?
-> Log error가 학습하기 더 좋은 데이터?

submission_eyle_ne5_it200_lr0.03_de7_ll4.csv 2 days ago by Jong-Yeol Yang SETUP: error_type: le, num_ens: 5, iter: 200, lr: 0.030000, depth: 7, l2_leaf_reg: 4	0.0642289	<input type="checkbox"/>
submission_eyle_ne5_it200_lr0.03_de6_ll3.csv 3 days ago by Jong-Yeol Yang catboost, 2017, baseline	0.0642569	<input type="checkbox"/>
submission_eyle_ne5_it200_lr0.03_de6_ll3.csv 3 days ago by Jong-Yeol Yang catboost, 2017, baseline	Error	<input type="checkbox"/>
submission_eyle_ne5_it200_lr0.03_de6_ll3.csv 3 days ago by Jong-Yeol Yang catboost baseline predict w.r.t date	0.0643487	<input type="checkbox"/>
submission_002.csv 3 days ago by Jong-Yeol Yang catboost, relative error, iter: 400	0.0643156	<input type="checkbox"/>
submission_002.csv 3 days ago by Jong-Yeol Yang catboost, relative error, iter: 600	0.0643339	<input type="checkbox"/>
submission_001.csv 3 days ago by Jong-Yeol Yang catboost iter: 600	0.0643535	<input type="checkbox"/>
submission_001.csv 4 days ago by Jong-Yeol Yang catboost: iteration: 300	0.0643369	<input type="checkbox"/>
submission_001.csv 4 days ago by Jong-Yeol Yang catboost baseline	0.0643486	<input type="checkbox"/>

Best score

115	▼ 38	XArena	  	0.0641720	245	9m
116	▼ 13	vashista.nobaub		0.0641724	117	2d
117	▼ 75	mimba		0.0641729	19	1mo
118	▲ 66	czy	  	0.0641732	42	13h
119	▼ 76	Validation	  	0.0641737	61	17d
120	▼ 76	Rudolph		0.0641740	56	5d
121	▼ 76	danieleewww		0.0641752	322	3h
122	▼ 40	Alan		0.0641753	43	4d
123	▲ 1133	m1in		0.0641753	17	5h
124	▲ 568	Jong-Yeol Yang		0.0641758	35	1h
Your Best Entry ⬆						
Your submission scored 0.0641758, which is an improvement of your previous score of 0.0641819. Great job!  Tweet this!						
125	▼ 57	Igor Praznik		0.0641758	43	5d
126	▼ 36	220V	  	0.0641764	164	9h
127	▲ 30	Yez121		0.0641768	33	4h
128	▼ 82	levelcrosser	  	0.0641780	225	4h
129	▲ 144	Oscar Takeshita		0.0641781	66	8h

의 결과..



결론

- 결과에 영향을 주지 않는 피처는 빼는게 더 좋다
 - Missing value가 0.98 이상인 데이터는 삭제
- Missing value를 분명하게 구분해주는 것이 더 좋다
 - -999로 세팅
- Log error를 그대로 사용하는 것이 좋다
 - Relative error보다 성능 우세
- 튜닝(실험)은 많이 하는게 좋다
 - 머신러닝과 PPT는 시간에 비례..



Q&A