

Проект

По теме:

**Разработка модели для
улучшение качества и скорости
совершения сделок риелторами**

Актуальность задачи

- ▶ В наше время, все больше и больше людей начинают использовать помощь компьютерных технологий. Те, кто их не использует, оказываются в проигрышном положении. Внедрение машинного обучения в отрасль торговли благоприятно скажется на эффективности и скорости работы в этой области

Цель работы:

- Разработать модель, качественно предсказывающую цены домов, основываясь на истории их предложений

В ходе работы я применял метрику `r2_score`

- ▶ Лист метрик для задач регрессии очень ограничен
- ▶ Среди этих метрик - среднеквадратичная ошибка и т. п., по которым тяжело определить качество модели
- ▶ `r2` позволяет легко оценить качество/изменение качества модели.


Этапы работы над задачей

- ▶ Предварительная обработка данных
- ▶ Получение дополнительных данных из полей таблицы
- ▶ Анализ и очистка данных
- ▶ Выбор оптимального алгоритма и подбор параметров
- ▶ Обучение модели и оценка полученного результата

Предварительная обработка данных

```
#Рассмотрим stories
print(data['stories'].unique())
```

```
[nan '2.0' '1.0' '3.0' 'One' '2' 'Multi/Split' '4.0' '0.0' '0' 'One Level'  
    '1' '9.0' '3' '1 Level, Site Built' 'One Story' '3.00' '1.00' '14.0'  
    'Two' '3+' '1 Story' '5.0' '2 Story' 'Ranch/1 Story' 'Condominium'  
    'Stories/Levels' '7.0' '2 Level, Site Built' '2 Level' '15'  
    '3 Level, Site Built' '4' '22.0' '2.00' '6.0' '1.0000' 'Lot' '3 Story'  
    'Three Or More' '1.5' '1 Level' 'Two Story or More'  
    'Site Built, Tri-Level' '54.0' '23' 'Farm House' '8.0' '16.0' '1.50' '18'  
    '9' '21' '8' '12.0' 'Split Level w/ Sub' '11.0' '18.0' '1.5 Stories' '7'  
    '11' 'Townhouse' '12' '21.0' '16' '1.5 Story/Basement' '28.0'  
    'Traditional' '2.5 Story' '17' '2.0000' '63.0' 'Acreage'  
    'Ground Level, One' '6' 'Split Foyer' '2 Stories' '27.0' '19.0' '2.50'
```

replace 

regular expr ✓

Получение дополнительных данных

```
print(data['homeFacts'][0])  
print(type(data['homeFacts'][0]))
```

```
{'atAGlanceFacts': [{'factValue': '2019', 'factLabel': 'Year built'}, {'factValue': '', 'factLabel': 'Remodeled year'}, {'factValue': 'Central A/C, Heat Pump', 'factLabel': 'Heating'}, {'factValue': '', 'factLabel': 'Cooling'}, {'factValue': '', 'factLabel': 'Parking'}, {'factValue': None, 'factLabel': 'lotsize'}, {'factValue': '$144', 'factLabel': 'Price/sqft'}]}
```

Тип данных: <class 'str'>

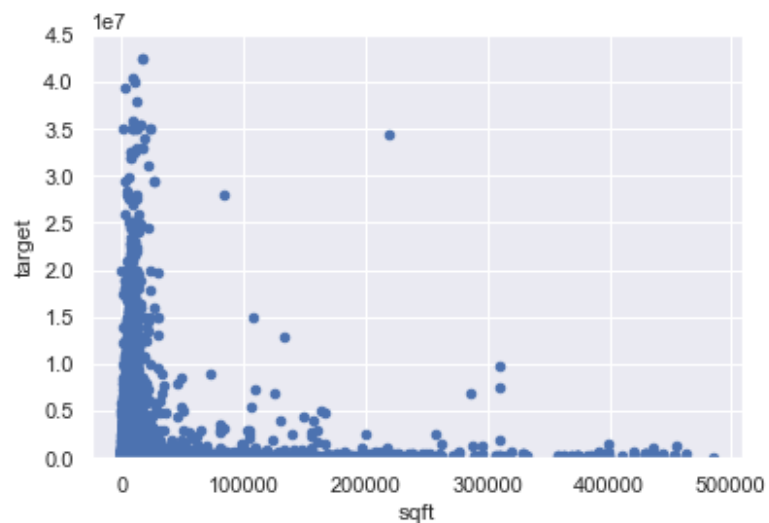
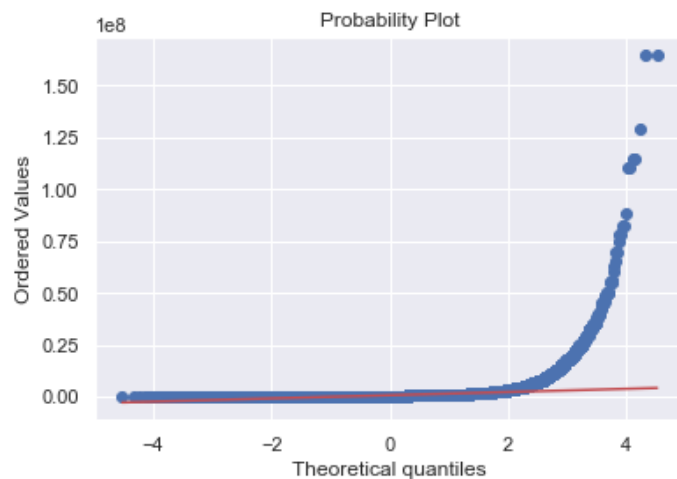
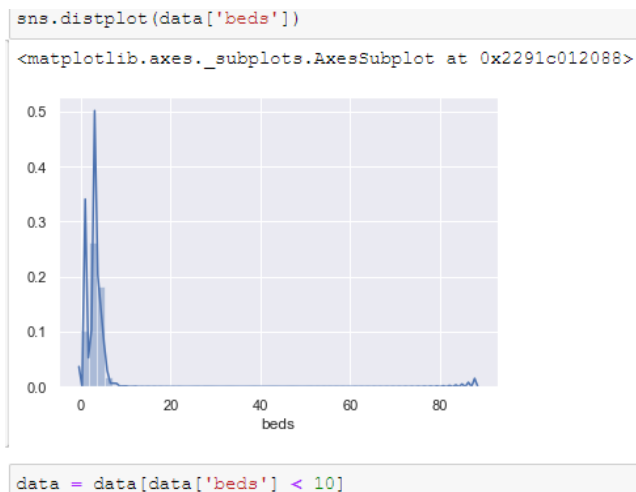
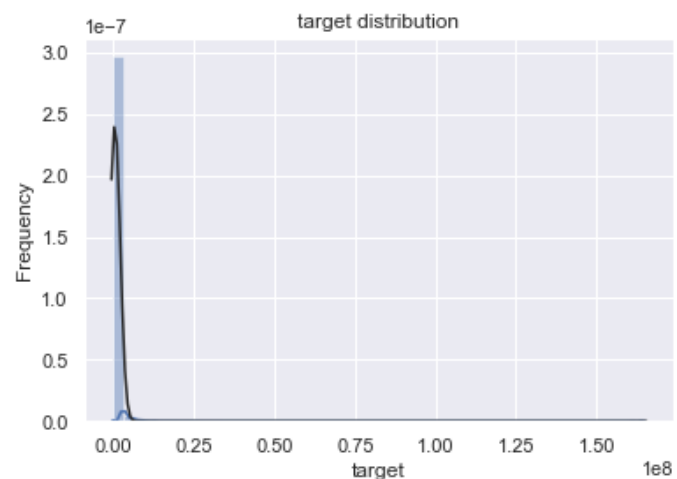
ast.literal_eval

```
: ex = ast.literal_eval(data['homeFacts'][0])  
print(ex)|  
print('Тип данных: {}'.format(type(ex)))
```

```
{'atAGlanceFacts': [{'factValue': '2019', 'factLabel': 'Year built'}, {'factValue': '', 'factLabel': 'Remodeled year'}, {'factValue': 'Central A/C, Heat Pump', 'factLabel': 'Heating'}, {'factValue': '', 'factLabel': 'Cooling'}, {'factValue': '', 'factLabel': 'Parking'}, {'factValue': None, 'factLabel': 'lotsize'}, {'factValue': '$144', 'factLabel': 'Price/sqft'}]}
```

Тип данных: <class 'dict'>

Анализ данных



target dispersion: 2726881339200.9546

Выбор алгоритма и подбор параметров

- ▶ R2 Xgboost: 0.98586
 - ▶ R2 GradientBoosting: 0.98248
 - ▶ R2 DecisionTree: 0.96734
 - ▶ R2 Kneighbors: 0.87198
- R2 RandomForest: 0.98107
 - R2 LGBMRegressor: 0.95281
 - R2 catboost: 0.93742
 - R2 LinearRegression: 0.20581

```
alg_thrd_model = GradientBoostingRegressor(random_state=21)
alg_thrd_params = [{
    "n_estimators": [75, 150, 225],
    "min_samples_split": [2],
    "min_samples_leaf": [2],
    "max_depth": [4, 5, 6]
}]
alg_thrd_grid = GridSearchCV(alg_thrd_model, alg_thrd_params, cv=cv, refit=True, n_jobs=-1)
alg_thrd_grid.fit(X_train, y_train)
alg_thrd_best = alg_thrd_grid.best_estimator_
print("gradient boosting: {} with params {}".format(
    alg_thrd_grid.best_score_, alg_thrd_grid.best_params_))

gradient boosting: 0.9858726548221832 with params {'max_depth': 4, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 75}
```

Обучение модели и оценка результата

- ▶ Gradient Boosting:
- ▶ `max_depth = 5,`
- ▶ `min_samples_leaf = 2,`
- ▶ `min_samples_split = 2,`
- ▶ `n_estimators = 300`

```
print(r2_score(y_test, y_pred))
```


```
0.9908028531942629
```

```
data = X_test.join(pred)  
data.head()
```

	baths	fireplace	sqft	beds	state	stories	PrivatePool	built	rebuilt	heating	cooling	parking	lotsize	price	distance	target
0	3	0	2900	3	0	1	0	2019	2019	1	1	1	8398	138	2	399448.675269
1	3	0	1768	3	1	1	0	1956	1975	1	0	1	7800	396	0	684784.855264
2	2	0	984	2	0	2	0	1924	1962	1	1	1	3675	44	0	45942.627604
3	3	1	3472	4	1	2	0	2004	2004	1	1	1	14375	158	1	560066.571122
4	3	0	2902	4	0	1	1	2000	2002	1	1	1	7967	95	0	271812.836297

Github

<https://github.com/Brilliance1512/dataproject>

 Brilliance1512 / **dataproject**

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security 0

Insights

Settings

No description, website, or topics provided.

Edit

[Manage topics](#)

23 commits

1 branch

0 packages

0 releases

1 contributor

Branch: master ▾


New pull request

Create new file





Upload files

Find file

Clone or download ▾

 Brilliance1512 Add files via upload

Latest commit 93940a0 2 days ago

 Описание проекта.txt	Update Описание проекта.txt	7 days ago
 Презентация.pdf	Add files via upload	8 days ago
 Проект.ipynb	Add files via upload	7 days ago
 Прототип.ipynb	Add files via upload	2 days ago

Help people interested in this repository understand your project by adding a README.

Add a README