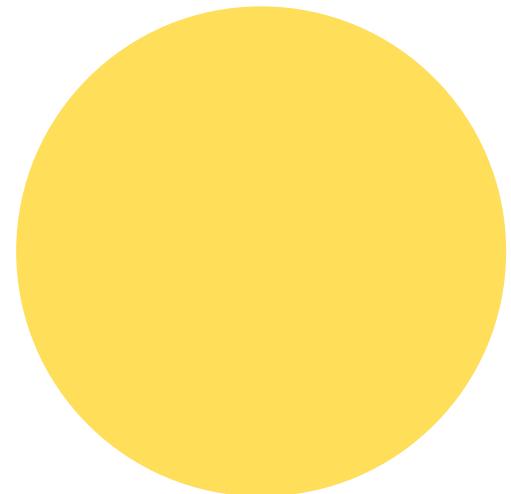


Дипломный проект

Разработка сервиса для предсказания стоимости домов, на основе истории предложений.

Спикер:
Колян Павел
Ментор:
Сидоров Никита



Задача проекта

Очистить и подготовить данные из предоставленного датасета для использования моделью машинного обучения.

Построить модели машинного обучения сравниТЬ их и выбрать лучшую на основе выбранной метрики.

Описание данных

status	Статус недвижимости
private pool	Наличие бассейна
propertyType	Тип недвижимости
street	Адрес
baths	Описание санузла
homeFacts	Описание недвижимости
fireplace	Наличие камина и его описание
city	Город
schools	Наличие школ
sqft	Площадь объекта.
zipcode	Почтовый индекс
beds	Спальные
state	Площадь объекта.
stories	Количество этажей
mls-id	Номер в реестре
PrivatePool	Наличие бассейна
MlsId	Номер в реестре
target	Целевая переменная

Предобработка (очистка) данных

- Очень загрязненный датасет.
- Большое количество по разному записанных одинаковых по значению данных.
- Все столбцы имеют пропуски.
- Столбцы `homeFacts` и `schools` это вложенные датасеты в формате json

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 377185 entries, 0 to 377184
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   status            337267 non-null   object 
 1   private pool      4181 non-null   object 
 2   propertyType     342452 non-null   object 
 3   street            377183 non-null   object 
 4   baths             270847 non-null   object 
 5   homeFacts         377185 non-null   object 
 6   fireplace          103115 non-null   object 
 7   city              377151 non-null   object 
 8   schools            377185 non-null   object 
 9   sqft              336608 non-null   object 
 10  zipcode           377185 non-null   object 
 11  beds               285903 non-null   object 
 12  state              377185 non-null   object 
 13  stories            226470 non-null   object 
 14  mls-id             24942 non-null   object 
 15  PrivatePool       40311 non-null   object 
 16  MlsId              310305 non-null   object 
 17  target              374704 non-null   object 
dtypes: object(18)
memory usage: 51.8+ MB
```

Датасеты из столбцов homeFacts и schools

	rating	name	data.Distance	data.Grades
0	[4', '4', '7', 'NR', '4', '7', 'NR', 'NR']	['Southern Pines Elementary School', 'Southern...']	['2.7 mi', '3.6 mi', '5.1 mi', '4.0 mi', '10.5...']	['3-5', '6-8', '9-12', 'PK-2', '6-8', '9-12', ...]
1	['4/10', 'None', '4/10']	['East Valley High School&Extension', 'Eastval...']	['1.65mi', '1.32mi', '1.01mi']	['9-12', '3-8', 'PK-8']
2	['8/10', '4/10', '8/10']	['Paul Revere Middle School', 'Brentwood Scien...']	['1.19mi', '2.06mi', '2.63mi']	['6-8', 'K-5', '9-12']

	Year built	Remodeled year	Heating	Cooling	Parking	lotsize	Price/sqft
0	2019	NaN	Central A/C, Heat Pump	NaN	NaN	NaN	\$144
1	2019	NaN	NaN	NaN	NaN	5828 sqft	\$159/sqft
2	1961	1967.0	Forced Air	Central	Attached Garage	8,626 sqft	\$965/sqft

Итоговый датасет без dummy получил 14 категориальных и 9 числовых признаков

Датасет с dummy получил 449 признаков

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 341561 entries, 0 to 341560
Columns: 449 entries, status to mean_city_PrivatePool
dtypes: float64(316), int64(117), object(16)
memory usage: 1.1+ GB
```

status	object
propertyType	object
street	object
fireplace	object
city	object
zipcode	object
state	object
heating	object
cooling	object
parking	object
rating	object
name	object
data_distance	object
data_grades	object
beds	float64
stories	float64
PrivatePool	float64
baths	float64
sqft	float64
year_built	float64
remodeled_year	float64
price_sqft	float64
lotsize	float64
dtype: object	

График до удаления пропусков

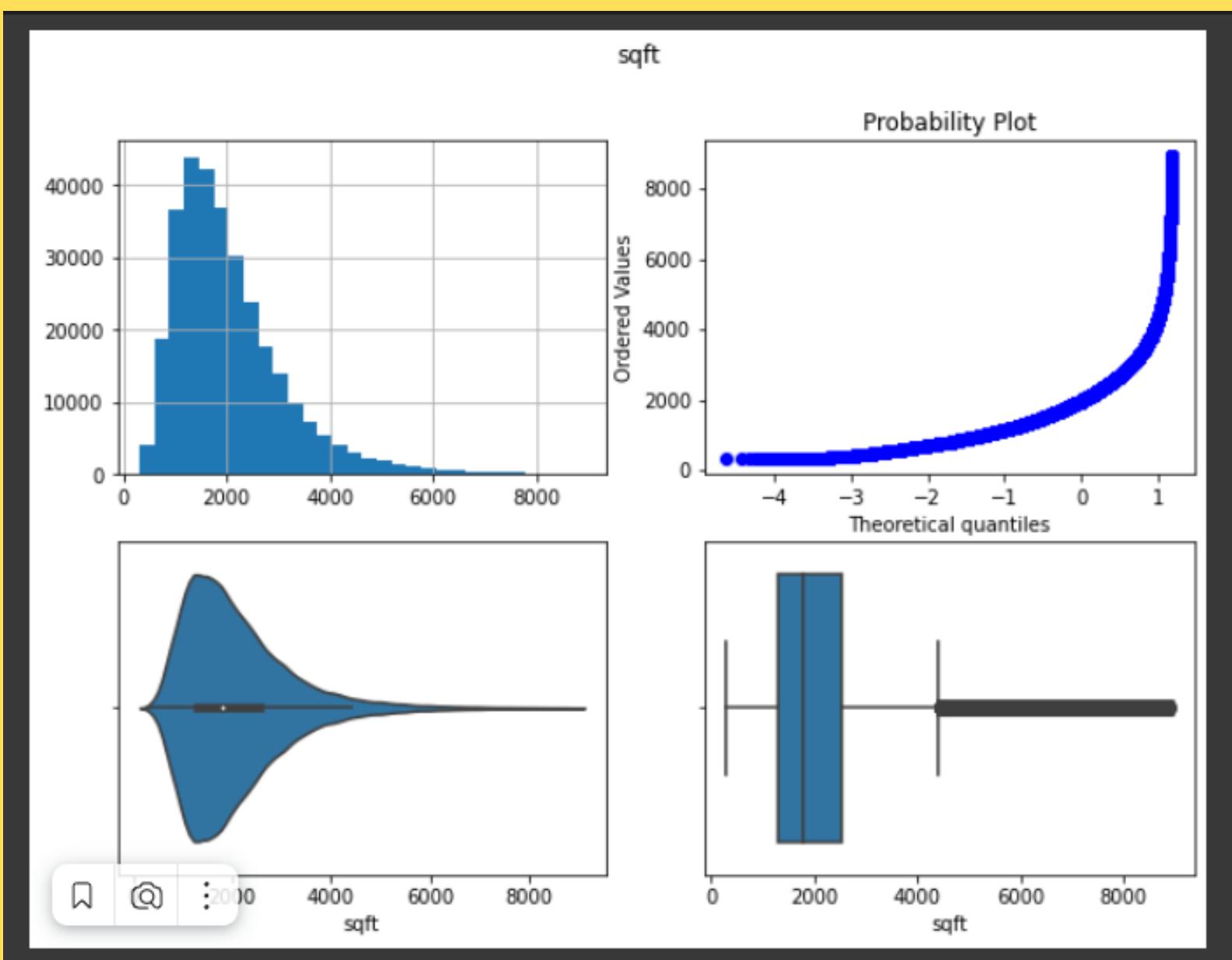
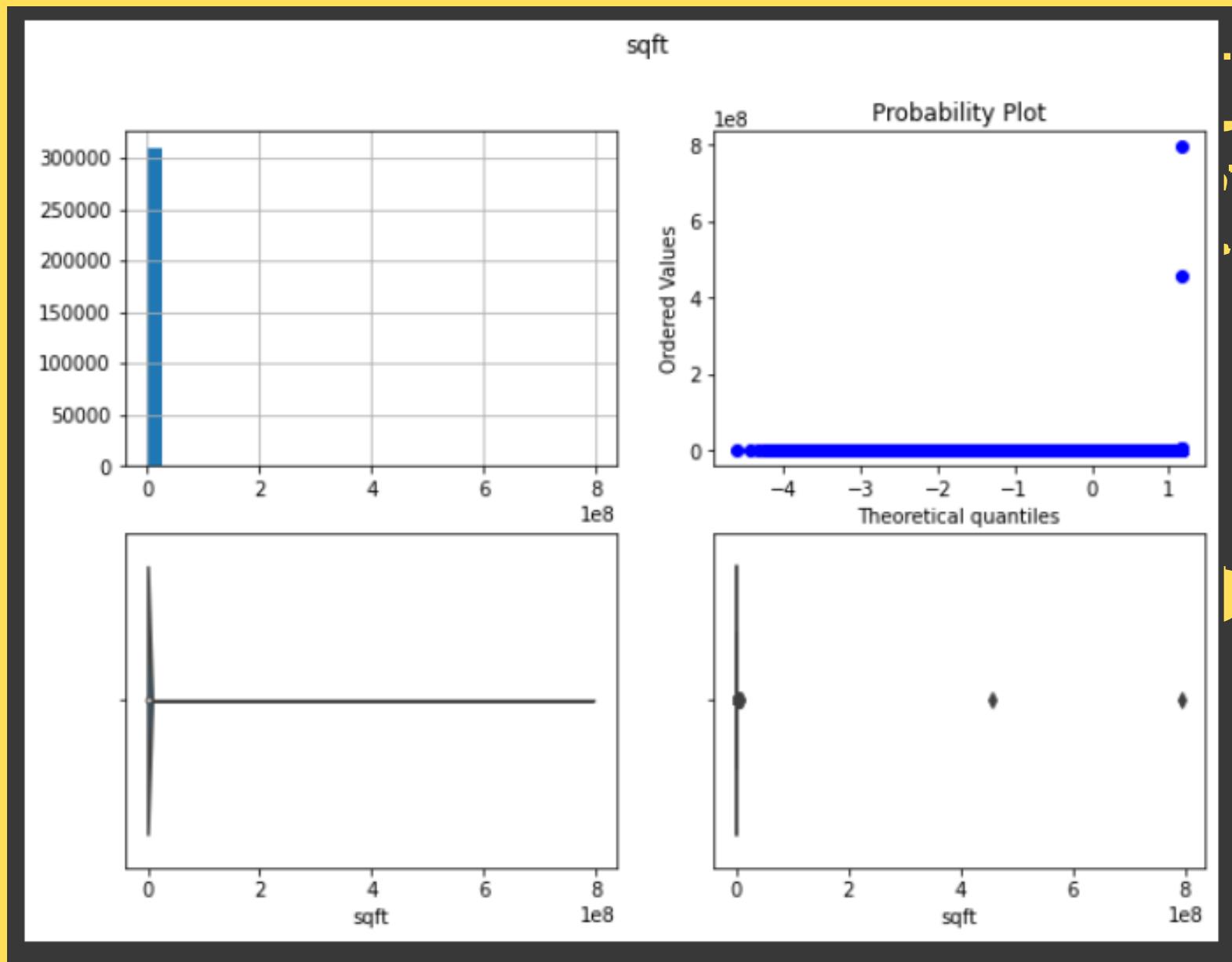
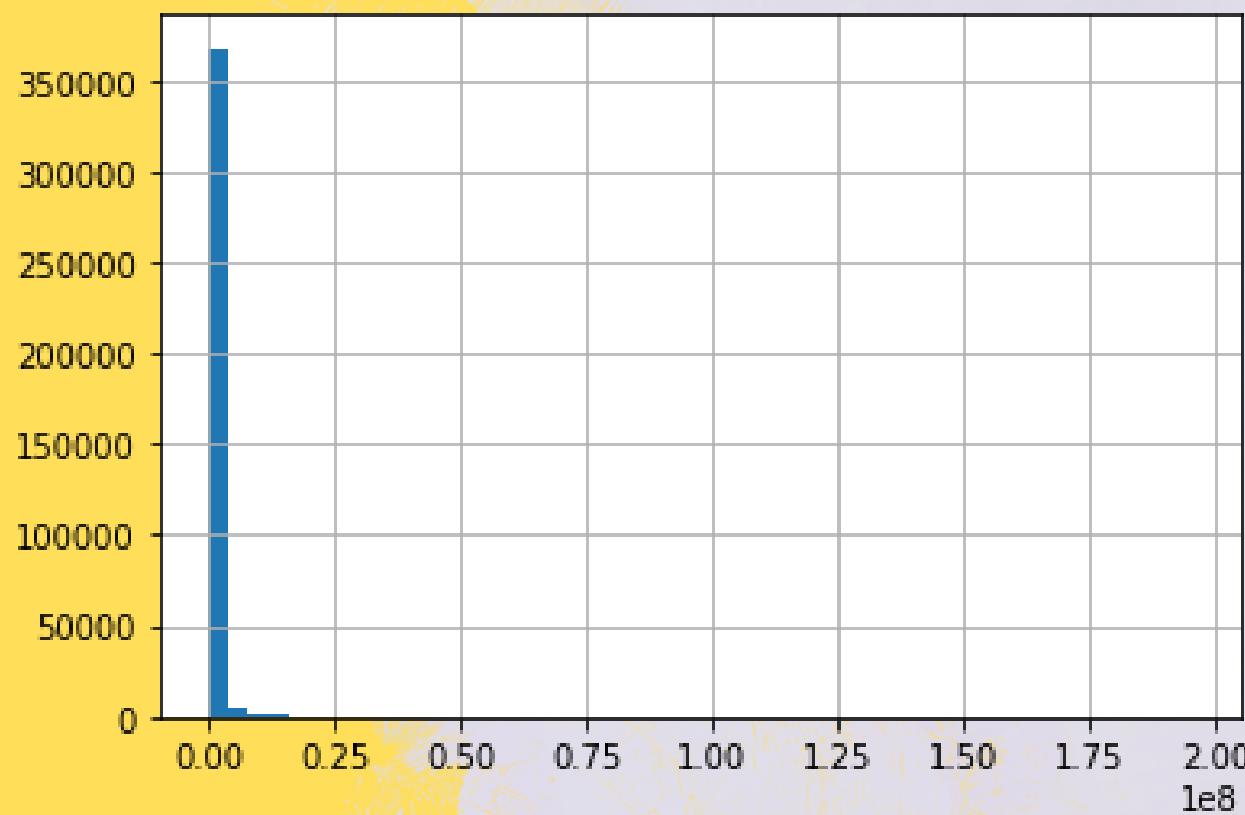
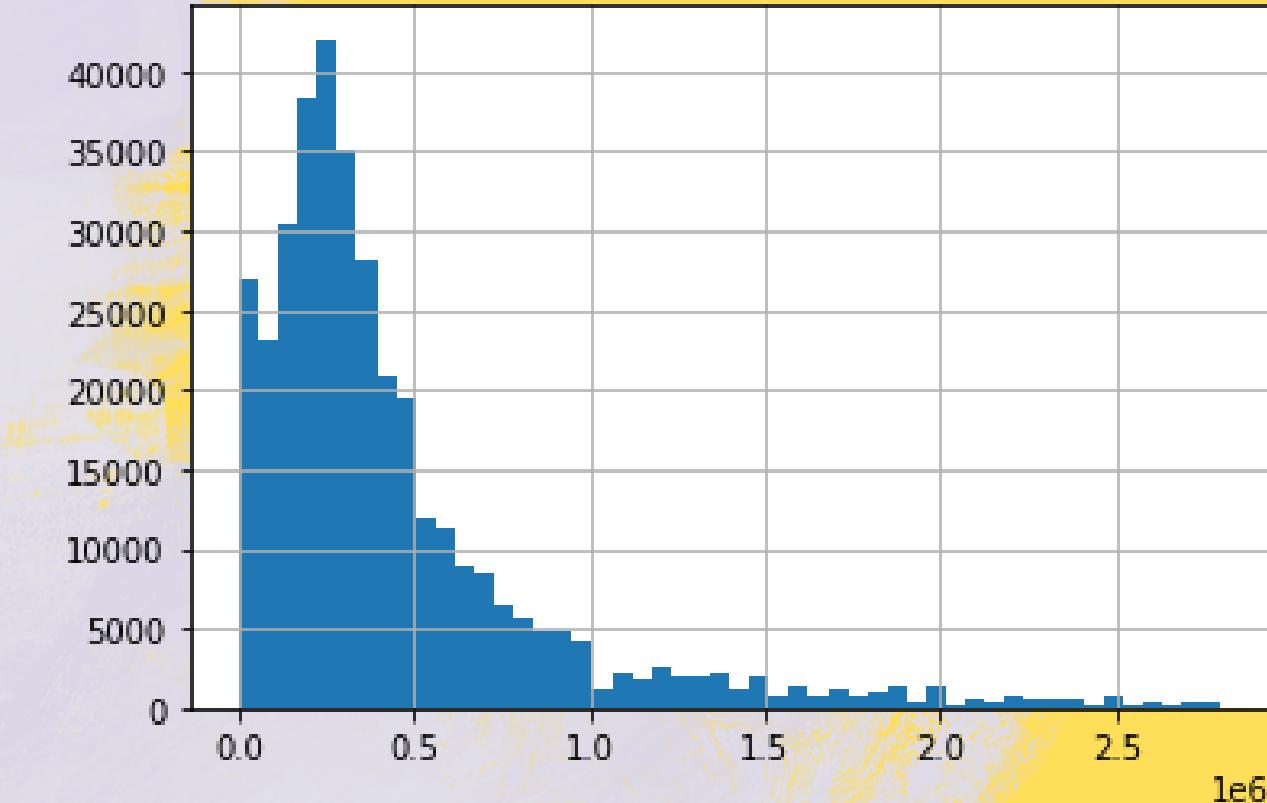


График после удаления пропусков

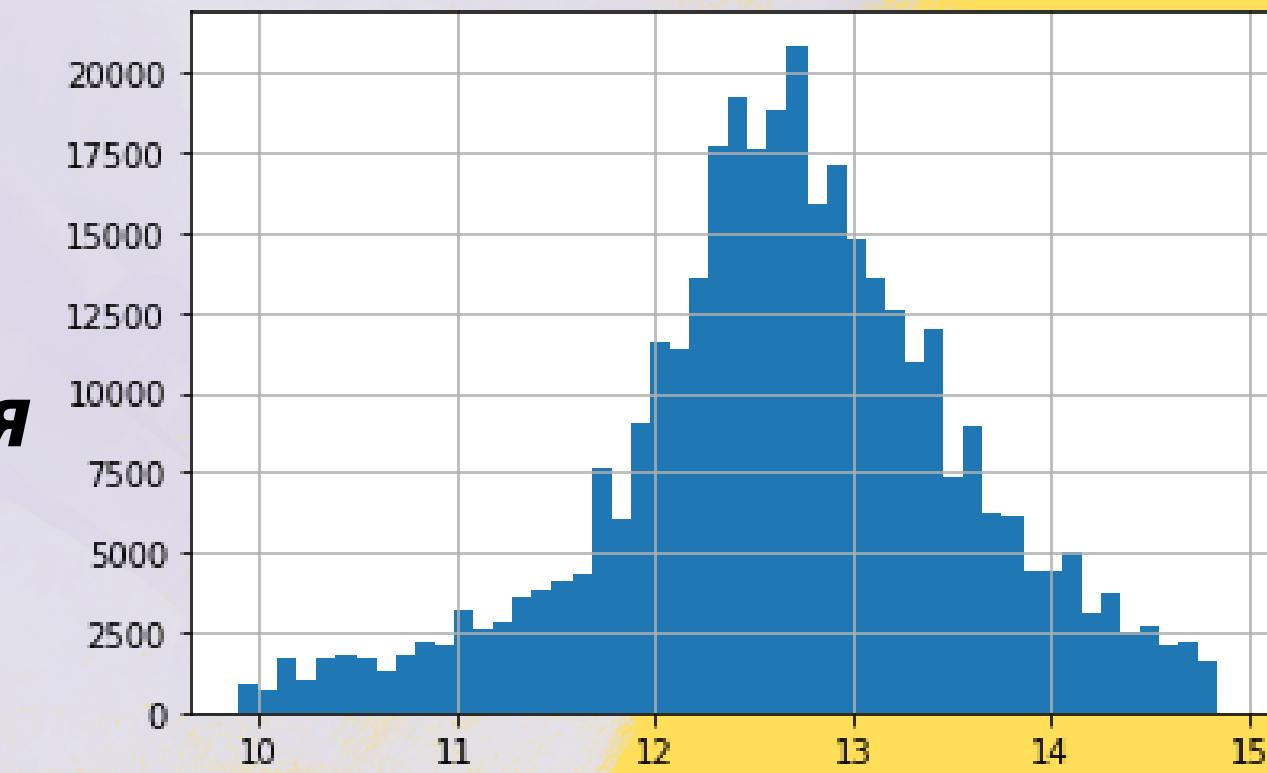
График распределения целевой переменной с выбросами



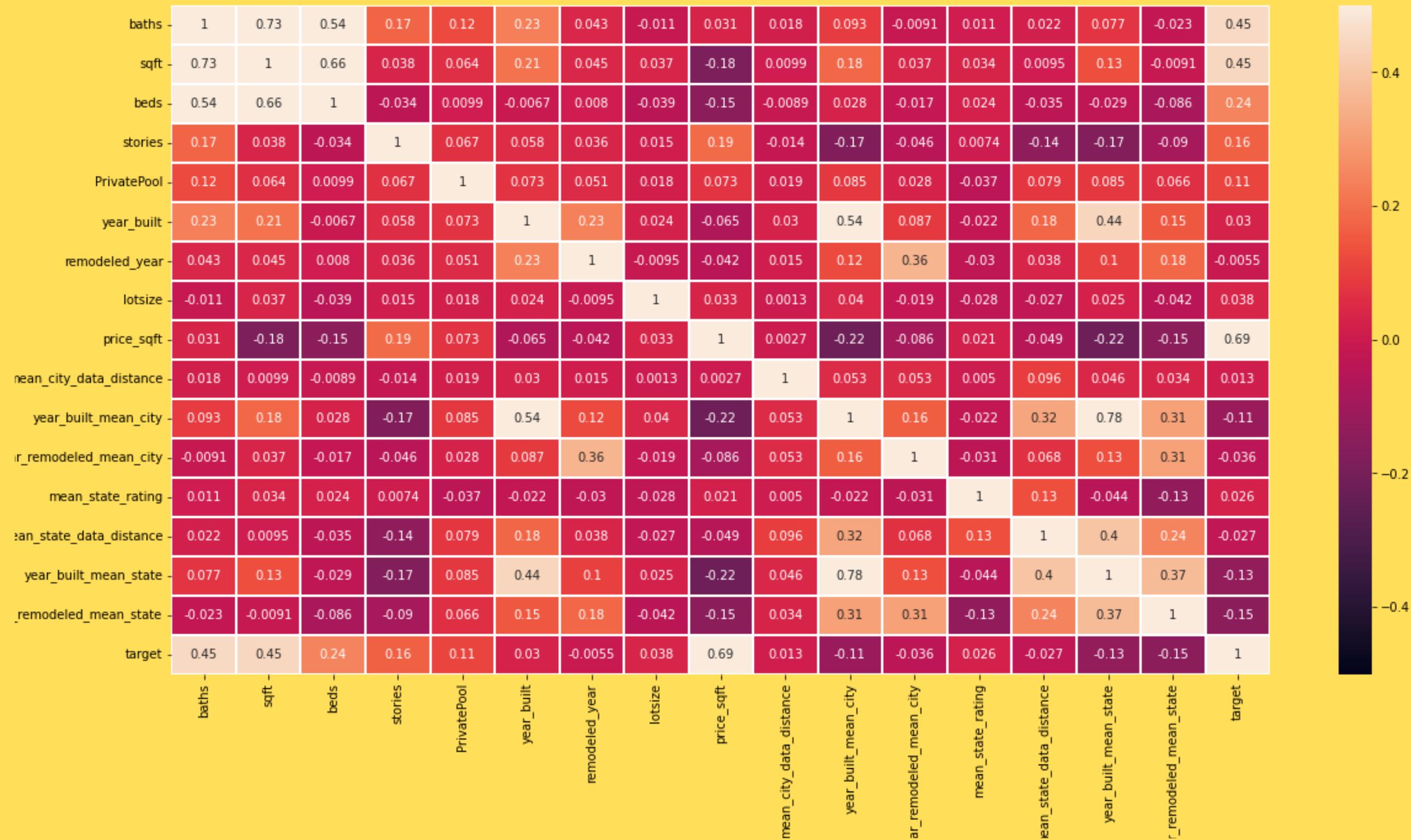
Без выбросов



*после
логарифмирования*



Heatmap



**Был извлечен ряд
новых признаков**



rating_sum	int64
rating_mean	float64
data_distance_sum	float64
data_distance_mean	float64
data_distance_median	float64
mean_city_rating	float64
mean_city_data_distance	float64
year_built_mean_city	float64
year_remodeled_mean_city	float64
mean_state_rating	float64
mean_state_data_distance	float64
year_built_mean_state	float64
year_remodeled_mean_state	float64
price_sqft_mean_state	float64
min_state_sqft	float64
mean_city_sqft	float64
median_city_sqft	float64
min_city_sqft	float64
max_city_sqft	float64
beds_mean_state	float64
mean_state_sqft	float64
min_state_stories	float64
mean_city_stories	float64
median_city_stories	float64
max_city_stories	float64
mean_city_PrivatePool	float64
dtype: object	

Модели для теста:

- CatBoost
- RandomForest
- GradientBoosting
- StackingRegressor

После подбора гиперпараметров был произведен ряд тестов

и применен blending

Результаты



	Score на train	MAPE	MAE
stacking_train	9.41	29446.0	
rand_forest_train	9.47	27667.0	
blending_stack_train	9.63	29184.0	
blending_grad_train	9.92	29733.0	
grad_boost_train	10.35	30422.0	
cat_train	11.40	36524.0	

	Score на val	MAPE	MAE
cat_val	10.86	42094.0	
blending_stack_val	11.67	38697.0	
blending_grad_val	11.79	39180.0	
stacking_val	12.94	41954.0	
rand_forest_val	12.95	39836.0	
grad_boost_val	13.41	42936.0	

Выводы и результаты работы

- Данные разделены на числовые и категориальные признаки.
- Извлечен ряд новых признаков и добавлены dummy признаки.
- Протестировано несколько моделей.
- Определена лучшая модель на основе выбранной метрики.

Лучшая модель - *CatBoost*

score MAPE: 10.86/11.40