Государственное бюджетное профессиональное
образовательное учреждение Московской области
«Физико-технический колледж»

«Анализ цены квартиры на вторичном рынке по Московскому региону: Москва, Новая Москва, Московская область»

Работу выполнил:
студент группы № ИСП-22
Кузнецов Арсений
Проверил:
Преподаватель информатики
Базяк Г.В. Коновалов И. В.

Долгопрудный, 2024

Оглавление

1. Введение

В этом отчете представлены результаты анализа цен на рынке недвижимости Московского региона, включая Москву, Новую Москву и Московскую область. Исследование направлено на выявление ключевых факторов, влияющих на стоимость квартир, что в дальнейшем может использоваться для разработки прогнозных моделей, оценивающих цену квадратного метра жилья.

2. Цели и задачи

Цель: сбор и анализ данных для создания прогностической модели, оценивающей стоимость недвижимости в Московском регионе.

Задачи:

1. Сбор данных о стоимости квартир из открытых источников.
2. Очистка и подготовка данных для анализа, включая обработку пропусков и аномальных значений.
3. Визуализация данных и выявление ключевых факторов, влияющих на стоимость.

3. Основная часть

3.1. Сбор данных

Для анализа использовался интернет-ресурс ЦИАН. Данные о квартирах были собраны с использованием Python и библиотеки CianParser. В итоге был сформирован датафрейм с ключевыми характеристиками квартир.

3.2. Очистка и подготовка данных

1. **Удаление ненужных столбцов**:
    - Удалены столбцы, которые не нужны для анализа (Рис. 1)
2. **Обработка пропусков**:
    - Заменены значения -1 на NaN для корректной обработки пропусков. (Рис. 2)
    - Удалены строки с пропусками в ключевых столбцах, удалены столбцы с большим количеством пропусков, в некоторых столбцах пропуски заменены на «0» (Рис. 3)
    - Заполнены пропуски в некоторых столбцах средним значениям (Рис. 4)
3. **Удаление выбросов**:
    - Удалены строки с аномальными значениями в столбцах (Рис. 5)
4. **Форматирование данных**:
    - Преобразованы строковые значения в числовые, где это необходимо. (Рис. 6)
5. **Кодирование категориальных переменных**:

- Закодированы категориальные переменные числовыми значениями. (Рис. 7)

3.3. Визуализация данных

Для визуального анализа использовались библиотеки matplotlib и seaborn, с помощью которых были созданы следующие графики:

1. **Тепловая карта пропусков**: этот график показывает распределение пропусков в датафрейме (Рис. 8)
2. **Матрица корреляции**: показывает взаимосвязь между различными числовыми столбцами в датафрейме. Цветные ячейки указывают на силу и направление корреляции. (Рис. 9)
3. **Графики зависимости цены за квадратный метр от различных факторов**: эти графики показывают, как различные факторы (например, количество этажей, этаж, год постройки, тип парковки, количество комнат) влияют на цену за квадратный метр. Каждый график представляет собой столбчатую диаграмму, где по оси X отложены значения фактора, а по оси Y — средняя цена за квадратный метр. (Рис. 10-14)

3.4. Аналитика данных

В результате анализа получены следующие ключевые выводы:

1. **Влияние количества этажей в доме на цену за квадратный метр**:

- **Вывод**: Дома с большим количеством этажей, как правило, имеют более высокую цену за квадратный метр. Это может быть связано с тем, что такие дома часто находятся в более престижных районах или имеют лучшую инфраструктуру.

2. **Влияние этажа на цену за квадратный метр**:
    - **Вывод**: Цены за квадратный метр на первых и последних этажах, как правило, ниже, чем на средних этажах. Это может быть связано с тем, что первые этажи часто подвержены шуму и не имеют видовых преимуществ, а последние этажи могут страдать от проблем с отоплением и доступом к лифту.
3. **Влияние года постройки на цену за квадратный метр**:
    - **Вывод**: Дома, построенные в последние годы, как правило, имеют более высокую цену за квадратный метр. Это может быть связано с использованием современных технологий и материалов, а также с более высоким уровнем комфорта и безопасности.
4. **Влияние типа парковки на цену за квадратный метр**:
    - **Вывод**: Наличие парковки, особенно подземной или закрытой, значительно повышает цену за квадратный метр. Это может быть связано с тем, что парковка является важным фактором для многих покупателей.
5. **Влияние количества комнат на цену за квадратный метр**:
    - **Вывод**: Цены за квадратный метр в квартирах с большим количеством комнат, как правило, выше. Это может быть связано с тем, что такие квартиры часто имеют большую общую площадь и более высокий уровень комфорта.

4. Заключение

В ходе работы были собраны и подготовлены данные, а также созданы визуализации, выявляющие основные зависимости. Данный анализ подтвердил, что факторы, такие как местоположение, площадь, год постройки и наличие удобств, оказывают ключевое влияние на стоимость недвижимости. Эти данные могут быть использованы для дальнейшей разработки моделей, прогнозирующих стоимость недвижимости на вторичном рынке Московского региона.

Удаляем ненужные для анализа колонки

```
df.columns # Смотрим колонки
```

```
Index(['author', 'author_type', 'url', 'location', 'deal_type',
       'accommodation_type', 'floor', 'floors_count', 'rooms_count',
       'total_meters', 'price', 'year_of_construction', 'object_type',
       'have_loggia', 'parking_type', 'house_material_type', 'heating_type',
       'finish_type', 'living_meters', 'kitchen_meters', 'phone',
       'ceiling_height', 'district', 'street', 'house_number', 'underground',
       'residential_complex'],
      dtype='object')
```

```
df.drop(['author', 'author_type', 'deal_type', 'accommodation_type', 'phone', 'house_number'], axis=1, inplace=True)
print(f"Осталось {df.shape[0]} строк и {df.shape[1]} колонок")
```

```
Осталось 6728 строк и 21 колонок
```

Рис. 1

```
df.replace('-1', np.nan, inplace=True) # Заменяем -1 (пустое значение при парсинге) на NaN

df.head(10)
```

Python

	url	location	floor	floors_count	rooms_count	total_meters	price	year_of_construction	object_type	have_loggia	...	house_material_type	heating_type	finish_type	living_meters
0	https://serpukhov.cian.ru/sale/flat/301136001/	Серпухов	6	6	1	20.7	2350000	1917	Вторичка	NaN	...	NaN	NaN	NaN	18 м²
1	https://serpukhov.cian.ru/sale/flat/305145433/	Серпухов	1	9	1	33.0	4150000	1975	Вторичка	NaN	...	NaN	NaN	NaN	18 м²
2	https://serpukhov.cian.ru/sale/flat/308145954/	Серпухов	4	4	1	14.0	1700000	NaN	Вторичка	NaN	...	NaN	NaN	NaN	NaN
3	https://serpukhov.cian.ru/sale/flat/309086009/	Серпухов	5	5	1	17.5	2500000	1968	Вторичка	NaN	...	NaN	NaN	NaN	NaN
4	https://serpukhov.cian.ru/sale/flat/308401669/	Серпухов	4	7	1	25.9	3000000	2009	Вторичка	NaN	...	NaN	NaN	NaN	20 м²
5	https://serpukhov.cian.ru/sale/flat/304233768/	Серпухов	2	3	1	46.7	6599999	NaN	Вторичка	NaN	...	NaN	NaN	NaN	25 м²
6	https://serpukhov.cian.ru/sale/flat/304697154/	Серпухов	3	5	1	29.0	3350000	1917	Вторичка	NaN	...	NaN	NaN	NaN	25 м²
7	https://serpukhov.cian.ru/sale/flat/304332318/	Серпухов	3	6	1	19.1	3600000	1917	Вторичка	NaN	...	NaN	NaN	NaN	NaN
8	https://serpukhov.cian.ru/sale/flat/295915722/	Серпухов	1	2	1	27.1	1900000	1969	Вторичка	NaN	...	NaN	NaN	NaN	18 м²
9	https://serpukhov.cian.ru/sale/flat/304548494/	Серпухов	6	9	2	42.7	3990000	1980	Вторичка	1 лоджия	...	NaN	NaN	NaN	29,7 м²

10 rows × 21 columns

Рис. 2

```
df.drop(['heating_type', 'house_material_type', 'residential_complex', 'district', 'finish_type'], axis=1,inplace=True)

df = df.dropna(subset=['url', 'rooms_count', 'street'])

df['have_loggia'] = df['have_loggia'].fillna('0')
df['parking_type'] = df['parking_type'].fillna('0')
df['underground'] = df['underground'].fillna('0')

print(f"Стало {df.shape[1]} колонок и {df.shape[0]} строк")
```

```
Стало 16 колонок и 5240 строк
```

Рис. 3

```
# Заполняем пропуски средним значением для каждого оставшегося столбца с пропусками
df['living_meters'].fillna(df['living_meters'].mean(), inplace=True)
df['kitchen_meters'].fillna(df['kitchen_meters'].mean(), inplace=True)
df['ceiling_height'].fillna(df['ceiling_height'].mean(), inplace=True)
```

Рис. 4

```
#Удаляю всё неестественное
df = df.drop(df[(df['total_meters'] > 300)].index)
df = df.drop(df[(df['price'] > 400000000)].index)
df = df.drop(df[(df['living_meters'] > 300)].index)
df = df.drop(df[(df['kitchen_meters'] > 300)].index)
df = df.drop(df[(df['ceiling_height'] > 4)].index)
df = df.drop(df[(df['year_of_construction'] < 1950) | (df['year_of_construction'] > 2026)].index)
```

Рис. 5

```
df['price'] = pd.to_numeric(df['price'], errors='coerce')
df = df.dropna(subset=['price'])
df['price'] = df['price'].astype(float)
df['price'].info()

df['rooms_count'] = pd.to_numeric(df['rooms_count'], errors='coerce')
df = df.dropna(subset=['rooms_count'])
df['rooms_count'] = df['rooms_count'].astype(int)

df['floor'] = pd.to_numeric(df['floor'], errors='coerce')
df['floors_count'] = pd.to_numeric(df['floors_count'], errors='coerce')
df['total_meters'] = pd.to_numeric(df['total_meters'], errors='coerce')
```

```
<class 'pandas.core.series.Series'>
Index: 5236 entries, 0 to 8391
Series name: price
Non-Null Count  Dtype
--------------  -----
5236 non-null   float64
dtypes: float64(1)
memory usage: 81.8 KB
/tmp/ipykernel_12835/1426250752.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['price'] = pd.to_numeric(df['price'], errors='coerce')
```

Делаем с year_of_construction тоже самое, что и с price. Выводим уникальные значения, чуть позже удалим выбросы

```
df['year_of_construction'] = pd.to_numeric(df['year_of_construction'], errors='coerce')
df = df.dropna(subset=['year_of_construction'])
df['year_of_construction'] = df['year_of_construction'].astype(int)
print(df['year_of_construction'].dtype)
print(df['year_of_construction'].unique())
```

Рис. 6

```
# Функция, котораякодирует категориальные переменные в датафрейме числовыми значениями
def number_encode_features(init_df):
    result = init_df.copy()  # Создаем копию датафреймв
    encoders = {}

    # Список столбцов, которые нужно закодировать
    categorical_columns = [col for col in result.columns if result[col].dtype == object]

    for column in categorical_columns:
        result[column] = result[column].astype(str)  # Преобразуем значения в строки
        encoders[column] = preprocessing.LabelEncoder()  # Создаем кодировщик для столбца
        result[column] = encoders[column].fit_transform(result[column])  # Применяем кодировку

    return result, encoders  # Возвращаем закодированный датафрейм и словарь кодировщиков

# Применение функции ко всей копии датафрейма
df2, encoders = number_encode_features(df)
df2.head()
```

	location	floor	floors_count	rooms_count	total_meters	price	year_of_construction	object_type	have_loggia	parking_type	living_meters	kitchen_meters	ceiling_height	street	underground
3	26	5	5	1	17.50	2500000.0	1968	0	0.0	0	35.0	11.0	3.0	457	0
12	26	15	17	1	20.55	2600000.0	2025	3	1.0	2	134.0	3.0	3.0	137	0
18	26	2	2	1	31.30	2700000.0	1970	0	0.0	2	35.0	5.0	3.0	217	0
19	26	5	5	1	31.00	3090000.0	1984	0	1.0	2	15.0	73.0	3.0	42	0
30	26	7	24	3	77.60	12700000.0	2011	1	1.0	0	35.0	5.0	3.0	331	0

Рис. 7

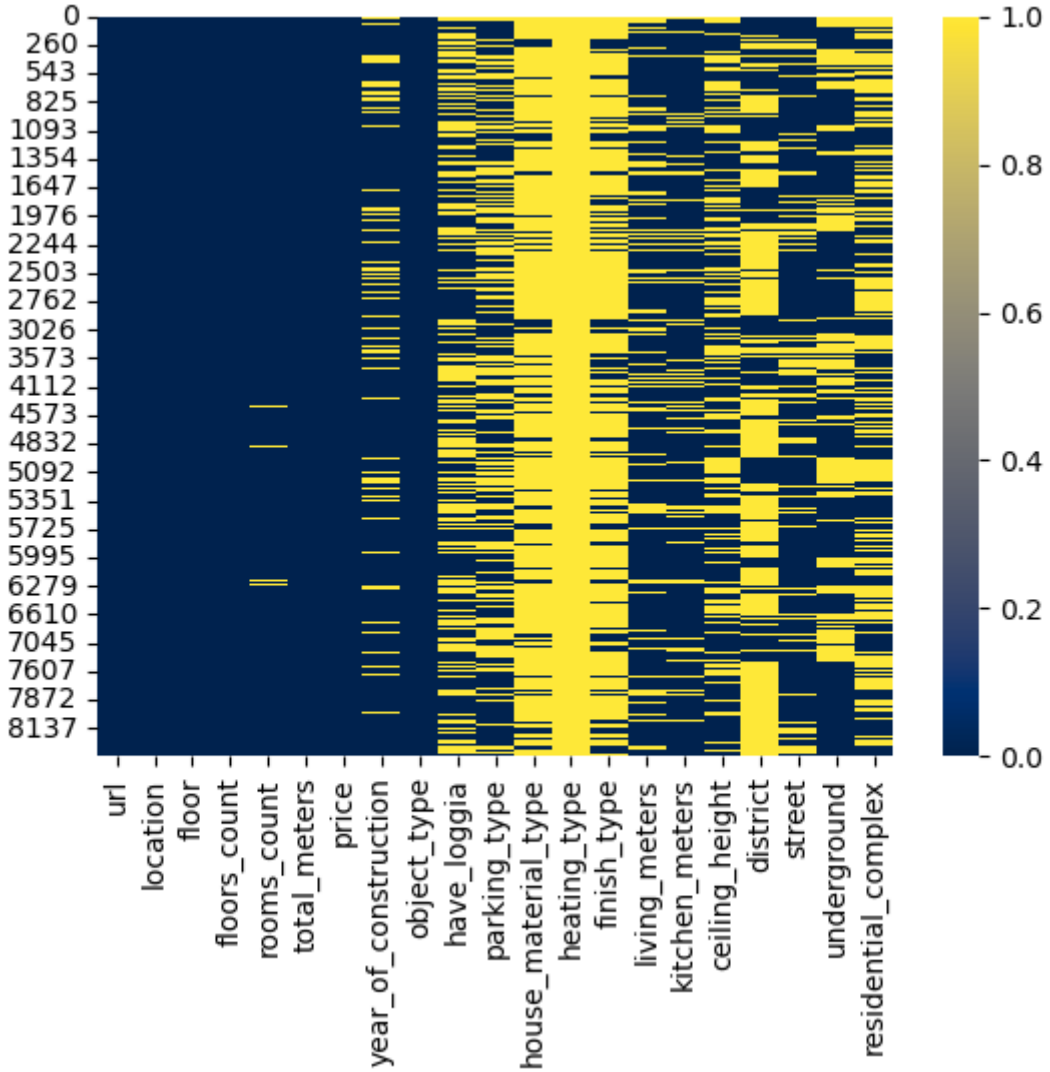
0
260
543
825
1093
1354
1647
1976
2244
2503
2762
3026
3573
4112
4573
4832
5092
5351
5725
5995
6279
6610
7045
7607
7872
8137
url
location
floor
floors_count
rooms_count
total_meters
price
year_of_construction
object_type
have_loggia
parking_type
house_material_type
heating_type
finish_type
living_meters
kitchen_meters
ceiling_height
district
street
underground
residential_complex
1.0
0.8
0.6
0.4
0.2
0.0

Рис. 8

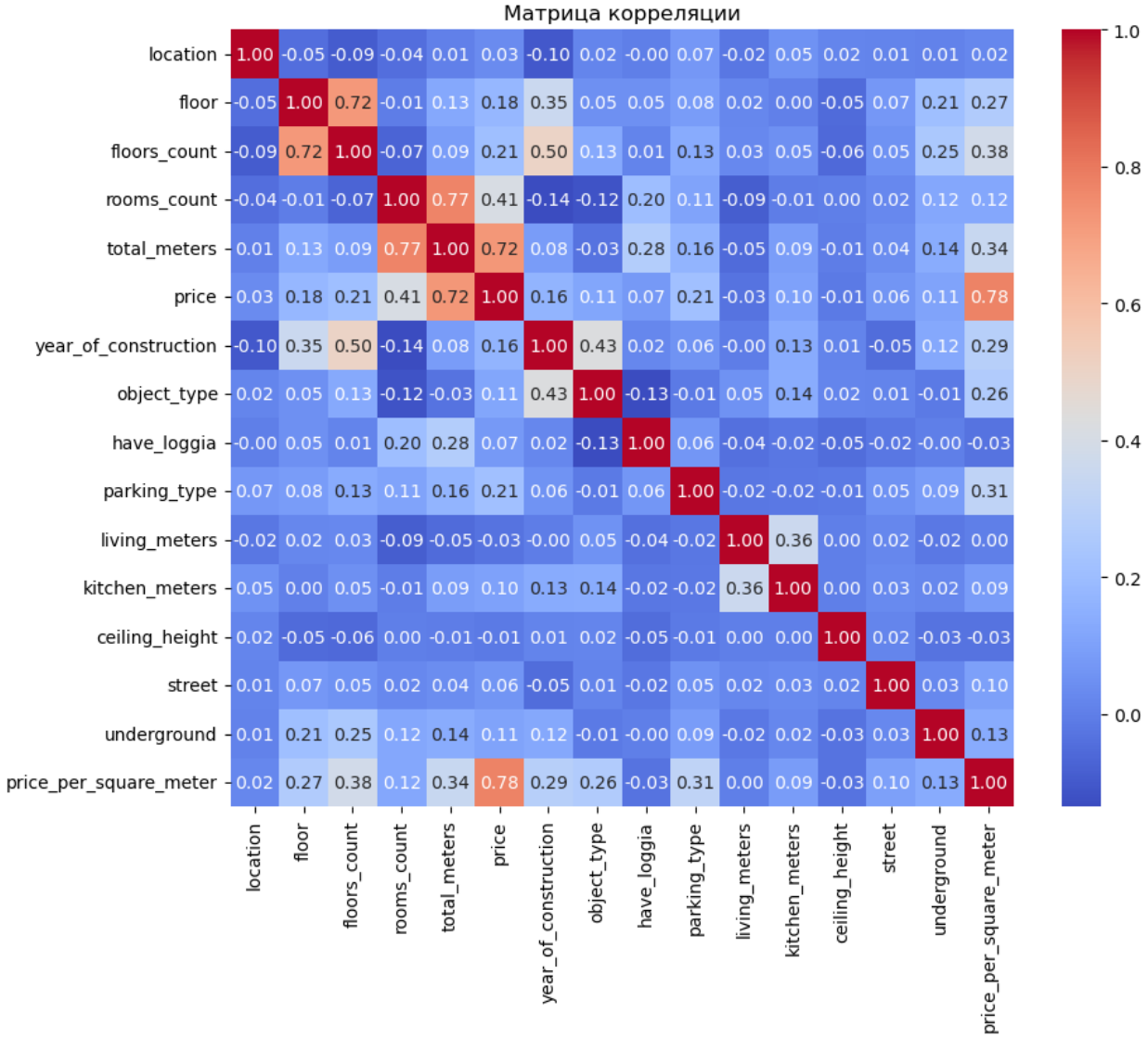
Матрица корреляции
location 1.00 -0.05 -0.09 -0.04 0.01 0.03 -0.10 0.02 -0.00 0.07 -0.02 0.05 0.02 0.01 0.01 0.02
floor -0.05 1.00 0.72 -0.01 0.13 0.18 0.35 0.05 0.05 0.08 0.02 0.00 -0.05 0.07 0.21 0.27
floors_count -0.09 0.72 1.00 -0.07 0.09 0.21 0.50 0.13 0.01 0.13 0.03 0.05 -0.06 0.05 0.25 0.38
rooms_count -0.04 -0.01 -0.07 1.00 0.77 0.41 -0.14 -0.12 0.20 0.11 -0.09 -0.01 0.00 0.02 0.12 0.12
total_meters 0.01 0.13 0.09 0.77 1.00 0.72 0.08 -0.03 0.28 0.16 -0.05 0.09 -0.01 0.04 0.14 0.34
price 0.03 0.18 0.21 0.41 0.72 1.00 0.16 0.11 0.07 0.21 -0.03 0.10 -0.01 0.06 0.11 0.78
year_of_construction -0.10 0.35 0.50 -0.14 0.08 0.16 1.00 0.43 0.02 0.06 -0.00 0.13 0.01 -0.05 0.12 0.29
object_type 0.02 0.05 0.13 -0.12 -0.03 0.11 0.43 1.00 -0.13 -0.01 0.05 0.14 0.02 0.01 -0.01 0.26
have_loggia -0.00 0.05 0.01 0.20 0.28 0.07 0.02 -0.13 1.00 0.06 -0.04 -0.02 -0.05 -0.02 -0.00 -0.03
parking_type 0.07 0.08 0.13 0.11 0.16 0.21 0.06 -0.01 0.06 1.00 -0.02 -0.02 -0.01 0.05 0.09 0.31
living_meters -0.02 0.02 0.03 -0.09 -0.05 -0.03 -0.00 0.05 -0.04 -0.02 1.00 0.36 0.00 0.02 -0.02 0.00
kitchen_meters 0.05 0.00 0.05 -0.01 0.09 0.10 0.13 0.14 -0.02 -0.02 0.36 1.00 0.00 0.03 0.02 0.09
ceiling_height 0.02 -0.05 -0.06 0.00 -0.01 -0.01 0.01 0.02 -0.05 -0.01 0.00 0.00 1.00 0.02 -0.03 -0.03
street 0.01 0.07 0.05 0.02 0.04 0.06 -0.05 0.01 -0.02 0.05 0.02 0.03 0.02 1.00 0.03 0.10
underground 0.01 0.21 0.25 0.12 0.14 0.11 0.12 -0.01 -0.00 0.09 -0.02 0.02 -0.03 0.03 1.00 0.13
price_per_square_meter 0.02 0.27 0.38 0.12 0.34 0.78 0.29 0.26 -0.03 0.31 0.00 0.09 -0.03 0.10 0.13 1.00
location
floor
floors_count
rooms_count
total_meters
price
year_of_construction
object_type
have_loggia
parking_type
living_meters
kitchen_meters
ceiling_height
street
underground
price_per_square_meter
1.0
0.8
0.6
0.4
0.2
0.0

Рис. 9

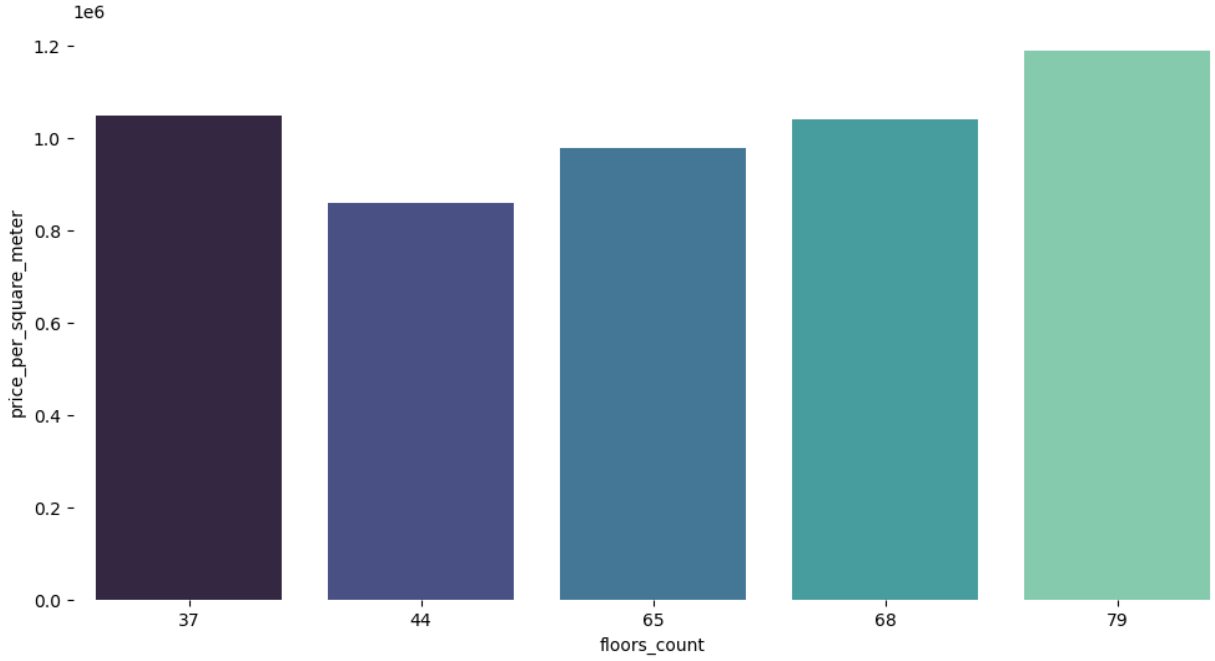
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
price_per_square_meter
37
44
65
68
79
floors_count

Рис. 10

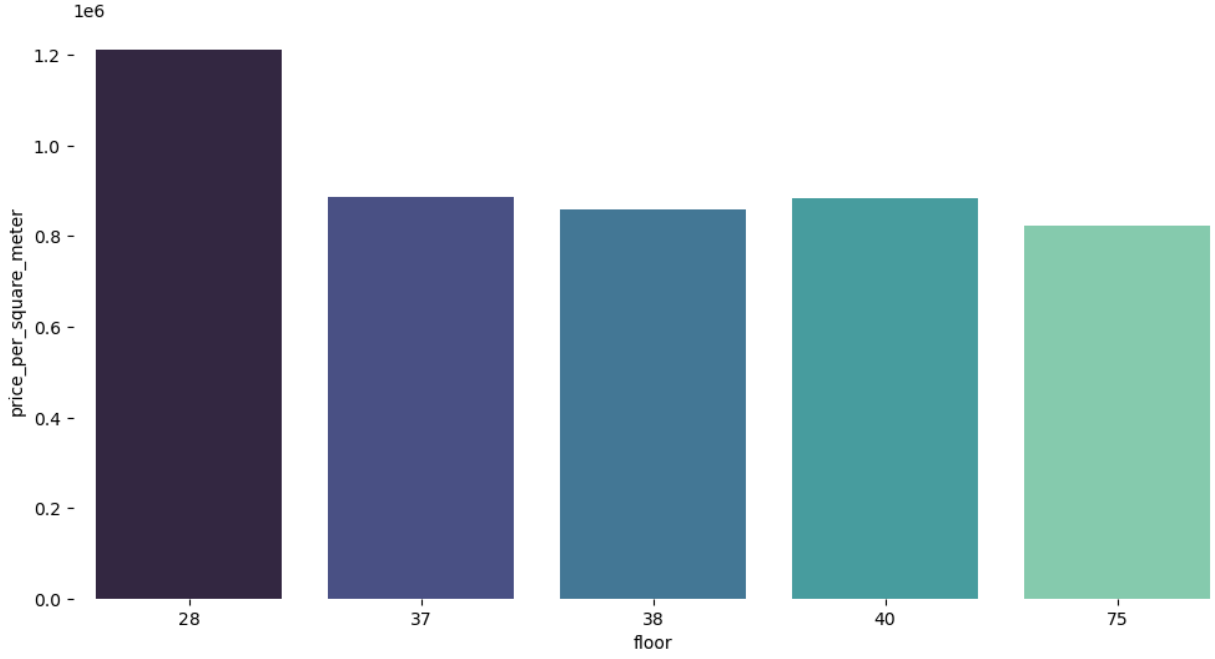
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
price_per_square_meter
28
37
38
40
75
floor

Рис. 11

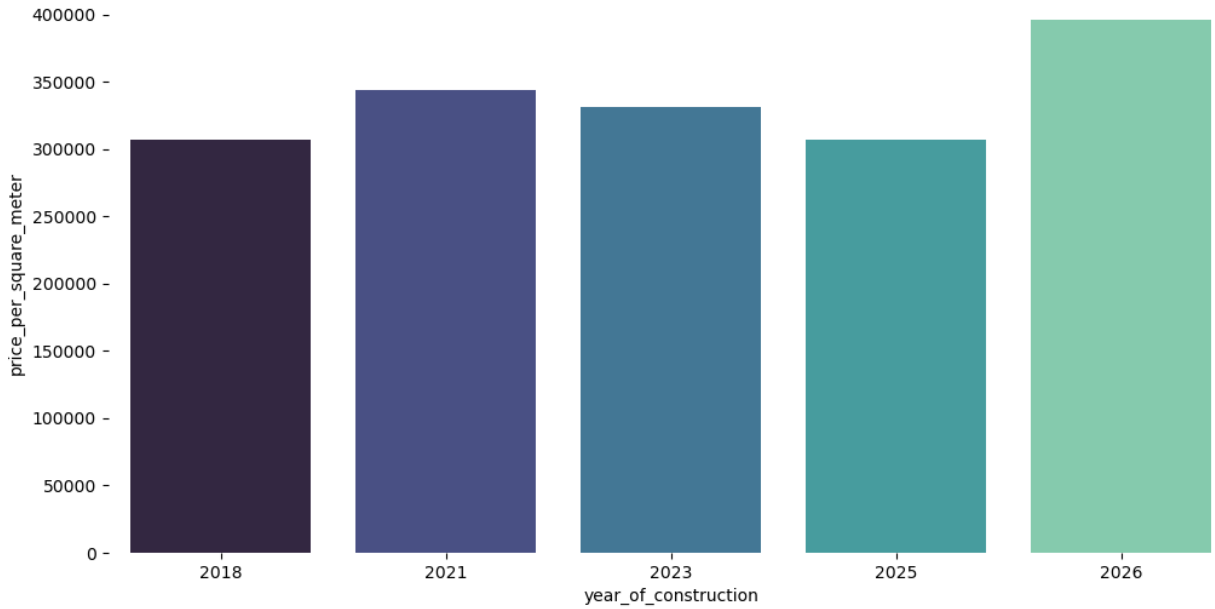
400000
350000
300000
250000
200000
150000
100000
50000
0
price_per_square_meter
2018
2021
2023
2025
2026
year_of_construction

Рис. 12

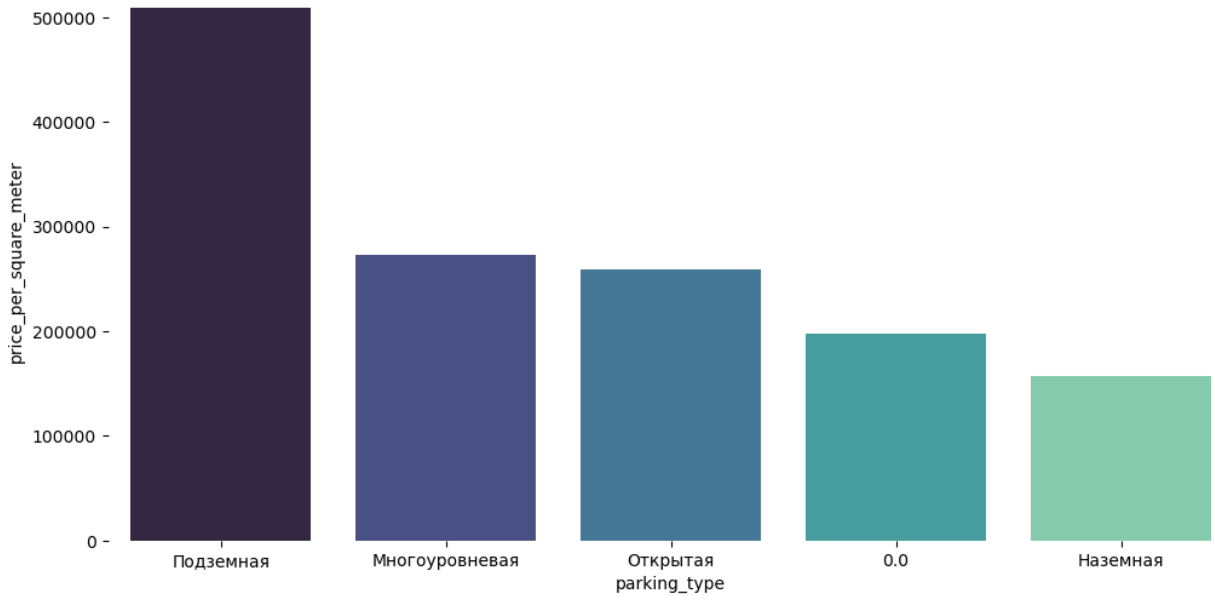
500000
400000
300000
200000
100000
0
price_per_square_meter
Подземная
Многоуровневая
Открытая
0.0
Наземная
parking_type

Рис. 13

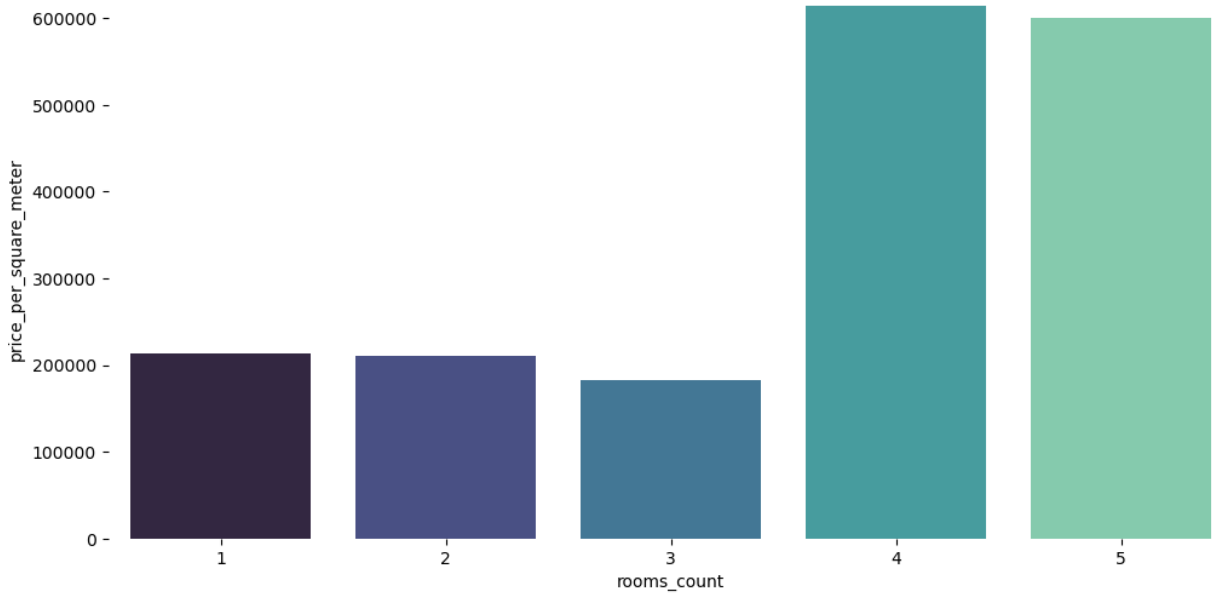
600000
500000
400000
300000
200000
100000
0
price_per_square_meter
1
2
3
4
5
rooms_count

Рис. 14