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

Отчёт по кейсу «Самолёт»:

Работу выполнил:
Студент группы № ИСП-21
Новасельский Артём

Долгопрудный, 2024

Введение

В данном отчёте рассматриваются выводы, полученные после анализа данных в области «Квартиры в Москве и Московской области».

Цель

Собрать данные и проанализировать их для будущего использования, например, обучение машины на основе выводов.

Задачи

- Собрать данные с помощью данных инструментов.
- Совершить работу над ними, а точнее удаление ненужных данных, дополнение необходимых и т.д.
- Визуализация данных. Нахождение взаимосвязи между данными или её полное отсутствие для отчёта.

Основная часть

Существует небольшой выбор источников, для сбора данных, нам был предложен интернет-ресурс «Циан». С помощью языка Python и библиотеки “CianParser” было собрано свыше десяти тысяч объявлений в нужных регионах.

После сбора всей информации воедино и уборки дубликатов, можно посмотреть, какого типа наши данные (рис.1).

Теперь мы смотрим, какие данные у нас не смогли собраться(рис.2) при помощи библиотеки seaborn и функции heatmap. Как мы видим, например, колонка residential_complex сильно пустеет, что означает мы вынуждены их удалить, так как строить анализ будет невозможно.

Далее фильтруем ненужные данные и форматируем столбцы, чтобы их было можно анализировать(рис.3). После объёмной чистки данных, нужно проверить их состояние – смотрим внутрь файла и бегло проверяем на аномалии, в случае их отсутствия приступаем к кодовой проверке данных.

После полной очистки данных, вручную и программно, можно сохранить очищенную базу данных, а после, приступать к постройке графиков и аналитической работе при помощи библиотеки matplotlib для вывода графических изображений. Например, будет 5 графиков:

1. Цена за м^2 в зависимости от этажа, на котором квартира.
2. Цена за м^2 по району.
3. Цена за м^2 по городу.
4. Цена за м^2 в зависимости от года постройки самой квартиры.
5. Количество объявлений по количеству комнат.

(рис.4-6)

Ещё можно вывести корреляционную матрицу, или матрицу корреляций, которая напрямую показывает зависимость значений друг от друга(рис 7).

В итоге получаются графики(рис.10), на основе которых уже можно проводить анализ.

Аналитика данных

Смотря на графики, делаем выводы, что цена в основном зависит от количества комнат и метража. Меньше всего на цену влияют год постройки и количество этажей.

Заключение

В результате работы были собраны около одиннадцати тысяч квартир, отсортированы, почищены данные, после чего, на удивление осталось почти десять тысяч, простроены облегчающие анализ графики, которые помогли понять главные критерии в оценивании стоимости недвижимости в Москве и Московской Области. Основными факторами, влияющие на стоимость, выявились метраж и количество комнат.

В теории можно было бы и закодировать категориальные данные, на что мне не хватило времени, знаний и сил, и тщательней прочесать данные, ибо тот факт, что в годе постройки были поля “Напишите Автору” и “Аукцион” показывает качество данных в негативном ключе, и скорее всего было что-то пропущено, ещё, наверно, можно было бы и использовать второй инструмент для анализа, что б точно закрепить выводы о данных.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10460 entries, 0 to 10459
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   author                10289 non-null  object
 1   author_type           10286 non-null  object
 2   url                   10456 non-null  object
 3   location              10057 non-null  object
 4   deal_type             10459 non-null  object
 5   accommodation_type    10459 non-null  object
 6   floor                 10459 non-null  float64
 7   floors_count          10459 non-null  float64
 8   rooms_count           10459 non-null  float64
 9   total_meters          10459 non-null  float64
 10  price                 10426 non-null  float64
 11  year_of_construction  10459 non-null  float64
 12  object_type           10456 non-null  float64
 13  house_material_type   10456 non-null  object
 14  heating_type          10456 non-null  float64
 15  finish_type           10456 non-null  object
 16  living_meters         10458 non-null  object
 17  kitchen_meters        10458 non-null  object
 18  phone                 10456 non-null  float64
 19  district              5900 non-null   object
 20  street                8981 non-null   object
 21  house_number          9366 non-null   object
 22  underground           6634 non-null   object
 23  residential_complex   4733 non-null   object
dtypes: float64(9), object(15)
```

рис.1

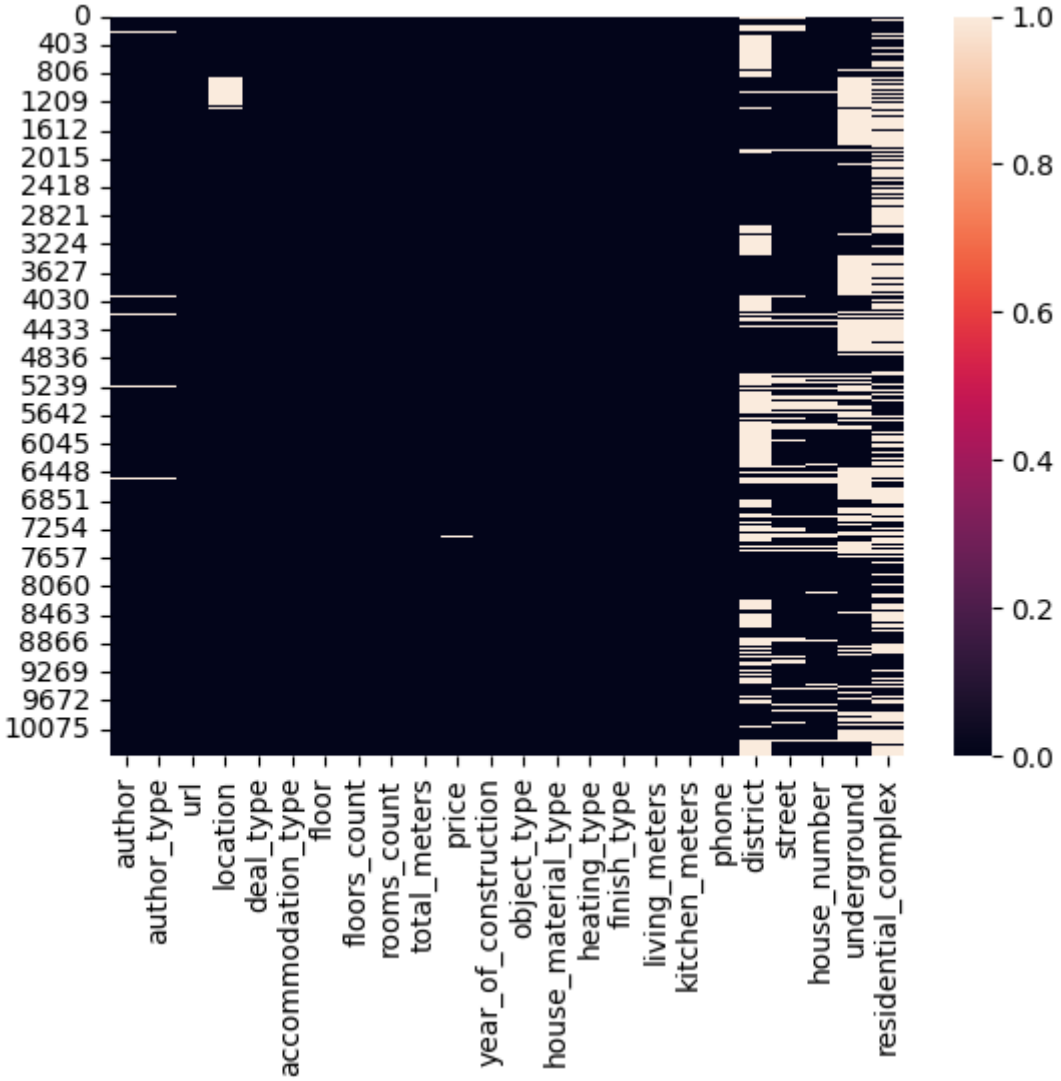
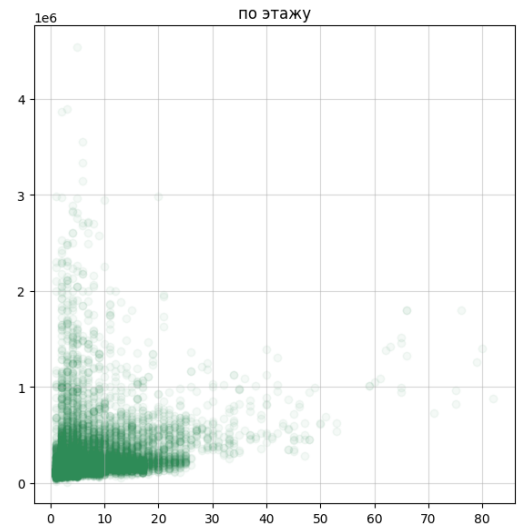
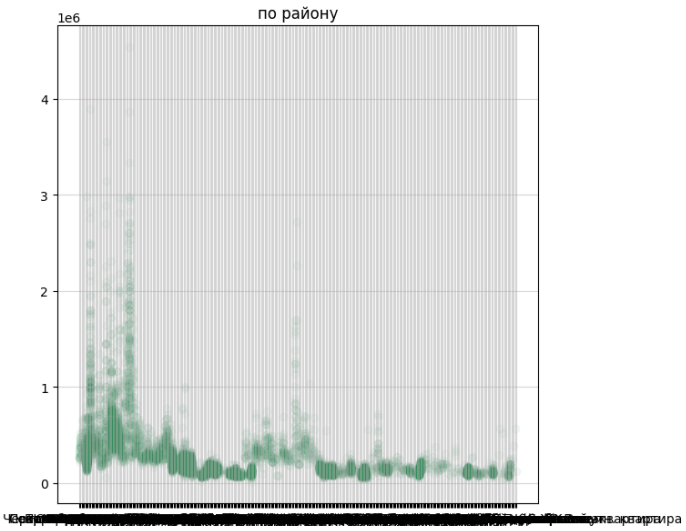

0
403
806
1209
1612
2015
2418
2821
3224
3627
4030
4433
4836
5239
5642
6045
6448
6851
7254
7657
8060
8463
8866
9269
9672
10075
1.0
0.8
0.6
0.4
0.2
0.0
author
author_type
url
location
deal_type
accommodation_type
floor
floors_count
rooms_count
total_meters
price
year_of_construction
object_type
house_material_type
heating_type
finish_type
living_meters
kitchen_meters
phone
district
street
house_number
underground
residential_complex

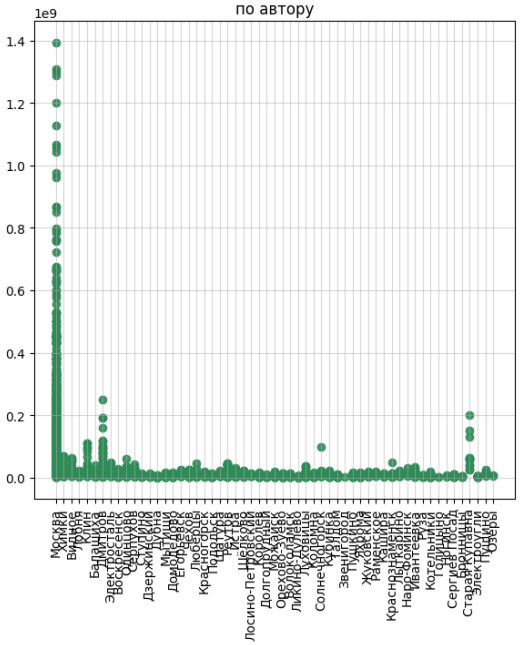
рис.2

```
print(df["deal_type"].value_counts(),           #везде sale
      df["accommodation_type"].value_counts(), #везде flat
      df["object_type"].value_counts(),         #везде -1
      df["house_material_type"].value_counts(),#много пропусков, заменить нечем
      df["finish_type"].value_counts(),         #много пропусков, заменить нечем
      df["street"].value_counts(),              #много пропусков, заменить нечем
      df["house_number"].value_counts(),        #много пропусков, заменить нечем
      df["heating_type"].value_counts())        #везде -1
useless_columns=["author","author_type","url","deal_type",
                 "accommodation_type","object_type",
                 "house_material_type","finish_type","phone",
                 "heating_type","street","house_number",
                 "residential_complex"]
df=df.drop(columns=useless_columns).drop_duplicates()
Run Cell | Run Above | Debug Cell
#%%заполняем или удаляем пропуски
df.dropna(subset=["location","price"],inplace=True)
df.loc[df["district"].isna(),"district"]=df["location"]
df.loc[df["underground"].isna(),"underground"]=df["location"]
df.loc[df["living_meters"]=="-1","living_meters"]=df["total_meters"]
df.loc[df["living_meters"].isna(),"living_meters"]=df["total_meters"]
df.loc[df["kitchen_meters"]=="-1","kitchen_meters"]=0
df.loc[df["kitchen_meters"].isna(),"kitchen_meters"]=0
df.replace(["-1",-1,"-1.0",-1.0],df['year_of_construction'].median(),inplace=True)
df=df[df['rooms_count']!=2006.0]
Run Cell | Run Above | Debug Cell
#%%приводим в порядок числа
df["living_meters"]=df["living_meters"].str.replace("\xa0м²","").str.replace(",",".").astype(float)
df["kitchen_meters"]=df["kitchen_meters"].str.replace("\xa0м²","").str.replace(",",".").astype(float)
```

рис.3


по этажу
1e6
0
1
2
3
4
0
10
20
30
40
50
60
70
80


по району
1e6
0
1
2
3
4


по автору
1e9
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4

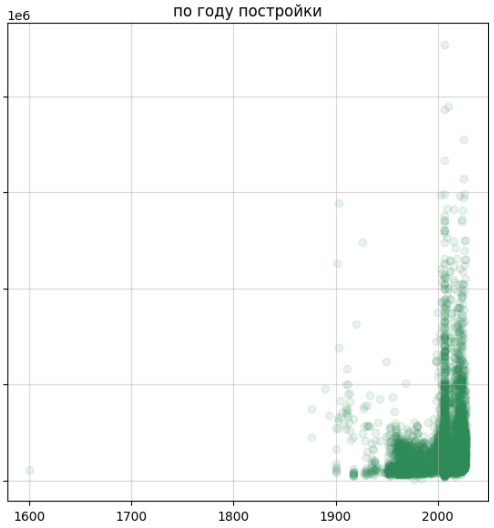

по году постройки
1e6
0
1
2
3
4
1600
1700
1800
1900
2000

рис.4

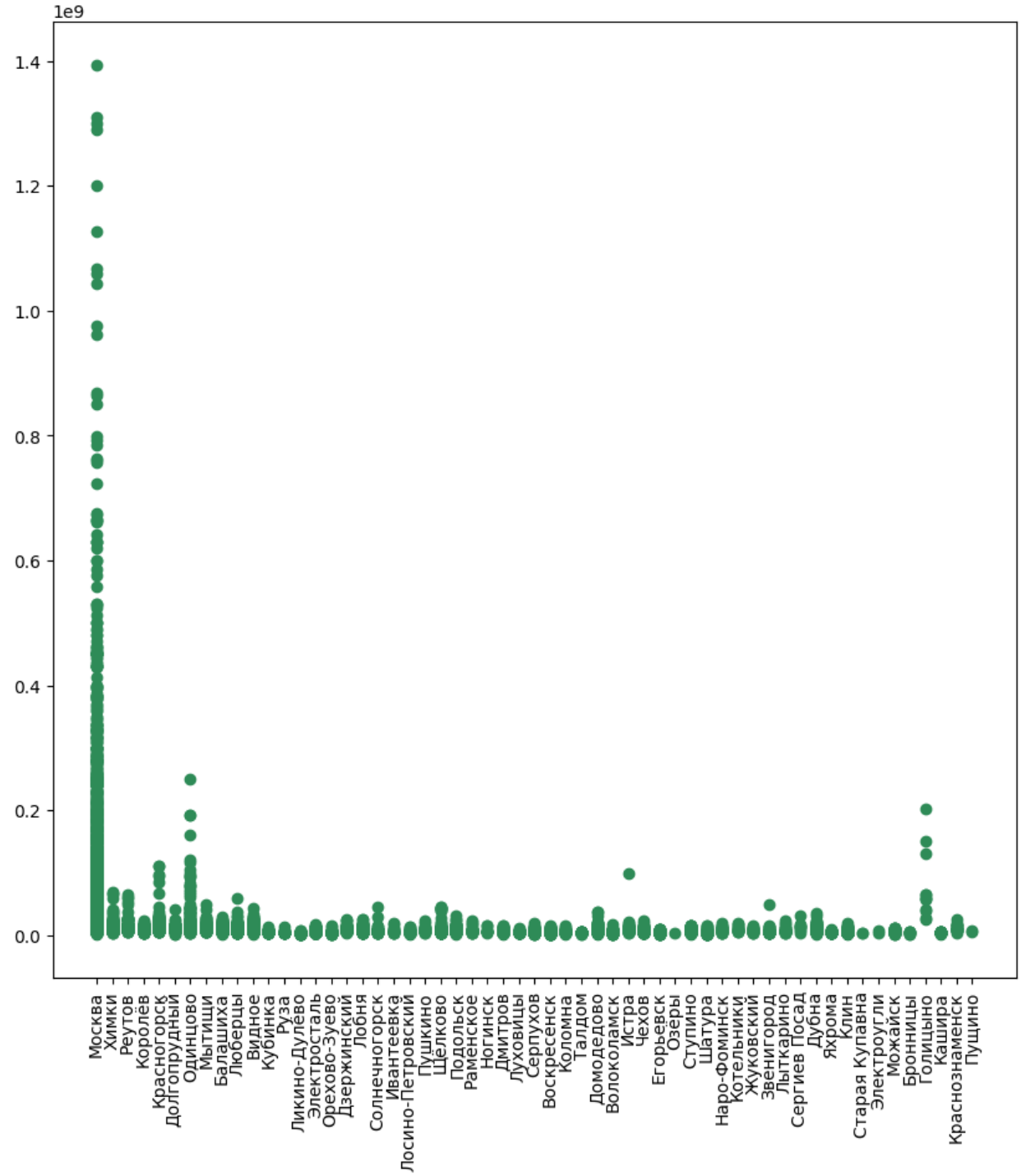

1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Москва
Химки
Реутов
Королёв
Красногорск
Долгопрудный
Одинцово
Мытищи
Балашиха
Люберцы
Видное
Кубинка
Руза
Ликино-Дулёво
Электросталь
Орехово-Зуево
Дзержинский
Лобня
Солнечногорск
Ивантеевка
Лосино-Петровский
Пушкино
Щёлково
Подольск
Раменское
Ногинск
Дмитров
Луховицы
Серпухов
Воскресенск
Коломна
Талдом
Домодедово
Волоколамск
Истра
Чехов
Егорьевск
Озёры
Ступино
Шатура
Наро-Фоминск
Котельники
Жуковский
Звенигород
Лыткарино
Сергиев Посад
Дубна
Яхрома
Клин
Старая Купавна
Электроугли
Можайск
Бронницы
Голицыно
Кашира
Краснознаменск
Пущино

рис.5

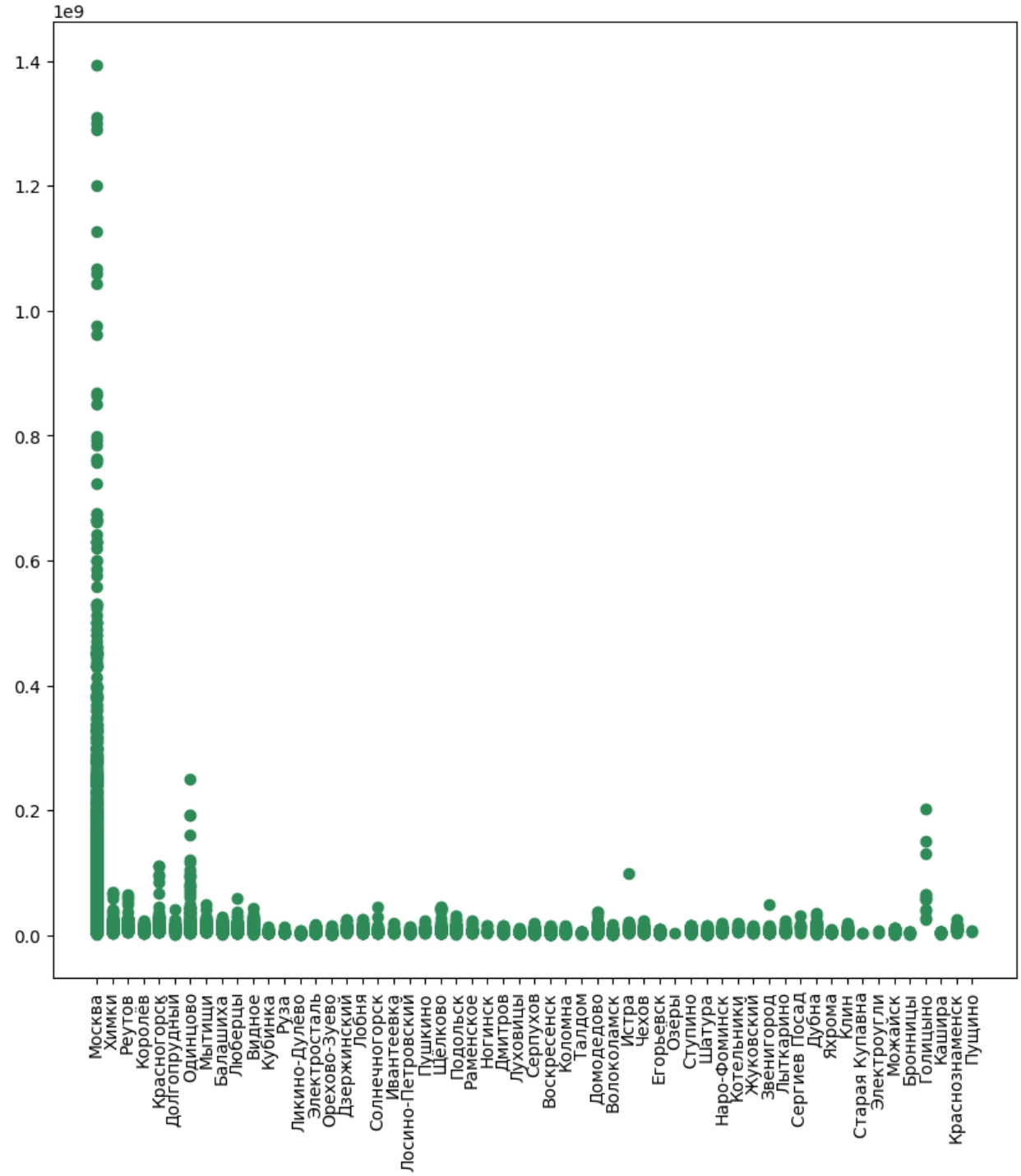

1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Москва
Химки
Реутов
Королёв
Красногорск
Долгопрудный
Одинцово
Мытищи
Балашиха
Люберцы
Видное
Кубинка
Руза
Ликино-Дулёво
Электросталь
Орехово-Зуево
Дзержинский
Лобня
Солнечногорск
Ивантеевка
Лосино-Петровский
Пушкино
Щёлково
Подольск
Раменское
Ногинск
Дмитров
Луховицы
Серпухов
Воскресенск
Коломна
Талдом
Домодедово
Волоколамск
Истра
Чехов
Егорьевск
Озёры
Ступино
Шатура
Наро-Фоминск
Котельники
Жуковский
Звенигород
Лыткарино
Сергиев Посад
Дубна
Яхрома
Клин
Старая Купавна
Электроугли
Можайск
Бронницы
Голицыно
Кашира
Краснознаменск
Пущино

рис.6

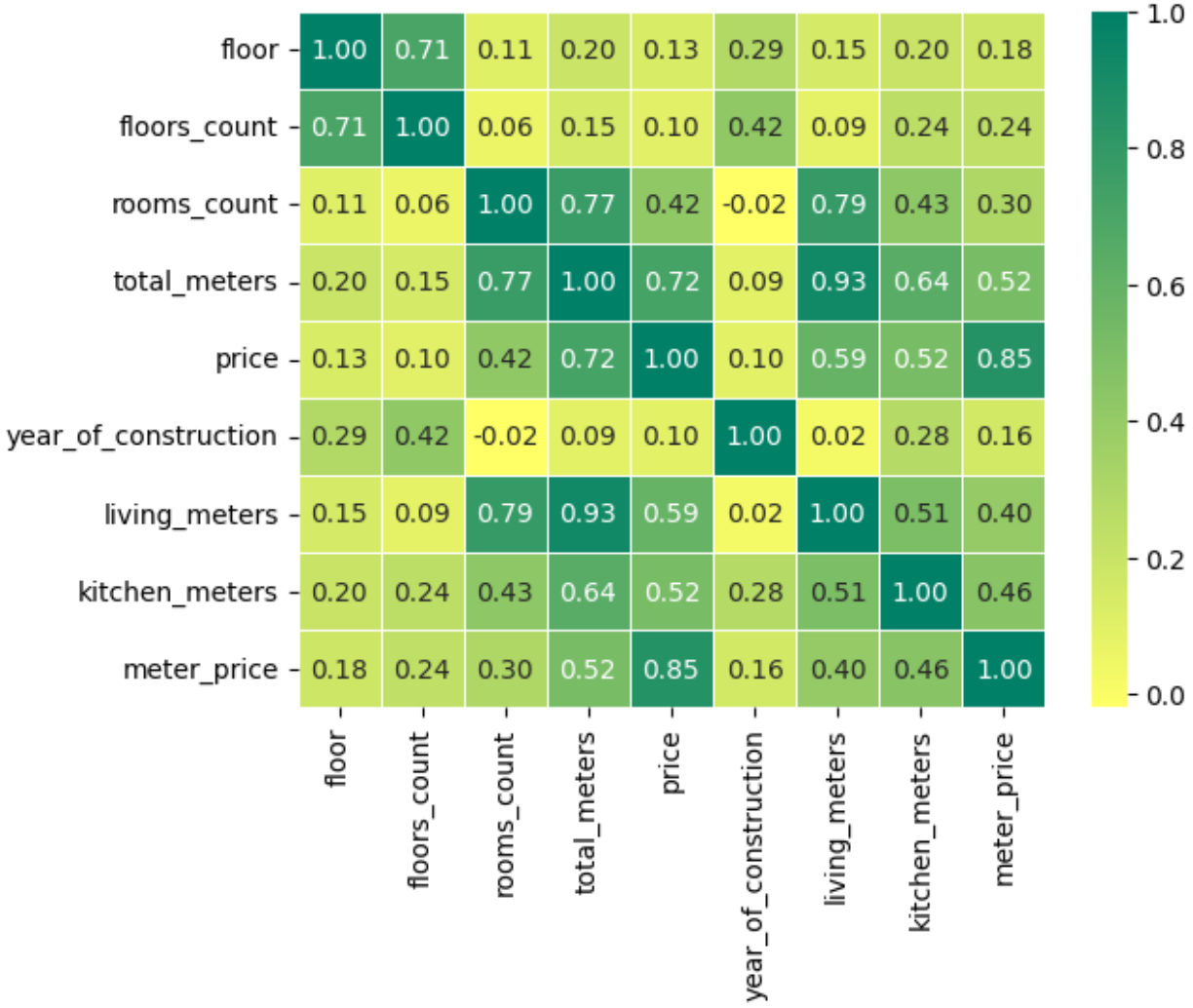

floor 1.00 0.71 0.11 0.20 0.13 0.29 0.15 0.20 0.18
floors_count 0.71 1.00 0.06 0.15 0.10 0.42 0.09 0.24 0.24
rooms_count 0.11 0.06 1.00 0.77 0.42 -0.02 0.79 0.43 0.30
total_meters 0.20 0.15 0.77 1.00 0.72 0.09 0.93 0.64 0.52
price 0.13 0.10 0.42 0.72 1.00 0.10 0.59 0.52 0.85
year_of_construction 0.29 0.42 -0.02 0.09 0.10 1.00 0.02 0.28 0.16
living_meters 0.15 0.09 0.79 0.93 0.59 0.02 1.00 0.51 0.40
kitchen_meters 0.20 0.24 0.43 0.64 0.52 0.28 0.51 1.00 0.46
meter_price 0.18 0.24 0.30 0.52 0.85 0.16 0.40 0.46 1.00
floor
floors_count
rooms_count
total_meters
price
year_of_construction
living_meters
kitchen_meters
meter_price
1.0
0.8
0.6
0.4
0.2
0.0

рис.7