Государственное бюджетное профессиональное образовательное учреждение Московской области

«Физико-технический колледж

Отчет по интенсиву №1

Работу выполнил:

Студент группы ИСП-22

Салимов Динислам

Преподаватель Базяк Г.В.

**Введение**

Цены на квартиру – это одно из самых не предсказуемых событий. Из-за постоянной инфляции, изменений цен, изменений заработных плат, изменений гео-данных нельзя точно предугадать как изменяться цены даже на ближайший год. Но благодаря науке Data Science можно предугадать даже такие данные. В этой работе я рассмотрю работу парсера и проведу анализ данных, для дальнейшей работы с этими данными. Парсер будет обхватывать Москву и Московскую область.

**Цель:** Цель данной работы спросить как можно больше данных, а затем провести чтение данных, удаление NaN значений, удаление выбросов и анализ данных.

**Задачи:**

1. Парсинг данных, для дальнейшей обработки.
2. Подготовка данных к дальнейшей работе: чтение, очистка и анализ.
3. Исследовательский анализ данных (EDA): Мы выполним исследовательский анализ данных, включающий построение распределения основных параметров, визуализацию взаимосвязей между ними, а также определение признаков, оказывающих наиболее сильное влияние на целевую переменную.

**Используемые инструменты:**

1. Язык программирования Python.
2. Библиотека для парсинга cianparser.
3. Google colab.
4. Браузер Chrome.
5. Библиотеки для очистки и анализа данных: numpy, pandas, seaborn, matplotlib.

**Парсинг данных**

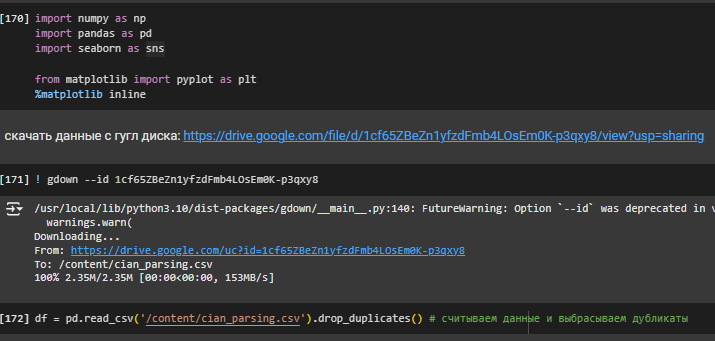
Для парсинга данных я использовал библиотеку cianparser. Это довольно простой и быстрый парсер, который может легко считывать данные с циана. В общей сложности на парсинг у меня ушло дней 5. И было спаршено около 8000 с чем-то данных.

**Основная часть**

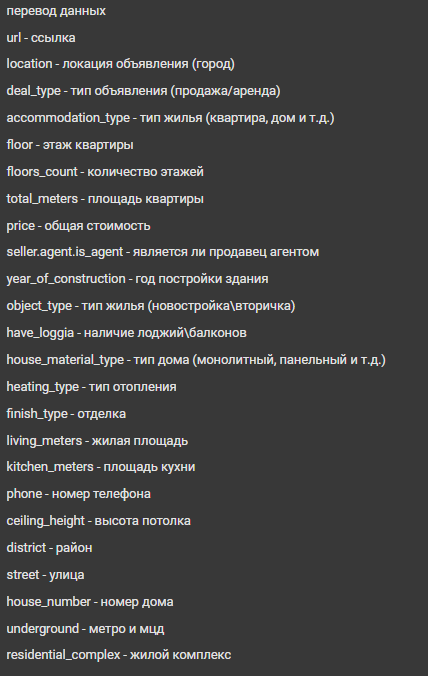
1. **Загрузка данных и библиотек**

Первым делом для работы с нашими данными нужно их загрузить в colab, я это сделал через гугл диск, благодаря этому каждый желающий при просмотре моего кода может скачать мои данные и сам проверить каждую строку.

Далее я скачал все нужные мне библиотеки. (см. фото снизу)

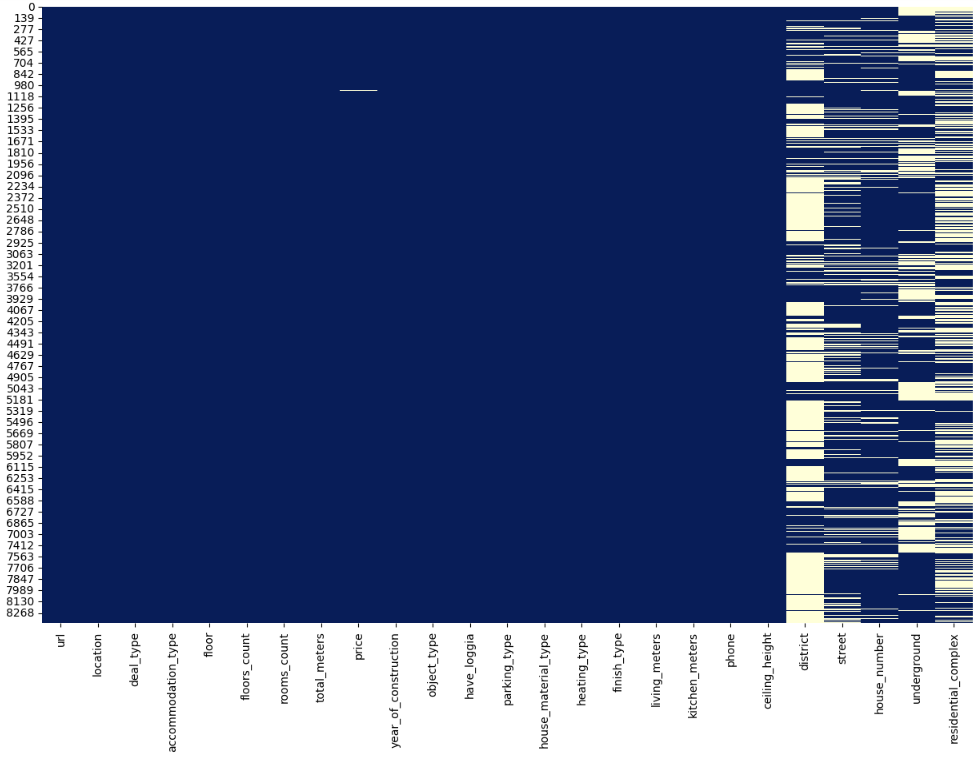


Для удобства чтения я сделал перевод всех столбцов в моём датафрейме. (см. фото снизу)

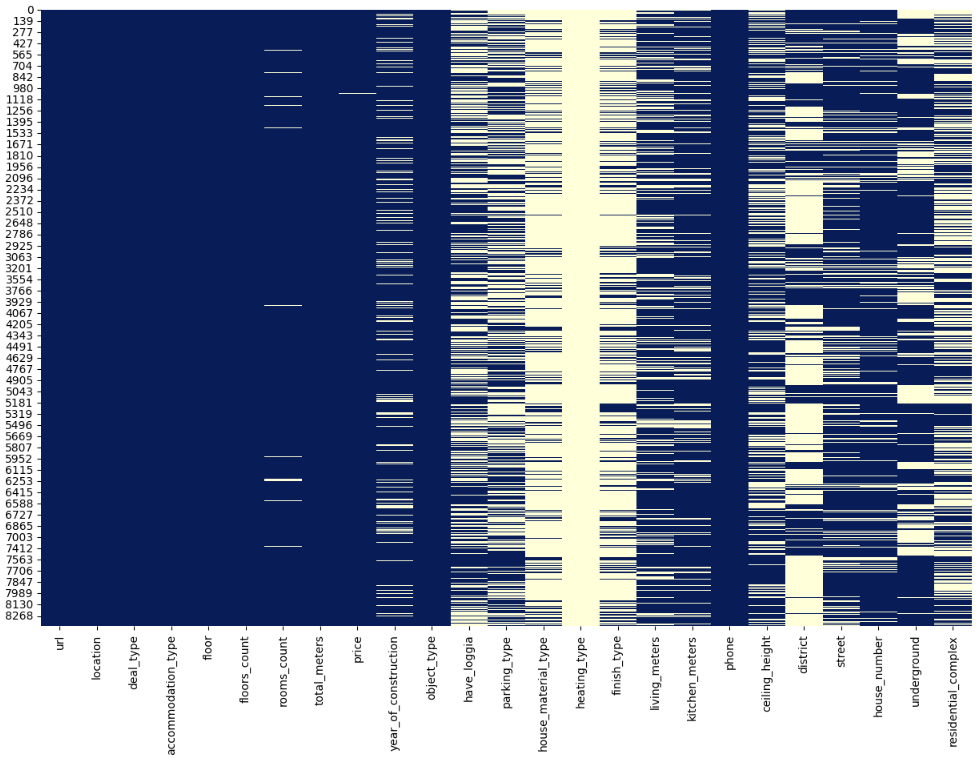


1. **Проверка на выбросы и NaN значения**

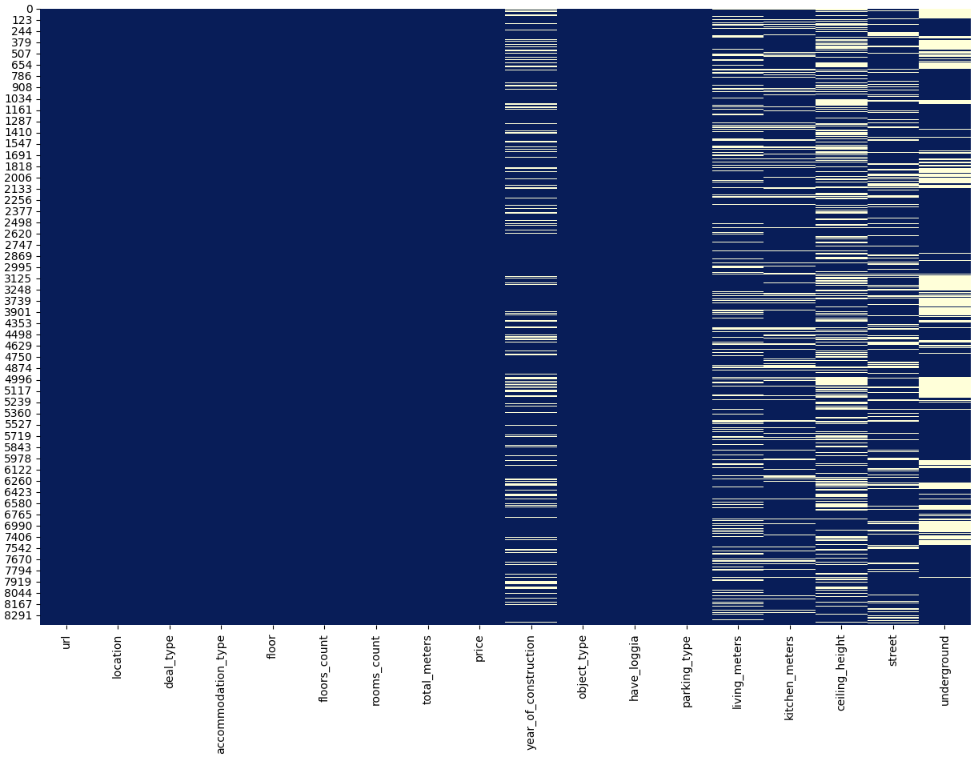
Первым делом удалил столбцы автор и тип автора, так как они нам не нужны. Далее я вывел тепловую карту для просмотра всех пропусков. (см. фото снизу)

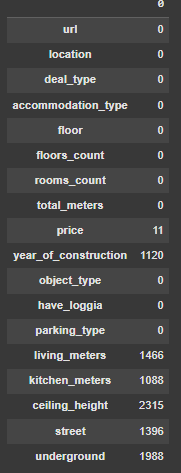


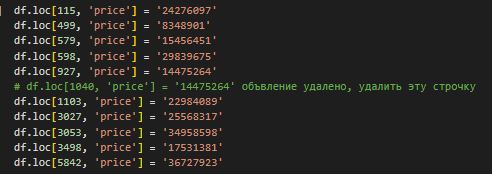
Замечаю, что значений с ‘-1’ очень много, поэтому решаю их всех заменить на NaN. После этого тепловая карта стала выглядить так. (см. фото снизу)



Видим, что у нас появилось очень много пропусков и с ними нужно работать. Первым делом удаляем heating\_type, finish\_type, house\_material\_type и district.

В столбцах have\_loggia и parking\_type заменяем пустые данные на 0.(тепловая карта см. фото снизу)

Далее выводим все пропуски в данных и понимаем, что в price всего 11 пропусков, который можно заменить в ручную через url. 

Меняем их таким образом. (см. фото снизу)

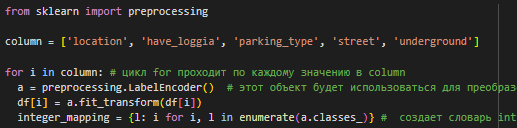
В столбце year\_of\_construction видно 3 неправильных значения: 'Аукцион', 'Напишите автору', 'Позвоните автору'. Удаляем все строки, которые содержат данную информацию. При помощи таблицы понимаем, что в столбце year\_of\_construction есть выбросы, избовляем от них и пустым значениям присваеваем медиану. (код см. фото снизу)

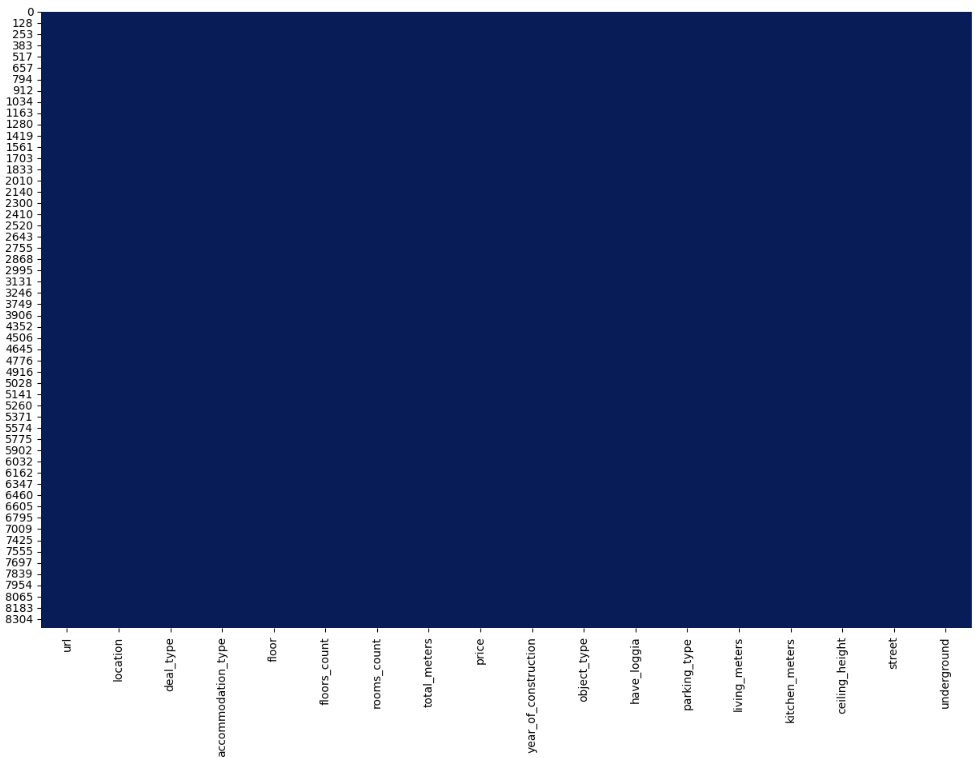


Не забываем привести наши значения к числовым, для этого используем код под каждый столбец с и так числовыми данными, просто помеченными как object. (код см. фото снизу)



Таким образом приводим к числовым данным наши значения. Для данных, который изначально не числовые используется другой метод, но его лучше использовать в конце после очистки данных. (см. фото снизу)

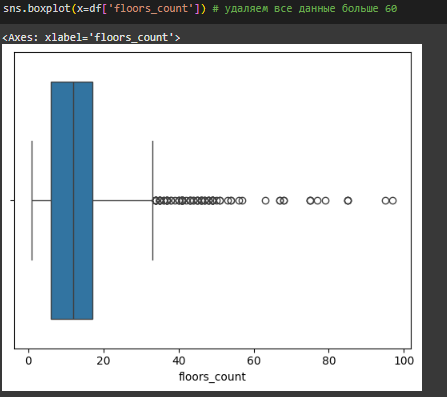


После всей очистки и перевода данных в числовые у вас должна получится такая тепловая карта. (см. фото снизу)

1. **Выбросы**

Сперва смотрим на данные количества этажей в общем и через

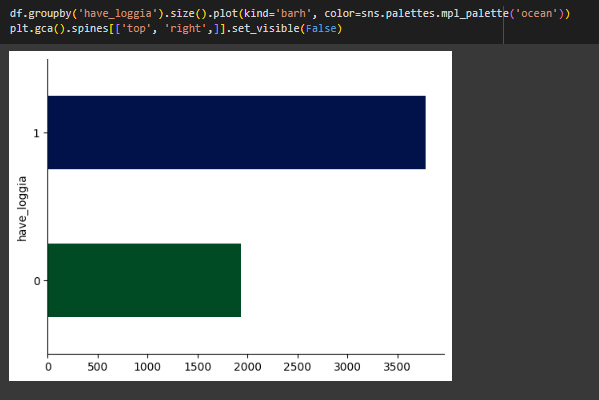
метод коробки с усами, видим, что есть выбросы, удаляем их. (см. фото снизу)



Также поступаем с total\_meters, в остальных данных выбросов у нас нет. Это проверяется через df['total\_meters'].describe().

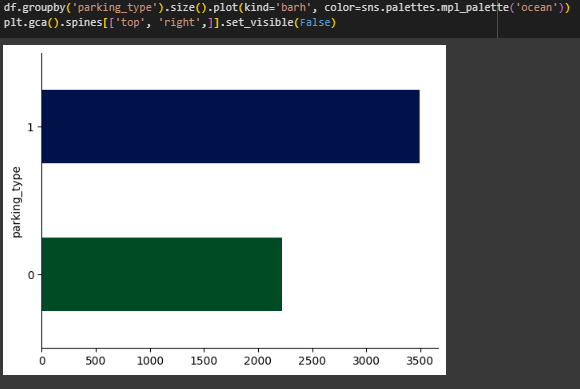
1. **Анализ данных**

Выводим количество квартир с лоджиями и балконами, и без

них, смотрим на соотношения. (см. фото снизу)

По графику видно, что количество квартир с лоджиями или балконами больше, чем без них.

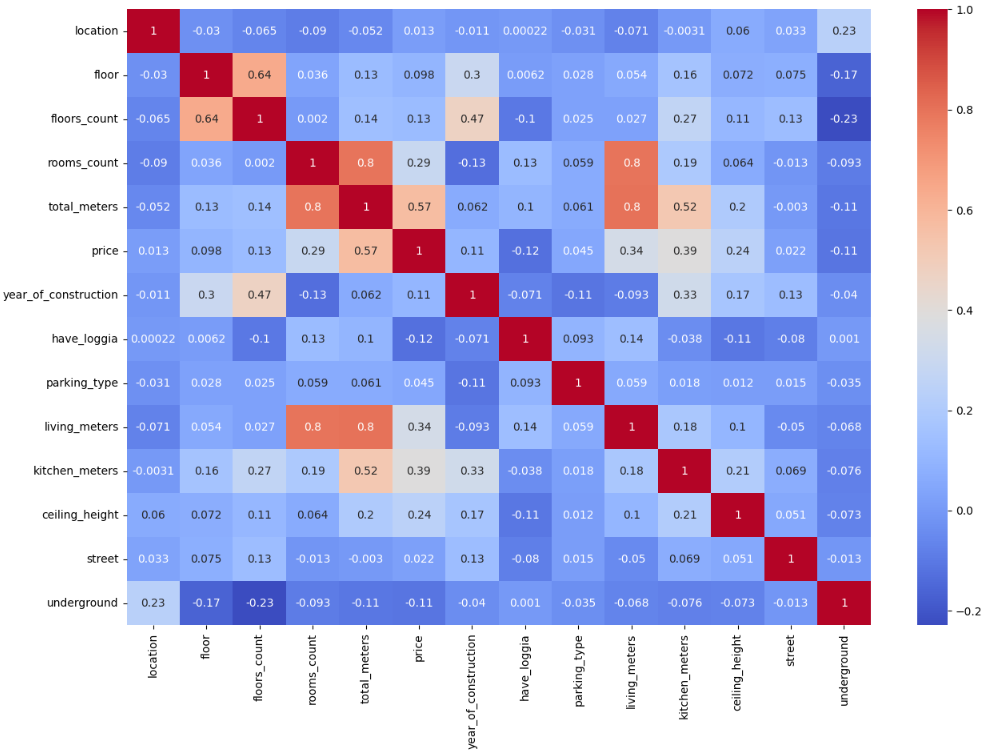
Также смотрим и анализируем квартиры с парковками и без. (см. фото снизу)



По графику видно, что квартир имеющих рядом парковки(наземные, подземные и тд) больше, чем других.

Незабываем удалять бесполезные данные, например я на этом этапе удалил url, deal\_type, accommodation\_type, object\_type.

Далее выводим матрицу корреляции. (см. фото снизу)



По матрице корреляции можно заметить, что столбец количество комнат и площадь квартиры сильно коррелируют с жилой площадью. Также количество комнат и площадь квартиры коррелируют между собой.

С ценой коррелирует площадь дома, это означает, что чем больше площадь дома, тем дороже квартира и наоборот.

Более или менее с чем коррелирует цена - это living\_meters, kitchen\_meters и total\_meters. Из этого приходят выводы, что самую большую роль на цену влияет площадь. А самую меньшую значимость имеет количество этажей, метро и мцд.

**Заключение**

После очистки и аналза данных можно сказать, что цена квартир, довольно сильно и хорошо коррелирует с другими данными, поэтому в будущем будет очень удобно создать модель на основе этих данных.

Матрица корреляции сыграла огромную роль в анализе и обработке данных. Не менее важным оказалась функция яшика с усами, благодаря которому легко находить выбросы.

**Список литературы**

1. <https://www.dmitrymakarov.ru/python/files-04/#14-issledovatelskiy-analiz-dannykh-eda>
2. <https://habr.com/ru/companies/ruvds/articles/494720/>
3. <https://proglib.io/p/moem-dataset-rukovodstvo-po-ochistke-dannyh-v-python-2020-03-27>
4. <https://www.dmitrymakarov.ru/data-analysis/outliers-09/#6-boxplot>
5. <https://sky.pro/wiki/python/udalenie-dublikatov-v-pandas-data-frame-po-vybrannym-kolonkam/>
6. <https://www.excelguide.ru/2021/12/find-nan-dataframe.html>
7. <https://www.cian.ru/>
8. <https://ru.piliapp.com/symbols/squared/>
9. <https://pythonru.com/biblioteki/seaborn-plot>
10. <https://seaborn.pydata.org/tutorial/color_palettes.html>