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

Исследовательский аналитический отчёт:

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

Работу выполнил:
студент группы ИСП - 21
Ермаков Григорий
Проверил:
преподаватель
Базяк Г.В.

Долгопрудный, 2024

Введение

Оценка недвижимости - важная составляющая девелоперского бизнеса. Информация, о реальной цене квартиры исходя из рынка, интересна для покупателей продавцов, застройщиков, агентов и др.

Целью анализа является: собрать данные и провести разведочный исследовательский анализ данных (EDA) для построения модели, которая будет оценивать цену квадратного метра недвижимости в Московском регионе (Москва, Новая Москва, Московская область).

Задачи:

1. Определить параметры, влияющие на цену квадратного метра жилья
2. Собрать данные о продающихся квартирах
3. Подготовить данные для анализа

Для сбора данных использовалась библиотека cianparser, с её помощью удалось собрать данные более чем о 11000 квартир.

```
import cianparser
from time import sleep


a = 0
while a  52:
    moscow_parser = cianparser.CianParser(location="Москва")
    data = moscow_parser.get_flats(deal_type="sale", rooms=(3), with_saving_csv=True,with_extra_data=True, additional_settings = {
        "start_page":1 + a,
       "end_page": 2 + a,
})
sleep(40)
a += 2
```

После парсинга переходим к работе с данными, сначала их нужно подготовить, удалить лишние столбцы, избавиться от неверных значений, заполнить или удалить недостающие данные.

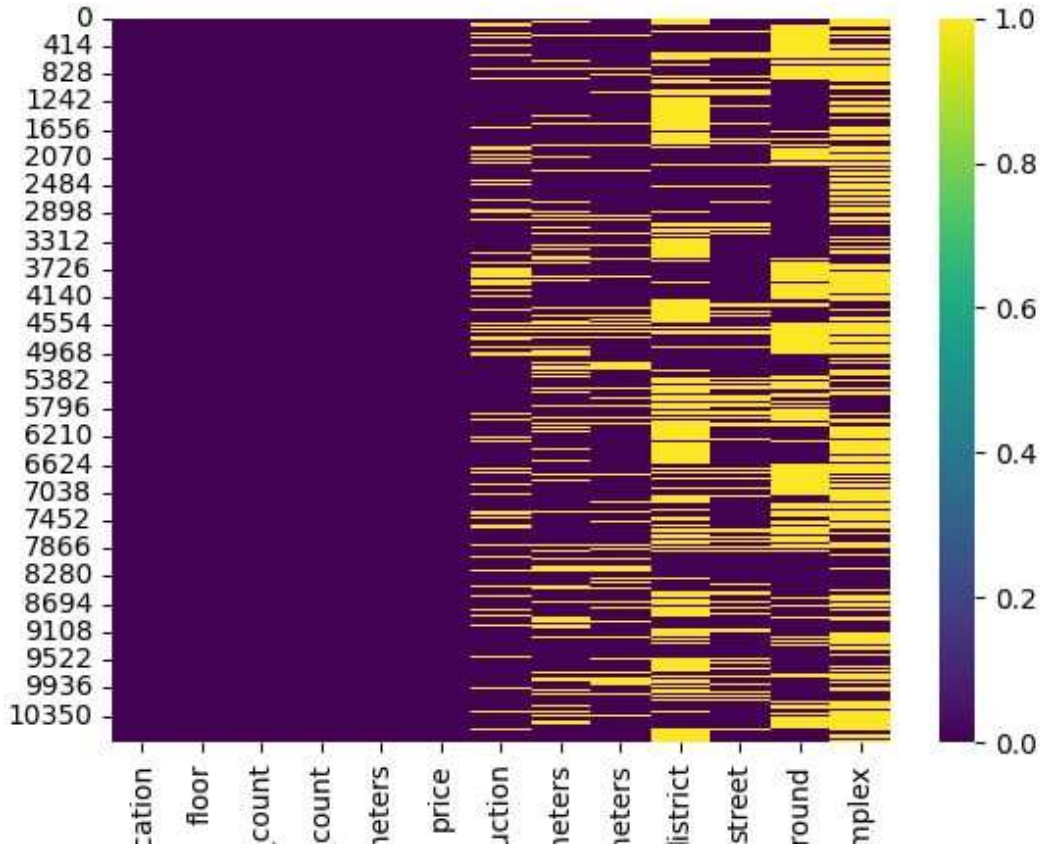
```
# убираем значения -1
df = df.replace(-1, np.nan)
df = df.replace("-1", np.nan)
df = df.replace(-1.0, np.nan)
df = df.replace("-1.0", np.nan)
```

```
# меняем тип данных где нужно
df['floor'] = df['floor'].astype(int)
df['floors_count'] = df['floors_count'].astype(int)
df['total_meters'] = df['total_meters'].astype(float)
```

```
# удаляем не нужные столбцы
df.drop([
    'house_number',
    'author_type',
    'phone',
    'deal_type',
    'accommodation_type',
    'object_type',
    'heating_type',
    'house_material_type',
    'finish_type',
    'author',
    'url'],
    axis=1, inplace=True)
```

```python
# убираем лишние символы
df['living_meters'] = df['living_meters'].replace(to_replace='NBSPм²', value='')

# заполняем пропущенные значения медианой
df['living_meters'] = df['living_meters'].fillna(df['living_meters'].median())
df['kitchen_meters'] = df['kitchen_meters'].fillna(df['kitchen_meters'].median())

# заполняем недостающие значения района значением локации
df['temp_district'] = df['district']
msk_condition = (df['location'] == 'Москва') & df['district'].isna()
df.loc[msk_condition & df['underground'].notna(), 'temp_district'] = df['underground']
df.loc[msk_condition & df['underground'].isna(), 'temp_district'] = 'Москва'
other_condition = df['district'].isna() & (df['location'] != 'Москва')
df.loc[other_condition, 'temp_district'] = df['location']
df['district'] = df['temp_district']
df.drop(columns=['temp_district'], inplace=True)
```

Удалив не нужные столбцы, я визуализировал пропущенные значения

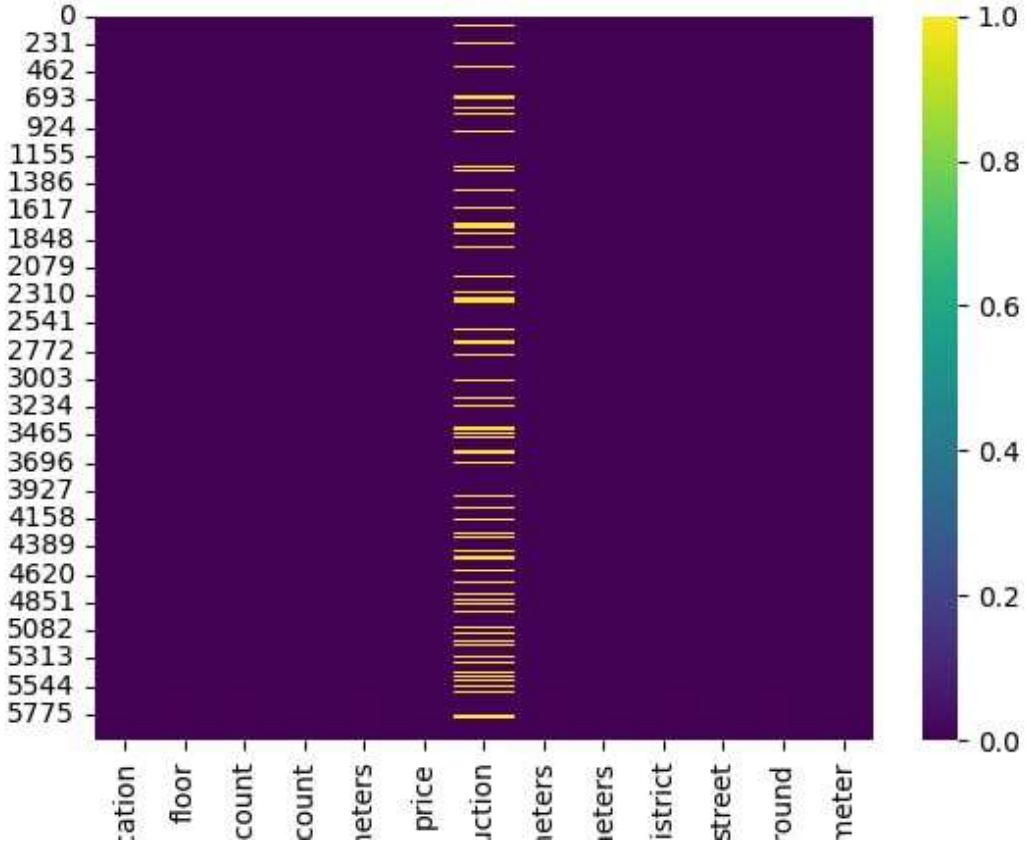


После этого пришел к решению об удалении столбцов, в которых отсутствует более половины значений

Потом заполнил некоторые строки данными

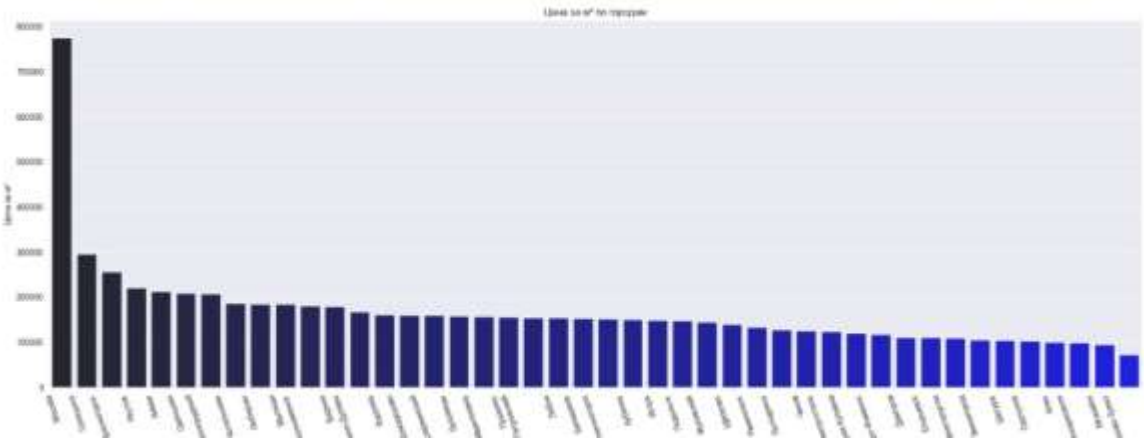
Также были удалены лишние символы из ячеек и сменён тип данных

Вот так выглядят пропущенные значения после проведенных действий



Пришло время найти цену квадратного метра

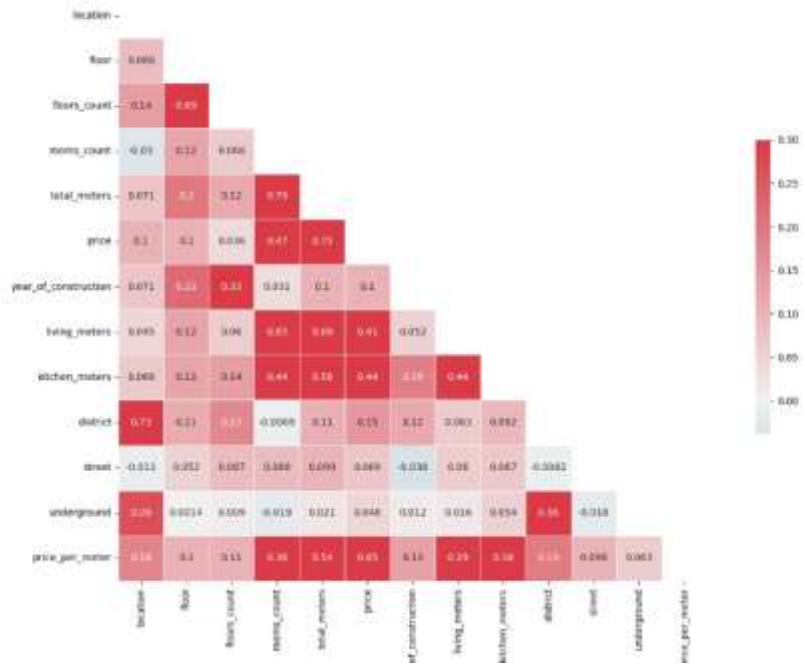
```python
# создаём новый файл в который запишем среднюю цену за квадратный метр по городам
dict_city = df['location'].unique()


1 usage
def price_for_meter(location):
    city = df[df['location'] == location]
    price_for_city = city['price'].sum()

    clean_data = city['total_meters'].sum()

    return round(price_for_city / clean_data, 2)


with open('info.csv', 'w', newline='', encoding='UTF-8') as csvfile:
    names = ['city', 'price_for_meter']
    writer = csv.DictWriter(csvfile, fieldnames=names)
    writer.writeheader()
    for city in dict_city:
        writer.writerow({'city': city, 'price_for_meter': price_for_meter(city)})

# отсортируем по возрастанию
info = pd.read_csv('info.csv')
info_sorted = info.sort_values(by='price_for_meter', ascending=False)
info_sorted.to_csv("info.csv", index=False)

# визуализируем Цена за м² по городам
sns.set_style("darkgrid")
info = pd.read_csv('.csv')
plt.figure(figsize=(24, 8))
sns.barplot(hue='city', legend=False, x='city', y='price_for_meter', data=info, color='Blue')
plt.title('Цена за м² по городам')
plt.xlabel('Города')
plt.ylabel('Цена за м²')
plt.xticks(rotation=110, ha='right')
plt.show()
```



Перекодируем данные для того чтобы создать матрицу корреляции, чтобы оценить взаимосвязи между переменными

```python
# функция, которая принимает на вход наши данные, кодирует числовыми значениями категориальные признаки
# и возвращает обновленный данные и сами кодировщики


1 usage
def number_encode_features(init_df):
    result = init_df.copy() #копируем нашу исходную таблицу
    encoders = {}
    for column in result.columns:
        if result.dtypes[column] == object: # np.object -- строковый тип / если тип столбца - строка, то нужно его закодировать
            encoders[column] = preprocessing.LabelEncoder() #для колонки column создаем кодировщик
            result[column] = encoders[column].fit_transform(result[column]) #применяем кодировщик к столбцу и перезаписываем столбец
    return result, encoders


encoded_data, encoders = number_encode_features(df) #теперь encoded_data содержит закодированные категориальные признаки
print(encoded_data.head()) #проверяем

encoded_data.to_csv("main5.csv", index=False)
```

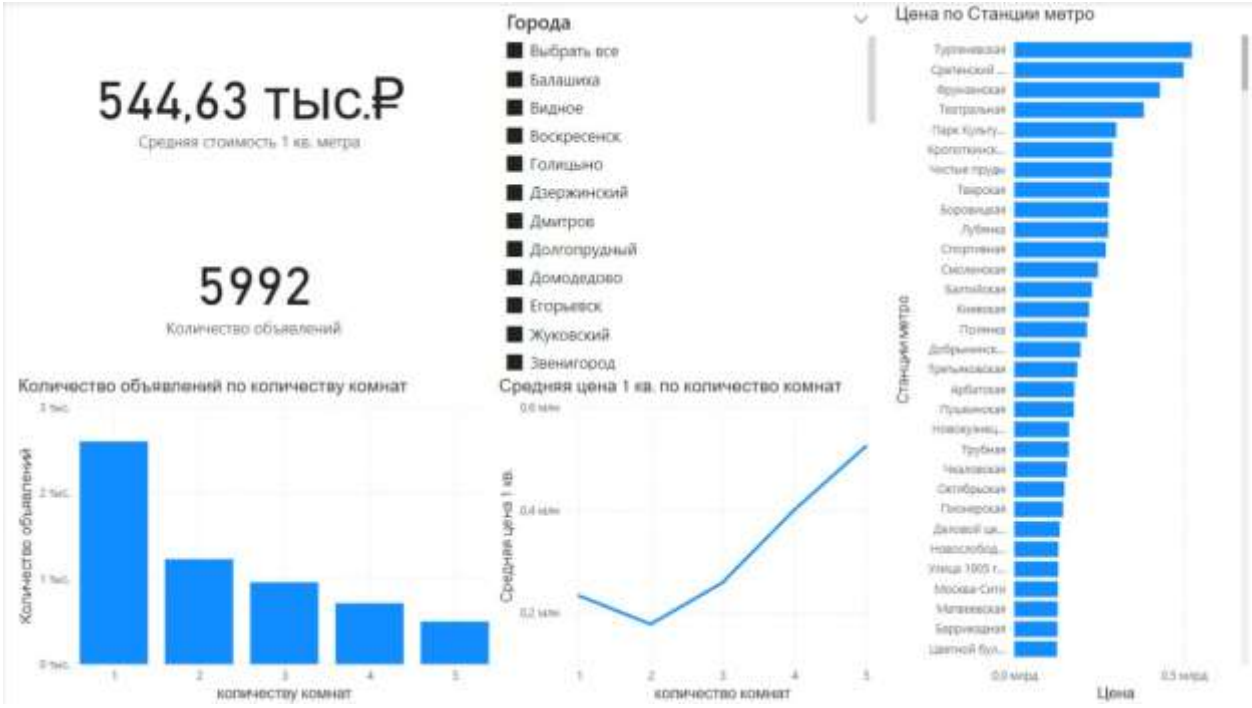
```python
#выведем матрицу кореляции
temp3 = df.copy()
corr = temp3.corr()
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(18, 11))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0, annot=True,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})

plt.show()
```



Заметим взаимосвязь между ценой метражом и количеством комнат в квартире

Переходим в Power BI

Здесь я визуализировал среднюю стоимость квадратного метра по городам, по количеству комнат, и по ближайшей станции метро



Чем показал зависимость цены от:

1. Города
2. Количества комнат
3. Ближайшей станции метро

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

В ходе анализа были собраны и обработаны данные, и создана визуализация, в которой были выявлены ключевые критерии для оценки стоимости квадратного метра недвижимости в Московском регионе. Основными факторами, влияющими на цену жилья, стали близость к станциям метро, расположение в крупных городах, размер квартиры и количество комнат.

. Полученные результаты могут быть использованы для построения модели для предсказания цен на недвижимость.