**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и системы управления»

Кафедра «Теоретическая информатика и компьютерные технологии»

**Лабораторная работа № 4**

**по курсу «Моделирование»**

ОБРАБОТКА И АНАЛИЗ БОЛЬШИХ ДАННЫХ

Студент: Яровикова А. С.

Группа: ИУ9-81Б

Преподаватель: Домрачева А. Б.

Москва, 2024

# **ЦЕЛЬ И ПОСТАНОВКА ЗАДАЧИ**

**Цель**

Целью данной работы является анализ больших данных и построение прогностической модели на основе предварительно обработанных данных. Необходимо провести очистку от выбросов, обработку пропусков, отбор целевого признака и признаков, подаваемых на вход модели, а также непосредственно выполнить обучение модели.

**Постановка задачи**

Рассматривается датасет по недвижимости в России за 2021 год (<https://www.kaggle.com/datasets/mrdaniilak/russia-real-estate-2021/data>).

Необходимо:

- выделить регион Санкт-Петербург

- очистить данные от выбросов:

- построить тепловую карту

- построить гистограммы распределения и диаграммы рассеяния

- отрезать выбросы по стандартному отклонению (z-индекс)

- построить корреляционную матрицу

- в отчет обязательно включить описание данных:

- использовать встроенные методы pandas

- есть ли пропуски были ли обнаружены аномальные значения

- как обрабатывались выбросы и как принималось решение, что это выброс

- разделить выборку на несколько в зависимости от стоимости и площади

- взять модель множественной линейной регрессии

- отобрать признаки, по которым будем обучать

- применить масштабирование данных с помощью StandardScaler. Попробовать как масштабировать таргет, так и нет (если масштабируем то можно, к примеру, применить логарифмирование)

- разделить выборку на обучающую и тестовую (метод из sklearn)

- обучить модель

- посчитать коэффициент- и среднюю абсолютную ошибку

# **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Обработка дубликатов и пропусков**

Часто при анализе данных возникают проблема наличия дубликатов записей. Если не найти дубликаты, то анализ данных может привести к некорректным результатам исследования. Для эффективного выявления дублированных записей применяются различные методы.

Один из методов основан на использовании функции **duplicated()**, которая возвращают логические значения, указывающие на присутствие или отсутствие дубликатов в наборе данных. Этот метод может быть дополнен функцией **sum()** для подсчета общего количества обнаруженных дубликатов.

Второй метод основан на применении функции **value\_counts()**, которая выдает список уникальных значений с их частотой появления в данных, упорядоченный по убыванию. Это позволяет выделить наиболее часто встречающиеся дубликаты и провести дальнейший анализ данных.

Выявление и обработка пропущенных значений также является необходимым шагом подготовки данных. Для получения списка всех уникальных значений в заданном столбце используется функция **unique()**. Удаление строк с пропущенными значениями осуществляется с помощью метода **dropna()**, который исключает строки с пропущенными значениями из исходного набора данных. После удаления строк возможно провести переназначение индексов строк данных в соответствии с новым порядком данных с помощью метода **reset\_index(drop=True)**.

Особые значения **NaN** и **None** обозначают отсутствие значений в ячейке. **NaN** отвечает за отсутствующее в ячейке число. Его тип данных float, поэтому с NaN можно проводить математические операции. А **None** относится к нечисловому типу **NoneType**, и математические операции с ним неосуществимы.

Важно отметить, что пропущенные значения могут искажать результаты анализа данных и требуют специальной обработки, в том числе иногда удаления.

Метод **value\_counts()** возвращает уникальные значения с их количеством в наборе данных. Метод **isnull()** возвращает булевский список, показывающий присутствие или отсутствие пропущенных значений в столбце (True означает, что значение в колонке пропущено). Для замены пропусков на какое-то значение, применяется метод **fillna(value)**.

**Обработка выбросов**

Выбросы в данных представляют собой аномальные значения, которые существенно отклоняются от основной массы наблюдений и могут внести искажения в результаты анализа.

Одним из методов выявления и обработки выбросов является анализ, основанный на стандартном отклонении. Стандартное отклонение (обозначается как ) является мерой разброса значений относительно их среднего значения , и определяется формулой:

где

* – отдельное наблюдение,
* – среднее значение,
* – количество наблюдений.

Выбросы могут быть идентифицированы как значения, находящиеся за пределами заданного диапазона от среднего значения, например, более чем на стандартных отклонений от среднего. Это можно выразить следующим образом:

где - множитель, определяющий ширину диапазона для выявления выбросов.

Применение данного метода позволяет выделить и удалить аномальные значения, уменьшая их влияние на статистические показатели и результаты анализа. Это способствует повышению точности и достоверности интерпретируемости выводов, полученных в результате обработки данных.

**Модель множественной линейной регрессии**

Множественная линейная регрессия является статистическим методом, используемым для прогнозирования значения зависимой переменной на основе нескольких независимых переменных.

В контексте нашей задачи множественная регрессия позволяет моделировать зависимость стоимости недвижимости от разных факторов, таких как площадь помещения, количество этажей в доме, близость к центру города и т.п.

Модель множественной регрессии определяется уравнением:

где:

* – зависимая переменная,
* – независимые переменные,
* ​ – коэффициенты регрессии, параметры модели,
* – случайная ошибка.

Идея состоит в том, чтобы оценить значения коэффициентов регрессии, которые минимизируют сумму квадратов остатков (расхождений между фактическими и прогнозируемыми значениями), что обеспечивает наилучшее приближение модели к данным.

**Оценки точности модели**

Для выбора наиболее подходящей модели и оценки ее производительности часто используются различные метрики. Например, коэффициент и средняя абсолютная ошибка (MAE).

**Коэффициент**  представляет собой меру соответствия модели данным и указывает на то, какую долю дисперсии зависимой переменной объясняет модель. Он вычисляется по формуле:

где

* ​ – фактические значения зависимой переменной,
* – предсказанные значения зависимой переменной моделью,
* – среднее значение зависимой переменной, - количество наблюдений.

Коэффициент может принимать значения от 0 до 1, где 1 указывает на идеальное соответствие модели данным, а значение 0 означает, что модель не объясняет никакой доли вариации.

**Средняя абсолютная ошибка MAE** представляет собой среднее абсолютное значение разницы между фактическими и предсказанными значениями. Она вычисляется как среднее значение абсолютных значений остатков:

где

* ​ – фактическое значение зависимой переменной,
* – предсказанное значение,
* – количество наблюдений.

MAE позволяет оценить среднюю величину ошибки модели и ее способность предсказывать значения зависимой переменной. Чем ниже значение MAE, тем лучше качество модели.

**Масштабирование данных**

Масштабирование данных – процесс преобразования значений признаков таким образом, чтобы они находились в определенном диапазоне или имели определенное распределение.

Одним из распространенных методов масштабирования является стандартизация, или масштабирование по стандартному методу.

Стандартизация основана на преобразовании значения признака по формуле:

где

* - исходное значение признака,
* - среднее значение признака,
* - стандартное отклонение признака,
* ​ - масштабированное значение признака.

Процесс стандартизации делает значения признаков такими, что они имеют среднее значение 0 и стандартное отклонение 1. Это полезно для сравнения и анализа признаков, которые изначально имеют разные единицы измерения. Также это улучшает работу алгоритмов машинного обучения, которые зависят от масштаба признаков.

# **ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ**

В данном задаче были использованы выборки мужских и женских доходов из набора данных по покупке беговых дорожек для фитнеса. Всего было 104 и 76 записей в выборках для мужчин и женщин соответственно. Данные были поделены на обучающие и контрольные выборки – 78 и 26 значений для мужских доходов и 57 и 19 – для женских, соответственно.

Исходный код лабораторной работы представлен ниже.

Импортирование необходимых библиотек:

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

Загрузка датасета:

df = pd.read\_csv('dataset/input\_data.csv', sep = ';')

df.head()

Извлечение Необходимого региона – Санкт-Петербруга:

saint\_petersburg\_df = df[df.id\_region == 78]

saint\_petersburg\_df

Добавление новых столбцов и удаление неактуальных для удобства анализа:

saint\_petersburg\_df['date'] = pd.to\_datetime(saint\_petersburg\_df['date'])

saint\_petersburg\_df['year'] = saint\_petersburg\_df['date'].dt.year

saint\_petersburg\_df['month'] = saint\_petersburg\_df['date'].dt.month

saint\_petersburg\_df = saint\_petersburg\_df.drop(['date'], axis = 1)

saint\_petersburg\_df = saint\_petersburg\_df.drop(['id\_region'], axis = 1)

saint\_petersburg\_df

Для построения тепловой карты используется библиотека *folium*:

import folium

from folium.plugins import HeatMap

mapa = folium.Map(location=[saint\_petersburg\_df.geo\_lat.mean(), saint\_petersburg\_df.geo\_lon.mean()], zoom\_start=6)

map\_values = saint\_petersburg\_df[['geo\_lat', 'geo\_lon', 'price']]

data = map\_values.values.tolist()

hm = HeatMap(data, min\_opacity=0.05, max\_opacity=0.9, radius=25).add\_to(mapa)

mapa

Для построения гистограмм распределения и диаграмм рассеяния используется библиотека *seaborn*. Метод *subplots()* позволяет расположить несколько диаграмм на одном графике, что помогает эффективно проводить анализ и сравнение:

fig, axes = plt.subplots(4, 2, figsize = (25, 25))

sns.set()

sns.histplot(data = saint\_petersburg\_df.price, ax = axes[0, 0])

axes[0, 0].set\_title('Распределение цены', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.area, ax = axes[2, 0])

axes[2, 0].set\_title('Распределение площадей квартир', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.building\_type, ax = axes[3, 0])

axes[3, 0].set\_title('Распределение типов домов', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.object\_type, ax = axes[3, 1])

axes[3, 1].set\_title('Распределение типов квартир', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.level, ax = axes[0, 1])

axes[0, 1].set\_title('Распределение количества этажей', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.levels, ax = axes[1, 0])

axes[1, 0].set\_title('Распределение этажей квартир в объявлениях', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.rooms, ax = axes[1, 1])

axes[1, 1].set\_title('Распределение количества комнат', loc = 'left')

sns.histplot(data = saint\_petersburg\_df.kitchen\_area, ax = axes[2, 1])

axes[2, 1].set\_title('Распределение площадей кухонь', loc = 'left')

plt.show()

fig, axes = plt.subplots(4, 2, figsize = (25, 25))

sns.set()

fig.suptitle('Санкт-Петербург')

sns.scatterplot(x='level', y='price', data=saint\_petersburg\_df, ax = axes[0, 0], alpha = 0.01)

sns.scatterplot(x='levels', y='price', data=saint\_petersburg\_df, ax = axes[0, 1], alpha = 0.01)

sns.scatterplot(x='rooms', y='price', data=saint\_petersburg\_df, ax = axes[1, 0], alpha = 0.01)

sns.scatterplot(x='area', y='price', data=saint\_petersburg\_df, ax = axes[1, 1], alpha = 0.01)

sns.scatterplot(x='kitchen\_area', y='price', data=saint\_petersburg\_df, ax = axes[2, 0], alpha = 0.01)

sns.scatterplot(x='building\_type', y='price', data=saint\_petersburg\_df, ax = axes[2, 1], alpha = 0.01)

sns.scatterplot(x='object\_type', y='price', data=saint\_petersburg\_df, ax = axes[3, 0], alpha = 0.01)

plt.show()

Построение корреляционной матрицы:

# корреляционная матрица

plt.figure(figsize=(15, 10))

sns.heatmap(saint\_petersburg\_df.corr(), center=0, cmap='mako', annot=True)

plt.title('Корреляционная матрица')

plt.show()

Информация о датасете:

saint\_petersburg\_df.info(verbose = True, show\_counts = True)

Информация о пропусках:

saint\_petersburg\_df.isna().sum()

Удаление пропусков. Столбцы, в которых наблюдаются пропуски, удаляются, так как не имеют корреляции с стоимостью недвижимости:

saint\_petersburg\_df.replace([np.inf, -np.inf], np.nan, inplace=True)

saint\_petersburg\_df.isna().sum()

# Удаляем ненужные колонки с пропусками

saint\_petersburg\_df.drop(columns = ['postal\_code', 'street\_id', 'house\_id', 'year'], inplace = True)

saint\_petersburg\_df

Информация о дубликатах:

saint\_petersburg\_df.duplicated().sum()

Удаление дубликатов:

saint\_petersburg\_df = saint\_petersburg\_df.drop\_duplicates()

saint\_petersburg\_df.duplicated().sum()

Замена отрицательных значений для комнат и кухонь. Предположительно наличие отрицательных значений связано с ошибкой опечатки:

print(saint\_petersburg\_df.rooms.unique())

saint\_petersburg\_df['rooms'] = saint\_petersburg\_df["rooms"].apply(lambda x: -x if x < 0 else x)

print(saint\_petersburg\_df.kitchen\_area.unique())

saint\_petersburg\_df['kitchen\_area'] = saint\_petersburg\_df["kitchen\_area"].apply(lambda x: -x if x < 0 else x)

# Если значения этажей в доме больше, чем этаж квартиры, то меняем их местами (так как это скорее просто ошибка)

saint\_petersburg\_df.loc[saint\_petersburg\_df["level"] > saint\_petersburg\_df["levels"], "level"] = saint\_petersburg\_df["levels"]

Отсечение выбросов по стандартному отклонению:

# Отрезаем по стандартному отклонению

def remove\_outliers(df, columns):

    for column in columns:

        z\_scores = (df[column] - df[column].mean()) / df[column].std()

        df = df[(z\_scores.abs() < 1.5)]

    return df

numeric\_columns = ['geo\_lat', 'geo\_lon', 'price', 'area', 'kitchen\_area']

clean\_data = remove\_outliers(saint\_petersburg\_df, numeric\_columns)

Масштабирование:

# Масштабирование

from sklearn.preprocessing import StandardScaler

X = data[['level', 'levels', 'rooms', 'area', 'kitchen\_area', 'geo\_lat', 'geo\_lon', 'building\_type', 'object\_type']]

Y = data['price']

scaler = StandardScaler()

# Масштабирование данных

X = scaler.fit\_transform(X)

Деление на обучающую и тестовую выборку и обучение модели:

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

print('Accuracy of Linear Regression on training data', model.score(X\_train, y\_train))

print('Accuracy of Linear Regression on testing data', model.score(X\_test, y\_test))

y\_pred = model.predict(X\_test)

Метрики:

from sklearn.metrics import mean\_absolute\_error, r2\_score

# коэффициент R^2

r2\_score(y\_test, y\_pred)

# средняя абсолютная ошибка

mean\_absolute\_error (y\_tests, y\_pred)

saint\_petersburg\_df['price'].mean()

# **РЕЗУЛЬТАТЫ**

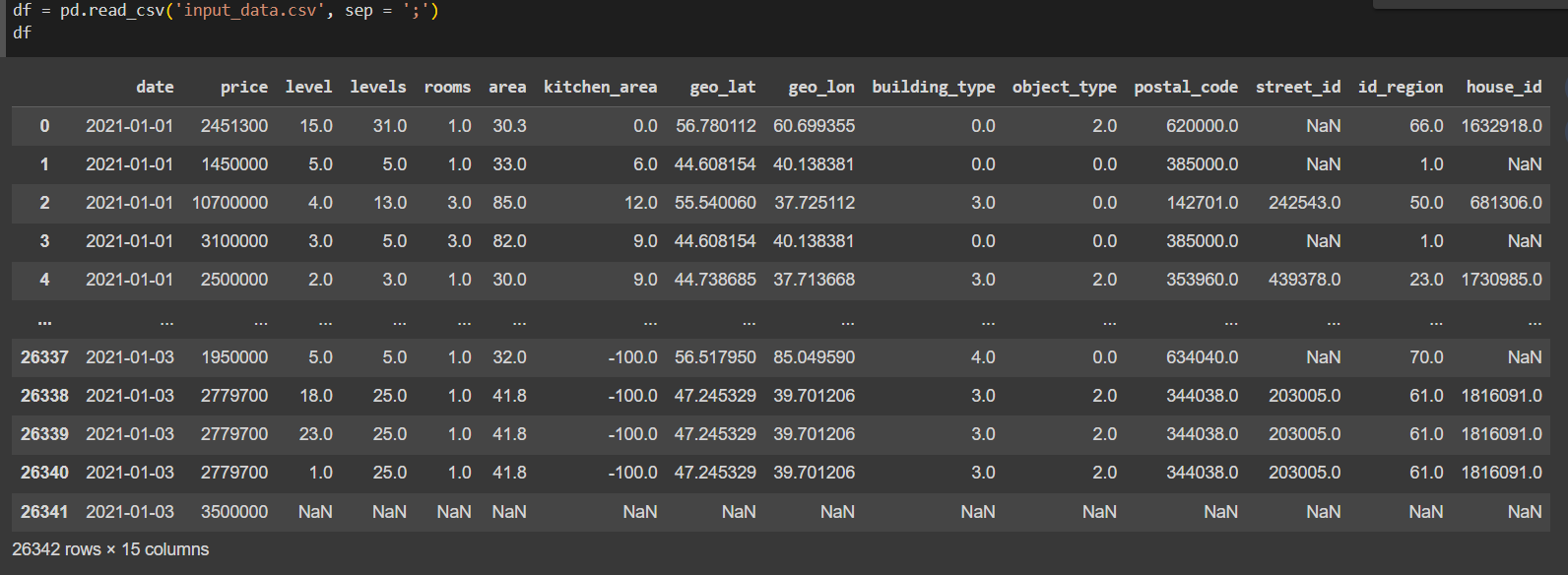
Данные загружены с помощью библиотеки *pandas,* после чего из них выделена выборка в соответствии с регионом 78 – городом Санкт-Петербург. 

Рисунок 1 - Данные из датасета

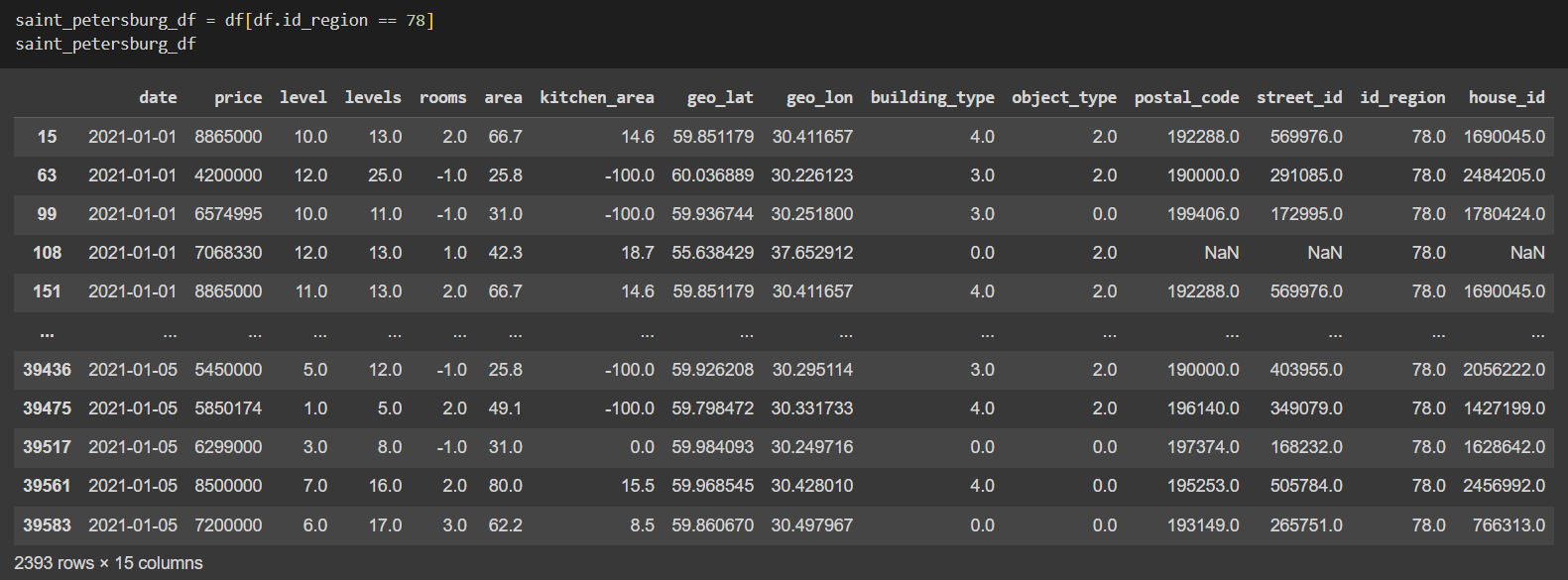


Рисунок 2 - Данные из выборки по Санкт-Петербургу

Целевым признаком было принято решение выбрать стоимость жилья – признак *price* в выборке.

Для анализа выбросов была построена тепловая карта по выборке. Она представлена на рисунках 3 и 4.

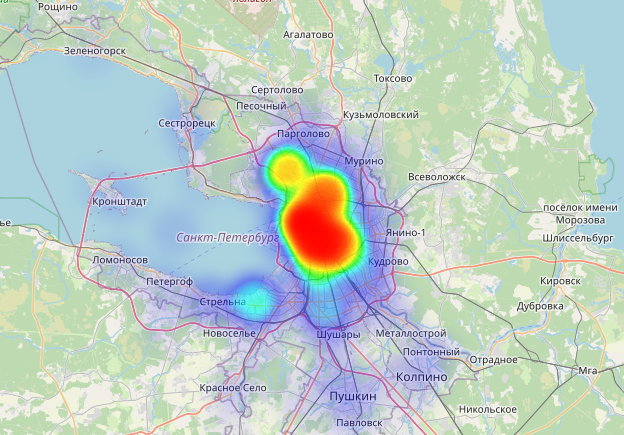


Рисунок 3 - Тепловая карта до очистки

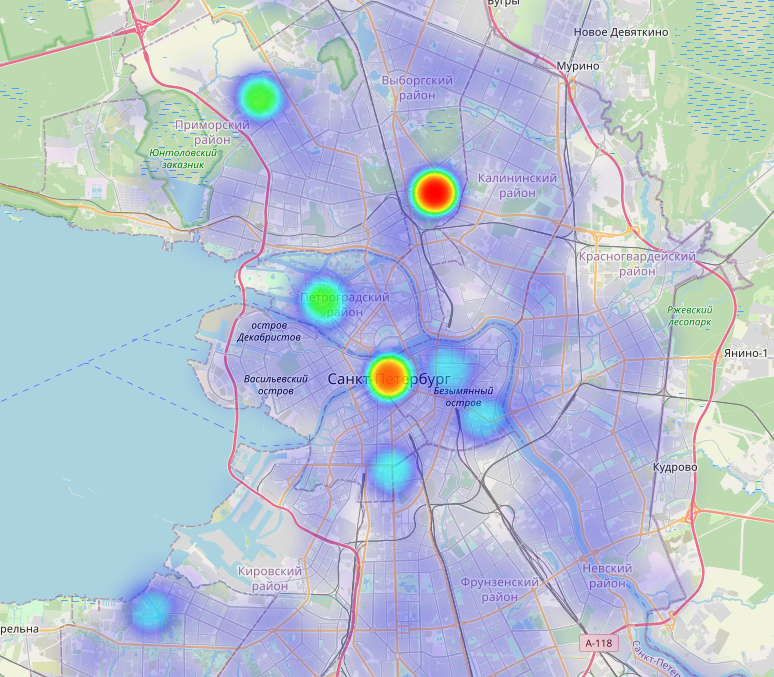


Рисунок 4 - Тепловая карта до очистки. Вид ближе

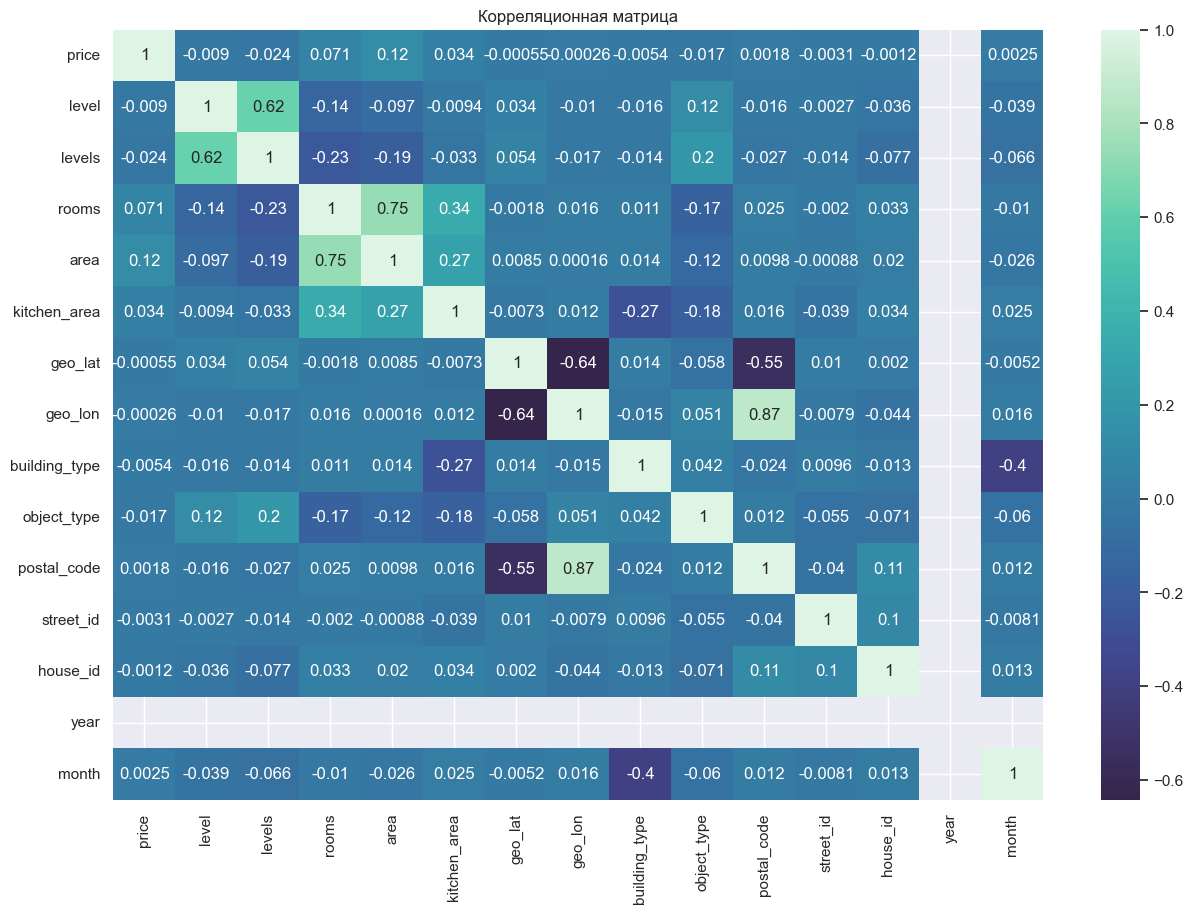


Рисунок 5 - Корреляционная матрица до очистки

Далее были построены гистограммы распределения и диаграммы рассеяния, на которых можно также наблюдать выбросы. Графики представлены на рисунках 6 и 7.

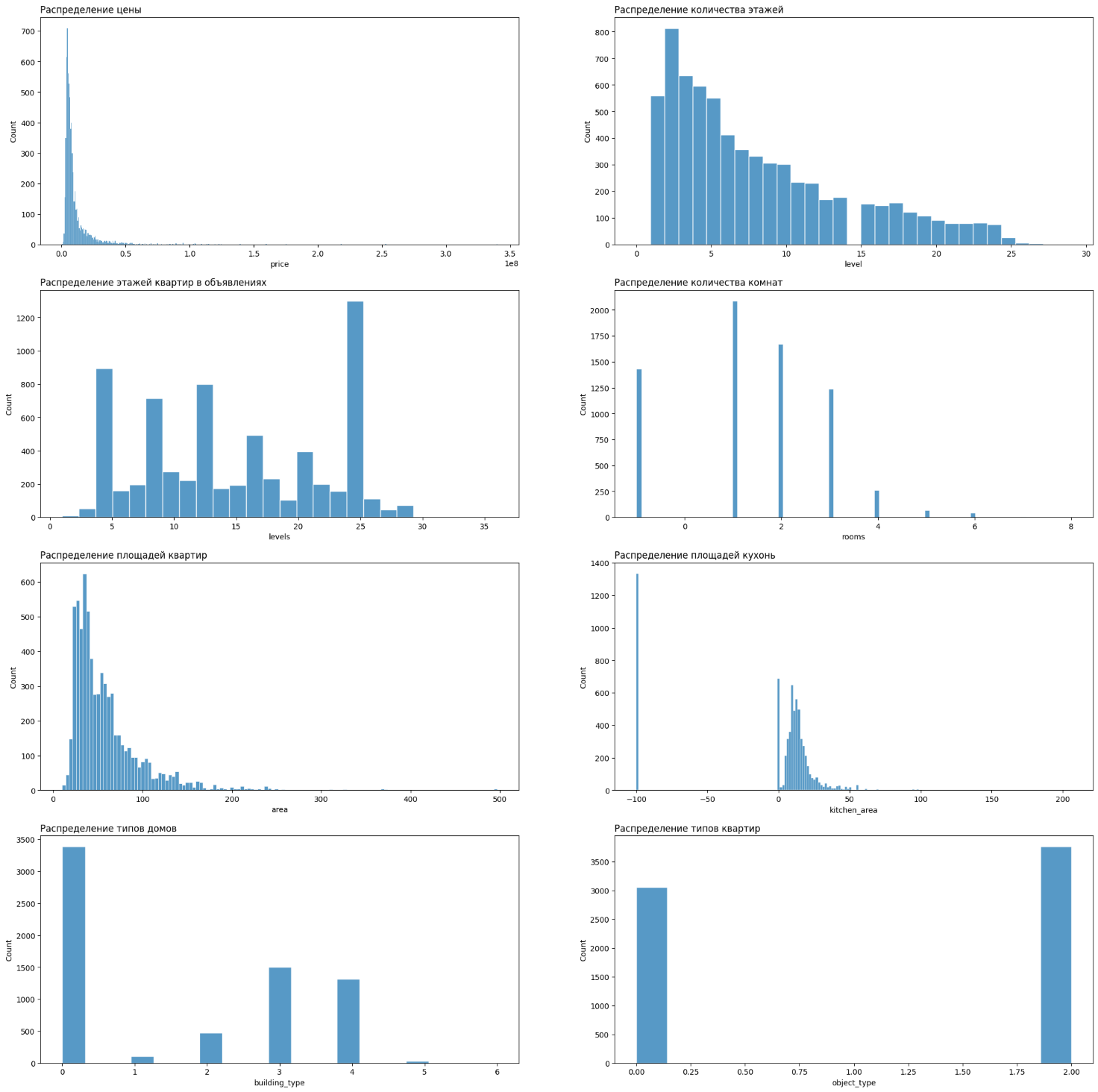


Рисунок 6 - Гистограммы распределения до очистки

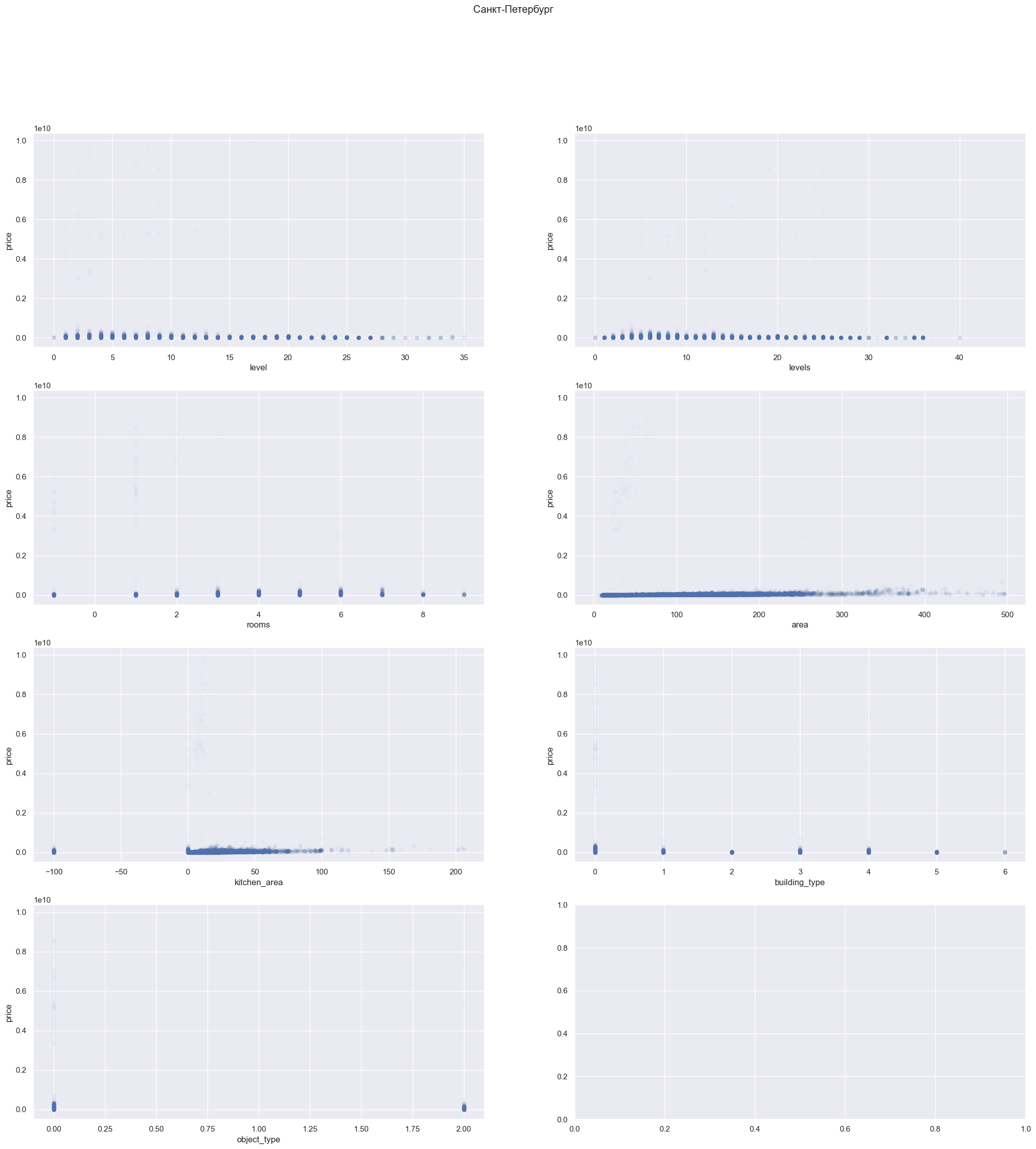


Рисунок 7 - Диаграммы рассеяния до очистки данных

На рисунке ниже представлена информация о данных по выборке по Санкт-Петербургу. Представлены следующие признаки:

* price – стоимость в рублях
* level – этаж квартиры
* levels – количество этажей в доме
* rooms – количество комнат в квартире (скольки-комнатная квартира)
* area – общая площадь квартиры
* kitchen\_area – площадь кухни
* geo\_lat – географическая долгота
* geo\_lon – географическая широта
* building\_type – тип (неизвестен, другой, панельный, монолит, кирпичный, блочный, деревянный). Категориальный признак представлен цифрами от 0 до 6
* object\_type – тип квартиры (вторичный рынок, новостройка). Категориальный признак представлен цифрами от 1 до 2
* postal\_code – почтовый индекс дома
* street\_id – идентификатор улицы города
* house\_id – идентификатор дома в городе
* year и month – год и месяц публикации объявления. Колонки созданы искусственно из начальной колонки date, поскольку день публикации практически не коррелирует со стоимостью жилья.

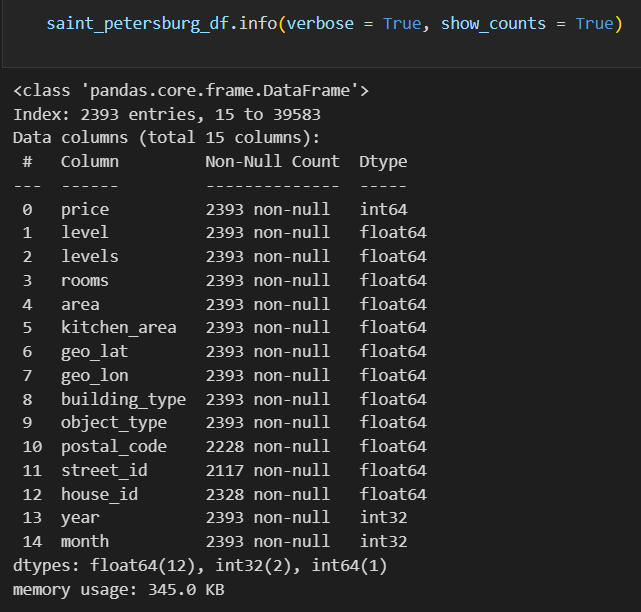


Рисунок 8 - Информация о данных

Данные были проверены на наличие пропусков. Пропуски были

найдены в колонках postal\_code, street\_id, house\_id, как видно на рисунке 9. Перечисленные признаки практически не имеют корреляции со стоимостью недвижимости, поэтому эти колонки были удалены.

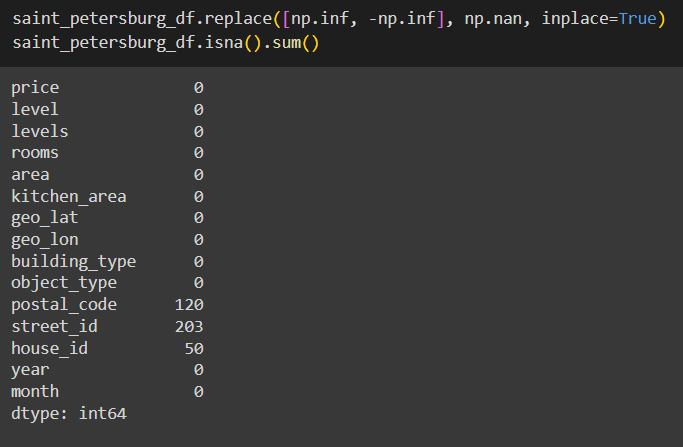
 

Рисунок 9 - Пропущенные значения

Далее была проверка на наличие дублированных записей. Найденные дубликаты были удалены.

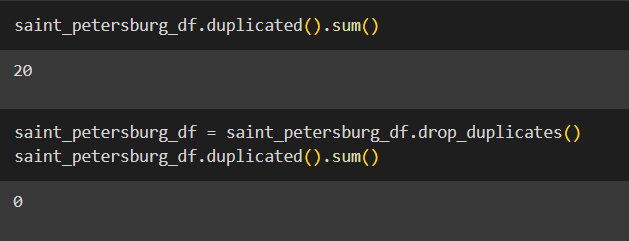


Рисунок 10 - Удаление дубликатов

Далее были обработаны аномальные отрицательные значения в колонке kitchen\_area и rooms. Поскольку отрицательные значения в данном случае рассматриваются как аномалия, они были заменены на противоположные по знаку положительные значения.

Затем выбросы в данных были отрезаны по стандартному отклонению (z-index): для широты, долготы, площади и количества комнат использовалось пороговое значение 1.5, для целевого признака стоимости – пороговое значение 0.15 в связи с достаточно большим разбросом значений.

По результатам обработки выбросов, пропусков и дубликатов, были заново созданы тепловая карта, диаграммы рассеяния и гистограммы распределения, на которых видно, что выбросы и аномалии устранены.

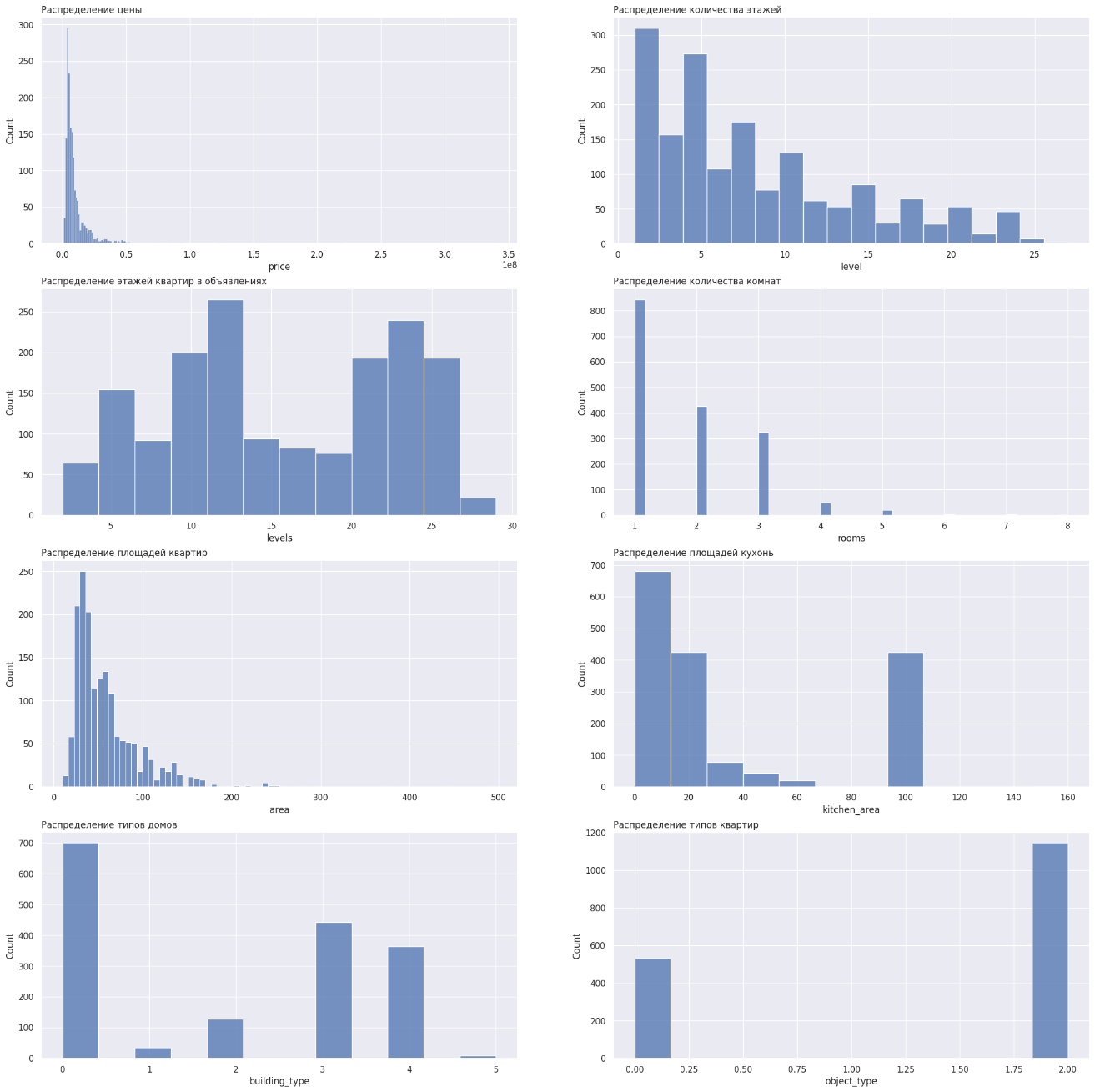


Рисунок 11 - Гистограммы распределения после очистки

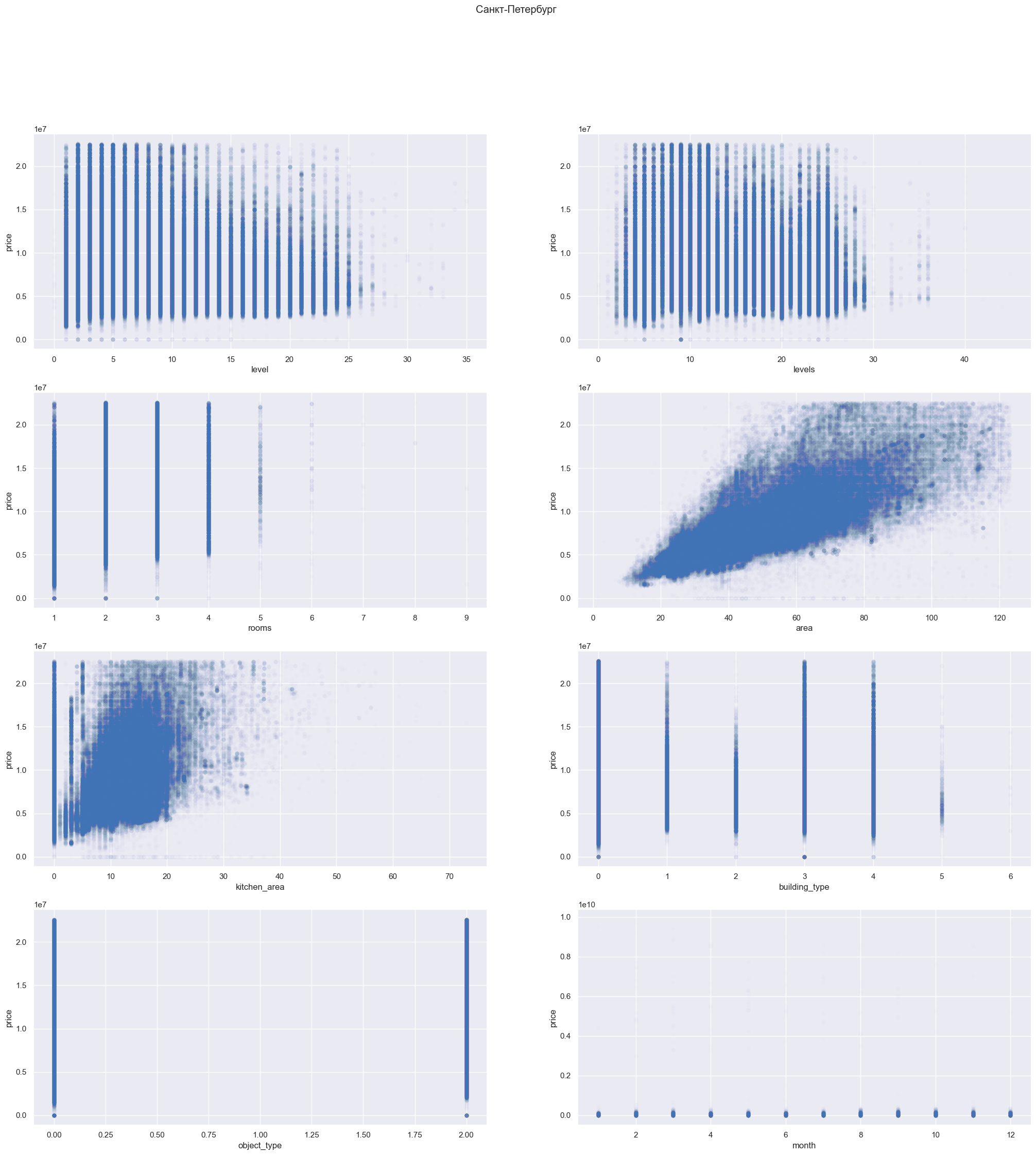


Рисунок 12 - Гистограммы рассеяния после очистки. Увеличенный масштаб

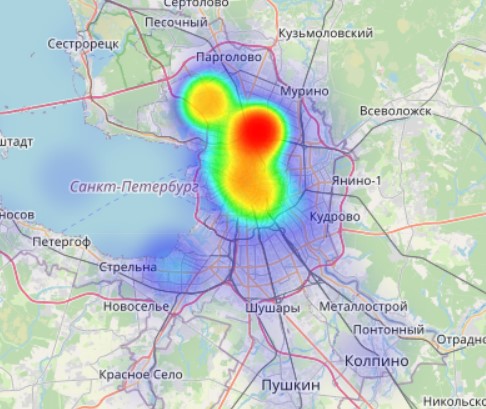


Рисунок 13 - Тепловая карта после очистки

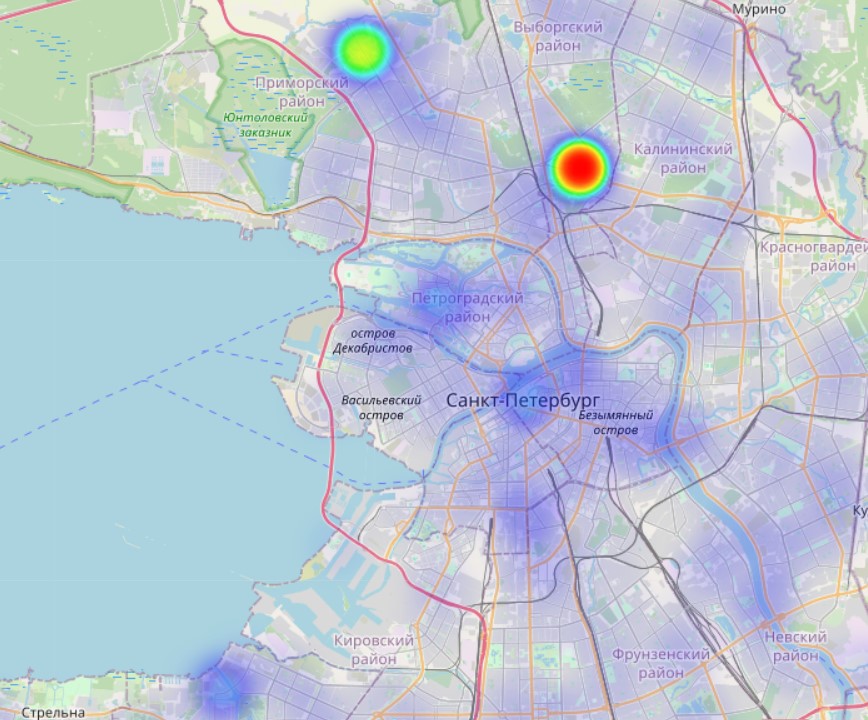


Рисунок 14 - Тепловая карта после очистки. Вид ближе

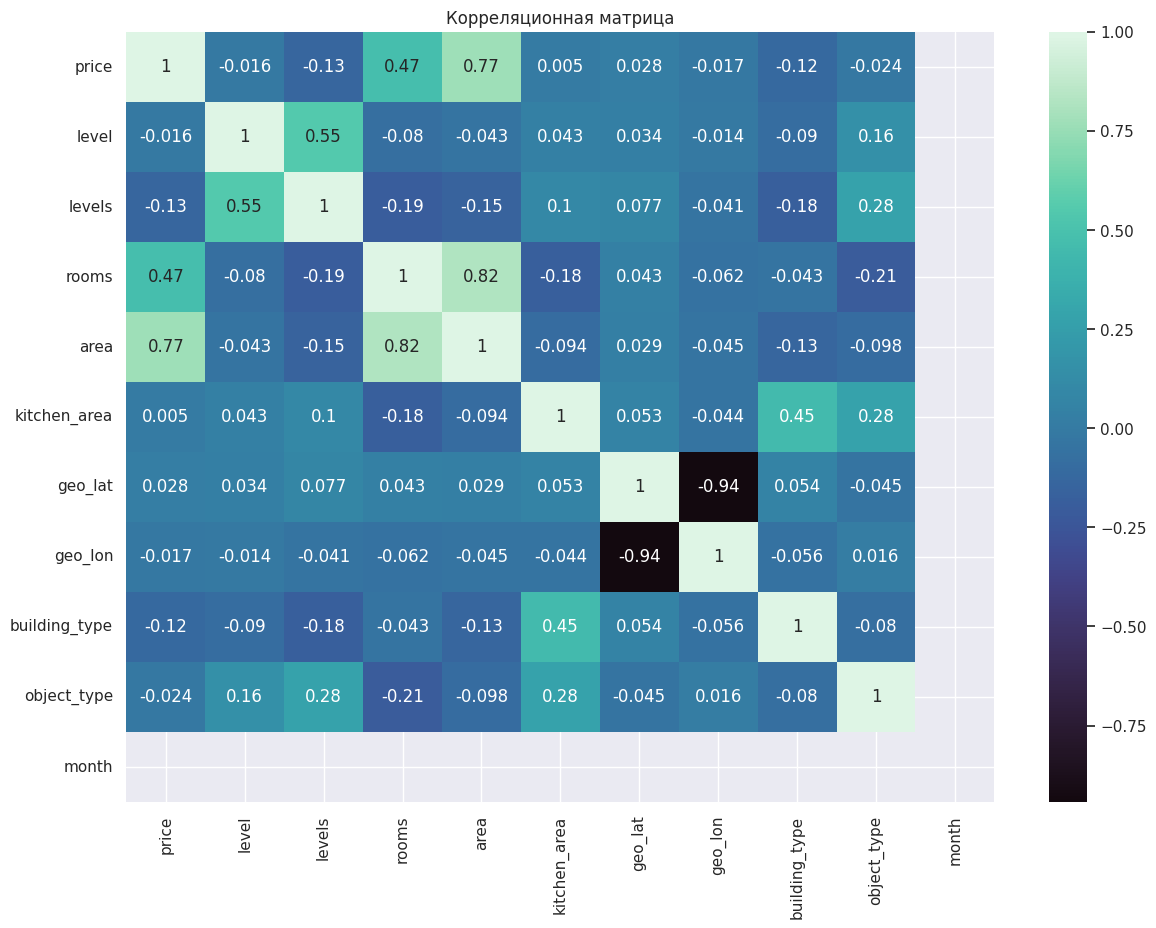


Рисунок 15- Корреляционная матрица после очистки

По обновленной корреляционной матрице можно сделать вывод на наибольшей корреляции между стоимостью и площадью и количеством комнат объекта недвижимости. Поэтому эти признаки были выбраны для обучения модели.

Перед обучением, данные были масштабированы по стандартному методу и разделены на тренировочную и валидационные выборку.

Точность модели на тренировочных данных составила приблизительно 0.65, в то время как на валидационной выборке – 0.74.

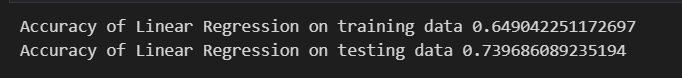


Рисунок 16 – Точность

Оценки метрик получились следующими:.

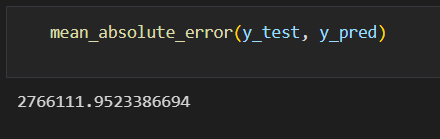
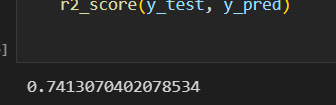
 

Рисунок 17 – Метрики

# **ВЫВОДЫ**

В ходе выполнения лабораторной работы был проведен анализ выборки о недвижимости в Санкт-Петербурге. Данные были обработаны и подготовлены для обучения построенной модели множественной линейной регрессии для прогнозирования стоимости жилья в данном регионе.

Результаты работы позволили сделать вывод о том, что модель делает достаточно точные предсказания.

Коэффициент детерминации равный 0.74 указывает на то, что 74% дисперсии целевой переменной (стоимости объектов недвижимости) объяснены с помощью используемых признаков модели. Это может признать хорошим результатом, поскольку значение близкое к 1 свидетельствует о хорошем соответствии модели выбранным данным.

Значение средней абсолютной ошибки MAE, равное можно интерпретировать, опираясь на среднее значение предсказываемых стоимостей на объекты недвижимости. Поскольку средняя стоимость составляет , то среднюю абсолютную ошибку со значением порядка двух-трех миллионов можно считать приемлемой.

Таким образом, основываясь на предыдущих рассуждениях и значениях рассмотренных метрик, делаем вывод о том, что модель демонстрирует хорошую способность предсказывать.