**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ**

**УНИВЕРСИТЕТ»**

**(СПбГЭУ)**

**ОТЧЕТ**

**по Производственной практике**

Наименование организации прохождения практической подготовки:

**ООО «Актив-Инвест»**

*(наименование организации)*

Направление **38.03.01 «Экономика»**

*(шифр, наименование)*

Направленность: **Математическое моделирование и анализ данных в экономике** *(наименование)*

Обучающийся **Дмитриев Александр Ростиславович**

*(Ф.И.О. полностью)*

Группа **Э-2010** Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(номер группы)*

Руководитель

по практической подготовке от СПбГЭУ

**Полякова Светлана Петровна**, **к.э.н., доцент, доц.**

*(Ф.И.О., ученая степень, должность, ученое звание)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Оценка по итогам защиты отчета\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024 г.

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc164382372)

[**1.** **РАБОТА С ДАННЫМИ** 4](#_Toc164382373)

[**1.1.** **ПОСТАНОВКА ЗАДАЧИ И ОПРЕДЕЛЕНИЕ ИСТОЧНИКА ДАННЫХ** 4](#_Toc164382374)

[**1.2.** **СБОР ДАННЫХ** 7](#_Toc164382375)

[**1.3.** **ОБРАБОТКА ДАННЫХ** 11](#_Toc164382376)

[**1.4.** **РАБОТА С КООРДИНАТНЫМИ ДАННЫМИ** 13](#_Toc164382377)

[**2.** **АНАЛИЗ ДАННЫХ И МАШИННОЕ ОБУЧЕНИЕ** 16](#_Toc164382378)

[**2.1.** **ГЕОГРАФИЧЕСКИЙ АНАЛИЗ** 16](#_Toc164382379)

[**2.2.** **АНАЛИЗ ЗНАЧИМОСТИ ПАРАМЕТРОВ** 18](#_Toc164382380)

[**2.3.** **ОБРАБОТКА ДАННЫХ И ПОДГОТОВКА К ПОСТРОЕНИЮ МОДЕЛЕЙ** 22](#_Toc164382381)

[**2.4.** **ВИЗУАЛИЗАЦИЯ ОСОБЕННОСТЕЙ ДАННЫХ** 22](#_Toc164382382)

[**2.5.** **ПОСТРОЕНИЕ МОДЕЛЕЙ** 26](#_Toc164382383)

[**ЗАКЛЮЧЕНИЕ** 30](#_Toc164382384)

[**СПИСОК ИСТОЧНИКОВ** 31](#_Toc164382385)

# **ВВЕДЕНИЕ**

В условиях современной экономики, роль рынка коммерческой недвижимости несомненна. Она обусловлена динамичным развитием рынка последние годы, только за 2023 год объем инвестиций вырос в 1,5 раза по всему коммерческому сектору недвижимости, по торговой – в 2. Эффективное управление активами, точная и объективная оценка и прогнозирование стоимости коммерческой недвижимости могут значительно повысить инвестиционную привлекательность и экономическую стабильность компаний и частных инвесторов, не говоря уже об улучшении качества оказываемых предприятиями услуг в рамках работы с недвижимостью. Сбор данных, их анализ и построение адекватных моделей прогнозирования являются ключевыми для обоснования инвестиционных решений и оптимизации портфелей недвижимости.

Развитие навыков и компетенций в области анализа, оценки и прогнозирования стоимости коммерческой торговой недвижимости являются одними из основополагающих целей практики. Это включает в себя освоение на практике методик сбора и анализа данных, работы с экономической интерпретацией, а также применение статистических и аналитических инструментов для работы над созданием надежных моделей прогнозирования. Практика направлена на получение реального опыта работы в специализированном отделе, умение анализировать рынок, делать обоснованные выводы и рекомендации для внутренних и внешних клиентов.

Предметом практики стоит понимать анализ и оценку рынка коммерческой торговой недвижимости. Это включает в себя изучение структуры и динамики рынка, анализ спроса и предложения, изучение факторов, влияющих на стоимость недвижимости, а также применение статистических методов для анализа и прогнозирования цен на объекты коммерческой недвижимости.

Объектом практики же является коммерческая торговая недвижимость. В ходе практики основное внимание уделяется объектам торговой недвижимости, таким как помещения свободного типа, будущие магазины и помещения в коммерческих комплексах. Работа включает анализ различных параметров этих объектов, таких как координатное местоположение, характеристика здания и объекта, удаленность от определенных точек, размеры, текущее состояние рынка, что в итоге позволяет формировать базу данных для моделей оценки стоимости.

Очень важно для проведения такого рода анализов осуществить доскональное изучение доступных источников данных, провести полноценный анализ сферы коммерческой недвижимости и принципов оценки объектов.

# **РАБОТА С ДАННЫМИ**

## **ПОСТАНОВКА ЗАДАЧИ И ОПРЕДЕЛЕНИЕ ИСТОЧНИКА ДАННЫХ**

Для анализа рынка и построения в дальнейшем модели – необходимо собрать основную базу данных недвижимости. В организации, как правило, используется собственная база данных, однако в рамках конфиденциальности собственности компании возникает необходимость использовать данные с открытых источников, тем не менее, в самой компании регулярно практикуется сбор и использование открытых источников, в особенности для обновления базы данных и понимания функционирования рынка. Также, стоит отметить, в рамках работы будут использованы данные именно по Санкт-Петербургу и ближайшей части Ленинградской области, но это не отменяет возможностей использования данного подхода к любой местности, при наличии основных параметров.

На нынешнем рынке есть несколько крупных открытых баз недвижимости. В первую очередь это такие сайты как «Авито», «Циан», «Яндекс Недвижимость». Однако есть и другие, достаточно подробные информационные источники, однако они принадлежат определенным агентствам недвижимости, то есть подразумевают узконаправленность имеющихся данных и не предоставляют необходимый для данного рода работы объем и спектр объектов.

Из указанных источников необходимо выбрать один основной, соответствующий основным параметрам:

* Полнота данных – наличие максимума информации об объекте;
* Целостность и корректность данных;
* Отсутствие ложных данных;
* Наличие критически важных параметров: адрес, квадратура, стоимость.

Стоит рассмотреть конкретные примеры объектов на данных площадках. В таком случае будет возможность изучить возможный набор данных, который можно будет собрать с каждого объекта.

На данном рисунке видно пример объекта с сайта «Яндекс Недвижимость». Если изучить код страницы, а также визуальную составляющую – со страницы объявления можно извлечь следующую информацию:

* URL;
* Стоимость;
* Площадь;
* Стоимость за квадратный метр;
* Адрес;
* Этаж;
* Назначение (офис, склад, торговое);
* Фиктивные переменные, т.е. подробная информация;
* Агентство / Собственник;
* Расстояние до метро в минутах.

Далее рассмотрим пример страницы объявления Авито:

* Стоимость;
* Площадь;
* Стоимость за квадратный метр;
* Фиктивные переменные, т.е. подробная информация;
* Адрес;
* Этаж;
* Агентство / Собственник;

Как последний подробно-изученный источник для будущей базы данных – рассмотрим Циан.

* URL;
* Стоимость;
* Площадь;
* Стоимость за кв метр;
* Дом;
* Этаж;
* Назначение (офис, склад, торговое);
* Тип здания;
* Фиктивные переменные;
* Агентство;
* Расстояние до метро в минутах;

С учетом проведенного анализа, включая изучение корректности данных – было решено, что основным источником для данных стоит использовать Циан. Яндекс и Авито не так подробно изучают объявления и оставляют большое количество ложных объявлений, которые размещают некоторые агентства недвижимости для привлечения клиентов.

## **СБОР ДАННЫХ**

Для сбора данной информации придется обратиться к написанию программы-парсера. Парсер — это программа, сервис или скрипт, который собирает данные с указанных веб-ресурсов, обрабатывает их и выдает в нужном формате. Использование парсеров очень распространено в современном мире, во многом это один из основных способов получения информации из открытого доступа. Более того, в сфере недвижимости это цифровой фундамент для изучения рынка. Все современные агентства стараются постоянно следить за новыми данными и собирать максимум из них. Такой задачей и занимается один из отделов компании «Актив-Инвест».

После изучения основных источников на тему парсеров, их типов, принципов использования и основных библиотек на языке программирования python возникла необходимость также и в изучении основ написания сайтов – html, для понимания структуры страниц и данных на этих страницах. Проконсультировавшись со специалистами в «Актив-Инвест» – была проделана работа по написанию парсера специально для данного сайта, с целью собрать максимум возможной информации.

Для данной задачи были изучены основные библиотеки для парсинга на языке программирования python – “Beautifull4Soup” и “Selenium”. Используя основные API этих библиотек, были изучены возможности для сбора данных.

Выводом стало, что самым оптимальным и эффективным способом стоит считать использование библиотеки “Selenium”, во многом благодаря возможностям данной библиотеки более точно взаимодействовать с сайтами, используя имитацию нажатию кнопок, открытия ссылок и перехода от страницы к странице.

Для того, чтобы собрать все объекты, первоначально необходимо собрать все ссылки на них с предварительно настроенной фильтрацией сайта. Настроим среду для кода и импортируем необходимые библиотеки.

Был написан код для парсинга ссылок всех объектов в сфере коммерческой недвижимости в Санкт-Петербурге и Ленинградской области.

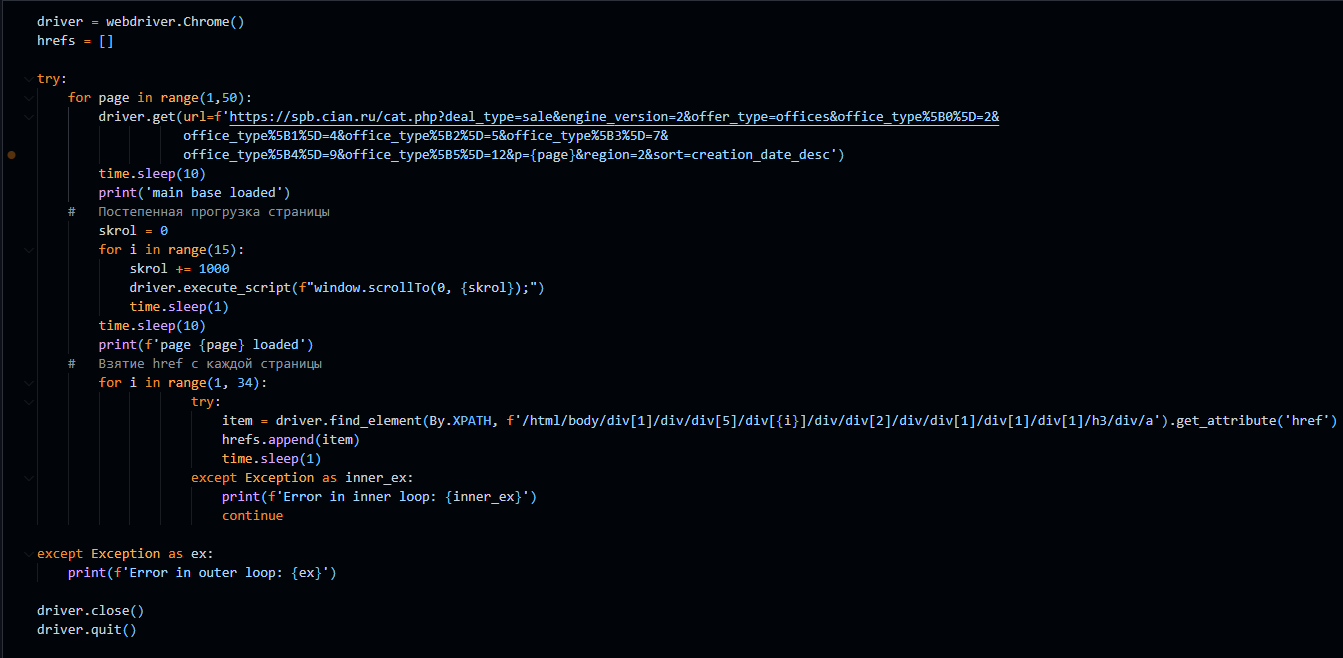


Рисунок 1 – Парсер ссылок объявлений.

Данный код включает в себя цикл с искусственными задержками, для прохождения по каждому из 34 объявлений на всех 50 страницах сайта с определенным нами заранее условиями фильтрации. В цикле предусмотрено использование скрипта для вертикального перемещения по странице, с целью загрузки и обработки всех объявления, взятие href-ссылок объектов, то есть url-ссылок опубликованных объявлений, учитывая структуру кода сайта «Циан».

Также было предусмотрено возникновение ошибок, почему после возникновения ошибки сам процесс парсинга не прекращается. Каждая ссылка объявления, при этом, записывается в список и затем сохраняется.

После успешного парсинга ссылок на объекты, возникает надобность парсинга самих объявлений, а для данной работы необходимо провести доскональное изучение кода страницы объявлений чтобы ответить на основные вопросы при такого рода парсинге.

Какие объекты размещены? Какой путь они имеют? В каком формате те или иные данные записываются? Имеет ли значение в каком порядке эти данные собираются? Обратившись к структуре кода одной из страниц была написана программа-парсер [приложение А].

В данном коде происходят действия сбора данных по циклу всех объектов. При открытии каждого объявления, происходит задержка в 10 секунд для полной загрузки страницы и, соответственно, для корректного функционирования парсера. На каждой странице происходит обращение к следующим объектам с извлечением текста и дальнейшим сохранением его, вместе с ссылкой на данное объявление в соответствующем с кодом порядке:

* Название объявления;
* Цена объекта;
* Цена за квадрат;
* Площадь м^2;
* Этаж;
* Свободно ли помещение;
* Город;
* Район;
* Округ;
* Улица;
* Номер дома;
* Ссылка на страницу продавца;
* Описание;
* Список ближайших станции метро (мин);
* Список ближайших выездов на шоссе (км).

Работа данного кода суммарно занимает приблизительно 7 часов с учетом обработки всех 1348 объявлений, полученных с работы предыдущего парсера. Такая длительность объясняется наличием задержек для прогрузки страниц и дополнительного времени для сбора информации с каждой страницы.

При изучении написанного парсера, необходимо обратить внимание на структуру внутри цикла. Для каждого из объектов используются операторы “try”, в случае если на какой-либо из страниц не окажется данного объекта или возникнет проблема с его интерпретацией библиотекой selenium. Такая система используется для всех объектов.

Однако ввиду особой структуры объектов, указывающих информацию о расположенных вблизи метро и выездов на шоссе, необходимо было сделать некоторые изменения, а именно:

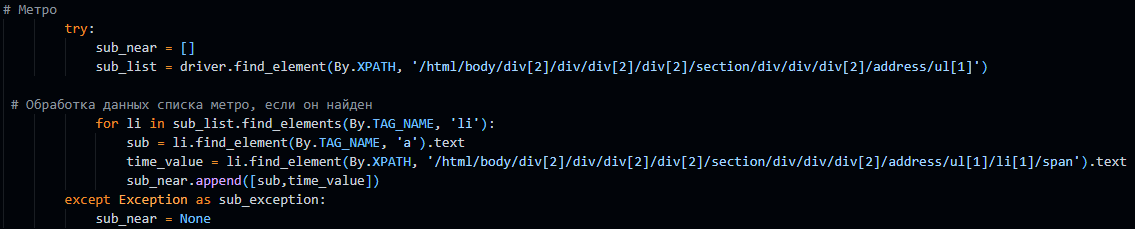


Рисунок 2 – Обработка списков метро в рамках парсера.

Поиск элемента, в котором содержатся данные о станциях метро (шоссе), запуск цикла для каждого из элементов в так называемом «списке», затем вывод названия станции метро в отдельную переменную и затем поиск элемента, указывающего на время (км) до метро (шоссе), после чего такой элемент добавляется в список станций метро (выездов на шоссе) у этого объекта.

В случае если у объекта не указаны станции метро (шоссе) – значения столбца остается пустым.

Также стоит отметить, что все эти данные нуждаются в тщательной обработке, ведь все они имеют на данный момент текстовый или даже списочный формат и не подразумевают какой-либо практической ценности.

Рассмотрим пример полученных нами данных:



Рисунок 3 – Пример данных.

## **ОБРАБОТКА ДАННЫХ**

Обработка первоначальных, так называемых «грязных» данных, является одной из важнейших частей Data-деятельности. Очень важно не потерять данные, обнаружить все ошибки и аномалии, ведь для построения дальнейших моделей нужно иметь подготовленные данные, не вызывающие, как и логических, так и технических конфликтов.

В первую очередь необходимо определить столбцы, которые могут быть полезны для нас и нести смысл, который позволит или более подробно изучить объект или окажется значимым фактором для построения модели.

Одним из важных факторов является площадь самого объекта. Такие данные нам нужны в числовом формате, а именно – float, ведь площадь может быть не целым числом. Для обработки данного столбца используем функции replace и strip и дополнительно библиотеку re для очищения символов единиц измерения площади и любых других символов возле. Используя указанные функции очистим столбец от лишних пробелов и заменим запятые на точки, после чего сможем перевести столбец из str-значения во float. Первоначальная обработка с площадью закончена. Похожие действия проведем и со столбцами полной цены и цены квадратного метра, только вместо единиц измерения площади будут символы рубля.

Обращаясь к следующему столбцу «free» - видим, что свободен ли объект описывается словами – заменяем соответствующие записи на 0 в случае, если объект свободен, и 1, если он занят. При подробном изучении было найдено определенное количество объектов, которые содержали не совсем точные данные касательно занятости объекта, поэтому предварительно были помечены как занятые.

Следующей задачей стояло обработка значений этажа. Согласно сайту «Циан», в объявлении указывался этаж самого объекта, а также максимальный этаж здания, в котором объект находился. Для корректного вывода информации и сохранении этажа и максимального этажа здания – напишем функцию [приложение Б] для вывода этих данных и создадим два отдельных столбца для них.

Наличие подробного адреса – один из важных фактов выбора именно площадки «Циан», ведь с помощью точного адреса есть возможность преобразовать такого рода данные в координаты и затем использовать их для геоспатического анализа и построения моделей. Для этого необходимо было объединить все столбцы с городом, районом, округом, улицей и домом в одну строку, приведя их при этом в формат, по которому можно было бы найти этот объект на картах.

При попытках обработки столбцов с данными по шоссе возникли проблемы вопроса интерпретации возможного формата обработанных данных. Представленные в виде текста станции метро не позволяли бы учесть их как фиктивные переменные, к тому же был важен фактор дистанции до метро, а циан обозначал его в минутах и мог указывать это время либо как на транспорте, либо пешим ходом, что добавляло проблемы в интерпретации. Такие же проблемы возникли и с шоссе, однако ввиду большего количества уникальных выездов на шоссе – решение такого рода проблем было найдено только для метро.

Учитывая работу с координатами, было решено собрать координаты станций Санкт-Петербургского Метрополитена и затем рассчитать расстояние до объектов недвижимости, а также количество станций метро в радиусе нескольких километров.

## **РАБОТА С КООРДИНАТНЫМИ ДАННЫМИ**

Для обработки адресов и перевода их в систему координат было решено обратиться к геокодеру от «Яндекс». Был изучен API геокодера и способы его применения. Первоначально стоит обратиться к термину.

Геокодер — это программа, которая переводит адрес текстового формата в географические координаты согласно открытой базе данных того или иного источника.

После ознакомления с особенностями геокодера – был создан ключ API для использования его внутри среды кода на языке программирования python и написана функция для отправления запросов в виде адреса и получения результата в формате координат. Также имеются определенные ограничения для бесплатного использования геокодера в рамках научной работы, одним из таких ограничений, оказавших влияние на обработку данных, был лимит на кол-во запросов в сутки – 1000.



Рисунок 4 – Код получения координат с помощью API геокодера.

Также были учтены возможные возникновения ошибок или факт ненайденных координат.

Данная функция была использована в цикле для получения координат по всему списку данных. Так как данных больше 1000, добавление координат было в 2 этапа. Вместе со вторым этапом также были собраны координаты открытых и ремонтирующихся станций метрополитена в Санкт-Петербурге. На момент отправления запросов – их число составляло 72.

Динамику и статистику запросов также можно рассмотреть в отдельном блоке сайта геокодера.

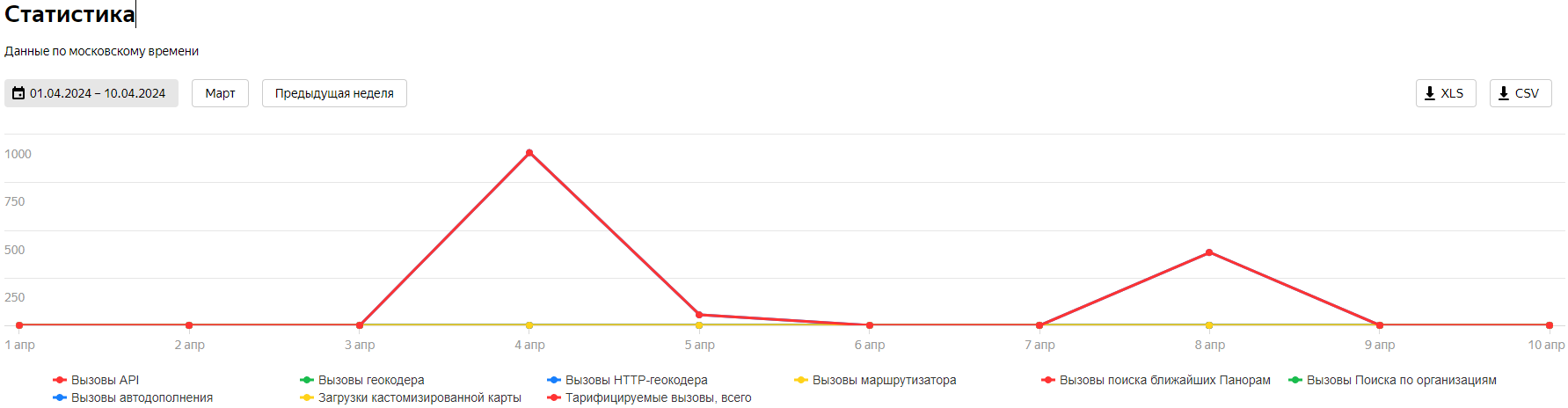


Рисунок 5 – Статистика запросов геокодера.

Для получения максимума пользы из информации по метро, рассчитаем к основным данным расстояние до всех станций метро, чтобы потом мы могли определить, какие станции оставить, а какие убрать. К тому же рассчитаем количество станций метро на определенных расстояниях от объекта. Для этого обратимся к формуле расчета расстояния между двумя точками на координатной плоскости Земли:

Где:

L — искомое расстояние между пунктами 1 и 2;

q — расстояние в одном градусе, константа ~ 111.2 км;

φ₁, φ₂ — географические широты пунктов 1 и 2;

λ₁, λ₂ — географические долготы пунктов 1 и 2.

Используя эту формулу, рассчитываем расстояние между каждым объектом и каждой станцией метро, используя циклы. Однако стоит учитывать возможное возникновение погрешностей и неточностей.

Помимо этого, предварительно была найдена платная база данных по Санкт-Петербургу, включающая основные объекты, по типу зданий, парковок, ж/д путей, дорог и так далее.

Для поставленной задачи – прогнозирования стоимости квадратного метра торгового помещения – наличия парковок является достаточно важным фактором. Это подтверждается и с точки зрения рассуждения, и при обращении к источникам, изучающим факторы влияния на стоимость объектов. Данные по координатам парковок были введены, а затем обработаны с учетом расстояния до объектов. Был добавлен столбец, содержащий количество парковок в радиусе одного километра от каждого из объектов.

К имеющимся данным было решено добавить разделение объектов на районы Санкт-Петербурга, используя фиктивные переменные. За базовое значение было решено взять Центральный район, при этом стоит отметить, что в выборке не оказалось данных по следующим районам: Колпинский, Курортный, Кронштадтский. В связи с чем, данные факторы были убраны из наших данных.

Для выполнения этой задачи был также использован платный набор геопространственных

данных по Санкт-Петербургу, разделяющий город на существующие 18 районов. Из имеющегося набора полигонов в рамках системы координат – были собраны названия районов в отдельный список. После чего были импортированы библиотека shapely, позволяющая работать с полигональными и координатными данными. С помощью метода loads данные были конвертированы в формат полигональных данных и затем, была написана функция для проверки, в какой из районов входит объект и последующего обновления столбцов соответствующих районов:

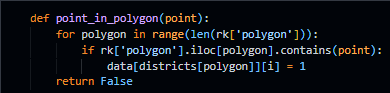


Рисунок 6 – Функция для определения района объекта.

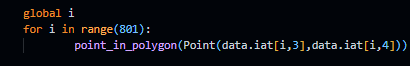


Рисунок 7 – Применение функции для всего набора данных.

После этого также убираем столбец с центральный районом, для корректного использования фиктивных переменных.

# **АНАЛИЗ ДАННЫХ И МАШИННОЕ ОБУЧЕНИЕ**

## **ГЕОГРАФИЧЕСКИЙ АНАЛИЗ**

С учетом имеющихся географических данных, с помощью свободной кроссплатформенной геоинформационной системы – QGIS, предназначенной для создания, редактирования, визуализации, анализа и публикации геопространственной информации, выведем их на координатную систему с наложенной на нее картой Санкт-Петербурга для проведения ознакомительного анализа.

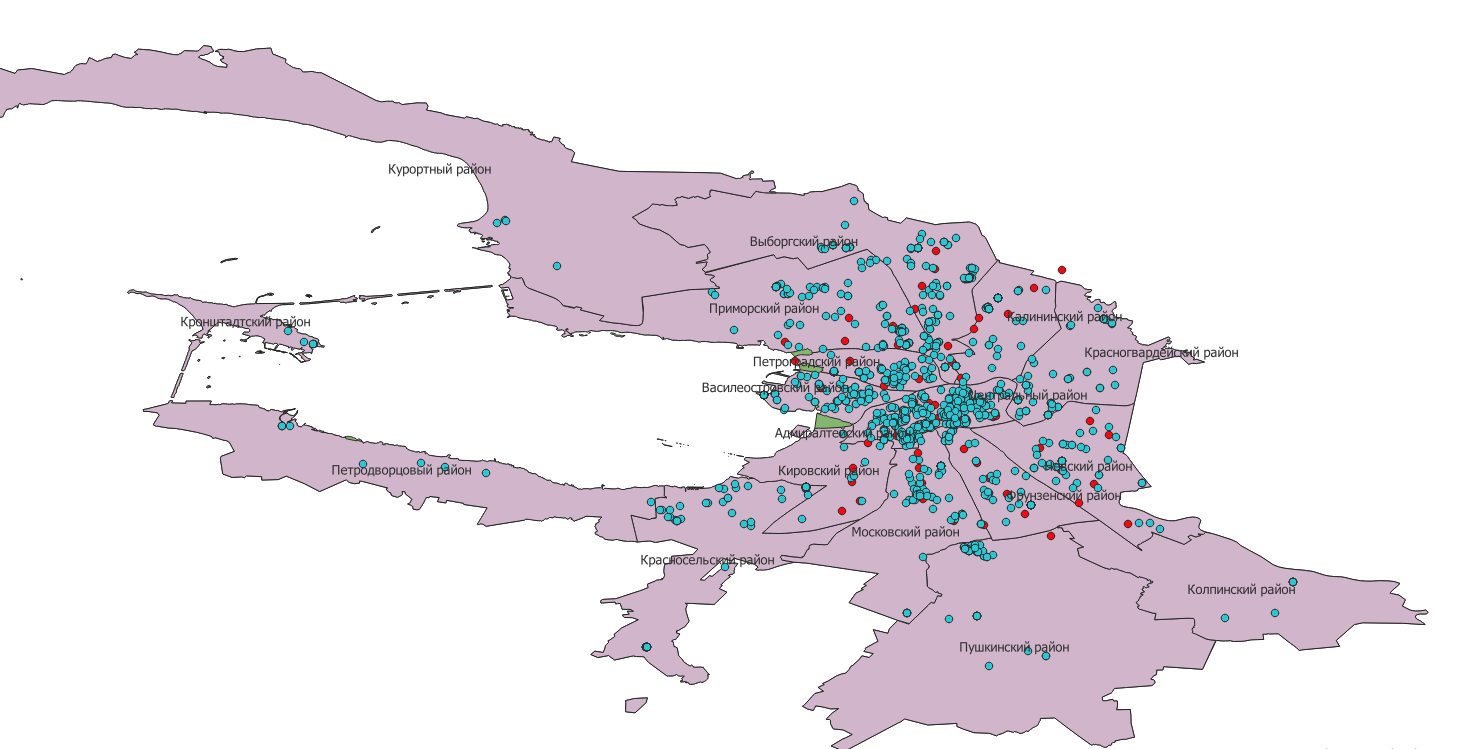


Рисунок 8 – Визуализация объектов в QGIS.

На данной карте расположен Санкт-Петербург, разделенный на административные деления – районы. Синим цветом на данном рисунке обозначены объекты, которые находятся в собранных данных, красным – станции метро.

При рассмотрении более подробно – можно также обнаружить, что на карте обозначены все здания и парковки, существовавшие в Санкт-Петербурге на 2022 год, это поможет нам проверить совпадение значений координат из данных с реальными зданиями.

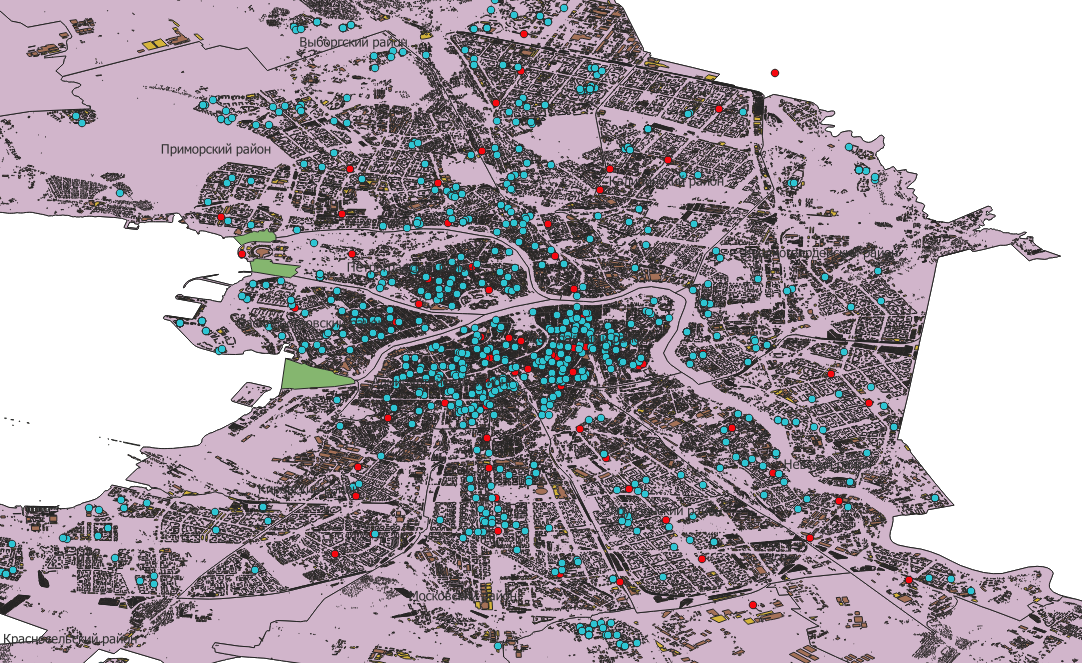


Рисунок 9 – Приближенная визуализация с полигонами зданий и парковок.

Предварительно можем отметить, что основная доля коммерческих объектов из нашей выборки располагается в Центральном и Адмиралтейском районах, в остальных районах распределение примерно одинаковое между собой, не считая, конечно же, удаленных от центра районов города.

При этом стоит отметить, что хоть QGIS критически важен в плане визуального и общего анализа и, соответственно, понимания данных – работа с таким ресурсом требует большого количества часов для изучения и использования на практике всех возможных методов, которые позволяют работать с данными, поэтому к более подробному изучению данной системы было решено вернуться позднее.

## **АНАЛИЗ ЗНАЧИМОСТИ ПАРАМЕТРОВ**

Теперь же возникает немного другого рода проблема – чрезмерное количество факторов в нашем наборе данных, которые при этом создают большую межфакторную корреляцию – дистанции до станций метро. Выведем корреляционную матрицу частей столбцов нашего Dataframe.

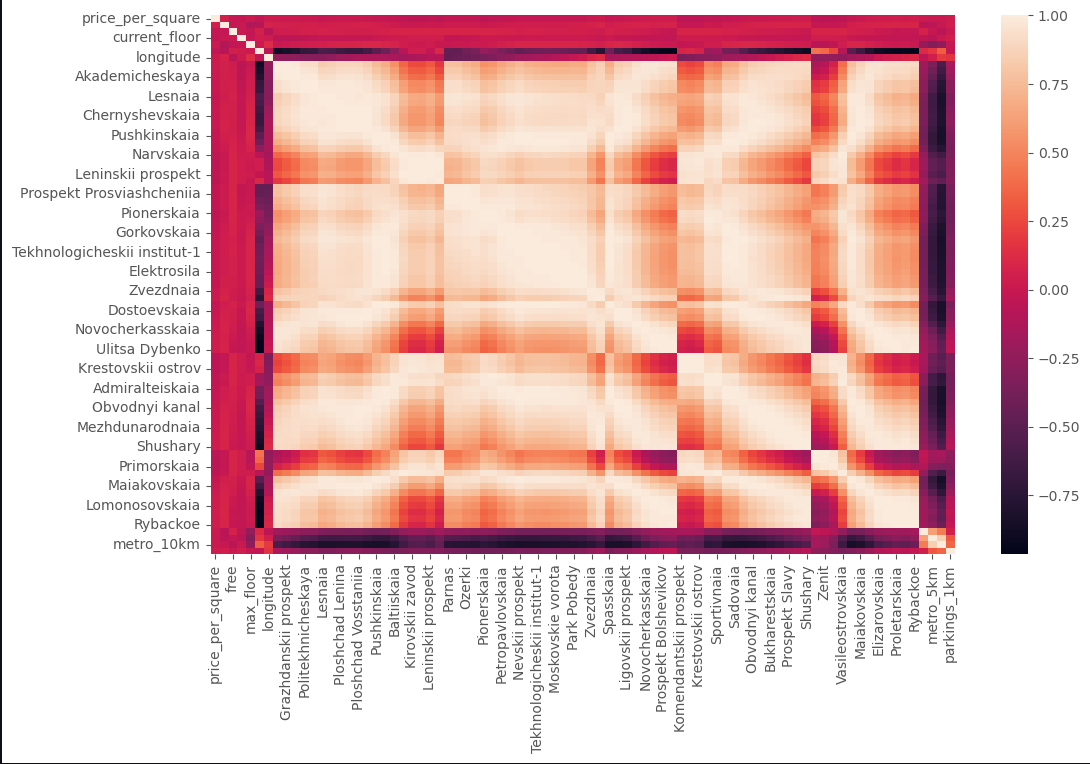


Рисунок 10 – Корреляционная матрица части параметров.

Было решено избавиться от большей части признаков на данном этапе модели и в случае необходимости – все же учесть некоторые станции. Также была изучена столбчатая диаграмма, показывающая корреляционную связь с целевым признаком:

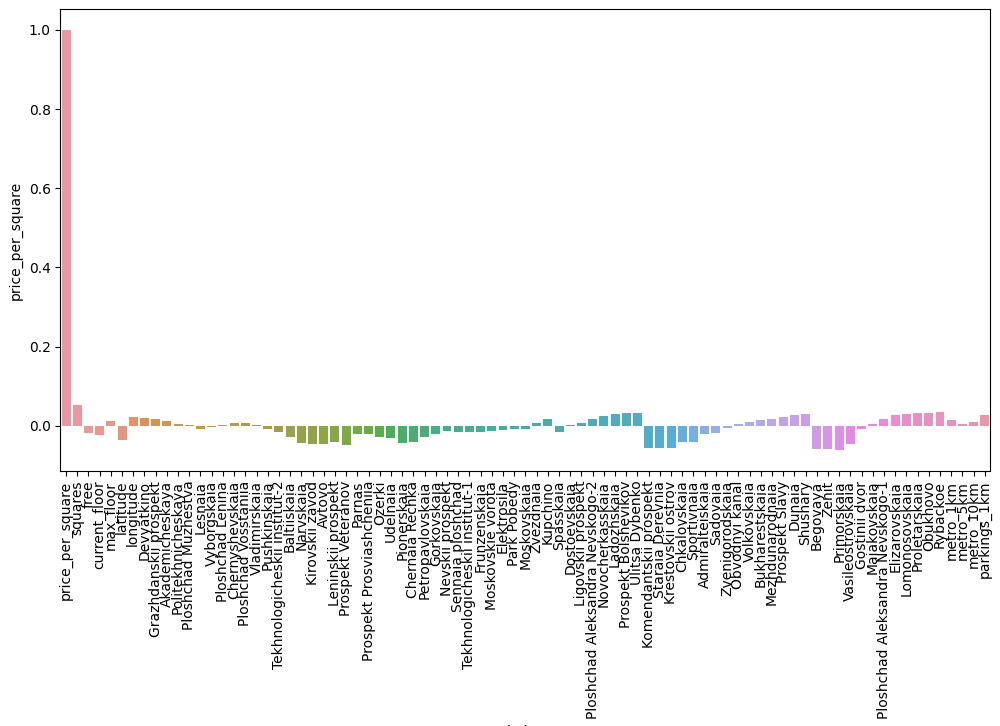


Рисунок 11 – Диаграмма корреляции параметров по метро.

По ней, во многом, можно сделать заключение о большой роли отрицательного влияния конечных станций на стоимость квадратного метра. Было испробовано несколько вариантов включения определенного количества факторов станций, в особенности использование только конечных станций метрополитена и Невского проспекта, однако во всех вариантах они вызывают корреляцию между собой и не оказывают значимого влияния на все параметры прогнозирования целевого параметра. Это же показывает и вывод регрессионной статистики, созданный для рассмотрения влияния параметров станций метрополитена и демонстрирующий незначимость параметров и неспособность объяснить целевой параметр.

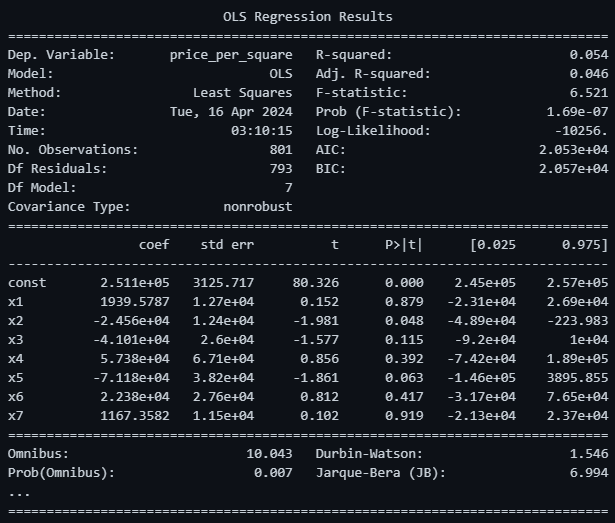


Рисунок 12 – Вывод регрессионной статистики по одному из наборов параметров.

Поэтому было решено на данный момент оставить только станцию метро - Невский проспект, ведь большинство объектов расположены в центре или вокруг центра, а данная станция во многом и представляет собой центр города, что позволит нам также в дальнейшем интерпретировать данный параметр как влияние расстояния от центра Санкт-Петербурга.

Рассмотрим наличие влияния фиктивных параметров между собой и с целевым параметром, а также проверим значимость параметров с учетом также и основных факторов в рамках поставленной задачи.

Для этого мы напишем код, создав отдельный объект – dataframe, в котором с помощью библиотеки scipy используем функцию pearsonr, позволяющую нам вывести p-value для всех используемых факторов:

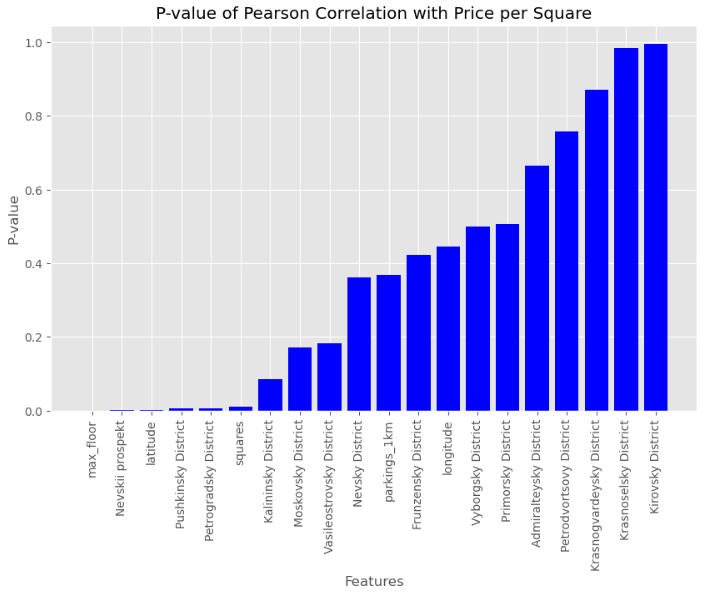


Рисунок 13 – Диаграмма значимости параметров и фиктивных переменных по районам.

Анализируя выявленные результаты, можно в целом утверждать, что никакой статистически различимой значимости параметров – фиктивных переменных районов Санкт-Петербурга – не может быть в рамках данной задачи. Возможно, причина лежит в необходимости большего объема равномерных данных – больше помещений в каждом из районов, для обнаружения серьезной зависимости в рамках разных районов.

Учитывая имеющиеся результаты проведенного анализа данных, стоит учесть только самые необходимые параметры, которые показывают наличие статистического смысла в виде зависимости с целевым показателем и наличие логического смысла в рамках данного бизнеса. На данный момент таковыми являются:

* price\_per\_square (Целевой признак, стоимость кв. метра);
* squares (Площадь объекта);
* max\_floor (Максимальный этаж здания);
* longitude (Долгота);
* latitude (Широта);
* Nevskii prospekt (Удаленность от центра);
* parkings\_1km (Количество парковок в радиусе 1 км).

## **ОБРАБОТКА ДАННЫХ И ПОДГОТОВКА К ПОСТРОЕНИЮ МОДЕЛЕЙ**

Для дальнейшей работы нам все-таки необходимо снова обратиться к методам обработки данных, однако уже не первичной – необходимо обработать выбросы, аномальные значения и пропуски в столбцах данных, которые могут иметь значимость для нас как исследователей.

При изучении объявлений и столбцов с данными были обнаружены ошибки в оформлении информации об объектах, в связи с чем цена за квадратный метр была ниже, чем само количество квадратных метров – в связи с чем есть необходимость в удалении указанных данных из нашей выборки. Также, учитывая тенденции рынка и его средние значения, а также консультацию с агентами по недвижимости на предприятии – было решено не учитывать в выборке торговые объекты с квадратурой больше 200 метров квадратных. Они представляют собой несколько иную подкатегорию, нежели изучаемую в данной работе, и имеют очень низкую востребованность ввиду своей дороговизны, в связи с чем гораздо дольше находятся на рынке и корректируют свою цену для привлечения инвесторов или покупателей.

Помимо этого, при определении целевого признака – цены за квадратный метр, возникла надобность в исключении полной стоимости объекта, она бы не позволяла рассматривать построение моделей без учета сильной межфакторной корреляции, по этой же причине данный фактор и не был выбран как целевой.

Целевой признак также нуждается в обработке, ведь помимо обработки аномальных значений, стоит задача максимальной нормализации данных – для улучшения результатов будущих моделей.

## **ВИЗУАЛИЗАЦИЯ ОСОБЕННОСТЕЙ ДАННЫХ**

При изучении целевого признака, обратим внимание в первую очередь на распределение параметра:

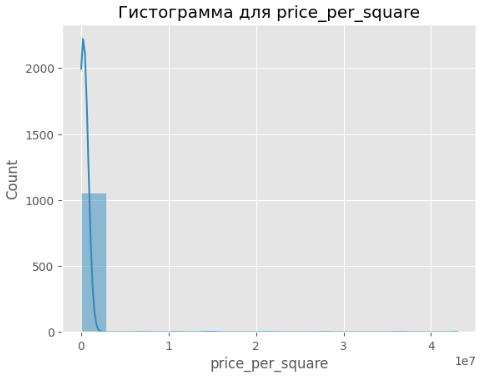


Рисунок 14 – Распределение целевого признака.

С учетом такого результата, обратим внимание на аномально высокие и критически низкие показателя стоимости квадратного метра. Стоит при этом учесть, что такого рода данные по большей части были ошибочны определены в исходном источнике, однако даже если некоторые из них не ошибочные – такие данные во всяком исходе не пригодны для нашего исследования, так как попадают кардинально в другое направление коммерческой сферы.

Поэтому мы обрабатываем данные, исключая объекты со стоимостью квадратного метра менее 30 000 рублей и более 450 000 рублей. При этом также убираем данные ниже 5% квантиля и выше 95%, чтобы даже с учетом исключенных данных – устранить разброс значений, нормализовать распределение. Выведем гистограмму и получим следующий результат:

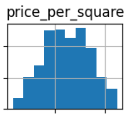


Рисунок 15 – Распределение обработанного целевого признака.

Выведем подобную статистику для каждого оставшегося параметра в наших данных:

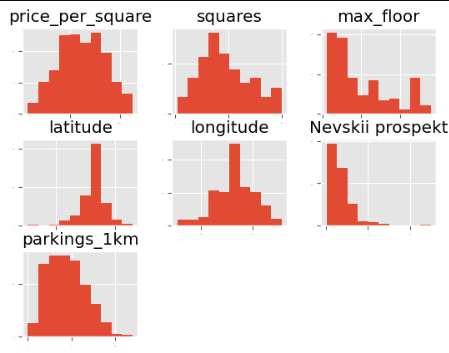


Рисунок 16 – Распределение используемых параметров.

Видим, что по большей части данные нормализованы или не могут быть откорректированы, за исключением расстояния от центра. Рассмотрим количество объявлений, объект которых находится на разных расстояниях от центра, определим с какого расстояния, можно считать, что объекты нерелевантны или не настолько востребованы, чтобы оставлять наличие выбросов.

Путем практического исследования было решено остановиться на 30 км от центра – таким образом мы не исключаем все объявления за чертой города, однако не учитываем чрезмерно удаленные, позволяя нам больше отталкиваться от цен города. Смотрим на изменения, а также на графики плотности данных параметров:

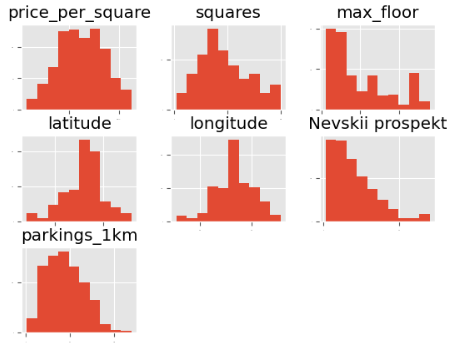


Рисунок 17 – Распределение обработанных параметров.

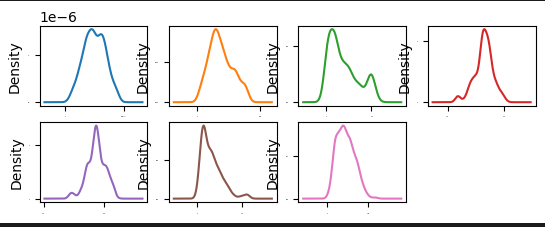


Рисунок 18 – Плотность обработанных параметров.

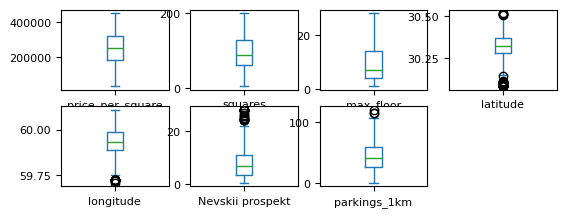


Рисунок 19 – Boxplot`ы обработанных параметров.

Не менее важно рассмотреть корреляционную зависимость между оставшимися параметрами. С учетом недостатка данных и основной зависимости стоимости объекта от координат, результаты приемлемы, но безусловно должен подниматься вопрос о нахождении большего количества параметров, которые могут оказать существенное влияние на целевой параметр – стоимость квадратного метра.

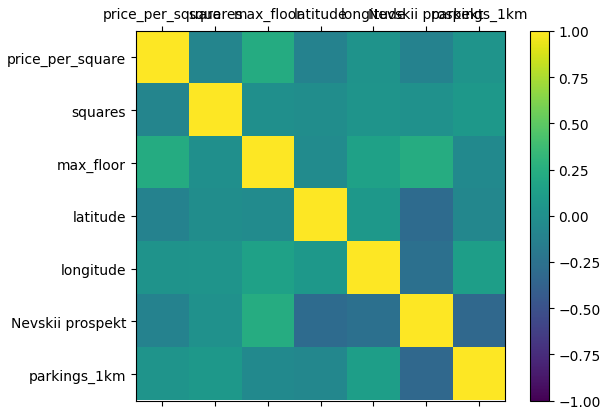


Рисунок 20 – Корреляция обработанных параметров.

После использования указанных методов разработки выводим количество строк с данными объектов, которые мы будем использовать для построения моделей. Их число - 801 объект.

## **ПОСТРОЕНИЕ МОДЕЛЕЙ**

При изучении источников, рассматривавших самые эффективные способы прогнозирования цен на коммерческую недвижимость – основной вектор направления стоял к случайному лесу или методу ближайших соседей - однако для структурного понимания данных необходимо рассмотреть всевозможные варианты.

Первоначально – разделим данные на тренировочную и тестовую выборки. В данном случае было выбрано распределение в 50%, а а основным параметром для сравнения результатов модели будем считать MAPE - среднее абсолютное процентное отклонение прогнозируемого значения от реального. Напишем код, для выведения данной статистики по основным регрессионным методам на обычных данных с использованием кросс-валидации с заранее предопределенными параметрам. Выведем результаты в виде boxplot`а:

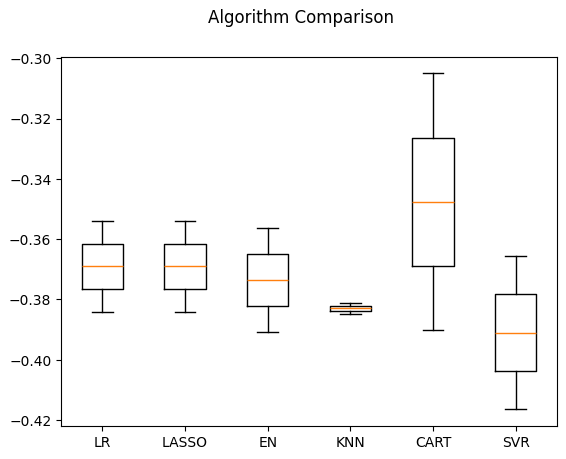


Рисунок 21 – Сравнение вывода MAPE регрессионных алгоритмов на нестандартизированных данных.

Видим, что в рамках прогнозирования на нестандартизированных данных примерно одинаково себя показывают данные методы, рассмотрим среднее значения:

LR: -0.369073;

LASSO: -0.369071;

EN: -0.373562;

KNN: -0.382985;

CART: -0.350150;

SVR: -0.390972.

Видим, что в среднем самый лучший (минимальный) результат показывает регрессор деревьев решений (CART), однако имеет достаточно большой разброс, поэтому в данном случае, предпочтительней было бы использовать метод ближайших соседей (KNN), ввиду его стабильности в показаниях.

Теперь стандартизируем данные и применим их для построения этих же моделей:

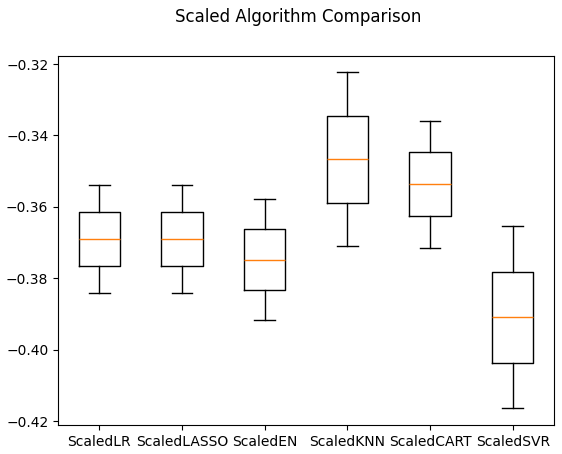


Рисунок 22 – Сравнение вывода MAPE регрессионных алгоритмов на стандартизированных данных.

Видим, что результаты не сильно улучшились по сравнению с не масштабированными данными, скорее даже ухудшились. Рассмотрим значения повнимательнее:

ScaledLR: -0.369073;

ScaledLASSO: -0.369073;

ScaledEN: -0.374864;

ScaledKNN: -0.346791;

ScaledCART: -0.353721;

ScaledSVR: -0.390965.

Все еще наиболее предпочтительным стоит считать метод ближайших соседей (ScaledKNN), хоть при стандартизированных данных стандартное отклонение значительно увеличилось по сравнению с прошлыми результатами. Линейная регрессия, L1 и комбинация L1&L2 не показали изменений по сравнению с предыдущими расчетами.

Перейдем к рассмотрению ансамблевых методов, а именно – Адабустинг, Градиентный бустинг, Случайный лес и Extra-Trees.

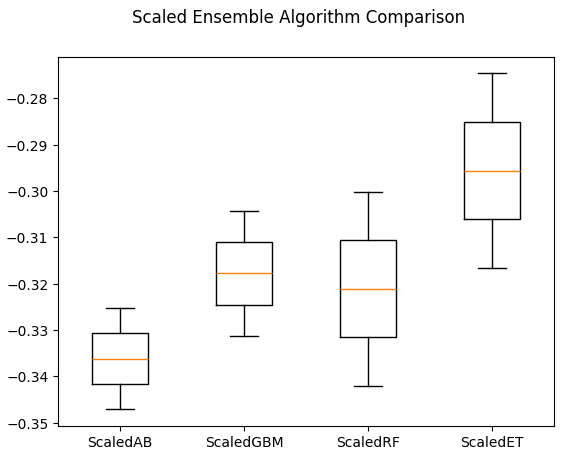


Рисунок 23 – Сравнение вывода MAPE ансамблевых алгоритмов на стандартизированных данных.

Лучший результат показывает Extra-Trees Regressor, со средним значением MAPE в 0.304085 и стандартной девиацией в 0.028873. Рассматривая другие параметры, видим –

ScaledAB: -0.343236 (0.019689);

ScaledGBM: -0.318066 (0.015337);

ScaledRF: -0.313031 (0.015420).

С учетом указанных результатов стоит присмотреться именно к Extra Trees, учитывая его значение MAPE. Стоит отметить и градиентный бустинг, у данного метода относительно небольшой разброс и его приемлимое значение процента абсолютного отклонения. Применим метод GridSearch для поиска наилучших параметров для Extra-Trees модели. Согласно выводу сети – лучший результат MAPE достигается при 150 базовых моделях на тестовых данных и равен 0.292196. После обучения модели с такими параметрами на тренировочной выборке и применении модели на тестовой получаем следующие результаты. Среднее абсолютное отклонение равняется ~56660 рубля, что в процентном отклонении равняется 35.3%.

Для понимания качества результата – обратимся к более подробному рассмотрению модели – влияние параметров. Ввиду невозможности интерпретировать ансамблевые методы настолько же точно, как регрессионные – обратимся к методам SHAP. Это метод интерпретации моделей машинного обучения, который предоставляет объяснения для индивидуальных прогнозов. Он основан на концепции кооперативных игр и использует значения из теории игр для определения вклада каждого признака в прогноз модели. Сами SHAP-значения — это оценки вклада каждого признака в прогноз модели для конкретного наблюдения данных. SHAP-значения представляют собой разницу между средним прогнозом модели и прогнозом модели для данного наблюдения при условии определенного набора признаков.

Рассмотрим предварительно важность признаков, используемых в модели.

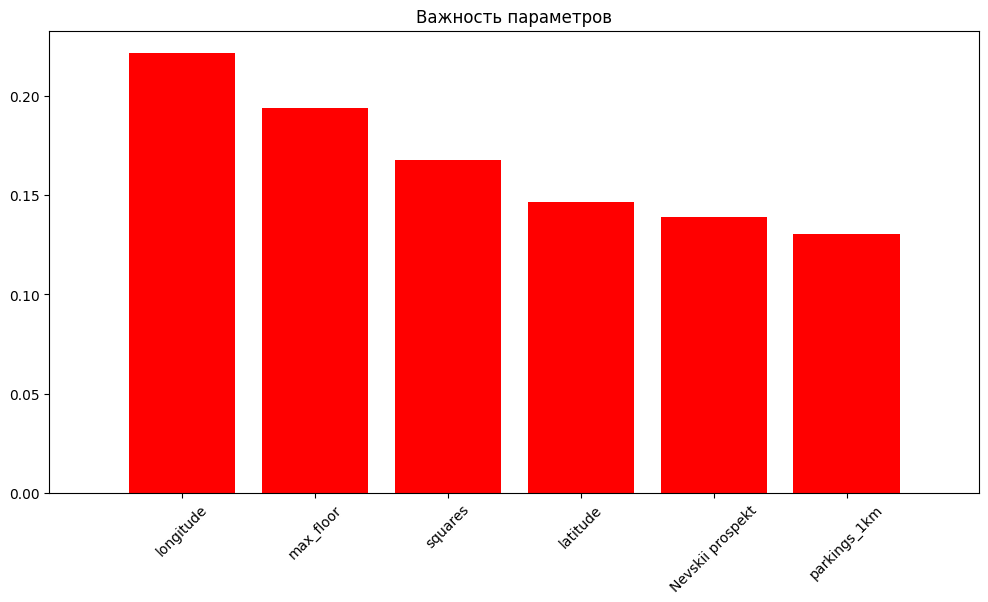


Рисунок 24 – Сравнение важности параметров для модели ETR.

Переходя к интерпретации роли параметров методами SHARP – стоит отметить основные принципы – на данном графике представлены по оси Y – ранжированные по важности параметры, важность отдельных наблюдений можно также заметить благодаря цветовой разметке, красные наблюдения оказались более важными, более влиятельными для модели, чем синие. По оси X же располагаются SHAP-значения, которые, как уже было сказано отображают вклад каждого признака, отрицательный или положительный, в прогноз модели относительно среднего прогноза. Распределение наблюдений по оси X позволяет оценить как в среднем каждый из параметров влияет на прогнозное значение.

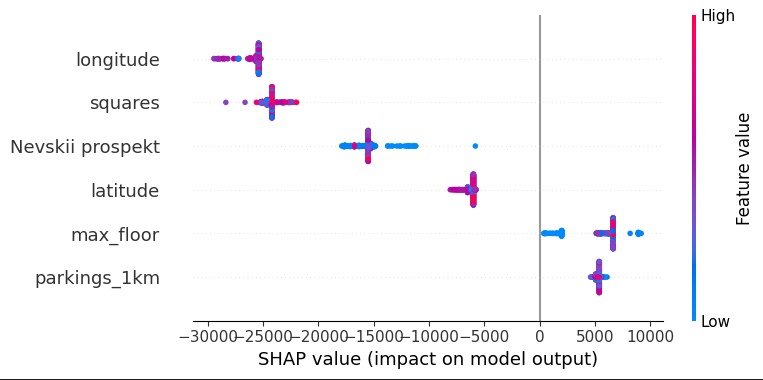


Рисунок 25 – Сравнение влияние параметров для модели ETR согласно SHAP.

Если рассматривать средние значение, получится выявить более структурную интерпретацию:

* squares -24237.017969;
* max\_floor 5622.295619;
* latitude -6181.290438;
* longitude -25707.471252;
* Nevskii prospekt -15534.455458;
* parkings\_1km 5290.673031.

Согласно модели, получается, что в среднем площадь объекта недвижимости снижает стоимость квадратного метра данного объекта на 24 тысячи 237 рублей; в среднем, максимальный этаж в здании объекта повышает стоимость его квадратного метра на 5 тысяч 622 рубля; в среднем, координаты широты объекта понижают стоимость квадратного метра объекта на 6 тысяч 181 рубль; в среднем, координаты долготы объекта понижают стоимость объекта на 25 тысяч 707 рублей; в среднем, расстояние объекта от центра Санкт-Петербурга понижают стоимость его квадратного метра на 15 тысяч 534 рубля; в среднем, количество парковок в радиусе одного километра от объекта, повышает стоимость квадратного метра объекта на 5 тысяч 290 рублей.

# **ЗАКЛЮЧЕНИЕ**

Полученные результаты говорят о недостаточности используемых данных в рамках решения вопроса создания модели прогнозирования, которая направленна на минимизацию ошибки. Это предполагает поиск, и использование более критически важных данных, осуществление более трудоемкой работы с их обработкой и подразумевает поиск новейших методов машинного обучения, способных нивелировать недостатки выборки и улучшить результаты прогнозов.

Тем не менее, проделанная работа оказала значимое воздействие на возможности применения практических способностей в рамках планирования деятельности, исследования темы и литературы узкой направленности, поисков данных, их обработки, исследования и дальнейшего построения моделей как для изучения рынка, так и для усовершенствования устоявшихся подходов к оценке недвижимости.

В дальнейшем планируется критическое усовершенствование результатов модели, добавление новых параметров и создание функционирующей среды для возможности практического использования.

# **СПИСОК ИСТОЧНИКОВ**

1. Beautiful Soup Documentation // beautiful-soup URL: https://beautiful-soup.readthedocs.io/en/latest/ (дата обращения: 20.03.2024).
2. Dr. Sweta R. Kumar, Swati Bhatt, Hasan Phudinawala Predicting House Price with Deep learning: A comparative study of Machine Learning Models // International Journal for Multidisciplinary Research. – 2023
3. git API // git URL: https://git-scm.com/docs/api-credentials (дата обращения: 01.04.2024).
4. QGIS geo-analysis // docs.scipy.org URL: https://qgis3-10-geoanalysis-un.readthedocs.io/en/latest/ (дата обращения: 13.04.2024).
5. SciPy API // docs.scipy.org URL: https://docs.scipy.org/doc/scipy/reference/index.html (дата обращения: 10.04.2024).
6. Unleashing the Potential of Random Forest Regression: A Python Implementation Guide with Hyperparameter Tuning. // DEV URL: https://dev.to/newbie\_coder/unleashing-the-potential-of-random-forest-regression-a-python-implementation-guide-with-hyperparameter-tuning-167o (дата обращения: 05.04.2024).
7. WebDriver API // selenium-python URL: https://selenium-python.readthedocs.io/api.html (дата обращения: 20.03.2024).
8. Астраханцева Ирина Александровна, Смирнова Наталья Владимировна Оценка стоимости коммерческой недвижимости на основе моделей машинного обучения // Научные труды ВЭО России. - 2022. - №237
9. Геокодер // Яндекс URL: https://yandex.ru/maps-api/products/geocoder-api/?utm\_source=yandex&utm\_medium=cpc&utm\_campaign=fmt\_textcmp\_brand\_searchprod\_api-mapsgeo\_ru&utm\_content=cid\_99950950\_gbid=5325032826\_aid\_15308233919\_kid=48232840543\_adp\_no\_pos\_premium1\_src\_search\_none\_dvc\_desktop\_rtg\_48232840543&utm\_term=геокодер%20яндекс%20api&yclid=11291778458353663999 (дата обращения: 02.04.2024).
10. О. В. Мезенцева Оценка коммерческой недвижимости. - Екатеринбург: УрФУ, 2011. - 115 с (дата обращения: 28.03.2024).

# **ПРИЛОЖЕНИЕ А**

driver **=** webdriver.Chrome()

data\_spec\_all **=** {}

**try**:

**for** k **in** range(655,1349):

*#         Инициализация селениума*

        driver.get(url**=**hrefs[k])

        time.sleep(10)

*#         Название товара*

**try**:

            title **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[1]/div/h1').text

**except** Exception **as** title\_exception:

            title **=** **None**

*#         Цены товара*

**try**:

            price **=** driver.find\_element(By.XPATH,"/html/body/div[2]/div/div[2]/div[3]/div/div[1]/div[1]/div[3]/div/div[1]/span").text

**except** Exception **as** price\_exception:

            price **=** **None**

**try**:

            square\_price **=** driver.find\_element(By.XPATH,"/html/body/div[2]/div/div[2]/div[3]/div/div[1]/div[3]/div/div/div[1]/span[2]").text

**except** Exception **as** sq\_price\_exception:

            square\_price **=** **None**

**try**:

            squares **=** driver.find\_element(By.XPATH,"/html/body/div[2]/div/div[2]/div[2]/div[3]/div[1]/div[2]/span[2]").text

**except** Exception **as** sq\_exception:

            squares **=** **None**

**try**:

            floor **=** driver.find\_element(By.XPATH,"/html/body/div[2]/div/div[2]/div[2]/div[3]/div[2]/div[2]/span[2]").text

**except** Exception **as** floor\_exception:

            floor **=** **None**

**try**:

            free **=** driver.find\_element(By.XPATH,"/html/body/div[2]/div/div[2]/div[2]/div[3]/div[3]/div[2]/span[2]").text

**except** Exception **as** free\_exception:

            free **=** **None**

**try**:

            city **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/div/div/a[1]').text

**except** Exception **as** city\_exception:

            city **=** **None**

**try**:

            district **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/div/div/a[2]').text

**except** Exception **as** distr\_exception:

            district **=** **None**

**try**:

            okrug **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/div/div/a[3]').text

**except** Exception **as** okrug\_exception:

            okrug **=** **None**

**try**:

            street **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/div/div/a[4]').text

**except** Exception **as** street\_exception:

            street **=** **None**

**try**:

            house\_number **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/div/div/a[5]').text

**except** Exception **as** house\_exception:

            house\_number **=** **None**

*#         Url продавца*

**try**:

            seller **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[3]/div/div[3]/div/div/div[1]/div/div[2]/div/div/div[1]/a').get\_attribute('href')

**except** Exception **as** seller\_exception:

            seller **=** **None**

**try**:

            description **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/div[5]/div/div/div/div/span').text

**except** Exception **as** description\_exception:

            description **=** **None**

*#         Метро*

**try**:

            sub\_near **=** []

            sub\_list **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/ul[1]')

*# Обработка данных списка, если он найден*

**for** li **in** sub\_list.find\_elements(By.TAG\_NAME, 'li'):

                sub **=** li.find\_element(By.TAG\_NAME, 'a').text

                time\_value **=** li.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/ul[1]/li[1]/span').text

                sub\_near.append([sub,time\_value])

**except** Exception **as** sub\_exception:

            sub\_near **=** **None**

*#         Шоссе*

**try**:

            highway\_near **=** []

            way\_list **=** driver.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/ul[2]')

*# Обработка данных списка, если он найден*

**for** li **in** way\_list.find\_elements(By.TAG\_NAME, 'li'):

                way **=** li.find\_element(By.TAG\_NAME, 'a').text

                dist\_value **=** li.find\_element(By.XPATH, '/html/body/div[2]/div/div[2]/div[2]/section/div/div/div[2]/address/ul[2]/li[2]/span').text

                highway\_near.append([way,dist\_value])

**except** Exception **as** highway\_exception:

            highway\_near **=** **None**

*# Добавляем данные с url товара*

        new\_data **=** {'url': hrefs[k],'seller\_url':seller,'title':title,'squares':squares, 'full\_price': price,

                    'price\_per\_square': square\_price,'city':city, 'district':district, 'okrug': okrug, 'street': street ,'house\_number':house\_number,

                    'description': description,'sub\_near':sub\_near,'highway\_near':highway\_near,'floor':floor,'free':free}

        cian **=** cian.append(new\_data, ignore\_index**=True**)

        time.sleep(5)

**except** Exception **as** ex:

    print(ex)

**finally**:

    driver.close()

    driver.quit()

# **ПРИЛОЖЕНИЕ Б**

**def** extract\_floors(text):

**if** pd.isna(text):

**return** np.nan

    match **=** re.match(**r**'(\d**+**)\s**\***из\s**\***(\d**+**)**|**(\d**+**)', str(text))

**if** match:

        current\_floor **=** int(match.group(1) **or** match.group(3))

        max\_floor **=** int(match.group(2) **or** match.group(3))

**return** current\_floor, max\_floor

**return** np.nan

df[['current\_floor', 'max\_floor']] **=** pd.DataFrame(df['floor'].apply(extract\_floors).tolist(), index**=**df.index)