Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования «Казанский (Приволжский) федеральный университет»

Институт математики и механики им. Н.И.Лобачевского Кафедра математического анализа

Направление: 02.03.01 – **Математика и компьютерные науки**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«СБОР ДАННЫХ И АНАЛИЗ ДАННЫХ С HTML-СТРАНИЦ»

Работа завершена:

Студент гр. 05-503

"__"_____2019 г. _____ Березин Н.С.

Работа допущена к защите:

Научный руководитель

Кандидат физико-математических наук,

ассистент кафедры математического анализа

"__"_____2019 г. _____ Новиков А.А.

Заведующий кафедрой

математического анализа

доктор физико-математических наук,

профессор

"__"_____2019 г. _____ Насыров С.Р.

Казань – 2019 год

Оглавление.

1. Введение.

Данная работа посвящена сбору данных с html страниц и последующим их анализом. Целью было собрать данные с объявлений на сайте недвижимости и построить регрессионную модель для их анализа. Регрессионная модель должна определить зависимость цены от других параметров указанных в объявлении, такие как район где расположена квартира, ближайшая станция метро, площадь квартиры. Также задачей было обучить данную модель с целью дальнейшего предсказания стоимости квартиры на основе собранных признаков.

Эта работа является актуальной, так как в настоящее время рынок недвижимости один из крупнейших рынков в мире. Только в России по данным Росреестра за первое полугодие 2018 года было продано более 4,2 миллионов объектов недвижимости. Из них более 3,9 миллионов было продано на вторичном рынке. [1] Исходя из этих данных можно сделать вывод о том, что система для рекомендации стоимости недвижимости была бы очень востребована на этом рынке.

В связи с этим был реализован алгоритм по сбору данных с объявлений на сайте по продаже недвижимости на языке PHP с последующей записью этих данных в базу данных с помощью системы управления базами данных MySQL и дальнейшим анализом этих данных на языке Python.

В пункте 2 приведены предварительные сведения в которых рассказывается, что такое язык HTML, как устроен HTML-документ, что такое язык PHP, что такое система управления базами данных MySQL и что такое язык Python.

В пункте 3 рассказано с помощью чего была сделана данная работа.

В пункте 4 расписано как был реализован алгоритм на языке PHP по сбору данных и записи их в базу данных используя СУБД MySQL, а также алгоритм анализа этих данных на языке Python.

В пункте 5 приведены итоги по проделанной работе.

В пункте 7 представлены код алгоритма на языке PHP, код базы данных на языке SQL и код алгоритма на языке Python.

2. Предварительные сведения.

2.1. Язык HTML.

HTML — язык разметки веб-документов. Позволяет создавать веб-страницы, которые потом отображаются в окне браузера компьютера или мобильного устройства. HTML 5 – последняя версия языка. В этой версии он имеет большое количество возможностей, таких как создание таблиц, отображение картинок, музыки или видео. [2]

2.2. Структура HTML-документа.

HTML-документ состоит из открывающихся и зарывающихся тегов. Каждый HTML-документ должен начинаться с тега DOCTYPE который определяет версию используемого HTML и также определяет соответствующий DTD-файл в интернет. DTD-файл – это XML-документ, который определяет какие теги, атрибуты и их значения действительны для конкретной версии HTML. После тега DOCTYPE, идет тег <html>. Элементы, которые находятся между тегами <html> и </html> образуют дерево элементов, где элемент <html> является корневым.

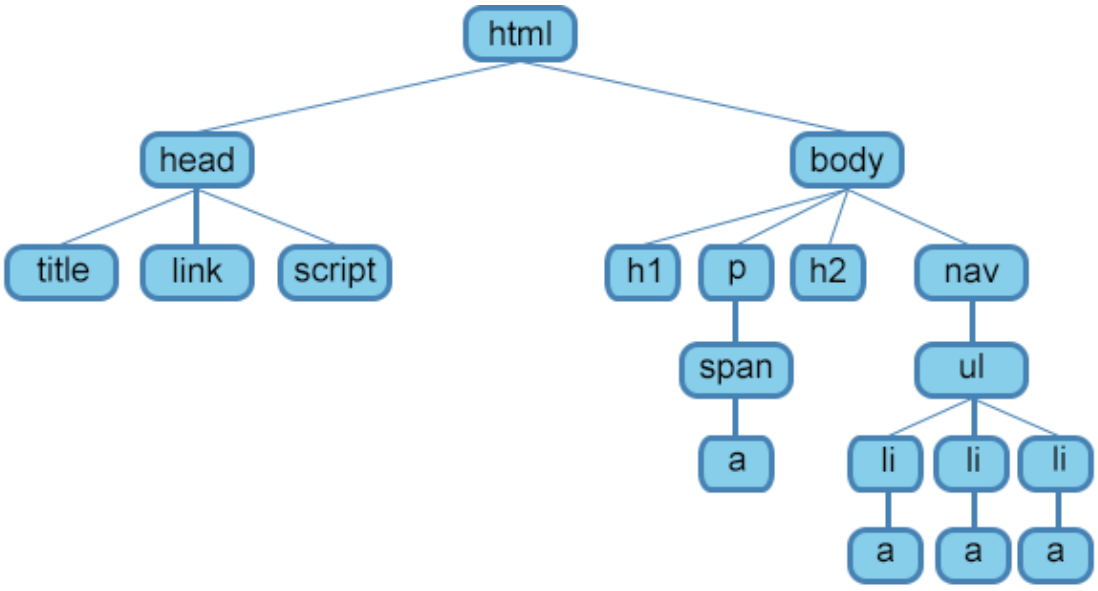


Рисунок 1. Простейшая структура HTML-документа.

Внутри тега <html> находятся два элемента <head> и <body>.

Внутри тега <head> содержится техническая информация, такая как кодировка, заголовки, описание, ключевые слова для поисковых систем, а также в этом теге происходит подключение файла отвечающего за оформление страницы CSS, и подключение файла, отвечающего за какую-либо динамику на веб-странице JavaScript. Вся информация, находящаяся внутри тега <head> не отображается в браузере, но содержит указания, как правильно браузеру отображать эту страницу.

Тег <body> содержит в себе данные которые должны отобразится в браузере.

У каждого элемента могут быть различные атрибуты, например, id= " " или class= " ". Эти атрибуты помогают обратится к этим элементам для того чтобы задать им стили с помощью CSS, либо придать им какую-то динамику с помощью JavaScript.

2.3. Система управления базами данных MySQL.

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, которая поддерживает одну или более областей применения. [3]

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие

полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. [4]

2.4. Язык PHP.

PHP — скриптовый язык, часто применяемый для разработки веб-приложений. Является самым популярным языком для создания динамических веб-сайтов. Чаще всего выполняет роль сервера, то есть обрабатывает команды, приходящие от html-страниц, и отправляет обработанную информацию. Также выводит и записывает данные из базы данных. Имеет большую популярность из-за простоты и удобства. Включает в себя много различных функций для простого и удобного создания веб-приложений. [5]

2.5. Язык Python.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты. [6]

3. Материалы и методы.

Алгоритм по сбору данных был написан на языке PHP версии 7.2. Выбор пал на язык PHP, потому что он наиболее удобен в работе с веб-страницами. Для написания программы была использована среда разработки PhpStorm версии 2018.1.2 от компании JetBrains. Эта среда разработки была выбрана в связи с ее удобством, так как она имеет приятный интерфейс и делает подсказки в написании кода. Так же эта среда разработки позволяет просматривать базы данных различных СУБД. Полученные в ходе работы алгоритма данные записывались в базу данных с помощью СУБД MySQL версии 8.0. Использование именно этой СУБД обусловлено тем, что она является одной из самых быстрых, в моей работе это важно, так как собираются и записываются в базу большие объемы данных. Для запуска алгоритма была использована платформа Open Server версии 5.2.2. Эта платформа представляет из себя локальный веб-сервер для операционной системы Windows. Open Server включает в себя большой набор серверного программного обеспечения. Для своей работы из платформы Open Server я использовал веб-сервер Apache версии 2.2 и веб-интерфейс для администрирования СУБД MySQL – phpMyAdmin версии 4.8. Алгоритм анализа данных был написан на языке Python версии 3.7. Python был выбран как наиболее удобный язык для математических вычислений. Для написания алгоритма на Python была использована среда разработки PyCharm версии 2018.2 от компании JetBrains. Была выбрана именно эта среда разработки так как она очень удобна в использовании, имеет приятный интерфейс и дает подсказки в написании кода.

4. Основная часть.

4.1. Сбор данных с HTML-страницы.

Сбор данных осуществлялся с сайта по продаже недвижимости с помощью написанного алгоритма на языке PHP. Помимо стандартного набора функций языка PHP для написания алгоритма также использовалась библиотека phpQuery.

Предварительно скачаем и установим локальный веб-сервер. Загрузим с официального сайта платформу OpenServer [7], включающую в себя сервер Apache и интерфейс для администрирования СУБД MySQL phpMyAdmin. Далее произведем установку этой платформы. После установки, в папке domain, создаем папку, ей можно дать любо название, назовём ее diplom. Потом внутри этой папки создадим php файл с названием index, в котором будет весь код алгоритма для сбора данных. Теперь запустим сервер и откроем phpMyAdmin. Далее нужно создать базу данных. Для этого нужно нажать на кнопку «Создать БД» (Рисунок 2).

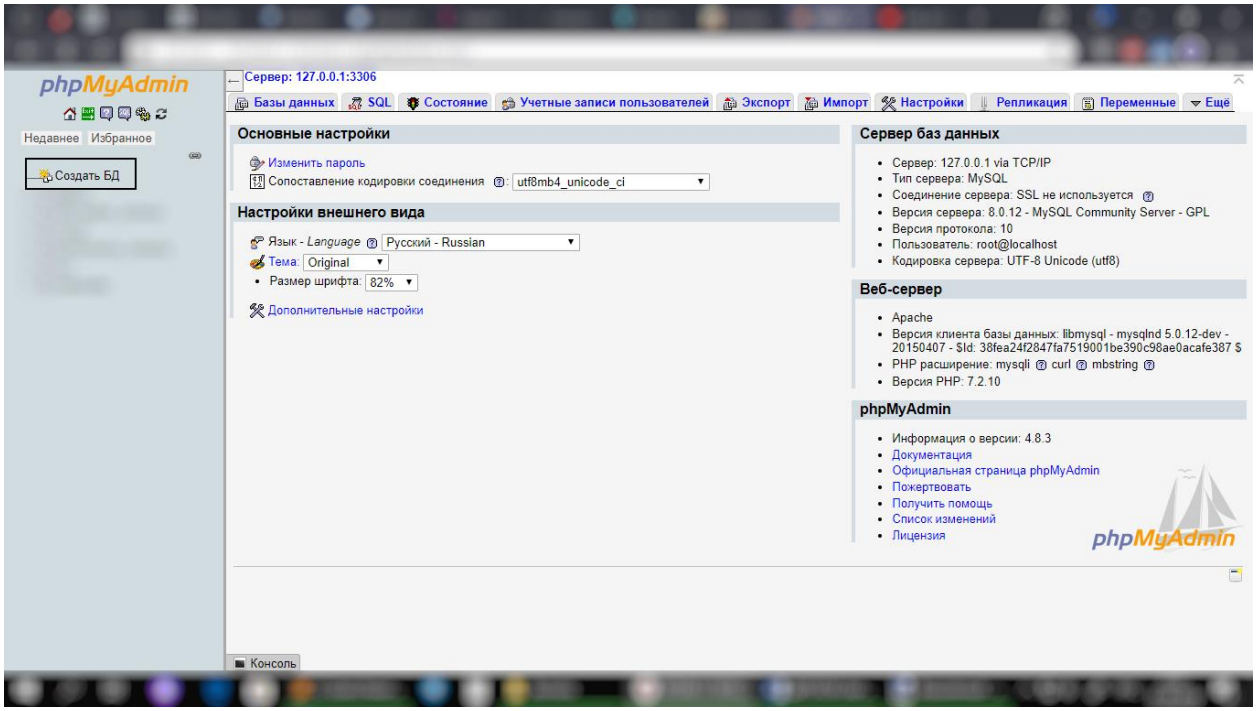


Рисунок 2. Создание БД в phpMyAdmin.

После этого откроется окно, нужно написать имя базы данных, назовем ее diplom, и нажмем на кнопку создать (Рисунок 3).

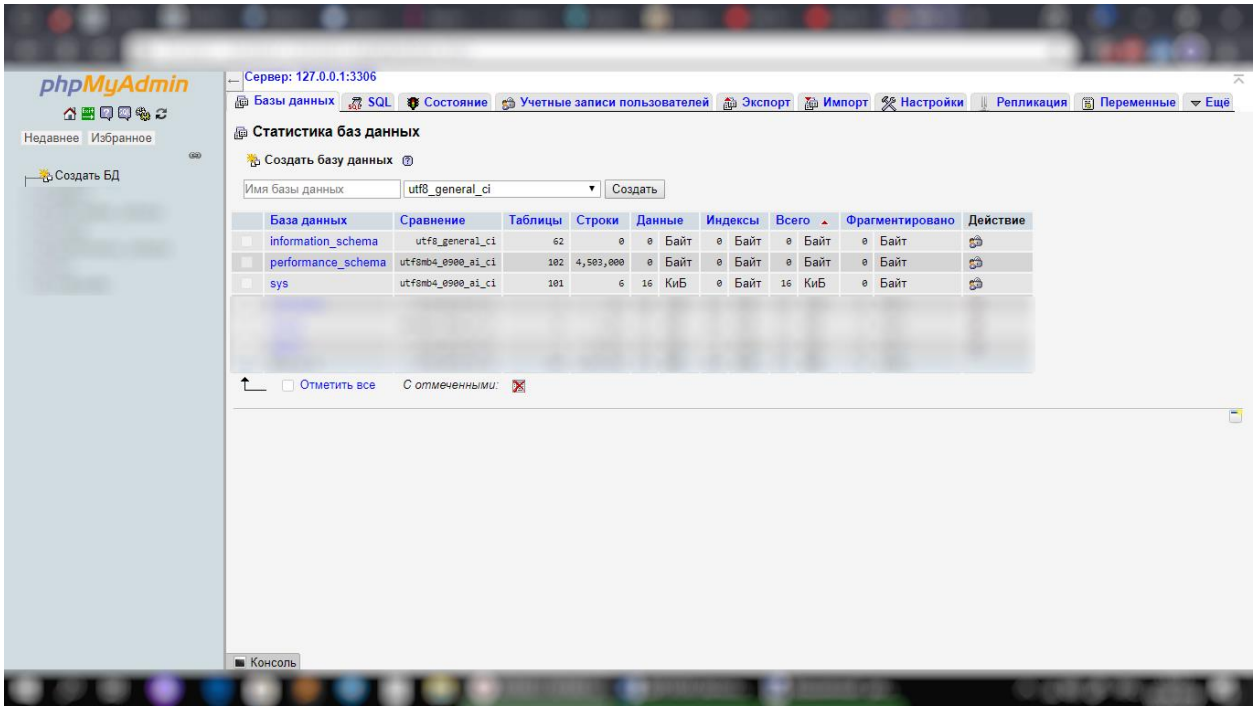


Рисунок 3. Создание БД в phpMyAdmin.

Далее создаем таблицу, у которой будет 11 столбцов и название apartaments (Рисунок 4).

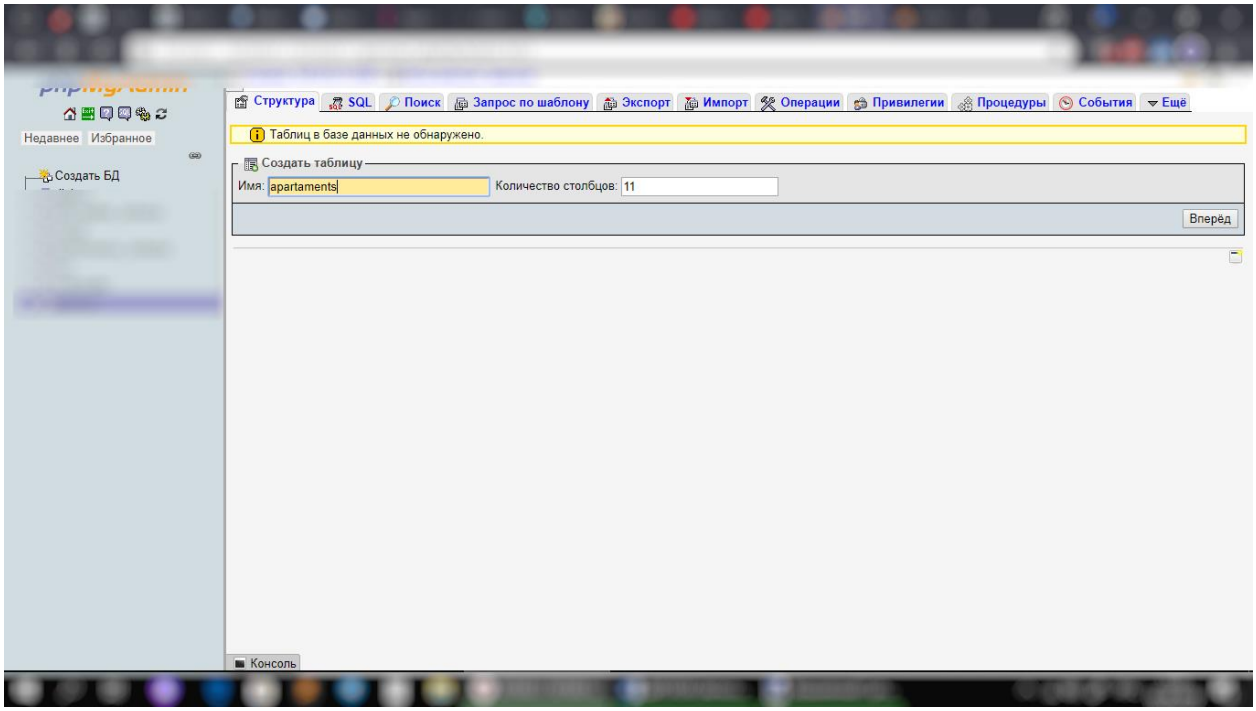


Рисунок 4. Создание таблицы.

Таблица apartaments содержит 11 столбцов (Рисунок 5):

1. Столбец id имеет тип int – в этот столбец записывается номер строки с данными. Пример записи: 1.
2. Столбец quantity_of_rooms имеет тип int – в этот столбец записывается количество комнат в квартире. Пример записи: 1.
3. Столбец area_of_apartaments имеет тип int – в этот столбец записывается площадь квартиры. Пример записи: 23.
4. Столбец floor имеет тип varchar – в этот столбец записывается на каком из скольких этажей находится квартира. Пример записи: 17/23 этаж.
5. Столбец metro имеет тип varchar – в этот столбец записывается название ближайшей к квартире станции метро. Пример записи: Охотный ряд.
6. Столбец time_to_metro имеет тип int – в этот столбец записывается время ходьбы или проезда до ближайшей станции метро. Пример записи: 5.
7. Столбец name_of_district имеет тип varchar – в этот столбец записывается название района, в котором находится квартира. Пример записи: Дмитровский.
8. Столбец price имеет тип int – в этот столбец записывается цена квартиры. Пример записи 8564987.
9. Столбец price_per_square_metr имеет тип int – в этот столбец записывается цена квартиры за квадратный метр. Пример записи: 372390.
10. Столбец date имеет тип date – в этот столбец записывается дата сбора данных. Пример записи: 2019-05-18.
11. Столбец collection number имеет тип int – в этот столбец записывается номер сбора данных. Пример записи: 1.

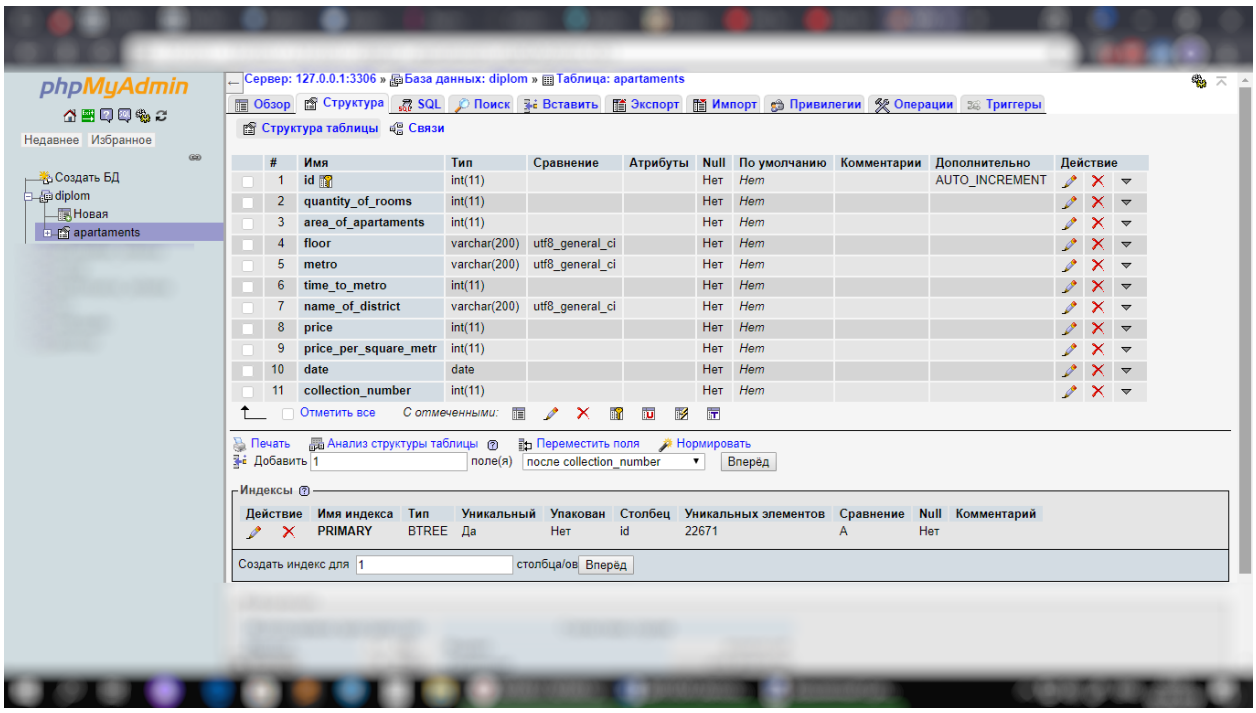


Рисунок 5. Структура таблицы apartaments.

Платформа OpenServer установлена, база данных создана. Далее подробно разберем алгоритм по сбору данных.

Для начала подключим библиотеку phpQuery, для этого нужно добавить файл «phpQuery-onefile.php» в папку с проектом, далее с помощью функции include() или с помощью функции require() подключаем библиотеку к проекту. Отличие этих функций заключается в том, что при неудачном подключении файла функция include() выдает предупреждение, но продолжает работу, а функция require() выдает ошибку и полностью останавливает работу программы. Так как, библиотека phpQuery играет важную роль в работе алгоритма, подключим ее с помощью функции require(), то есть вот так require(" phpQuery-onefile.php ").

После подключение библиотеки перейдем к написанию алгоритма. Первым делом получим весь HTML-код со страницы с помощью функции $file = file_get_contents($url), где $url это ссылка на нужный нам сайт, в данном случае это сайт по продаже недвижимости. Далее используя библиотеку

phpQuery, мы создаем документ с помощью $doc = phpQuery::newDocument($file), где $file это HTML-код, который мы получили с помощью функции file_get_contents($url). В результате мы получим объект, к которому можно будет применять определенные методы. Далее на нужной нам странице нужно выделить нужную информацию, для начала отделим каждое объявление. На одной странице располагается 25 объявлений. Объявления выглядят как блоки располагающиеся друг за другом. В каждом объявлении располагается основная информация о квартире. Такая как, фотографии дома и квартиры изнутри, количество комнат в квартире, площадь квартиры в квадратных метрах, на каком этаже располагается квартира и сколько всего этажей в доме, ближайшее метро, расстояние до метро в минутах, стоимость квартиры в рублях, стоимость за квадратный метр в рублях, адрес дома, небольшое описание и контакты для связи с владельцем квартиры (Рисунок 6).

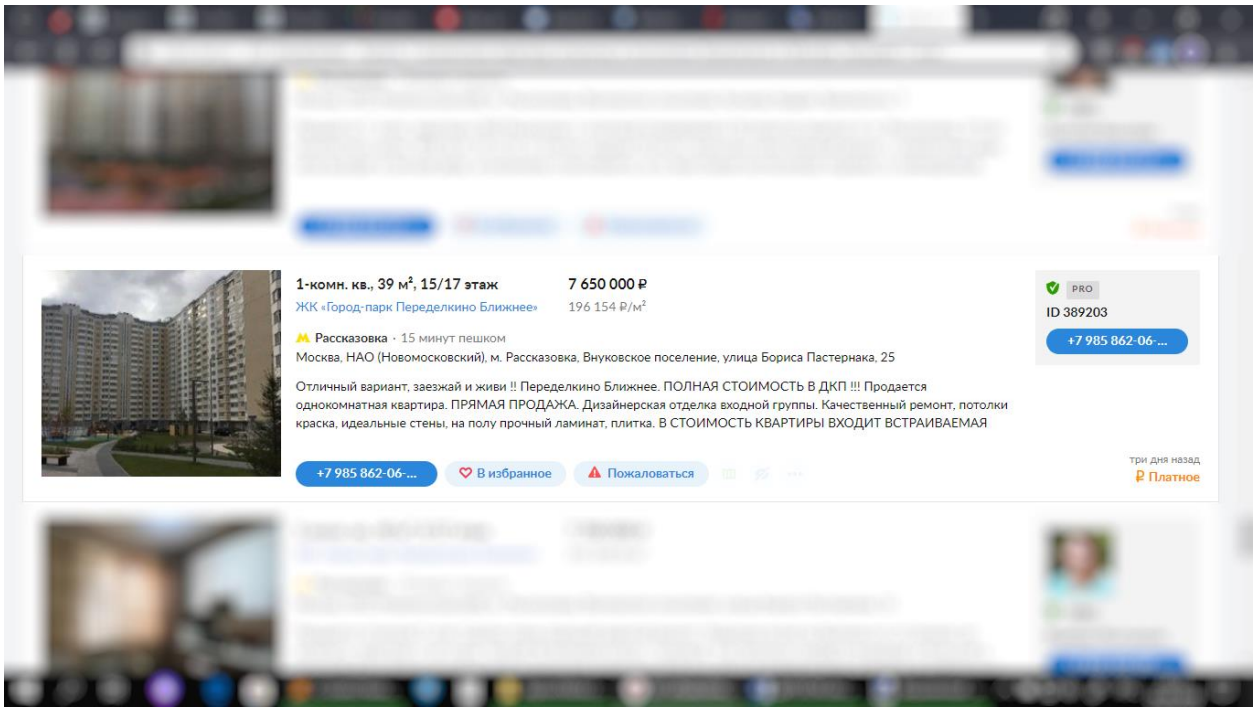


Рисунок 6. Вид объявления на сайте по продаже недвижимости.

Каждый блок объявления на сайте имеет одинаковый атрибут class. Здесь

каждое объявление имеет значение атрибута class = "_93444fe79c--card--_yguQ". В данном случае объявления имеют одинаковый атрибут для того чтобы можно было привести все объявления к одному стилю и задать одинаковое оформление.

Чтобы узнать значение атрибута у требуемого нам блока, необходимо зайти на выбранный нами сайт, нажать на нужный нам блок правой кнопкой мыши и выбрать пункт исследовать элемент (Рисунок 7).

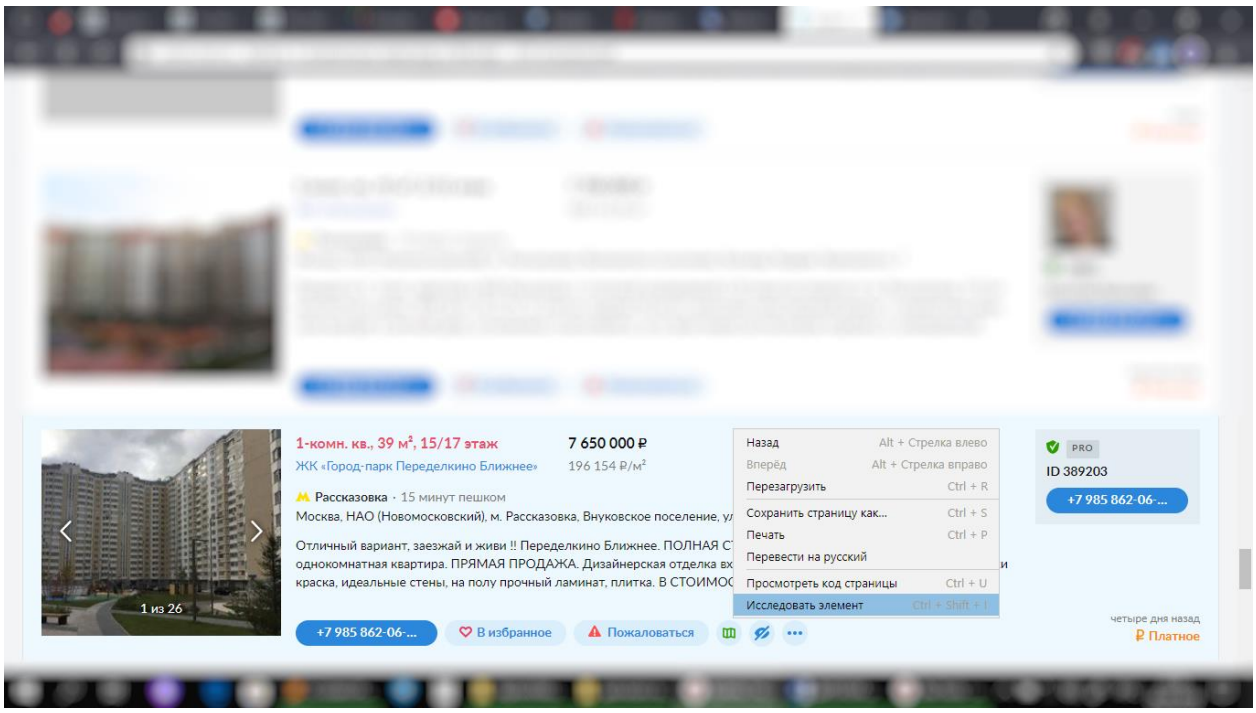


Рисунок 7. Исследование элемента.

После нажатие на пункт «Исследовать элемент» откроется окно DevTools (Инструменты разработчика), в котором во вкладке Elements можно посмотреть HTML-код всей страницы. Во вкладке Console показываются ошибки JavaScript кода, а также там можно писать JavaScript код, например, чтобы узнать значение переменной. Во вкладке Sources можно посмотреть все подключенные файлы к этой страннице, а также открыв JavaScript файл можно провести его пошаговое выполнение и сделать отладку. У нас по умолчанию откроется вкладка Elements и будет выделен нужный нам блок. Как мы видим (Рисунок 8), нужный нам блок имеет тег и атрибут class = "_93444fe79c--card--_yguQ". Также (Рисунок 8,

рисунок 9) видно, что другие объявления имеют такой же тег с таким же значением атрибута class = "_93444fe79c--card--_yguQ".

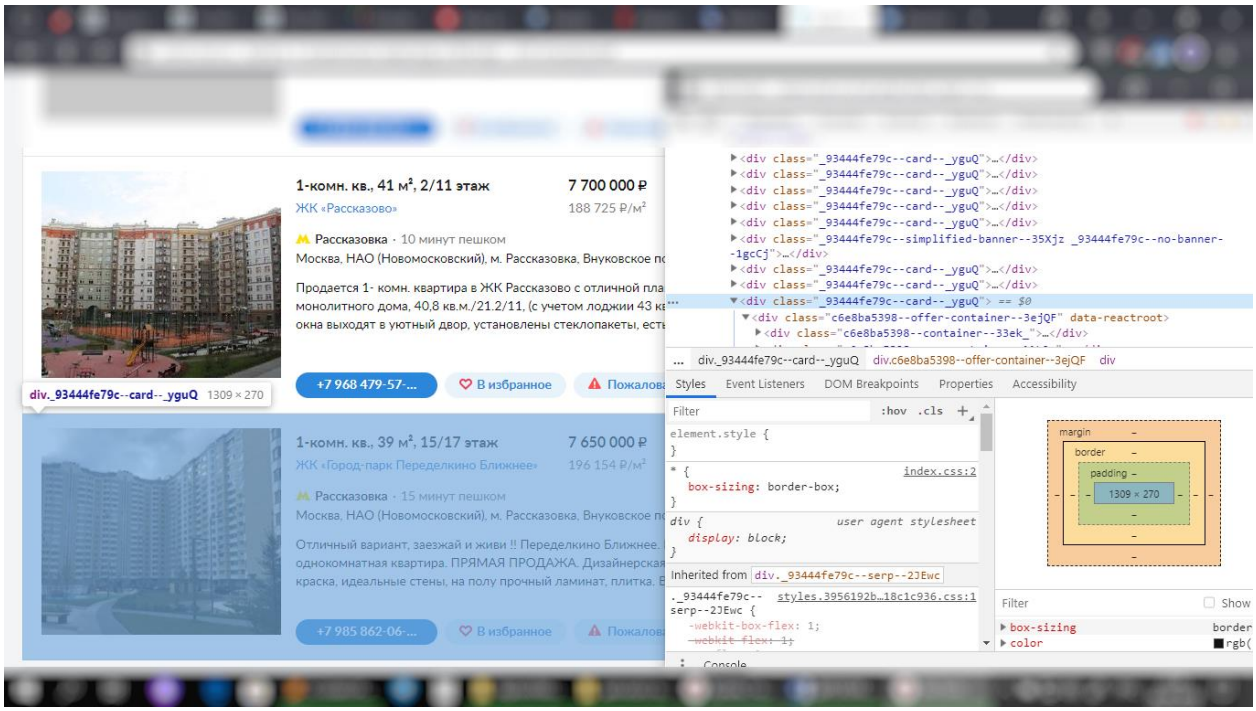


Рисунок 8. Исследование элемента.

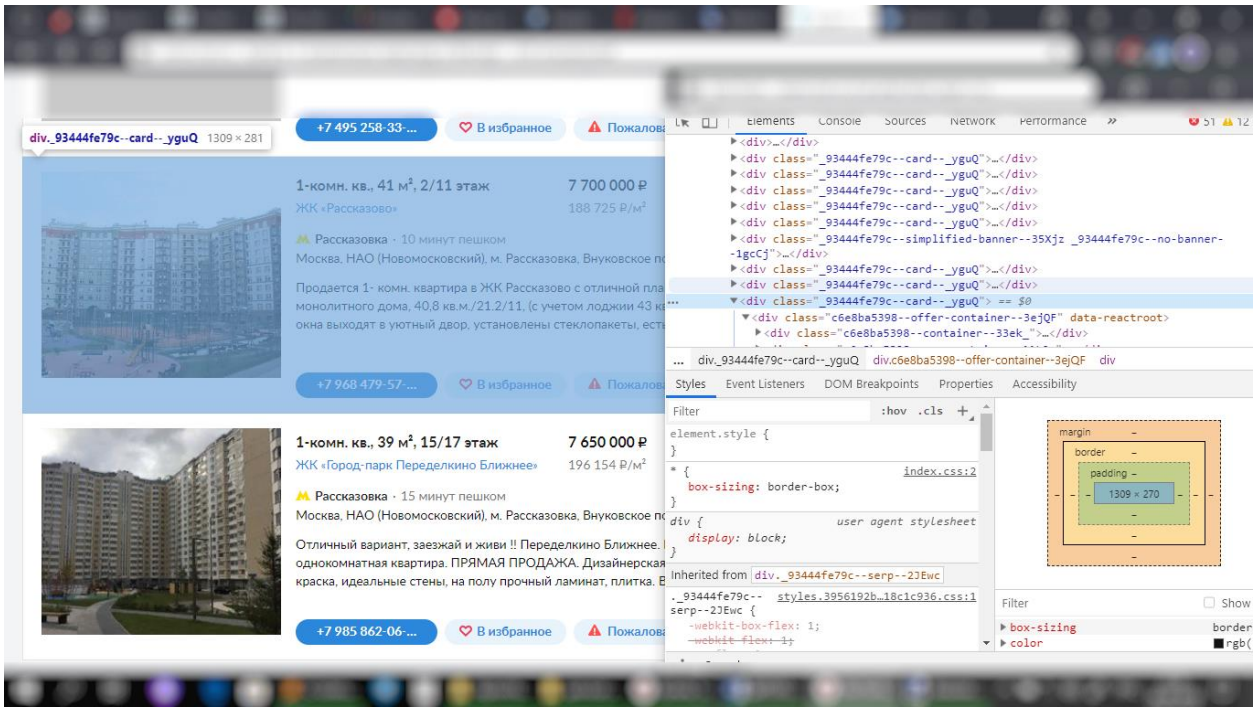


Рисунок 9. Исследование элемента.

Итак, так как мы узнали значение атрибута, у блока, который содержит информацию об объявлении, теперь его можно найти в нашем документе. Для

поиска по объекту, который вернулся с помощью phpQuery::NewDocument($file), мы используем метод $doc→find(), где $doc это сам документ. Чтобы найти нужное нам значение атрибута в документе, в скобках метода find(), мы пишем значение атрибута в кавычках и ставим точку в начале, чтобы показать, что это атрибут class, то есть так $dv1 = $doc→find("._93444fe79c--card--_yguQ"). В переменную $dv1 записываются только данные с объявлениями. С этого момента поиск нужных нам данных будет производится по переменной $dv1. С помощью алгоритма собираются следующие данные, количество комнат в квартире, площадь квартиры в квадратных метрах, на каком этаже располагается квартира и сколько всего этажей в доме, ближайшее метро, расстояние до метро в минутах, стоимость квартиры в рублях, стоимость за квадратный метр в рублях (Рисунок 10).

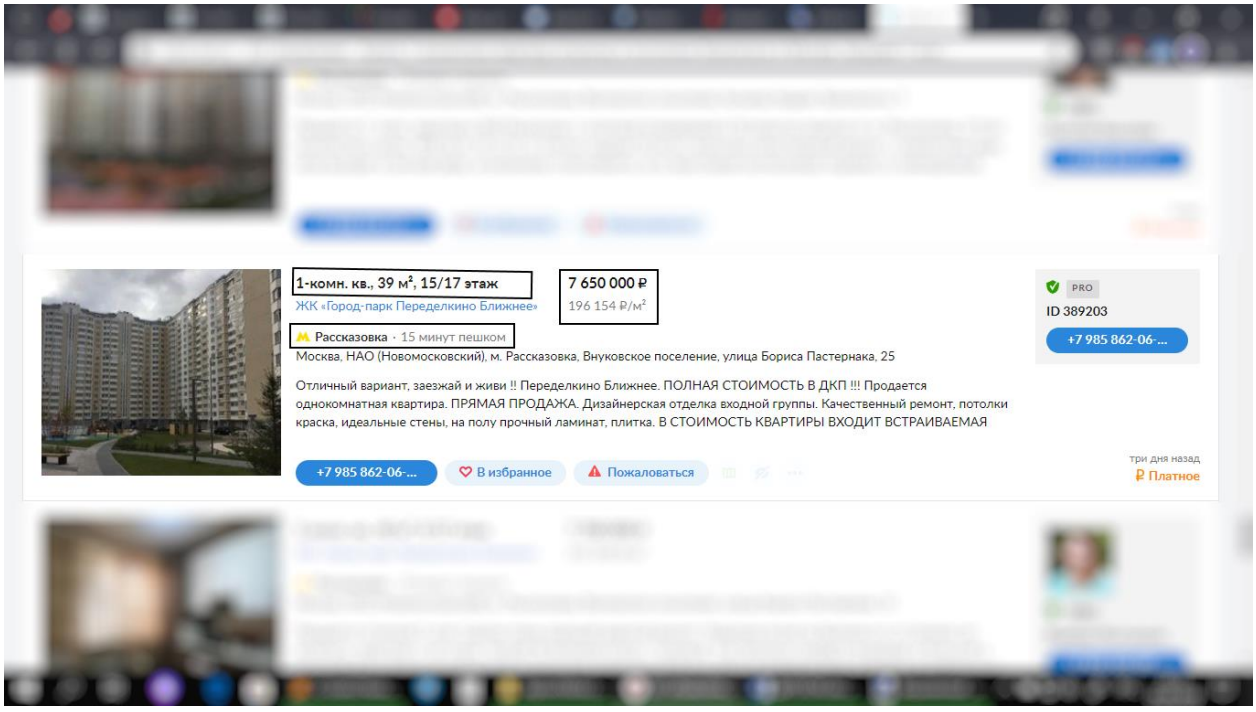


Рисунок 10. Собираемые данные (выделены).

Сперва с помощью окна DevTools узнаем значение атрибута у блока, который содержит в себе информацию о количестве комнат, площади квартиры и этажность. Нужный нам блок также имеет тег и обладает значением атрибута class="c6e8ba5398--title--2CW78" (Рисунок 11). Блок с этой

информацией содержится в каждом объявлении на странице, и имеет одинаковое значение атрибута class.

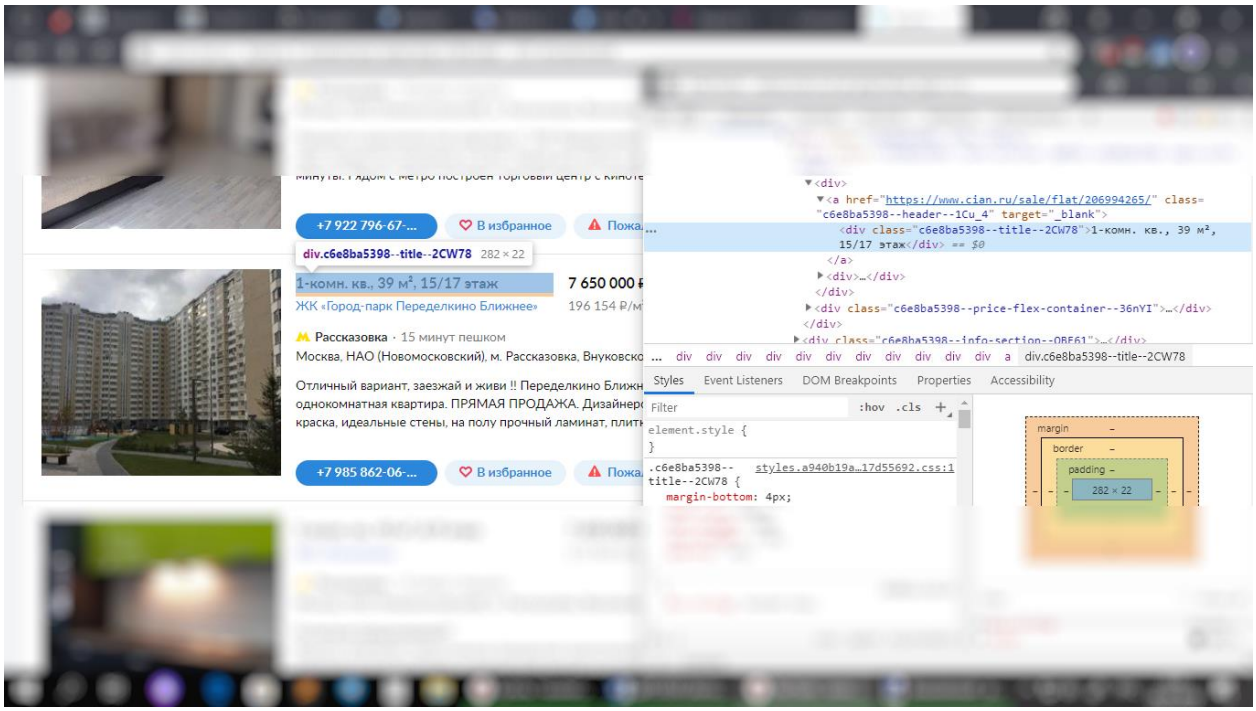


Рисунок 11. Исследование элемента с основными данными о квартире.

Теперь, зная значение атрибута, используем метод find (), чтобы найти его в переменной $dv1. Запишем найденные данные в переменную $dv2. $dv2 = $dv1→find('.c6e8ba5398--title--2CW78'). Если вывести на экран переменную $dv2 внутри тега <xmp>, то видно что данные там записаны вместе с тегами и атрибутами (Рисунок 12).

```
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 18 м², 1/2 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 20 м², 1/3 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 22 м², 1/11 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 40 м², 8/19 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 37 м², 12/13 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 42 м², 11/19 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 41 м², 8/13 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 35 м², 2/13 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 42 м², 7/13 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 28 м², 6/23 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 27 м², 9/23 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 30 м², 1/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 34 м², 1/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 26 м², 3/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 38 м², 2/3 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 39 м², 12/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 35 м², 2/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 34 м², 2/3 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 30 м², 14/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 38 м², 13/23 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 39 м², 1/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 35 м², 13/23 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 35 м², 9/23 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 30 м², 14/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 35 м², 14/14 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 39 м², 2/12 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. кв., 41 м², 1/17 этаж</div>
<div class="c6e8ba5398--title--2CW78">1-комн. апарт., 45 м², 3/8 этаж</div>
```

Рисунок 12. Значение переменной $dv2.

Для удобного анализа данных, необходимо убрать лишнюю информацию. Поэтому с помощью функции str_replace () уберем всё, кроме

нужной для записи в базу данных информации. После применения этой функции, тип переменной $dv2 поменяется с объекта на строку. Эта функция позволяет заменить определенную информацию на нужную. В данном случае заменим сначала открывающийся тег . $dv2 = str_replace (' ', '', $dv2). Эта запись означает, что все что в переменной $dv2 равно заменяется на пустоту, то есть по сути удаляется. Далее заменим закрывающий тег . $dv2 = str_replace(' ', '~', $dv2). Здесь закрывающий тег мы заменяем на тильду, чтобы в дальнейшем разделить строку по этому символу и создать массив для последующей записи в базу данных. Функция str_replace помогла избавится от лишних данных. И теперь в переменной $dv2 содержится только нужная информация (Рисунок 13).

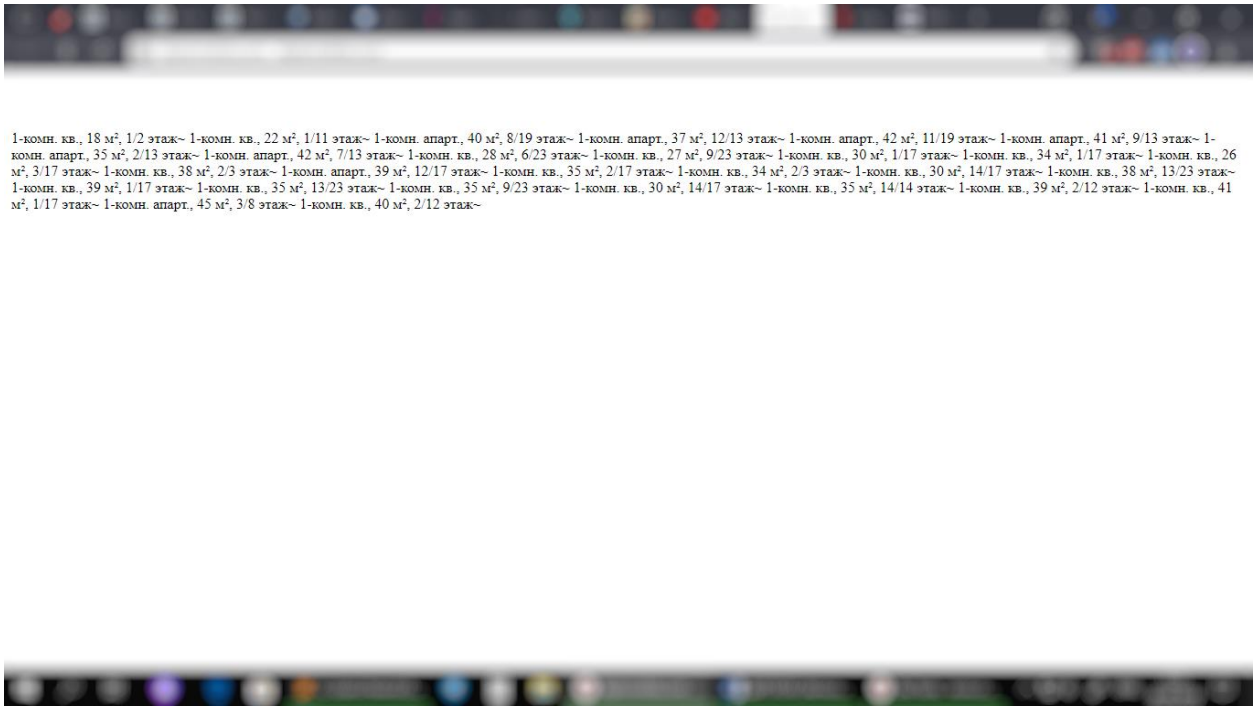


Рисунок 13. Значение переменной $dv2.

Теперь применим функцию preg_split (). $dv2 = preg_split('~', $dv2). С помощью неё мы поделили строку $dv2 по символу ~. Теперь $dv2 имеет тип переменной массив, каждый элемент которого имеет такой вид: 1-комн. кв., 18 м², 1/2 этаж. Также применим функцию array_pop($dv2). Эта функция позволяет удалить последний элемент массива. В данном случае нужно применить эту

функцию, так как последний элемент массива $dv2 пустой, и для последующего корректного использования массив не должен содержать пустых элементов.

Все остальные данные собираются по точно такому же алгоритму. То есть сначала с помощью окна DevTools нужно узнать значение атрибута class, далее применяем нужные методы и функции, в итоге получаем массив с данными.

Теперь, все данные нужно записать в базу данных. С помощью строчки $pdo = **new** PDO('mysql:host=localhost;dbname=diplom', 'root', ' ') выполним подключение к базе данных, в скобках указывается, вид СУБД, в этом проекте это MySQL, адрес сервера, здесь это локальный сервер, то есть localhost, а также название базы данных, логин и пароль от БД. Подключение выполнено, запись полученных данных будет происходить в цикле, который будет проходить по всем массивам и выполнять SQL запрос (Рисунок 14).

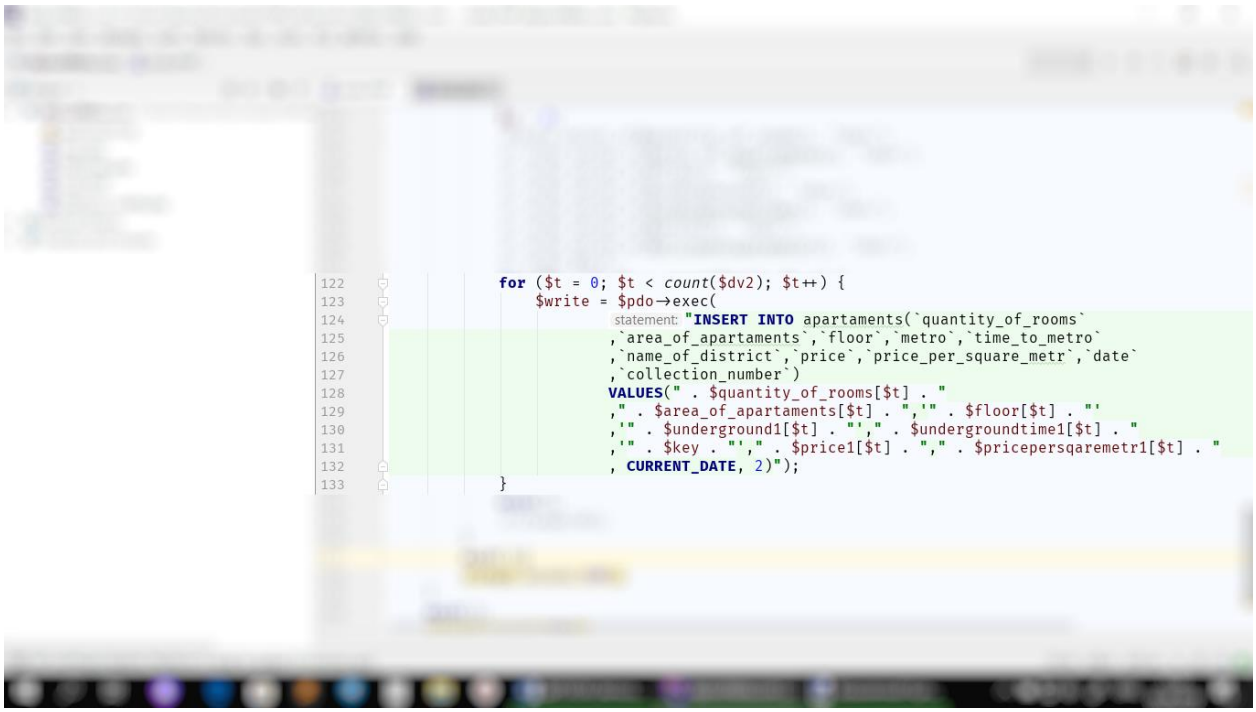


Рисунок 14. Цикл исполнения запроса.

Здесь (Рисунок 14) count($dv2) это количество объявлений на странице. в столбцы базы данных записываются значения массивов, в котором хранятся все данные по объявлению. Для перехода к следующей странице на сайте, для

последующего дальнейшего сбора данных находим с помощью окна DevTools значение атрибута class у страницы, которая открыта сейчас (Рисунок 15).

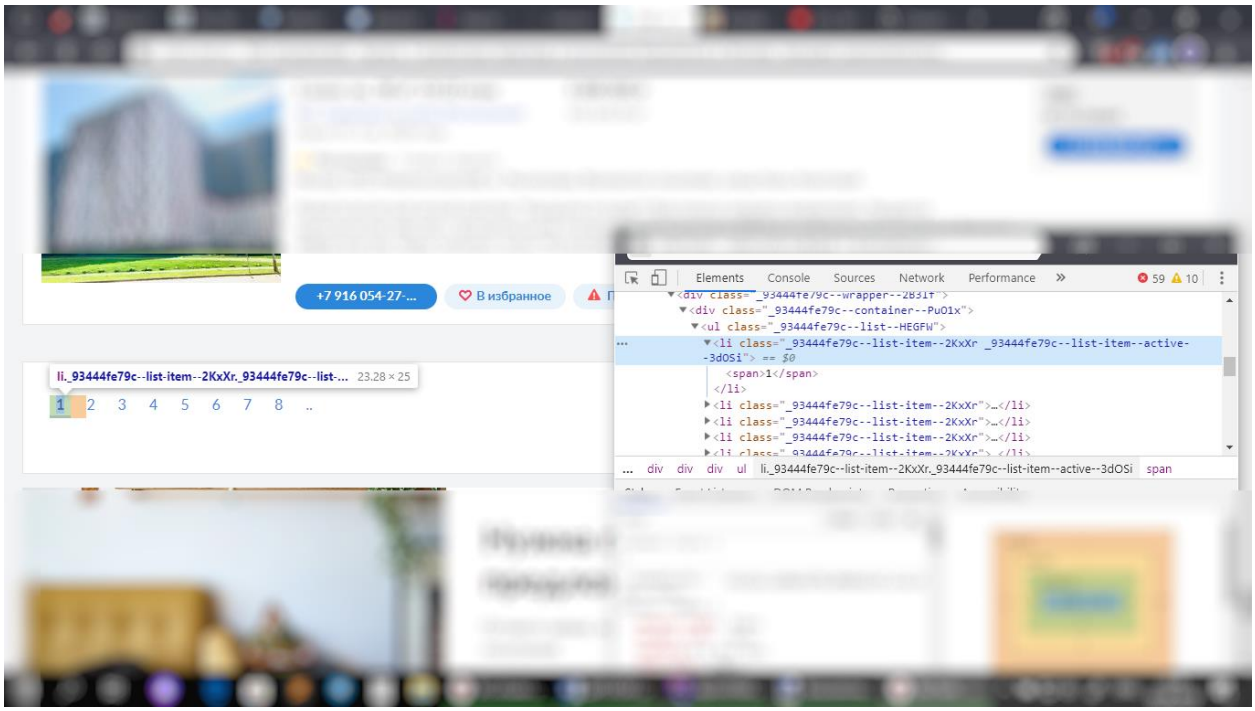
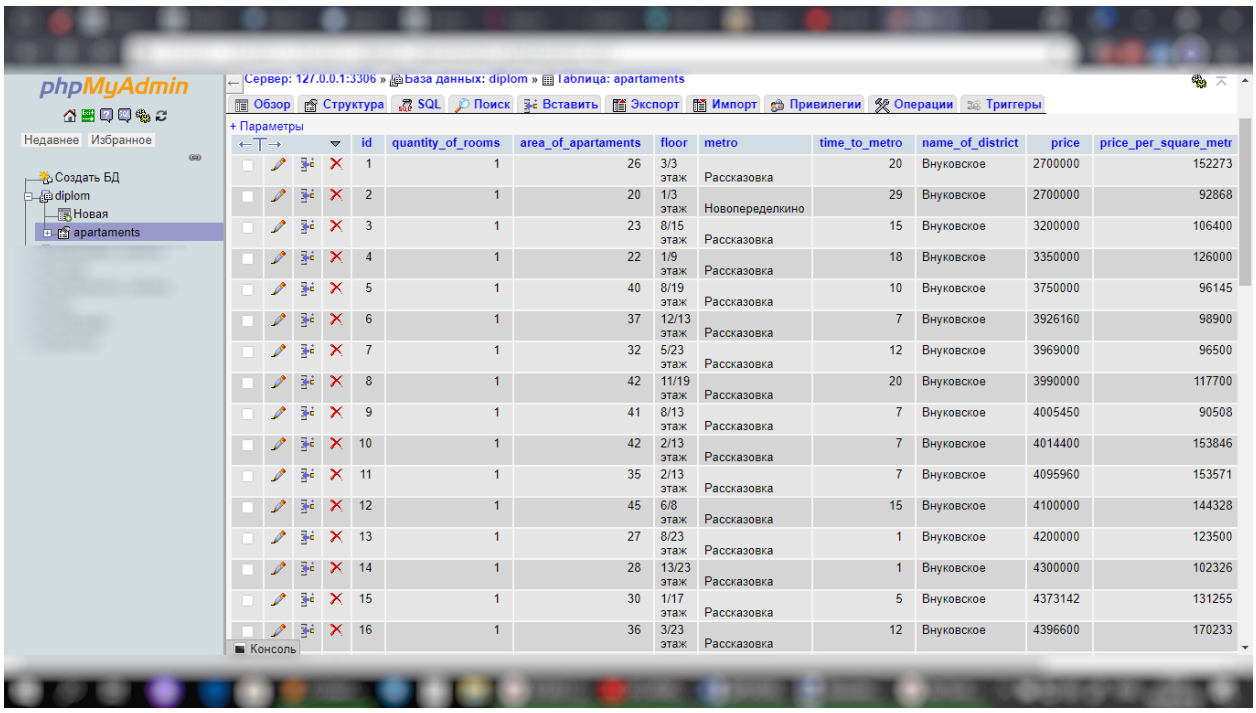


Рисунок 15. Исследование элемента.

У активной страницы значение атрибута равно class="_93444fe79c--list-item--2KxXr _93444fe79c--list-item--active--3dOSi". Далее с помощью методов phpQuery найдем следующую за этой страницу. Это делается так: $next = $doc->find('._93444fe79c--list-item--active--3dOSi')->next()->children()->attr('href'). Теперь в переменной $next хранится ссылка, на следующую страницу. Далее мы записываем значение переменной $next в переменную $url. Теперь мы оборачиваем весь алгоритм в цикл while. В итоге алгоритм будет производить сбор данных со страниц, пока они не закончатся.

Чтобы запустить алгоритм, включаем локальный веб-сервис, заходим в браузер и вводим в адресную строку название папки, которая была создана в папке domains, здесь она названа diplom. Далее нажимаем enter, теперь алгоритм заработал и собирает данные. Все данные будут записываться в таблицу в базу данных (Рисунок 16).



id	quantity_of_rooms	area_of_apartaments	floor	metro	time_to_metro	name_of_district	price	price_per_square_metr
1	1	26	3/3 этаж	Рассказовка	20	Внуковское	2700000	152273
2	1	20	1/3 этаж	Новопеределкино	29	Внуковское	2700000	92868
3	1	23	8/15 этаж	Рассказовка	15	Внуковское	3200000	106400
4	1	22	1/9 этаж	Рассказовка	18	Внуковское	3350000	126000
5	1	40	8/19 этаж	Рассказовка	10	Внуковское	3750000	96145
6	1	37	12/13 этаж	Рассказовка	7	Внуковское	3926160	98900
7	1	32	5/23 этаж	Рассказовка	12	Внуковское	3969000	96500
8	1	42	11/19 этаж	Рассказовка	20	Внуковское	3990000	117700
9	1	41	8/13 этаж	Рассказовка	7	Внуковское	4005450	90508
10	1	42	2/13 этаж	Рассказовка	7	Внуковское	4014400	153846
11	1	35	2/13 этаж	Рассказовка	7	Внуковское	4095960	153571
12	1	45	6/8 этаж	Рассказовка	15	Внуковское	4100000	144328
13	1	27	8/23 этаж	Рассказовка	1	Внуковское	4200000	123500
14	1	28	13/23 этаж	Рассказовка	1	Внуковское	4300000	102326
15	1	30	1/17 этаж	Рассказовка	5	Внуковское	4373142	131255
16	1	36	3/23 этаж	Рассказовка	12	Внуковское	4396600	170233

Рисунок 16. Данные в БД.

4.2. Построение регрессионной модели.

Для построения регрессионной модели решим задачу минимизации.

$$\sum_{i=1}^{n}(y_i - (\vec{a}, \overrightarrow{x_i}) - b)^2 \underset{a,b}{\rightarrow} min$$

Поскольку функция является выпуклой всюду в R^{n+1}, то она обладает единственным экстремумом. Для его нахождения возьмем производные по каждой из переменных a_1, a_2, ... , a_n и b и приравняем к нулю. Тогда получим систему уравнений:

$$\begin{cases} \sum_{i=1}^{n} 2x_i(y_i - (a, x_i) - b) = 0 \\ \sum_{i=1}^{n} 2x_i^2(y_i - (a, x_i) - b) = 0 \\ \dots \\ \sum_{i=1}^{n} 2x_i^n(y_i - (a, x_i) - b) = 0 \\ \sum_{i=1}^{n} 2(y_i - (a, x_i) - b) = 0 \end{cases}$$

Эта система сводится к матричному уравнению:

$$\begin{pmatrix} Cov(x_1, x_1) & \cdots & Cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ Cov(x_n, x_1) & \cdots & Cov(x_n, x_n) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} Cov(x_1, y) \\ \vdots \\ Cov(x_n, y) \end{pmatrix}$$

с дополнительным условием

$$b = \bar{y} - \left(\vec{a}, \bar{\vec{x}}\right) = \bar{y} - a_1\bar{x}_1 - a_2\bar{x}_2 - \cdots - -a_n\bar{x}_n$$

Находя a_1, a_2, ... , a_n получаем линейную регрессионную модель.

$$y \sim a_1x_1 + a_2x_2 + \cdots + a_nx_n + b$$

Для определения значимости параметров сделаем центрирование и нормировку, то есть преобразование

$$x \to x' = \frac{x - \overline{x}}{sd(x)}$$

Где

$$sd(x) = \sqrt{\frac{\sum(x_i - \overline{x})^2}{n - 1}}$$

Делаем нормировку данных и строим регрессионную модель по x_i' и находим a'. Пропорция между компонентами a_i' позволяет определить значимость параметров для модели. В качестве измерения качества модели возьмем индекс детерминированности R^2, который равен:

$$1 - \frac{var(y(x))}{var(x)}$$

Далее разберем алгоритм построения регрессионной модели на языке Python. Изначально считаем все собранные записи с базы данных, для этого нужно подключится к БД. Далее после запроса получим все эти данные. Для большей видимости результатов, возьмем однокомнатные квартиры, стоимостью от 1000000₽ до 50000000₽. Для анализа и визуализации данных воспользуемся библиотекой matplotlib. Так как в БД большинство полей – строки, нужно произвести классификацию данных. Для начала определим корреляцию признаков от цены, то есть узнаем насколько сильно взаимосвязаны признаки с ценой. Сделаем это с помощью функции corrcoef из библиотеки numpy (Рисунок 17, Рисунок 18).

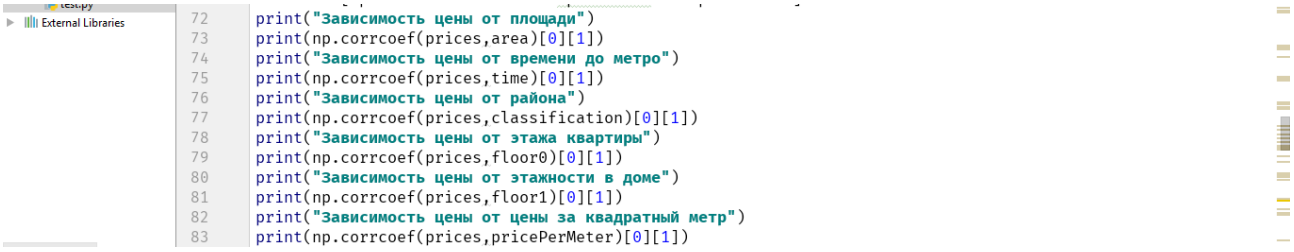


Рисунок 17. Вычисление коэффициента корреляции.

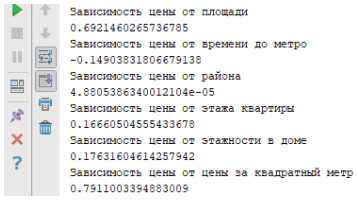


Рисунок 18. Значение коэффициента корреляции.

Видно, что цена наиболее взаимосвязана с такими признаками как площадь и цена за квадратный метр. В этих двух случаях коэффициент корреляции положительный и наиболее близок к единице. Это означает, что чем больше площадь квартиры или цена за квадратный метр, тем больше цена за квартиру. У признака время до метро отрицательный коэффициент корреляции, это означает, что чем ближе метро, тем выше цена. Построим графики зависимости цены от этих трех признаков (Рисунок 19).

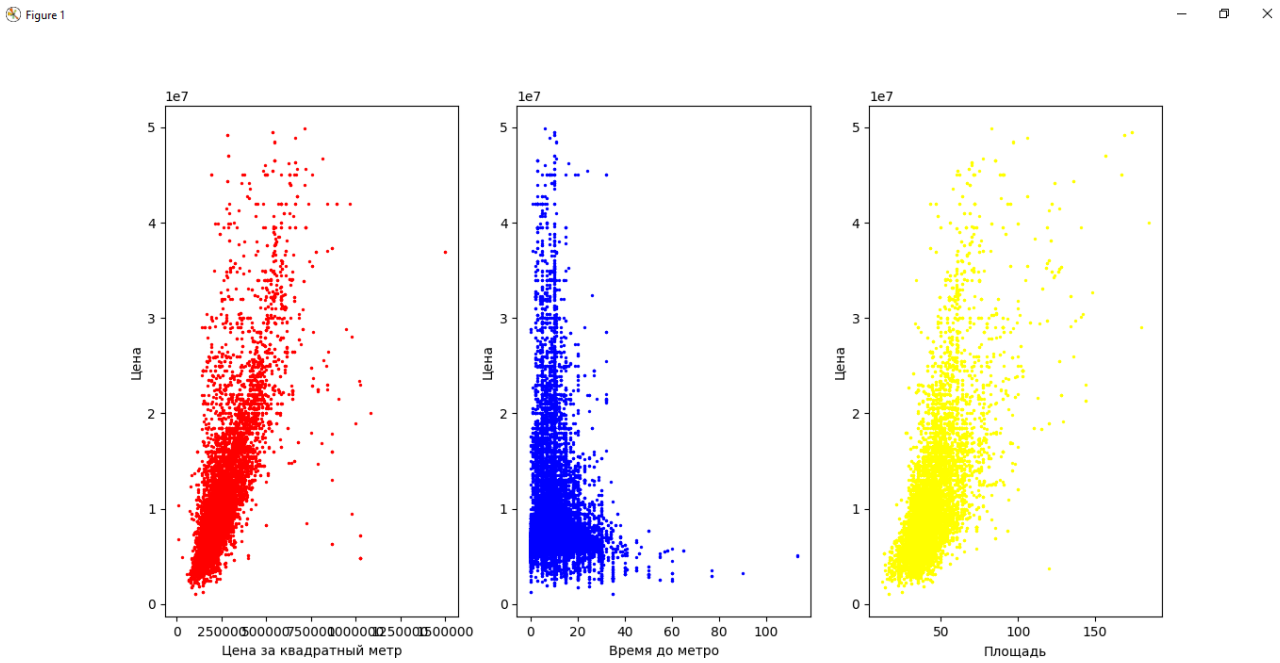


Рисунок 19. График зависимости цены от признаков.

На графиках также видно эти зависимости. Далее используя эти три признака обучим модель. В качестве модели для обучения была выбрана модель LinearRegression(). Обучаем используя функцию fit(). После обучения модели, сделаем предсказание по тем же трем признакам. Для этого используем функцию predict(). В итоге получаем точность предсказания равную 0,82 (Рисунок 20).

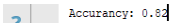


Рисунок 20. Точность предсказания.

Также на графике сопоставим значения предсказанной цены и реальной, зеленые точки — это предсказанная цена, а красные это реальная (Рисунок 21).

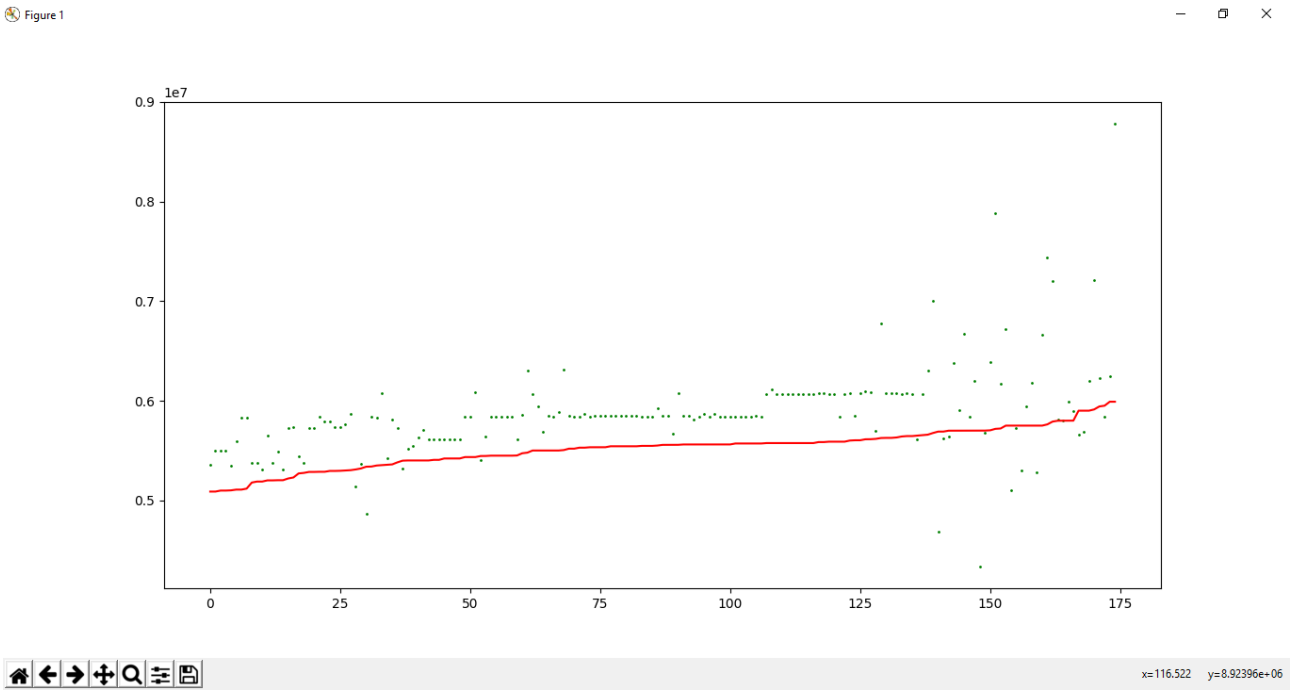


Рисунок 21. Сопоставление предсказанной цены и реальной.

5. Заключение и выводы.

В данной работе, успешно был осуществлен сбор данных с html-страниц с помощью языка PHP и библиотеки phpQuery. Все собранные данные были упорядочены и успешно записаны в базу данных. В дальнейшем производился анализ этих данных. В ходе анализа были выведены зависимости признаков от результативного признака. На основе этих признаков была обучена линейная регрессионная модель. В итоге после обучения на основе трех признаков, а именно площадь квартиры, цена за квадратный метр и расстояние до метро, получилось сделать предсказание цены основываясь на эти же признаки с точность 0,82.

6. Список литературы.

1) Ситуация на рынке жилья по регионам к середине 2018 года. [Электронный ресурс] URL: https://kazan.cian.ru/stati-situatsija-na-rynke-zhilja-po-regionam-k-seredine-2018-goda-283917/

2) Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание = HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2010. — 656 с. — ISBN 978-5-8459-1676-1.

3) Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с. — ISBN 978-5-94774-736-2.

4) В. Васвани. MySQL: использование и администрирование = MySQL Database Usage & Administration. — М.: «Питер», 2011. — 368 с. — ISBN 978-5-459-00264-5.

5) Кузнецов Максим, Симдянов Игорь. PHP на примерах. — 2-е изд. перераб. и доп. — СПб.: «БХВ-Петербург», 2011. — С. 400. — ISBN 978-5-9775-0445-4.

6) Фёдоров Д. Ю. Основы программирования на примере языка Python / Учебное пособие. — СПб.: Юрайт, 2018. — 167 с. — ISBN 978-5-534-04479-9.

7. Приложение.

7.1. Приложение 1. Код алгоритма на языке PHP.

Файл index.php.

```
<?php
ini_set('max_execution_time', 30000);
header('Content-type: text/html; charset:utf-8');
include("array.php");
require 'phpQuery-onefile.php';
$pdo = new PDO('mysql:host=localhost;dbname=diplom', 'root', '');
for ($i = 0; $i < count($districts); $i++) {
    for ($j = 0; $j < count($districts[$i]); $j++) {
        $flag = true;
        $url = array_values($districts[$i])[$j];
        $key = array_search(array_values($districts[$i])[$j], $districts[$i]);
        while ($flag) {
            $file = file_get_contents($url);
            $doc = phpQuery::newDocument($file);
            $dv1 = $doc->find('._93444fe79c--card--_yguQ');
            $dv2 = $dv1->find('.c6e8ba5398--title--2CW78');
            $dv2 = str_replace('<div class="c6e8ba5398--title--2CW78">', '', $dv2);
            $dv2 = str_replace('</div>', '~', $dv2);
            $dv2 = preg_split('#~#', $dv2);
            $elem = array_pop($dv2);
            $underground = $dv1->find('.c6e8ba5398--underground-name--3YjAi');
            $underground = str_replace('<div class="c6e8ba5398--underground-name--
3YjAi">', '', $underground);
            $underground = str_replace('</div>', '~', $underground);
            $underground = preg_split('#~#', $underground);
            while (count($underground) > count($dv2)) {
                $elem = array_pop($underground);
            }
            $undergroundtime = $dv1->find('.c6e8ba5398--remoteness--1MdE6');
            $undergroundtime = str_replace('<div class="c6e8ba5398--remoteness--
1MdE6">', '', $undergroundtime);
            $undergroundtime = str_replace('</div>', '~', $undergroundtime);
            $undergroundtime = preg_split('#~#', $undergroundtime);
            while (count($underground) > count($dv2)) {
                $elem = array_pop($undergroundtime);
            }
            $price = $dv1->find('.c6e8ba5398--header--1df-X');
            $price = str_replace('<div class="c6e8ba5398--header--1df-X">', '', $price);
            $price = str_replace('</div>', '~', $price);
            $price = preg_split('#~#', $price);
            $elem = array_pop($price);
            $pricepersqaremetr = $dv1->find('.c6e8ba5398--term--3kvtJ');
            $pricepersqaremetr = str_replace('<div class="c6e8ba5398--term--
3kvtJ">', '', $pricepersqaremetr);
            $pricepersqaremetr = str_replace('</div>', '~', $pricepersqaremetr);
            $pricepersqaremetr = preg_split('#~#', $pricepersqaremetr);
            for ($z = 0; $z <= 2; $z++) {
                $elem = array_shift($pricepersqaremetr);
                $elem = array_pop($pricepersqaremetr);
            }
            $elem = array_pop($pricepersqaremetr);
            $next = $doc->find('._93444fe79c--list-item--active--3dOSi')->next()-
>children()->attr('href');
            $next = str_replace('https://www.!!!.ru', '', $next);
            if (!emptyempty($next)) {
```

```
                $url = 'https://www.!!!.ru'.$next;
            } else {
                $flag = false;
            }
            $g = 0;
            $quantity_of_rooms = array();
            $area_of_apartaments = array();
            $floor = array();
            $underground1 = array();
            $undergroundtime1 = array();
            $price1 = array();
            $pricepersqaremetr1 = array();
            foreach ($dv2 as $value) {
                $value = preg_split('#,#', $value);
                foreach ($value as $value2) {
                    if ($g == 0) {
                        array_push($quantity_of_rooms, intval($value2));
                        $g = 1;
                    } else {
                        if ($g == 1) {
                            array_push($area_of_apartaments, intval($value2));
                            $g = 2;
                        } else {
                            array_push($floor, $value2);
                            $g = 0;
                        }
                    }
                }
            }
            $g = 0;
            foreach ($underground as $value) {
                array_push($underground1, $value);
            }
            $g = 0;
            foreach ($undergroundtime as $value) {
                array_push($undergroundtime1, intval($value));
            }
            $g = 0;
            foreach ($price as $value) {
                $value = str_replace(" ", '', $value);
                array_push($price1, intval($value));
            }
            $g = 0;
            foreach ($pricepersqaremetr as $value) {
                $value = str_replace(" ", '', $value);
                array_push($pricepersqaremetr1, intval($value));
            }
            $g = 0;
                for ($t = 0; $t < count($dv2); $t++) {
                    $write = $pdo-
>exec("INSERT INTO apartaments(`quantity_of_rooms`,`area_of_apartaments`,`floor`,`metro`,`
time_to_metro`,`name_of_district`,`price`,`price_per_square_metr`,`date`,`collection_numbe
r`) VALUES(" . $quantity_of_rooms[$t] . "," . $area_of_apartaments[$t] . ",'" . $floor[$t]
 . "','" . $underground1[$t] . "'," . $undergroundtime1[$t] . ",'" . $key . "'," . $price1
[$t] . "," . $pricepersqaremetr1[$t] . ", CURRENT_DATE, 2)");
                }
            }
        }
    }
```

7.2. Приложение 2. Часть кода базы данных на языке SQL.

Файл apartaments.sql.

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1:3306
-- Время создания: Май 22 2019 г., 16:13
-- Версия сервера: 8.0.12
-- Версия PHP: 7.2.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База данных: `diplom`
--

-- --------------------------------------------------------

--
-- Структура таблицы `apartaments`
--

CREATE TABLE `apartaments` (
  `id` int(11) NOT NULL,
  `quantity_of_rooms` int(11) NOT NULL,
  `area_of_apartaments` int(11) NOT NULL,
  `floor` varchar(200) NOT NULL,
  `metro` varchar(200) NOT NULL,
  `time_to_metro` int(11) NOT NULL,
  `name_of_district` varchar(200) NOT NULL,
  `price` int(11) NOT NULL,
  `price_per_square_metr` int(11) NOT NULL,
  `date` date NOT NULL,
  `collection_number` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `apartaments`
--

INSERT INTO `apartaments` (`id`, `quantity_of_rooms`, `area_of_apartaments`, `floor`, `metro`, `time_to_metro`, `name_of_district`, `price`, `price_per_square_metr`, `date`, `collection_number`) VALUES
(1, 1, 26, ' 3/3 этаж', '\nРассказовка', 20, 'Внуковское', 2700000, 152273, '2019-05-18', 1),
(2, 1, 20, ' 1/3 этаж', '\nНовопределкино', 29, 'Внуковское', 2700000, 92868, '2019-05-18', 1),
(3, 1, 23, ' 8/15 этаж', '\nРассказовка', 15, 'Внуковское', 3200000, 106400, '2019-05-18', 1),
```

```
(4, 1, 22, ' 1/9 этаж', '\nРассказовка', 18, 'Внуковское', 3350000, 126000, '2019-05-18', 1),
(5, 1, 40, ' 8/19 этаж', '\nРассказовка', 10, 'Внуковское', 3750000, 96145, '2019-05-18', 1),
(6, 1, 37, ' 12/13 этаж', '\nРассказовка', 7, 'Внуковское', 3926160, 98900, '2019-05-18', 1),
(7, 1, 32, ' 5/23 этаж', '\nРассказовка', 12, 'Внуковское', 3969000, 96500, '2019-05-18', 1),
(8, 1, 42, ' 11/19 этаж', '\nРассказовка', 20, 'Внуковское', 3990000, 117700, '2019-05-18', 1),
(9, 1, 41, ' 8/13 этаж', '\nРассказовка', 7, 'Внуковское', 4005450, 90508, '2019-05-18', 1),
(10, 1, 42, ' 2/13 этаж', '\nРассказовка', 7, 'Внуковское', 4014400, 153846, '2019-05-18', 1),
(11, 1, 35, ' 2/13 этаж', '\nРассказовка', 7, 'Внуковское', 4095960, 153571, '2019-05-18', 1),
(12, 1, 45, ' 6/8 этаж', '\nРассказовка', 15, 'Внуковское', 4100000, 144328, '2019-05-18', 1),
(13, 1, 27, ' 8/23 этаж', '\nРассказовка', 1, 'Внуковское', 4200000, 123500, '2019-05-18', 1),
(14, 1, 28, ' 13/23 этаж', '\nРассказовка', 1, 'Внуковское', 4300000, 102326, '2019-05-18', 1),
(15, 1, 30, ' 1/17 этаж', '\nРассказовка', 5, 'Внуковское', 4373142, 131255, '2019-05-18', 1),
(16, 1, 36, ' 3/23 этаж', '\nРассказовка', 12, 'Внуковское', 4396600, 170233, '2019-05-18', 1),
(17, 1, 43, ' 19/19 этаж', '\nРассказовка', 3, 'Внуковское', 4400000, 118421, '2019-05-18', 1),
(18, 1, 34, ' 1/17 этаж', '\nРассказовка', 5, 'Внуковское', 4423291, 135294, '2019-05-18', 1),
(19, 1, 26, ' 3/17 этаж', '\nРассказовка', 5, 'Внуковское', 4426056, 117207, '2019-05-18', 1),
(20, 1, 38, ' 2/3 этаж', '\nНовопеределкино', 21, 'Внуковское', 4500000, 138701, '2019-05-18', 1),
(21, 1, 34, ' 2/3 этаж', '\nРассказовка', 15, 'Внуковское', 4600000, 129500, '2019-05-18', 1),
(22, 1, 40, ' 12/16 этаж', '\nРассказовка', 23, 'Внуковское', 4700000, 132713, '2019-05-18', 1),
(23, 1, 30, ' 14/17 этаж', '\nРассказовка', 5, 'Внуковское', 5100000, 157576, '2019-05-18', 1),
(24, 1, 35, ' 13/23 этаж', '\nРассказовка', 2, 'Внуковское', 5100000, 134127, '2019-05-18', 1),
(25, 1, 35, ' 9/23 этаж', '\nРассказовка', 10, 'Внуковское', 5195000, 138588, '2019-05-18', 1),
(26, 1, 33, ' 13/17 этаж', '\nРассказовка', 5, 'Внуковское', 5200000, 129051, '2019-05-18', 1),
(27, 1, 39, ' 2/12 этаж', '\nРассказовка', 5, 'Внуковское', 5257792, 132500, '2019-05-18', 1),
(28, 1, 38, ' 4/23 этаж', '\nРассказовка', 5, 'Внуковское', 5266344, 134127, '2019-05-18', 1),
(29, 1, 41, ' 1/17 этаж', '\nРассказовка', 26, 'Внуковское', 5278202, 147383, '2019-05-18', 1),
(30, 1, 40, ' 19/19 этаж', '\nРассказовка', 5, 'Внуковское', 5300000, 127381, '2019-05-18', 1),
(31, 1, 40, ' 2/12 этаж', '\nРассказовка', 5, 'Внуковское', 5324855, 150000, '2019-05-18', 1),
(32, 1, 36, ' 18/23 этаж', '\nРассказовка', 17, 'Внуковское', 5350000, 134127, '2019-05-18', 1),
(33, 1, 42, ' 14/15 этаж', '\nРассказовка', 5, 'Внуковское', 5350000, 126293, '2019-05-18', 1),
(34, 1, 36, ' 22/24 этаж', '\nРассказовка', 5, 'Внуковское', 5400000, 146793, '2019-05-18', 1),
(35, 1, 41, ' 2/12 этаж', '\nРассказовка', 5, 'Внуковское', 5432157, 147521, '2019-05-18', 1),
(36, 1, 43, ' 1/12 этаж', '\nРассказовка', 5, 'Внуковское', 5455860, 165165, '2019-05-18', 1),
```

```
(37, 1, 37, ' 6/17 этаж', '\nРассказовка', 7, 'Внуковское', 5460700, 141388, '2019-05-18', 1),
(38, 1, 37, ' 8/17 этаж', '\nРассказовка', 20, 'Внуковское', 5487787, 130952, '2019-05-18', 1),
(39, 1, 33, ' 14/14 этаж', '\nРассказовка', 17, 'Внуковское', 5500000, 141026, '2019-05-18', 1),
(40, 1, 39, ' 17/19 этаж', '\nРассказовка', 5, 'Внуковское', 5500000, 137744, '2019-05-18', 1),
(41, 1, 42, ' 10/16 этаж', '\nРассказовка', 5, 'Внуковское', 5500000, 124570, '2019-05-18', 1),
(42, 1, 39, ' 17/19 этаж', '\nРассказовка', 5, 'Внуковское', 5500000, 144492, '2019-05-18', 1),
(43, 1, 40, ' 2/17 этаж', '\nРассказовка', 5, 'Внуковское', 5551099, 156239, '2019-05-18', 1),
(44, 1, 45, ' 1/17 этаж', '\nРассказовка', 5, 'Внуковское', 5568299, 144172, '2019-05-18', 1),
(45, 1, 39, ' 11/17 этаж', '\nРассказовка', 5, 'Внуковское', 5684268, 136081, '2019-05-18', 1),
(46, 1, 49, ' 12/23 этаж', '\nРассказовка', 5, 'Внуковское', 5695000, 143510, '2019-05-18', 1),
(47, 1, 42, ' 21/25 этаж', '\nРассказовка', 24, 'Внуковское', 5700000, 148718, '2019-05-18', 1),
(48, 1, 42, ' 3/17 этаж', '\nРассказовка', 5, 'Внуковское', 5715404, 152632, '2019-05-18', 1),
(49, 1, 40, ' 1/17 этаж', '\nРассказовка', 5, 'Внуковское', 5726053, 153034, '2019-05-18', 1),
(50, 1, 39, ' 6/17 этаж', '\nРассказовка', 27, 'Внуковское', 5800000, 161111, '2019-05-18', 1),
(51, 1, 38, ' 2/17 этаж', '\nРассказовка', 27, 'Внуковское', 5800000, 133028, '2019-05-18', 1),
(52, 1, 38, ' 9/9 этаж', '\nРассказовка', 24, 'Внуковское', 5800000, 159341, '2019-05-18', 1),
(53, 1, 36, ' 3/20 этаж', '\nРассказовка', 8, 'Внуковское', 5800000, 136081, '2019-05-18', 1),
(54, 1, 44, ' 7/24 этаж', '\nРассказовка', 28, 'Внуковское', 5800000, 155263, '2019-05-18', 1),
--
-- Индексы сохранённых таблиц
--

--
-- Индексы таблицы `apartaments`
--
ALTER TABLE `apartaments`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT для сохранённых таблиц
--

--
-- AUTO_INCREMENT для таблицы `apartaments`
--
ALTER TABLE `apartaments`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=23522;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

7.3. Приложение 3. Код алгоритма на языке Python.

```
import numpy as np
from sklearn.linear_model import LinearRegression
import pymysql
connection = pymysql.connect(
    host='localhost',
    user='root',
    password='',
    db='diplom',
    charset='utf8mb4',
    cursorclass=pymysql.cursors.DictCursor
)
class Apartament():
    def __init__(self,classification = None,pricePerMeter  = None,time = None,floor0 = None,
                 floor1 = None,price = None,area = None):
        self.price = price
        self.floor0 = floor0
        self.floor1 = floor1
        self.classification = classification
        self.time = time
        self.pricePerMeter = pricePerMeter
        self.area = area

apartaments = []
query = connection.cursor()
query.execute('SELECT * FROM apartaments WHERE price < 50000000 AND price > 1000000 AND id < 23000')
result = query.fetchall()
x = []
y = []
uniqueNames = {}
for res in result:
     if res['name_of_district'] not in uniqueNames.keys():
         uniqueNames[res['name_of_district']]=len(uniqueNames)
def distToClass(dist):
    return -1 if dist not in uniqueNames.keys() else uniqueNames[dist]
# print(uniqueNames)
for res in result[:-200]:
    floor = res['floor'].replace(' этаж','').split('/')

    appendToX = [res['price_per_square_metr'],
                 res['time_to_metro'],
                 res['area_of_apartaments']
                 ]
    appendToY = res['price']
    x.append(appendToX)
    y.append(appendToY)
    apartaments.append(Apartament(classification=distToClass(res['name_of_district']),
                 pricePerMeter=res['price_per_square_metr'],
                 time=res['time_to_metro'],area=res['area_of_apartaments'],
                 floor0=int(floor[0]),floor1=(int)(floor[1]),price=res['price']))

reg = LinearRegression()
print(len(y))
print(len(x))

reg.fit(x, y)


test = result[-175:]
print(len(test))
print(reg.score(x,y))
```

```
import matplotlib.pyplot as plt
apartaments.sort(key= lambda apartament: apartament.price)
prices = [apartament.price for apartament in apartaments]
classification = [apartament.classification for apartament in apartaments]
pricePerMeter = [apartament.pricePerMeter for apartament in apartaments]
time = [apartament.time for apartament in apartaments]
area = [apartament.area for apartament in apartaments]
floor0 = [apartament.floor0 for apartament in apartaments]
floor1 = [apartament.floor1 for apartament in apartaments]
print("Зависимость цены от площади")
print(np.corrcoef(prices,area)[0][1])
print("Зависимость цены от времени до метро")
print(np.corrcoef(prices,time)[0][1])
print("Зависимость цены от района")
print(np.corrcoef(prices,classification)[0][1])
print("Зависимость цены от этажа квартиры")
print(np.corrcoef(prices,floor0)[0][1])
print("Зависимость цены от этажности в доме")
print(np.corrcoef(prices,floor1)[0][1])
print("Зависимость цены от цены за квадратный метр")
print(np.corrcoef(prices,pricePerMeter)[0][1])
fig, f = plt.subplots(nrows=1,ncols=3,figsize=(40,15))
f[0].scatter(pricePerMeter,prices,color='red',s=2)
f[0].set_xlabel('Цена за квадратный метр')
f[0].set_ylabel('Цена')
f[1].scatter(time,prices,color='blue',s=2)
f[1].set_xlabel('Время до метро')
f[1].set_ylabel('Цена')
f[2].scatter(area,prices,color='yellow',s=2)
f[2].set_xlabel('Площадь')
f[2].set_ylabel('Цена')
plt.show()

testList = [Apartament(classification=distToClass(res['name_of_district']),
                  pricePerMeter=res['price_per_square_metr'],
                  time=res['time_to_metro'],area=res['area_of_apartaments'],
                  floor0=int(floor[0]),floor1=(int)(floor[1]),price=res['price']) for res i
n test]
testList.sort(key = lambda apartament: apartament.price)
x_test = []
      y_test = []
      y1 = []
      x1 = []
      for apartament in testList[:]:
          x_test = [np.array([
                    apartament.pricePerMeter,
                    apartament.time,
                    apartament.area
          ])]
          y_test = reg.predict(x_test)
          if y_test > 0:
              y1.append(y_test)
              x1.append(apartament.price)
          print(y_test,apartament.price)
      print(np.nanmean(prices))
      plt.scatter([i for i in range(len(y1))],x1, color = "red", s = 1)
      plt.scatter([i for i in range(len(y1))],y1, color = "green", s = 1)
      plt.show()
      allTrue = len(x1)
      x1 = np.array(x1)
      y1 = np.array(y1)
      trueCount = 0
      for i in range(len(y1)):
          if abs(y1[i]-x1[i]) < 500000:
              trueCount += 1
      print('Accurancy:',np.round(trueCount/allTrue,2))
```