Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

УДК 004.622

Голубко
Дмитрий Владимирович

Система менеджмента недвижимости и анализа цен на рынке

ДИССЕРТАЦИЯ
на соискание академической степени
магистра

по специальности 1-40 80 05 — Программная инженерия

Научный руководитель Смолякова О. Г.
к.т.н., доцент

Минск 2021

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 3
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ 4
ВВЕДЕНИЕ 5
1 Анализ существующих алгоритмов 7
1.1 Регрессионный анализ 7
1.2 Нейросетевой анализ 14
2 Обзор аналогов 24
2.1 Регрессионный анализ 24
2.2 Анализ с испольнованием нейросетей 25
3 Эксперементальная часть 30
3.1 Сбор данных 30
3.2 Подготовка данных 32
3.3 Регрессионный анализ 33
3.4 Анализ с использованием нейронных сетей 38
3.5 Сравнение результатов 43
ЗАКЛЮЧЕНИЕ 46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 47
Приложение А. Исходный код парсера и анализатора данных 49

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

МНК – Метод наименьших квадратов

ИНС – Искусственная нейронная сеть

MSE – Mean squared error, среднеквадратическая ошибка

MLP – Multilayer perceptron – Многослойный перцептрон

RBF network – Radial basis function network – Сеть радиально-базисных функций

SSR – Sum of squared residuals – Сумма квадратов остатков

SST – Sum of squares total – Общая сумма квадратов

GRNN – General regression neural network – Генерализованная регрессионная нейронная сеть

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Цель диссертационной работы – провести анализ применяемых моделей и алгоритмов, которые используются при прогнозировании цен на рынке недвижимости, проверить адекватность применения методов эконометрического анализа для оценки объектов недвижимости и построение на их основе модели стоимости, применить существующие алгоритмы на подготовленной выборке данных. На основе проведенного анализа выполнить сравнение алгоритмов, определить наиболее эффективный из них. Результаты исследования могут быть полезны для прогнозирования ценообразования на рынке недвижимости, а также при оценке стоимости объектов недвижимости.

Для достижения поставленной цели необходимо решить следующие задачи:

а) Провести анализ применяемых моделей и алгоритмов, которые используются при прогнозировании цен на рынке недвижимости.

б) Реализовать предложенные модели и провести экспериментальные исследования на основе накопленных данных.

в) На основе сравнительного анализа выбрать более эффективный и достоверный.

Объектом исследования выступает процесс формирования цен на недвижимость.

Предметом исследования являются модели и алгоритмы, которые используются для анализа процесса формирования цен.

Основной *гипотезой*, положенной в основу диссертационной работы, является сравнение результатов работы выбранных алгоритмов и выявление более эффективного способа оценки стоимости недвижимости.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, двух глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе представлен анализ существующих применяемых моделей и алгоритмов. Вторая глава посвящена проведению эксперименального исследования, реализацией выбранных алгоритмов анализа, проведению их сравнительного анализа.

ВВЕДЕНИЕ

Переход к рыночным отношениям в экономике и научно-технический прогресс чрезвычайно ускорили темпы внедрения во все сферы социально-экономической жизни общества последних научных разработок в области информационных технологий.

Рынок недвижимости представляет собой механизм, обслуживающий и регулирующий отношения по купле, продаже и аренде недвижимости на основе спроса и предложения. В мировой практике можно выделить следующие типы рынков недвижимости:

– рынок жилой недвижимости;

– рынок коммерческой недвижимости, приносящей доход ее владельцу (офисные, торговые, производственные, складские помещения);

– рынок земельных участков.

Рынок недвижимости делится на первичный и вторичный. Объектом сделок на первичном рынке является новая недвижимость, т.е. только что построенные дома, квартиры, офисные и другие помещения. Их могут продавать застройщики, инвесторы, финансировавшие строительство. На вторичном рынке предоставлено жилье и помещения, которыми уже пользовались по основному назначению. Первичный рынок отражает объемы созданной жилой недвижимости, а объем вторичного рынка определяется другими факторами:

– изменением благосостояния населения;

– доходностью различных инвестиционных объектов;

– мобильностью трудовых ресурсов;

– событиями человеческой жизни (свадьба, развод, рождение ребенка в семье, смена места жительства и др.).

Объекты недвижимости занимают значительную часть ресурсов экономики любой страны. Оценка стоимости – длительный и сложный процесс установления денежного эквивалента стоимости объекта недвижимости. Она требует высокой квалификации оценщика, владеющего методами и инструментарием оценочной деятельности, знающего состояние рынка недвижимости и особенно нужного сегмента, детального значения правовых особенностей сделок с недвижимостью и др [1]. Практика показывает, что для оценки стоимости объекта недвижимости, специалисту требуется значительное время.

С развитием теоретических подходов для создания адекватных моделей поведения рынка недвижимости в западных странах и США одновременно происходило активное внедрение новых интеллектуальных компьютерных

технологий в практику принятия финансовых и инвестиционных решений. Вначале в виде экспертных систем и баз знаний, а затем с конца 80-х - нейросетевых технологий, которые являются адекватным аппаратом для решения задач прогнозирования.

Начало исследования методов обработки информации, называемых сегодня нейросетевыми, было положено несколько десятилетий назад. С течением времени интерес к нейросетевым технологиям то ослабевал, то вновь возрождался. Такое непостоянство напрямую связано с практическими результатами проводимых исследований.

Цена на жилье - вопрос крайне сложный, изучение основных факторы влияния и выяснение правил изменения имеют важное теоретическое и практическое значение для способствования устойчивому и здоровому развитию рынка жилой недвижимости. Автоматизация позволит ускорить процесс принятия решения, учесть большее количество факторов и снизить уровень субъективности.

1 АНАЛИЗ СУЩЕСТВУЮЩИХ АЛГОРИТМОВ

1.1 Регрессионный анализ

В статистическом моделировании регрессионный анализ – это набор статистических процедур для изучения зависимостей между случайными переменными. Он включает в себя множество методов моделирования и анализа взаимосвязей между зависимой переменной и одной или несколькими независимыми переменными, называемых также предикторами или регрессорами.

Регрессионный анализ помогает понять, как значение зависимой переменной изменяется при изменении одной из независимых переменных, в то время как другие независимые переменные остаются фиксированными.

Чаще всего в регрессионном анализе оценивает условное математическое ожидание зависимой переменной с учетом значений, принимаемых независимыми переменными. Во всех случаях оцениваться функция математического ожидания зависимой переменной от независимых переменных, называемая функцией регрессии.

Регрессионный анализ широко используется для численного предсказания, классификации и прогнозирования, где его применение существенно перекрывается с областью машинного обучения.

В настоящее время разработано много методов регрессионного анализа. Наиболее популярными из них являются простая и множественная линейная регрессия, среднеквадратическая и логистическая регрессия. Эти модели, являются параметрическими в том смысле, что функция регрессии определяется конечным числом неизвестных параметров, которые оцениваются на основе данных.

В математической статистике линейная регрессия представляет собой метод аппроксимации зависимостей между входными и выходными переменными на основе линейной модели. Является частью более широкой статистической методики, называемой регрессионным анализом.

В регрессионном анализе входные (независимые) переменные называются также предикторными переменными или регрессорами, а зависимые переменные — критериальными.

Если рассматривается зависимость между одной входной и одной выходной переменными, то имеет место простая линейная регрессия. Для этого определяется уравнение регрессии и строится соответствующая прямая, известная как линия регрессии(рис. 1.1).

$$y = ax + b \tag{1.1}$$

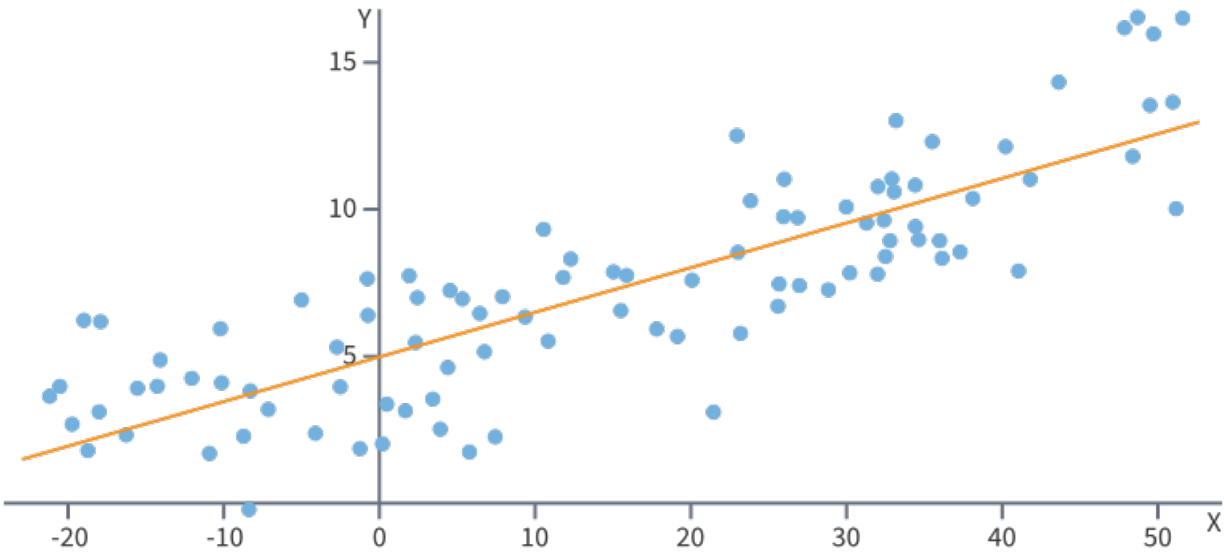

Рисунок 1.1 – Линия регрессии

Коэффициенты *a* и *b*, называемые также параметрами модели, определяются таким образом, чтобы сумма квадратов отклонений точек, соответствующих реальным наблюдениям данных, от линии регрессии была бы минимальной. Коэффициенты обычно оцениваются методом наименьших квадратов.

Метод наименьших квадратов (МНК) — математический подход для оценки параметров моделей (например, регрессионной) на основании экспериментальных данных, содержащих случайные ошибки.

Если данные известны с некоторой погрешностью, то вместо неизвестного точного значения параметра модели используется приближенное. Поэтому параметры модели должны быть рассчитаны так, чтобы минимизировать разницу между экспериментальными данными и теоретическими (вычисленными при помощи предложенной модели).

Мерой рассогласования между фактическими значениями и значениями, оцененными моделью в методе наименьших квадратов, служит сумма квадратов разностей между ними, т.е.:

$$\sum_{i=1}^{N}(\mathrm{y}' - \mathrm{y})^2 \qquad (1.2)$$

где y' — оценка, полученная с помощью модели;

y — фактическое наблюдаемое значение.

Очевидно, что лучшей будет та модель, которая минимизирует данную сумму.

Важнейшим применением МНК в анализе данных является линейная регрессия, где параметры регрессионной модели вычисляются таким образом, чтобы сумма квадратов расстояний от линии регрессии до фактических значений данных была минимальной [2].

Если ищется зависимость между несколькими входными и одной выходной переменными, то имеет место множественная линейная регрессия. Соответствующее уравнение имеет вид:

$$\mathrm{Y} = \mathrm{b}_0 + \mathrm{b}_1\mathrm{x}_1 + \mathrm{b}_2\mathrm{x}_2 + \cdots + \mathrm{b}_n\mathrm{x}_n \tag{1.3}$$

где n — число входных переменных.

Очевидно, что в данном случае модель будет описываться не прямой, а гиперплоскостью. Коэффициенты уравнения множественной линейной регрессии подбираются так, чтобы минимизировать сумму квадратов отклонения реальных точек данных от этой гиперплоскости.

Преимущество множественной линейной регрессии по сравнению с простой заключается в том, что использование в модели нескольких входных переменных позволяет увеличить долю объяснённой дисперсии выходной переменной, и таким образом улучшить соответствие модели данным. Т.е. при добавлении в модель каждой новой переменной коэффициент детерминации растёт.

Коэффициент детерминации – Статистический показатель, отражающий объясняющую способность регрессии *a:X -> Y* и равный отношению суммы квадратов регрессии SSR к общей вариации SST:

$$\mathrm{r}^2 = \frac{SSR}{SST} = \frac{\sum_i(\mathrm{a}(\mathrm{x}_i) - \bar{y})^2}{\sum_i(\mathrm{y}_i - \bar{y})^2} \tag{1.4}$$

где $\mathrm{x}_l = (\mathrm{x}_i, \mathrm{y}_i)_{i=1}^{l}$ — набор данных из *l* наблюдений.
x_i — вектор признаков *i*-го наблюдения,
$\mathrm{y}_i \in \mathrm{Y}$ — y_i принадлежит *Y*;

Данный показатель является статистической мерой согласия, с помощью которой можно определить, насколько уравнение регрессии соответствует реальным данным.

Коэффициент детерминации изменяется в диапазоне от 0 до 1. Если он равен 0, это означает, что связь между переменными регрессионной мо-

дели отсутствует и вместо нее для оценки значения выходной переменной можно использовать простое среднее ее наблюдаемых значений. Напротив, если коэффициент детерминации равен 1, это соответствует идеальной модели, когда все точки наблюдений лежат точно на линии регрессии, т.е. сумма квадратов их отклонений равна 0.

На практике, если коэффициент детерминации близок к 1, это указывает на то, что модель работает очень хорошо (имеет высокую значимость), а если к 0, то это означает низкую значимость модели, когда входная переменная плохо «объясняет» поведение выходной, т.е. линейная зависимость между ними отсутствует. Очевидно, что такая модель будет иметь низкую эффективность.

В некоторых случаях коэффициент детерминации может принимать небольшие отрицательные значения, если модель получилась «бесполезной» и ее предсказания хуже, чем оценки на основе среднего значения [3].

Линейная регрессия была первым видом регрессионного анализа, который был тщательно изучен и начал широко использоваться в практических приложениях. Это связано с тем, что в линейных моделях оценивание параметров проще, а также с тем, что статистические свойства полученных оценок легче определить.

Линейная регрессия имеет много практических применений. Большинство приложений попадают в одну из двух широких категорий:

– Если целью является прогнозирование, линейную регрессию можно использовать для подгонки модели к наблюдаемому набору данных.

– Если цель заключается в том, чтобы объяснить изменчивость выходной переменной, можно применить линейный регрессионный анализ для количественной оценки силы взаимосвязи между выходной и входными переменными.

Среднеквадратическая регрессия – разновидность регрессии, где при определении параметров модели используется обобщение метода наименьших квадратов (МНК) – метод наименьших средних квадратов.

Иными словами, в процессе подгонки модели к данным минимизируется не сумма квадратов остатков регрессии, а их средний квадрат:

$$E[(Y - F(X))^2] \tag{1.5}$$

где E — операция усреднения,
Y — зависимая переменная,
X — вектор независимых переменных.

Это становится возможным благодаря тому, что МНК допускает широкое обобщение, когда вместо минимизации суммы квадратов остатков можно минимизировать их некоторую положительно определённую квадратичную форму.

Смысл данного подхода заключается в том, что к результатам классического МНК (т.е. квадратам остатков) применяется дополнительное линейное преобразование - усреднение. Как известно, одним из предположений регрессии является предположение о нормальности остатков, которое в практических случаях не соблюдается. Усреднение позволяет снизить степень влияния отклонения остатков от нормального распределения на качество построенной модели.

Метод наименьших средних квадратов был сформулирован Бернардом Уидроу и Тедом Хоффом в 1960 году и применён при обучении нейронных сетей с помощью алгоритма обратного распространения ошибки.

В математическом и статистическом моделировании зависимой (выходной) называется переменная модели, которая зависит от входных переменных и случайных факторов, воздействующих на моделируемый процесс или объект.

Выходная переменная представляет результаты работы модели. Она изменяется (варьирует) под воздействием изменения входной переменной и случайных факторов. Изучение изменчивости выходной переменной при изменении входной и является целью моделирования.

В статистическом моделировании и машинном обучении независимой (входной) переменной называют величину, от которой зависит изменение выходной переменной, при этом целью построения модели является аппроксимация этой зависимости.

Аппроксимация – математический метод, в основе которого лежит замена одних математических объектов другими, близкими к исходным в том или ином смысле, но более простыми. [4]

Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, тех, параметры которых легко вычисляются или известны заранее).

Например, в линейной регрессии некоторая неизвестная функция, описывающая реальные наблюдения, аппроксимируется уравнением прямой, а если наблюдаемые данные носят нелинейный характер, то полиномами и т.д.

моделью, которая использует логистическую функцию для моделирования зависимости выходной переменной от набора входных в случае, когда первая является бинарной.

Это разновидность множественной регрессии, общее назначение которой состоит в анализе связи между несколькими независимыми переменными (называемыми также регрессорами или предикторами) и зависимой переменной. Регрессия в общем виде применяется, когда входные и выходная переменные непрерывные. А логистическая регрессия лучшим образом подходит, когда выходная переменная принимает только два значения.

Важность логистический регрессии обусловлена тем, что многие задачи анализа данных могут быть решены с помощью бинарной классификации или сведены к ней.

Например, с помощью логистической регрессии можно оценивать вероятность наступления (или не наступления) некоторого события: пациент болен (здоров), заемщик вернул кредит (допустил просрочку) и т.д. Благодаря этому логистическую регрессию можно рассматривать как мощный инструмент поддержки принятия решений.

Как известно, все регрессионные модели могут быть записаны в виде формулы:

$$y = F(x_1, x_2, \cdots, x_n) \tag{1.6}$$

Например, если рассматривается исход по займу, задается переменная y со значениями 1 и 0, где 1 означает, что соответствующий заемщик расплатился по кредиту, а 0 — что имел место дефолт.

Однако здесь возникает проблема: множественная регрессия не «знает», что переменная отклика бинарная по своей природе. Это неизбежно приведет к модели с предсказываемыми значениями большими 1 и меньшими 0. Но такие значения вообще не допустимы для первоначальной задачи. Таким образом, множественная регрессия просто игнорирует ограничения на диапазон значений для y.

Для решения проблемы задача регрессии может быть сформулирована иначе: вместо предсказания бинарной переменной мы предсказываем непрерывную переменную со значениями на отрезке [0,1] при любых значениях независимых переменных. Это достигается применением следующего регрессионного уравнения (логит-преобразование):

$$p = \frac{1}{1 + e^{-y}} \tag{1.7}$$

где p — вероятность того, что произойдет интересующее событие,
e — основание натуральных логарифмов 2,71…,
y — стандартное уравнение регрессии.

Зависимость, связывающая вероятность события и величину y, показана на следующем графике:

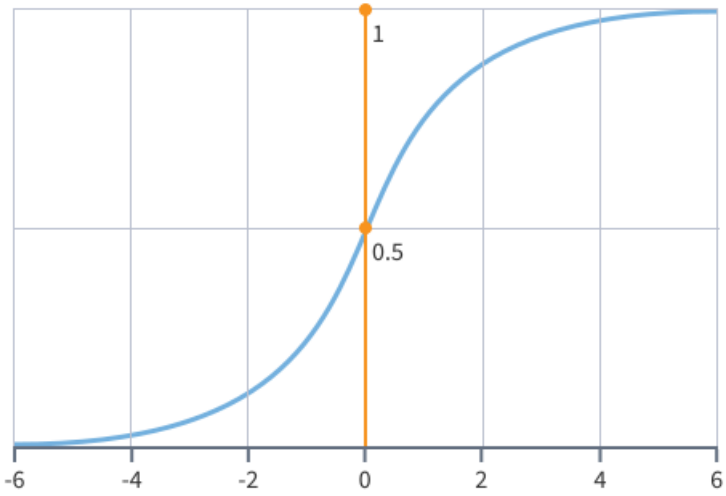

Рисунок 1.2 – Зависимость, связывающая вероятность события и величину y

Преобразование вида:

$$P' = \log_e(\frac{P}{1-P}) \tag{1.8}$$

называют логистическим, или логит-преобразованием.

Существует несколько способов нахождения коэффициентов логистической регрессии. На практике часто используют метод максимального правдоподобия. Он применяется в статистике для получения оценок параметров генеральной совокупности по выборочным данным.

Логистическая регрессия является традиционным статистическим инструментом для расчета коэффициентов (баллов) скоринговой карты на основе накопленной кредитной истории.

Генеральная совокупность — это совокупность всех объектов или наблюдений, относительно которых исследователь намерен делать выводы при решении конкретной задачи. В ее состав включаются все объекты, которые подлежат изучению.

Объем генеральной совокупности может быть очень велик, и на практике рассмотреть все ее элементы не представляется возможным. Поэтому

обычно из генеральной совокупности извлекаются выборки, на основе анализа которых аналитик пытается сделать вывод о свойствах всей совокупности, скрытых в ней закономерностях, действующих правилах и т.д. При этом выборки должны быть репрезентативными.

Регрессионный анализ является одним из наиболее распространенных методов обработки результатов экспериментов при изучении зависимостей в естественных науках, экономике, технике и др. областях.

В аналитических технологиях Data Mining элементы регрессионного анализа широко используются для решения задач прогнозирования, оценивания, классификации, выявления зависимостей между признаками.

Основы регрессионного анализа были заложены А. Лежандром и Карлом Гауссом в их работах по методу наименьших квадратов в начале 19 в. [5]

1.2 Нейросетевой анализ

Нейронная сеть (также искусственная нейронная сеть, ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы.

Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений подобных тем, что делает человеческий мозг. Самыми распространенными применениями нейронных сетей является:

– Классификация — распределение данных по параметрам. Например, на вход дается набор людей и нужно решить, кому из них давать кредит, а кому нет. Эту работу может сделать нейронная сеть, анализируя такую информацию как: возраст, платежеспособность, кредитная история и тд.

– Предсказание — возможность предсказывать следующий шаг. Например, рост или падение акций, основываясь на ситуации на фондовом рынке.

– Распознавание — в настоящее время, самое широкое применение нейронных сетей. Используется в Google, когда вы ищете фото или в камерах телефонов, когда оно определяет положение вашего лица и выделяет его и многое другое.

Классификация искусственных нейронных сетей представлена на рисунке 1.3

Задачи прогнозирования (предсказания) можно разбить на два основных класса: классификация и регрессия.

Рисунок 1.3 – Классификация нейронных сетей

В задачах классификации нужно бывает определить, к какому из нескольких заданных классов принадлежит данный входной набор. Примерами могут служить предоставление кредита (относится ли данное лицо к группе высокого или низкого кредитного риска), диагностика раковых заболеваний (опухоль, чисто), распознавание подписи (поддельная, подлинная). Во всех этих случаях, очевидно, на выходе требуется всего одна номинальная переменная. Чаще всего (как в этих примерах) задачи классификации бывают

двузначными, хотя встречаются и задачи с несколькими возможными состояниями.

В задачах регрессии требуется предсказать значение переменной, принимающей (как правило) непрерывные числовые значения: завтрашнюю цену акций, расход топлива в автомобиле, прибыли в следующем году и т.п.. В таких случаях в качестве выходной требуется одна числовая переменная.

Исскусственные нейронные сети обычно состоят из нейронов и связывающих их синапсов.

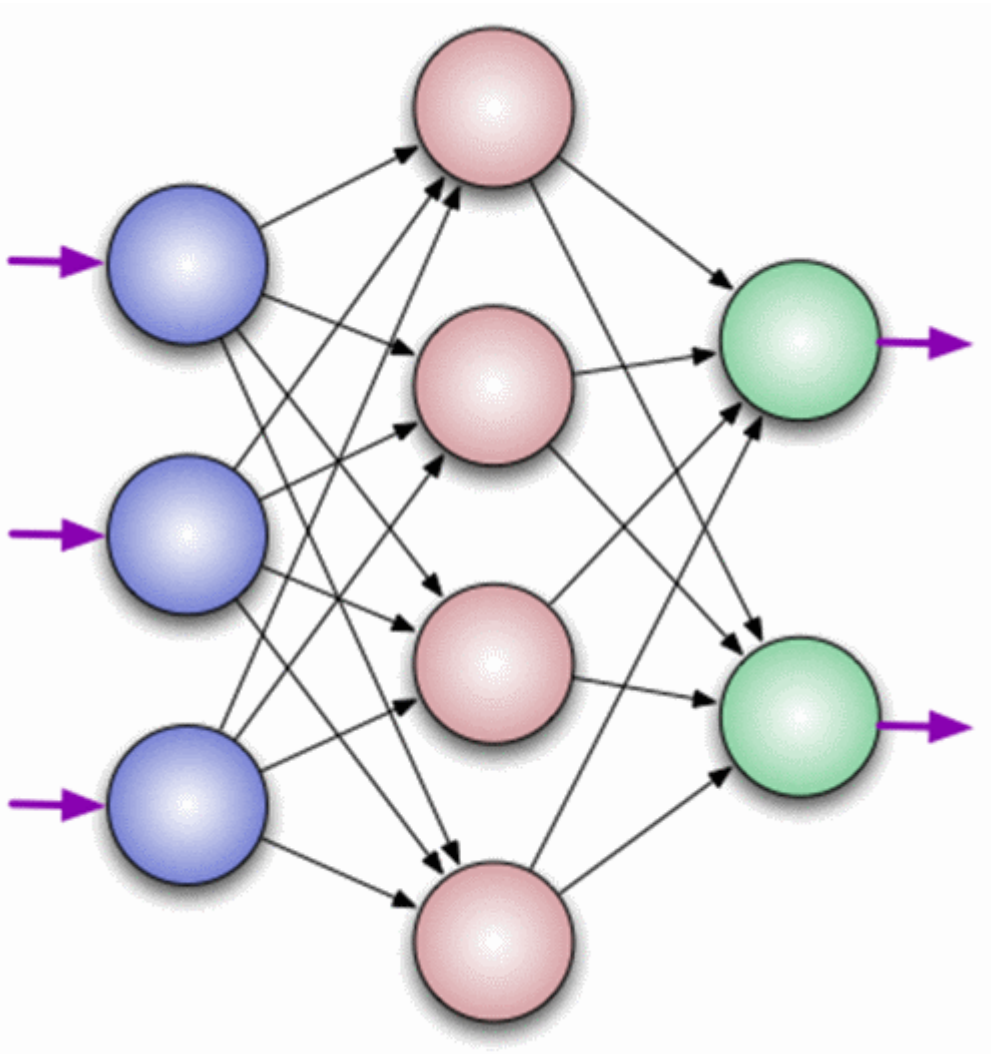

Рисунок 1.4 – Виды нейронов

Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной (слева), скрытый (посередине) и выходной (справа). Также есть нейрон смещения и контекстный нейрон. В том случае, когда нейросеть состоит из большого количества нейронов, вводят термин слоя. Соответственно, есть входной слой, который получает информацию, *n* скрытых слоев (обычно их не больше 3), которые ее обрабатывают

и выходной слой, который выводит результат. У каждого из нейронов есть 2 основных параметра: входные данные (input data) и выходные данные (output data). В случае входного нейрона: input=output. В остальных, в поле input попадает суммарная информация всех нейронов с предыдущего слоя, после чего, она нормализуется, с помощью функции активации и попадает в поле output [6].

Нейрон смещения или bias нейрон — это третий вид нейронов, используемый в большинстве нейросетей. Особенность этого типа нейронов заключается в том, что его вход и выход всегда равняются 1 и они никогда не имеют входных синапсов. Нейроны смещения могут, либо присутствовать в нейронной сети по одному на слое, либо полностью отсутствовать, 50/50 быть не может (красным на схеме обозначены веса и нейроны которые размещать нельзя). Соединения у нейронов смещения такие же, как у обычных нейронов — со всеми нейронами следующего уровня, за исключением того, что синапсов между двумя bias нейронами быть не может. Следовательно, их можно размещать на входном слое и всех скрытых слоях, но никак не на выходном слое, так как им попросту не с чем будет формировать связь.

Нейрон смещения нужен для того, чтобы иметь возможность получать выходной результат, путем сдвига графика функции активации вправо или влево.

Также нейроны смещения помогают в том случае, когда все входные нейроны получают на вход 0 и независимо от того какие у них веса, они все передадут на следующий слой 0, но не в случае присутствия нейрона смещения.

Синапс – это связь между двумя нейронами. У синапсов есть 1 параметр — вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть 3 нейрона, которые передают информацию следующему. Тогда у нас есть 3 веса, соответствующие каждому из этих нейронов. У того нейрона, у которого вес будет больше, та информация и будет доминирующей в следующем нейроне (пример — смешение цветов). На самом деле, совокупность весов нейронной сети или матрица весов — это своеобразный мозг всей системы. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат.

Функция активации – это способ нормализации входных данных. То есть, если на входе у вас будет большое число, пропустив его через функцию активации, вы получите выход в нужном вам диапазоне. Самые основные функции активации: Линейная, Сигмоид (Логистическая) и Гиперболический тангенс. Главные их отличия — это диапазон значений.

Линейная функция почти никогда не используется, за исключением

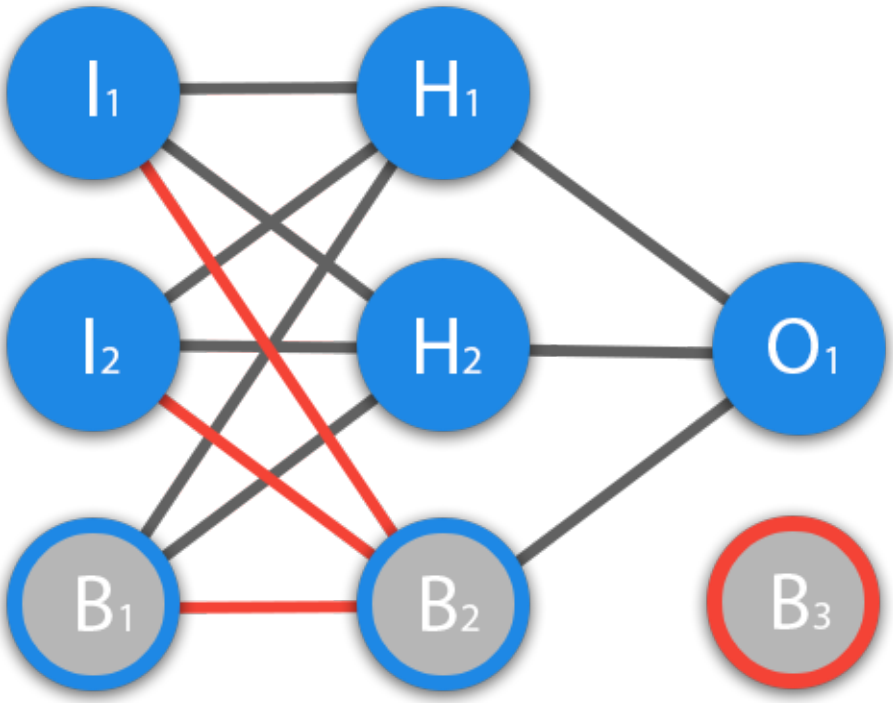

Рисунок 1.5 – Нейрон смещения

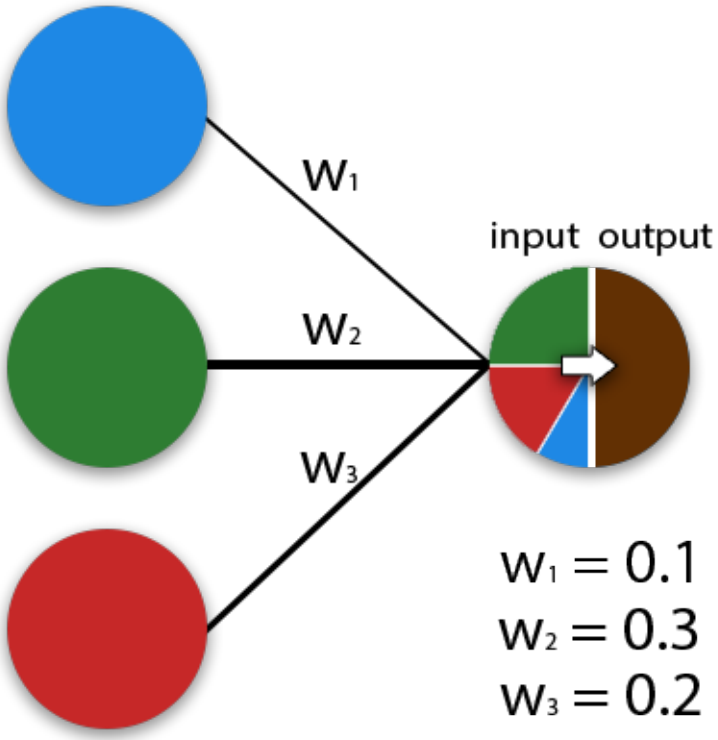

Рисунок 1.6 – Синапсы для связи нейронов

случаев, когда нужно протестировать нейронную сеть или передать значение без преобразований.

Сигмоид – это самая распространенная функция активации, ее диапа-

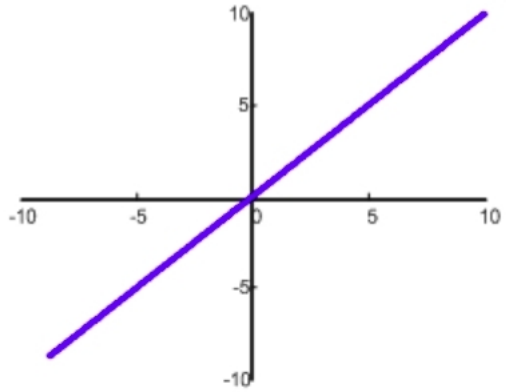

Рисунок 1.7 – Линейная функция активации

зон значений [0,1]. Именно на ней показано большинство примеров в сети, также ее иногда называют логистической функцией. Соответственно, если в вашем случае присутствуют отрицательные значения (например, акции могут идти не только вверх, но и вниз), то вам понадобиться функция которая захватывает и отрицательные значения.

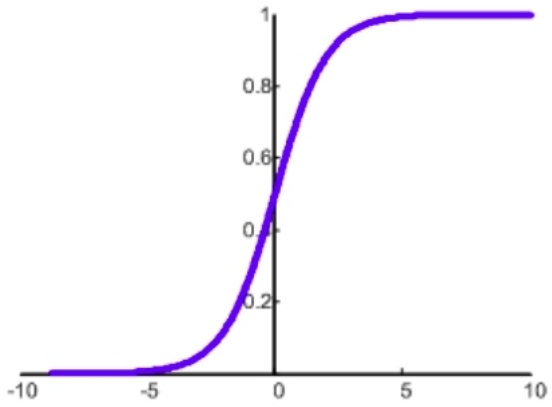

Рисунок 1.8 – Сигмоид

Имеет смысл использовать гиперболический тангенс, только тогда, когда ваши значения могут быть и отрицательными, и положительными, так как диапазон функции [-1,1]. Использовать эту функцию только с положительными значениями нецелесообразно так как это значительно ухудшит результаты вашей нейросети.

Ошибка — это процентная величина, отражающая расхождение между ожидаемым и полученным ответами. Ошибка формируется каждую эпоху и должна идти на спад. Ошибку можно вычислить разными путями, например: Mean Squared Error (далее MSE), Root MSE и Arctan. Здесь нет какого-либо ограничения на использование, как в функции активации, и можно использо-

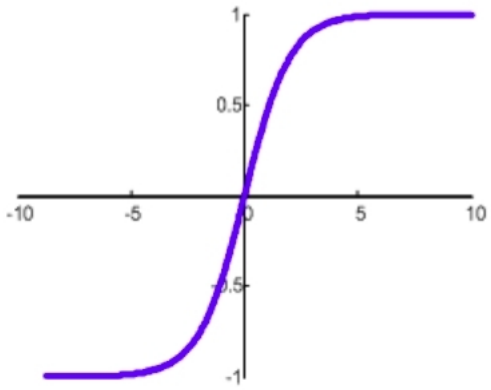

Рисунок 1.9 – Гиперболический тангенс

вать любой метод, который будет приносить вам наилучший результат. Стоит лишь учитывать, что каждый метод считает ошибки по разному. У Arctan, ошибка, почти всегда, будет больше, так как он работает по принципу: чем больше разница, тем больше ошибка. У Root MSE будет наименьшая ошибка, поэтому, чаще всего, используют MSE, которая сохраняет баланс в вычислении ошибки.

Обучение искусственных нейронных сетей происходит с помощью заранее подготовленной выборки данных (тренировочный сет).

Общее количество тренировочных сетов, пройденных нейронной сетью, называется итерацией.

Эпохой называется количество пройденных наборов тренировочных сетов. Чем больше эпоха, тем лучше натренирована сеть и соответственно, ее результат.

Для получения корректных результатов работы искусственной нейронной сети, её необходимо обучить. Существует несколько методов обучения нейронных сетей, самые распространенные из них: метод обратного распространения, метод упругого распространения, генетический алгоритм.

Метод обратного распространения, использует алгоритм градиентного спуска.

Применение алгоритма обратного распространения ошибки — один из известных методов, используемых для глубокого обучения нейронных сетей прямого распространения (такие сети ещё называют многослойными персептронами). Этот метод относят к методу обучения с учителем, поэтому требуется задавать в обучающих примерах целевые значения.

Сегодня нейронные сети прямого распространения используются для решения множества сложных задач. Если говорить об обучении нейронных сетей методом обратного распространения, то тут пользуются двумя проходами по всем слоям нейросети: прямым и обратным. При выполнении прямо-

го прохода осуществляется подача входного вектора на входной слой сети, после чего происходит распространение по нейронной сети от слоя к слою. В итоге должна осуществляться генерация набора выходных сигналов — именно он, по сути, является реакцией нейронной сети на этот входной образ. При прямом проходе все синаптические веса нейросети фиксированы. При обратном проходе все синаптические веса настраиваются согласно правил коррекции ошибок, когда фактический выход нейронной сети вычитается из желаемого, что приводит к формированию сигнала ошибки. Такой сигнал в дальнейшем распространяется по сети, причём направление распространения обратно направлению синаптических связей. Именно поэтому соответствующий метод и называют алгоритмом с обратно распространённой ошибкой. Синаптические веса настраивают с целью наибольшего приближения выходного сигнала нейронной сети к желаемому.

Цель обучения нейросети при использовании алгоритма обратного распространения ошибки — это такая подстройка весов нейросети, которая позволит при приложении некоторого множества входов получить требуемое множество выходов нейронов (выходных нейронов). Можно назвать эти множества входов и выходов векторами. В процессе обучения предполагается, что для любого входного вектора существует целевой вектор, парный входному и задающий требуемый выход. Эту пару называют обучающей. Работая с нейросетями, мы обучаем их на многих парах.

Также можно сказать, что алгоритм использует стохастический градиентный спуск и продвигается в многомерном пространстве весов в направлении антиградиента, причём цель — это достижение минимума функции ошибки.

При практическом применении метода обучение продолжают не до максимально точной настройки нейросети на минимум функции ошибки, а пока не будет достигнуто довольно точное его приближение. С одной стороны, это даёт возможность уменьшить количество итераций обучения, с другой — избежать переобучения нейронной сети.

Для реализации метода обратного распространения ошибки необходимо выполнить следующие действия:

а) Инициализировать синаптические веса случайными маленькими значениями.

б) Выбрать из обучающего множества очередную обучающую пару; подать на вход сети входной вектор.

в) Выполнить вычисление выходных значений нейронной сети.

г) Посчитать разность между выходом нейросети и требуемым выходом (речь идёт о целевом векторе обучающей пары).

д) Скорректировать веса сети в целях минимизации ошибки.

е) Повторять для каждого вектора обучающего множества шаги б-д, пока ошибка обучения нейронной сети на всём множестве не достигнет уровня, который является приемлемым.

Сегодня существует много модификаций алгоритма обратного распространения ошибки. Возможно обучение не «по шагам» (выходная ошибка вычисляется, веса корректируются на каждом примере), а «по эпохам» в offline-режиме (изменения весовых коэффициентов происходит после подачи на вход нейросети всех примеров обучающего множества, а ошибка обучения нейронной сети усредняется по всем примерам).

Обучение «по эпохам» более устойчиво к выбросам и аномальным значениям целевой переменной благодаря усреднению ошибки по многим примерам. Зато в данном случае увеличивается вероятность «застревания» в локальных минимумах. При обучении «по шагам» такая вероятность меньше, ведь применение отдельных примеров создаёт «шум», «выталкивающий» алгоритм обратного распространения из ям градиентного рельефа.

К плюсам можно отнести простоту в реализации и устойчивость к выбросам и аномалиям в данных, и это основные преимущества. Но есть и минусы:

– неопределенно долгий процесс обучения;

– вероятность «паралича сети» (при больших значениях рабочая точка функции активации попадает в область насыщения сигмоиды, а производная величина приближается к 0, в результате чего коррекции весов почти не происходят, а процесс обучения «замирает»;

– алгоритм уязвим к попаданию в локальные минимумы функции ошибки.

Появление алгоритма стало знаковым событием и положительно отразилось на развитии нейросетей, ведь он реализует эффективный с точки зрения вычислительных процессов способ обучения многослойного персептрона. В то же самое время, было бы неправильным сказать, что алгоритм предлагает наиболее оптимальное решение всех потенциальных проблем [7].

Одним из серьезных недостатков алгоритма с обратным распространением ошибки, используемого для обучения многослойных нейронных сетей, является слишком долгий процесс обучения, что делает неприменимым использование данного алгоритма для широкого круга задач, которые требуют быстрого решения. В настоящее время известно достаточное количество алгоритмов ускоряющих процесс обучения, одним из них является метод упругого распространения ошибки, который был предложен М. Ридмиллером (M.Riedmiller) и Г. Брауном (H.Braun).

Нейросеть можно обучать с учителем и без.

Обучение с учителем — это тип тренировок присущий таким проблемам как регрессия и классификация. Иными словами здесь вы выступаете в роли учителя, а НС в роли ученика. Вы предоставляете входные данные и желаемый результат, то есть ученик посмотрев на входные данные поймет, что нужно стремиться к тому результату который ему предоставлен.

Обучение без учителя — этот тип обучения встречается не так часто. Здесь нет учителя, поэтому сеть не получает желаемый результат или же их количество очень мало. В основном такой вид тренировок присущ НС у которых задача состоит в группировке данных по определенным параметрам.

Существует еще такой метод, как обучение с подкреплением. Такой способ применим тогда, когда можно основываясь на результатах полученных от НС, дать ей оценку. НС поредоставляется право найти любой способ достижения цели, до тех пор пока он будет давать хороший результат. Таким способом, сеть начнет понимать чего от нее хотят добиться и пытается найти наилучший способ достижения этой цели без постоянного предоставления данных “учителем”.

2 ОБЗОР АНАЛОГОВ

2.1 Регрессионный анализ

Исходя из проведенного анализа способов применения различных методов регрессии можно сделать вывод, что наиболее подходящим способом для анализа процесса формирования цен на недвижимость является множественный линейный регрессионный анализ.

Данное утверждение подтверждается анализом статьи [8].

В целях оценки недвижимости может применяться либо многофакторный, либо однофакторный регрессионный анализ. В первом случае строится множественная регрессионная модель, описывающая зависимость стоимости оцениваемого объекта от нескольких независимых определяющих факторов, значения которых определяются из анализа рыночных данных. Этими факторами могут быть как физические характеристики объекта (площадь, качество отделки и т.п.), так и характеристики его местоположения (удаленность от транспортных магистралей, экологическая обстановка и т.п.).

При однофакторном регрессионном анализе рассматривается зависимость переменной – стоимости единицы сравнения – от одной независимой (контролируемой) переменной. Значения остальных независимых переменных считаются фиксированными. В качестве независимой переменной X обычно используется показатель «общая площадь», за зависимую переменную Y принимается показатель «стоимость 1м кв. общей площади».

В случае, когда рассматривается зависимость между одной зависимой переменной Y и несколькими независимыми переменными X1, X2, ... ,Xn говорят о множественной линейной регрессии.

В данной статье достаточно хорошо показан объем данных и набор парамтерв, используемых в дальнейшем анализе, построена отдельные модели по количеству комнат в квартире.

Однако в данной статье проведен анализ рынка первичного жилья, который не всегда отражает реальную рыночную стоимость. Также отсутствует общая модель с учетом всех параметров, по которой невозможно сделать сравнение с отдельными моделями по количеству комнат.

Данные недостатки решено устранить, помимо отдельных моделей по количеству комнат, принято решение также построить дополнительно модели по району города, в котором продается объект недвижимости и провести сравнение полученных результатов с целью выбора наилучшей модели.

2.2 Анализ с испольнованием нейросетей

Приведенный выше анализ существующих типов нейронных сетей позволяет сделать вывод, что целесообразным является использование нейронной сети для задачи прогнозирования, обучение с учителем с применением алгоритма обратного распространения ошибки.

Данное удтверждение подтверждается анализом статьи [9]. В данной статье проведено исследование и разработка методики оценки стоимости недвижимости с использованием нейросетевых технологий. Были решены задачи предварительного отбора факторов, оказывающих влияние на рыночную стоимость квартир; подготовлена обучающая выборка и определены оптимальные типы и характеристики, а также метода ее обучениия.

Обучающая выборка построена для проектирования и обучения нейронной сети «с учителем», к реализации предполагается 3 типа сетей: многослойный персептрон (MLP); сеть радиально-базисных функций (RBF); обобщенно-регрессионная нейронная сеть (GRNN).

Многослойный персептрон - это класс искусственных нейронных сетей прямого распространения, состоящих как минимум из трех слоёв: входного, скрытого и выходного. За исключением входных, все нейроны использует нелинейную функцию активации.

При обучении MLP используется обучение с учителем и алгоритм обратного распространения ошибки.

Обобщенная схема многослойного перспетрона показана на рисунке 2.1

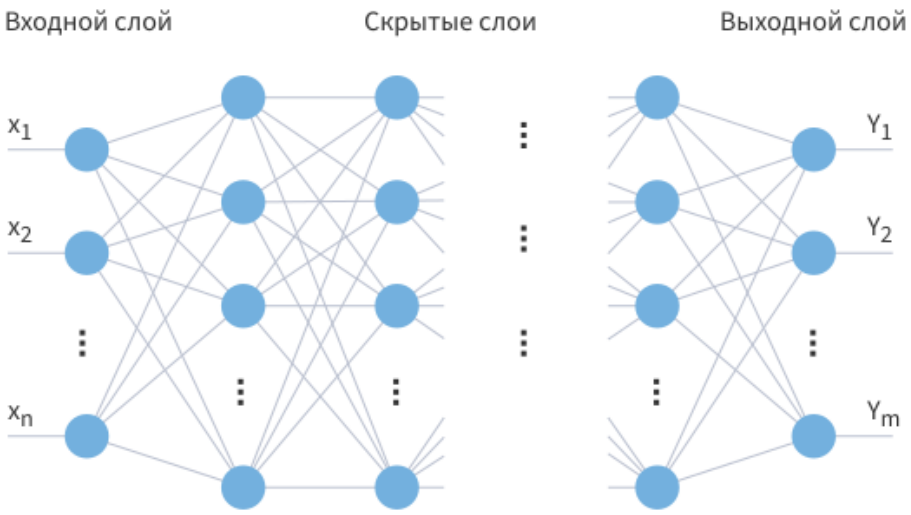

Рисунок 2.1 – Схема многослойного перспетрона

В качестве активационных функций нейронов используются сигмоидальные: логистическая или гиперболический тангенс.

MLP показали возможность находить приближённые решения для чрезвычайно сложных задач. В частности, они являются универсальным аппроксиматором функций, поэтому с успехом используются в построении регрессионных моделей. Поскольку классификацию можно рассматривать как частный случай регрессии, когда выходная переменная категориальная, на основе MLP можно строить классификаторы.

Пик популярности MLP в машинном обучении пришёлся на 1980-е годы в таких областях, как распознавание речи и изображений, системах машинного перевода. Однако позднее они столкнулись с конкуренцией с другими технологиями машинного обучения, такими, как машины опорных векторов. Интерес к многослойным персептронам вернулся благодаря успехам глубокого обучения.

Радиально-симметричные функции – простейший класс функций. В принципе, они могут быть использованы в разных моделях (линейных и нелинейных) и в разных сетях (многослойных и однослойных). Традиционно термин RBF сети ассоциируется с радиально-симметричными функциями в однослойных сетях, имеющих структуру, представленную на рисунке 2.2.

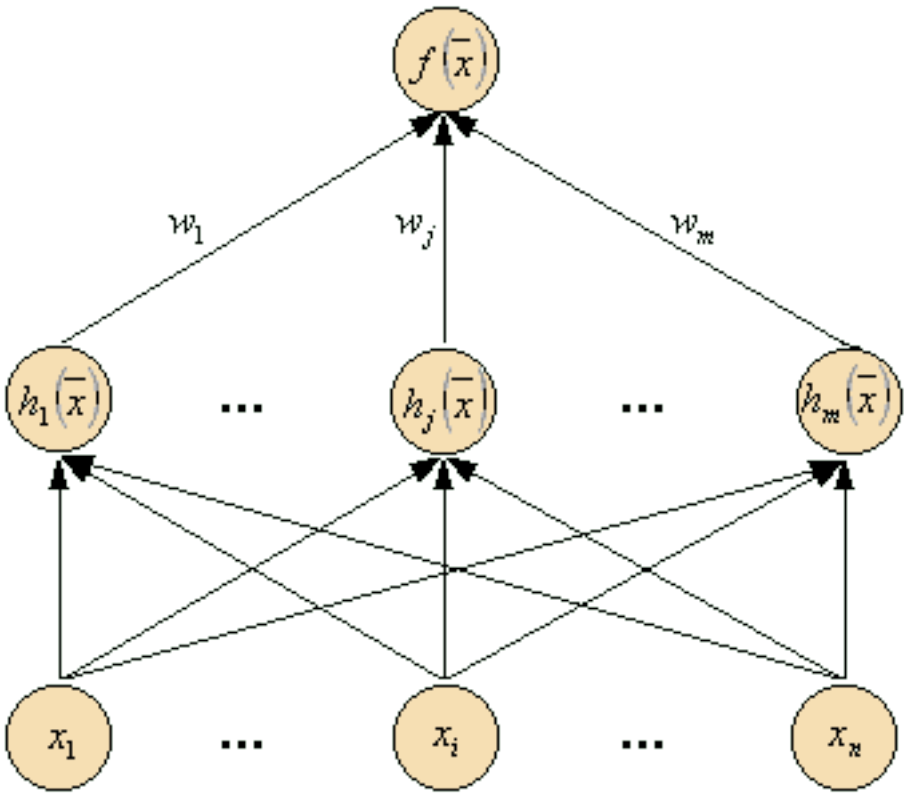

Рисунок 2.2 – Схема RBF сети

То есть, каждый из n компонентов входного вектора подается на вход

m базисных функций и их выходы линейно суммируются с весами.

Обобщенно-регрессионная нейронная сеть (GRNN) устроена аналогично вероятностной нейронной сети (PNN), но она предназначена для решения задач регрессии, а не классификации. Обобщенная схема GRNN сети представлена на рисунке 2.3. Как и в случае PNN-сети, в точку расположения каждого обучающего наблюдения помещается гауссова ядерная функция. Мы считаем, что каждое наблюдение свидетельствует о некоторой нашей уверенности в том, что поверхность отклика в данной точке имеет определенную высоту, и эта уверенность убывает при отходе в сторону от точки. GRNN-сеть копирует внутрь себя все обучающие наблюдения и использует их для оценки отклика в произвольной точке. Окончательная выходная оценка сети получается как взвешенное среднее выходов по всем обучающим наблюдениям, где величины весов отражают расстояние от этих наблюдений до той точки, в которой производится оценивание (и, таким образом, более близкие точки вносят больший вклад в оценку).

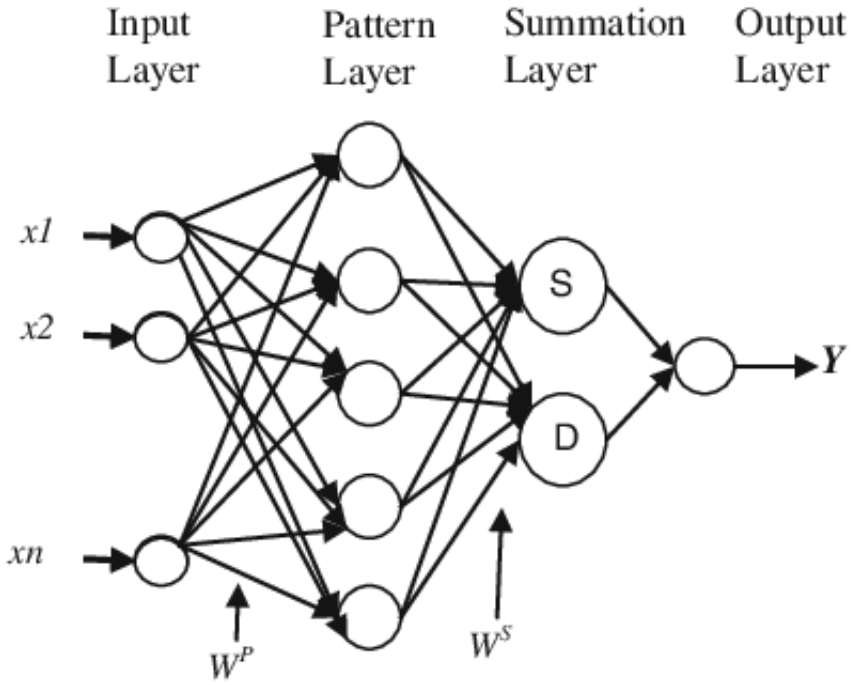

Рисунок 2.3 – Схема GRNN сети

Первый промежуточный слой сети GRNN состоит из радиальных элементов. Второй промежуточный слой содержит элементы, которые помогают оценить взвешенное среднее. Для этого используется специальная процедура. Каждый выход имеет в этом слое свой элемент, формирующий для него взвешенную сумму. Чтобы получить из взвешенной суммы взвешенное среднее, эту сумму нужно поделить на сумму весовых коэффициентов. Последнюю сумму вычисляет специальный элемент второго слоя. После этого в выход-

ном слое производится собственно деление (с помощью специальных элементов "деления"). Таким образом, число элементов во втором промежуточном слое на единицу больше, чем в выходном слое. Как правило, в задачах регрессии требуется оценить одно выходное значение, и, соответственно, второй промежуточный слой содержит два элемента.

Можно модифицировать GRNN-сеть таким образом, чтобы радиальные элементы соответствовали не отдельным обучающим случаям, а их кластерам. Это уменьшает размеры сети и увеличивает скорость обучения. Центры для таких элементов можно выбирать с помощью любого предназначенного для этой цели алгоритма (выборки из выборки, K-средних или Кохонена).

GRNN-сеть обучается почти мгновенно, но может получиться большой и медленной (хотя здесь, в отличие от PNN, не обязательно иметь по одному радиальному элементу на каждый обучающий пример, их число все равно будет большим). Как и сеть RBF, сеть GRNN не обладает способностью экстраполировать данные.

Проведенное сравнение полученных результатов(рис. 2.4), полученных в статье [9] позволяет сделать выводы.

Тип	Обучение	Контроль	Тест
MLP			
Средняя ошибка	-7815.80	57316.31	-15922.77
Абсолютная средняя ошибка	149701.10	176512.5	203957
Коэф. регрессии	0.19	0.26	0.20
Корреляция	0.98	0.96	0.98
РБФ-сеть			
Средняя ошибка	-4.168e-09	130071	-60229.71
Абсолютная средняя ошибка	356535.5	397528.1	390934.1
Коэф. регрессии	0.5297183	0.663065	0.43
Корреляция	0.84	0.76	0.92
GRNN-сеть			
Средняя ошибка	30.27681	-22907.96	-339579
Абсолютная средняя ошибка	32448.87	282364.5	551109
Коэф. регрессии	0.11	0.57	0.81
Корреляция	0.99	0.82	0.58

Рисунок 2.4 – Сравнение полученных результатов

GRNN-сеть показала очень хорошие результаты на тестовой выборке, в то время как на тестовой выборке ее эффективность оказалась значительно ниже, чем у прочих рассмотренных сетей. Наиболее вероятным здесь событием является нерешенная проблема переобучения. То есть минимизировалась не та ошибка, которая ожидается от сети при подаче совершенно новых значений. Другими словами, у данной сети отсутствует способность обобщать результаты работы на новые наблюдения.

РБФ-сеть не продемонстрировала высоких результатов, однако несомненным ее достоинством является более высокая скорость обучения.

Многослойный персептрон является наиболее подходящим вариантом решения задачи определения стоимости жилых квартир. Полученные данные позволяют с достаточной точностью прогнозировать стоимость квартир по заданным параметрам.

Среди достоинств данной статьи, которые будут использованы при проведении эксперимента, можно отметить построение нескольких типов нейронных сетей и дальнейшее сравнение их результатов. Рассмотрены и реализованы 3 типа сетей: многослойный персептрон (MLP); сеть радиально-базисных функций (RBF); обобщенно-регрессионная нейронная сеть (GRNN). Многослойный персептрон является наиболее подходящим вариантом решения задачи определения стоимости жилых квартир.

Среди недостатков данной статьи, которые предполагается устранить, можно отметить небольшой объем выборки, а также построение нескольких моделей отдельно по какому-либо признаку. Увеличение объема выборки и построение отдельных сетей должно повысить качетсво получаемых результатов.

3 ЭКСПЕРЕМЕНТАЛЬНАЯ ЧАСТЬ

3.1 Сбор данных

В связи с тем, что рынок вторичного жилья лучше соответствует рыночным принципам формирования цен на основе спроса и предложения, в отличие от цен, устанавливаемых компанией-застройщиком жилья на первичном рынке, определение рыночной стоимости, как наиболее вероятной цены продажи объекта недвижимости, более целесообразно провести на примере объектов недвижимости вторичного рынка жилья. При создании модели оценки жилой недвижимости в качестве входных параметров были включены факторы, представленные в таблице 3.1:

Таблица 3.1 – Характеристики недвижимости

Параметр	Описание
Price	Стоимость данного объекта
District	Район города
Rooms count	Количество комнат
Floor	Этаж
House Type	Тип дома(панельный, монолитный, кирпичный, каркасно-блочный)
Area full	Полная площадь, кв.м
Area living	Жилая площадь, кв.5
Build year	Год постройки дома
Bathroom type	Тип санузла(раздельный, совмещенный)
Finishing	Ремонт(хороший, плохой, без отделки, и т.д.)

Исходные данные были взяты из базы проданной недвижимости в г. Минске на момент конца 2020 года. Данные собраны с сайта realt.by с помощью технологии веб-скрапинга. Это технология получения данных путем извлечения их со страниц веб-ресурсов. Для скрапинга использовался язык программирования Ruby. Данный язык удобен тем, что поддерживает множество сторонних библиотек, помогающих в сборе данных. Всего было собрано данных о более чем 8 тысяч продающихся объектов недвижимости, которые затем были экспортированы в формате csv. Данные хранятся в виде таблицы, часть которой показана на рисунке 3.1

price	district	rooms_count	floor	house_type	area_full	area_living	build_year	bathroom_type	remont
550000	Маркса, Кирова	4	5	кирпичный	136.2	88.3	1950	2 сан.узла	без отделки
79900	Минск Мир (Minsk World)	3	14	каркасно-блочный	75.4	67.9	2020	раздельный	без отделки
45900	Маяковского	1	8	кирпичный	30.84	16.77	1972	раздельный	хороший ремонт
319000	Дрозды	4	3	каркасно-блочный	160	70.58	2008	2 сан.узла	евроремонт
125000	Маяковского	3	12	каркасно-блочный	89.1	51.1	2014	раздельный	без отделки
74500	Малиновка	2	8	панельный	54.7	28.7	2004	раздельный	нормальный ремонт
46900	Зеленый луг	1	6	панельный	35	16.93	1980	раздельный	нормальный ремонт
149900	Лебяжий (Ржавец)	3	14	каркасно-блочный	113.7	57.2	2015	2 сан.узла	строительная отделка
230800	Независимости, Кедышко, Волгоградск	2	1	монолитный	115.4	64.3	2016	совмещенный	без отделки
149900	Богдановича, Куйбышева, Веры Хоруж	3	2	силикатные блоки	102.3	57.8	2004	раздельный	евроремонт
53868	нач.Партизанского, пл.Ванеева	1	12	панельный	44.89	17.96	2021	раздельный	без отделки
45000	Минск Мир (Minsk World)	1	5	каркасно-блочный	31.8	24.4	2019	совмещенный	без отделки
52200	Михалово	1	3	силикатные блоки	41.74	27.56	2022	2 сан.узла	без отделки
169900	Макаенка	3	10	каркасно-блочный	94.6	83.4	2017	раздельный	отличный ремонт
105000	Уручье	2	8	каркасно-блочный	56.4	29.5	2012	раздельный	отличный ремонт
55000	Юго-Запад	1	6	кирпичный	37.2	17.2	1992	раздельный	хороший ремонт
155000	Лебяжий (Ржавец)	4	24	каркасно-блочный	114.2	65.2	2020	раздельный	без отделки
97600	Червякова, Шевченко	2	5	каркасно-блочный	61	32.6	2021	раздельный	без отделки
210000	Червякова, Шевченко	Фактически 3-ком	8	каркасно-блочный	107.3	65.3	2008	2 сан.узла	евроремонт
253460	Независимости, Кедышко, Волгоградск	4	11 этажей	монолитный	169	81.5	2010	2 сан.узла	без отделки
135000	Каменная горка	3	5	панельный	79.7	47.3	2012	совмещенный	отличный ремонт
74500	Веснянка	2	6	панельный	51.8	29.3	1989	раздельный	хороший ремонт
80000	Брилевичи	2	2	панельный	61.8	31.2	2008	раздельный	отличный ремонт
58262	Я.Коласа-Рига, Некрасова, Восточная	1	13	панельный	46.61	19	2021	совмещенный	без отделки
77662	Я.Коласа-Рига, Некрасова, Восточная	2	6	панельный	62.13	33	2021	раздельный	без отделки
99275	Я.Коласа-Рига, Некрасова, Восточная	3	7	панельный	79.42	47	2021	раздельный	без отделки
58000	Михалово	1	9	каркасно-блочный	50	18	2019	раздельный	без отделки
149900	Пушкина-Глебки-Притыцкого-Ольшевск	3	13	каркасно-блочный	97.3	54.7	2012	2 сан.узла	евроремонт
0	Минск Мир (Minsk World)	3	18	каркасно-блочный	88.6	81	2020	раздельный	без отделки

Рисунок 3.1 – Исходные данные в csv формате

3.2 Подготовка данных

Как можно заметить, полученные данные являются как количественными, так и качественными. Количественные данные остаются без изменений, для качественных(материал дома, качество ремонта) были введены числовые характеристики. Величина характеристики выставлялась в зависимотси от медианной стоимости всех объектов недвижимости с данной характеристикой. Так, например, для материалов строительства были выставлены значения от 1 до 4 в следующем порядке: Панельный дом - 1, монолитный - 2, каркасно-блочный - 3, кирпичный - 4.

Проведен анализ полученных данных. На рисунке 3.2 приведена гистограмма площадей полученных объектов недвижимости, которая позволяет говорить что наиболее популярная недвижмиость имеет площадь от 40 до 60 квадратных метров.

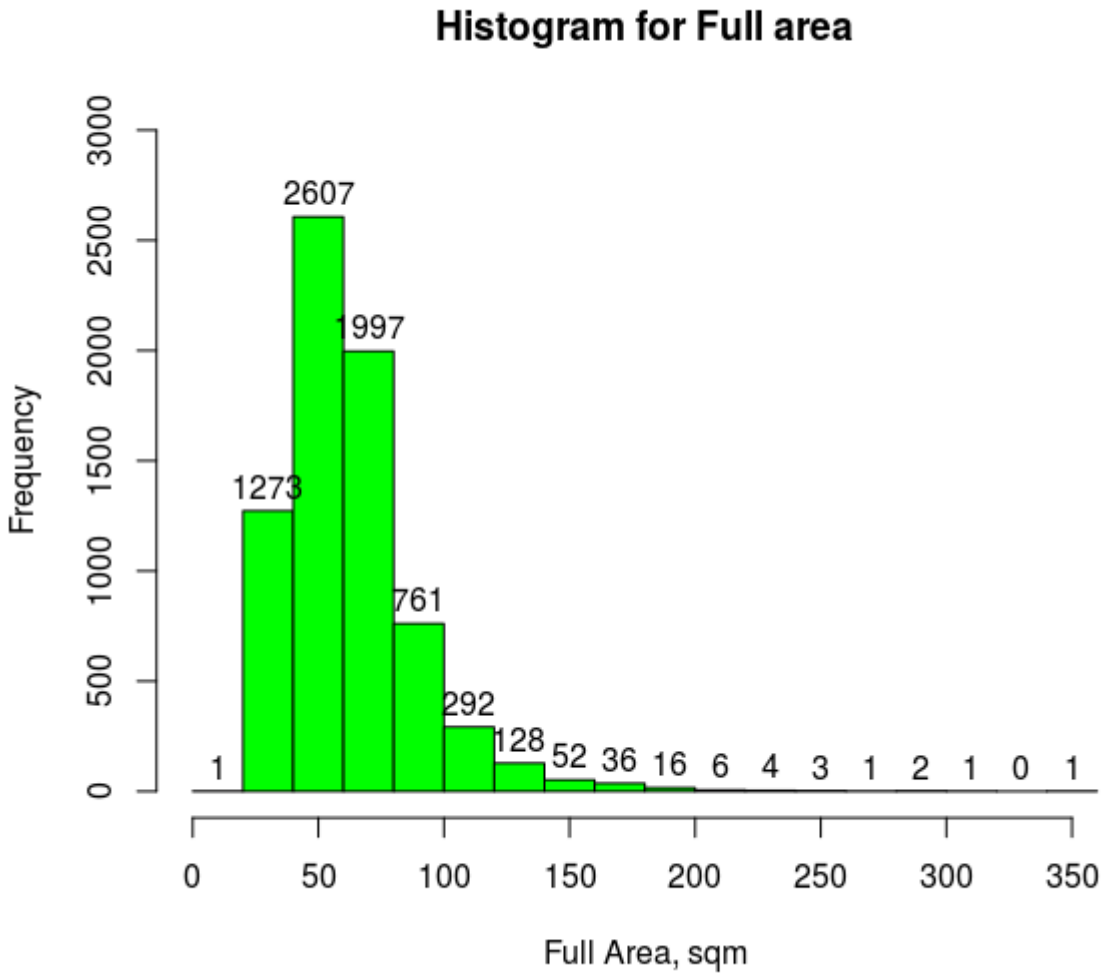

Рисунок 3.2 – Гистограмма площадей

На рисунке 3.3 приведена гистограмма санузлов. полученных объек-

тов недвижимости, на которой видно что подавляющее большинство квартир имеют раздельный санузел.

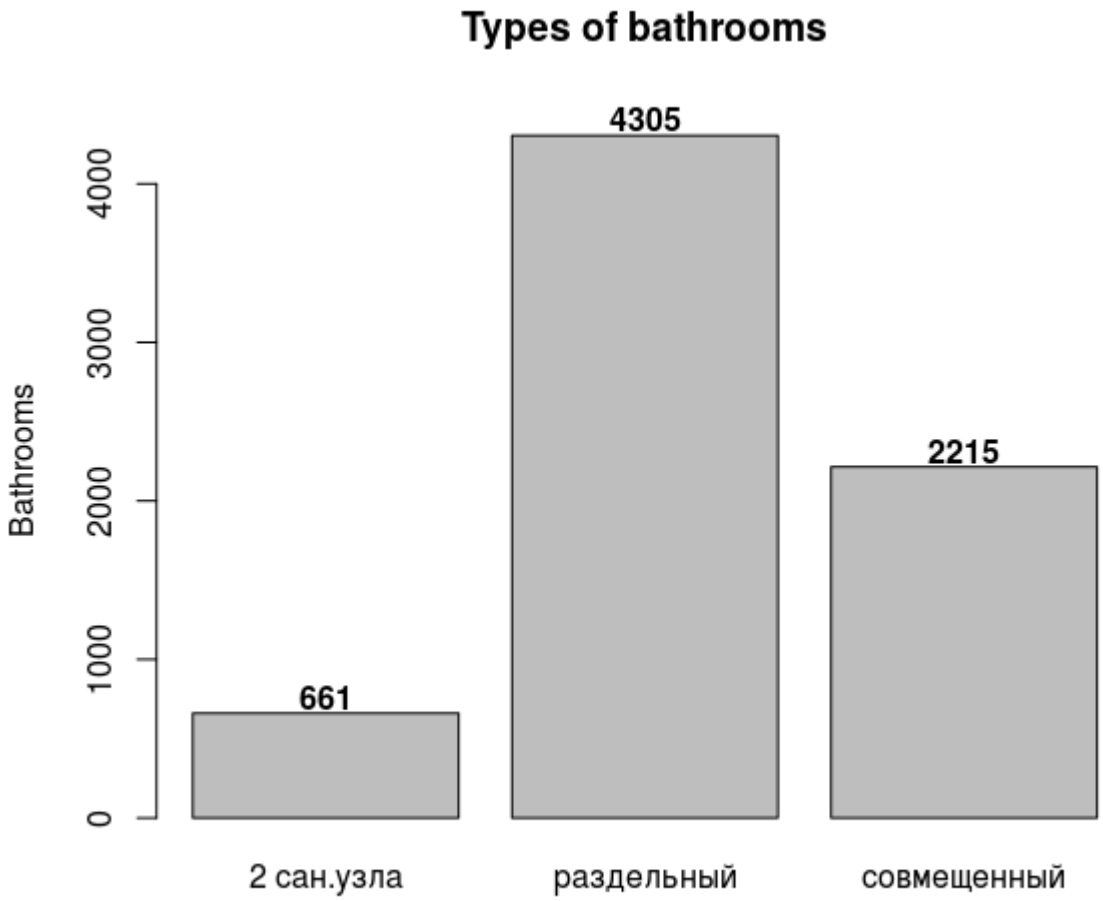

Рисунок 3.3 – Гистограмма санузлов

На рисунке 3.4 приведена гистограмма комнат полученных объектов недвижимости, на которой видно что большинство квартир имеют 2 комнаты.

На рисунке 3.5 можно увидеть, что большинство квартир продаются в диапазоне цен от 40 до 100 тысяч долларов.

3.3 Регрессионный анализ

В качестве первого способа анализа рынка недвижимости было решено использовать регрессионный анализ.

В качестве инструмента для анализа данных было решено использовать язык программирования R. R — язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта GNU. Язык

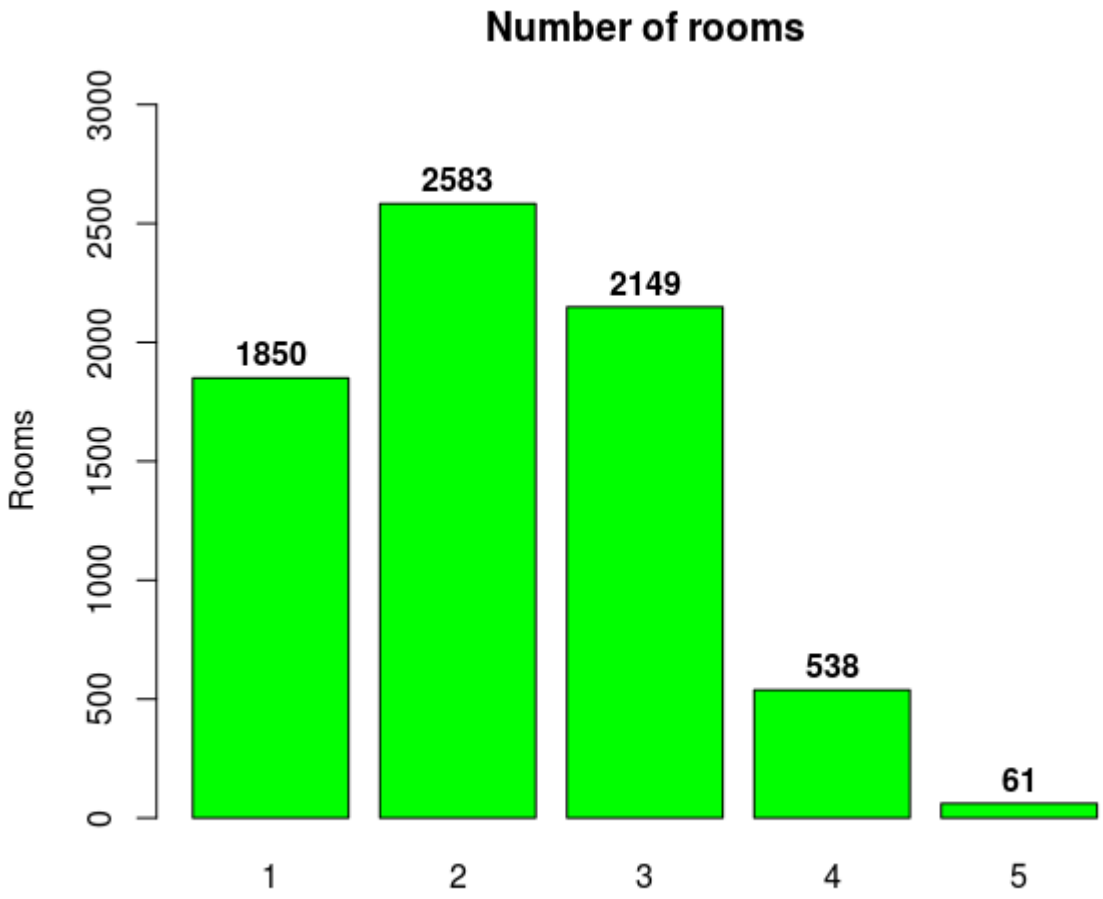

Рисунок 3.4 – Гистограмма комнат

создавался как аналогичный языку S, разработанному в Bell Labs, и является его альтернативной реализацией, хотя между языками есть существенные отличия. [10].

У R есть ряд преимуществ по сравнению с Python. Он интуитивно понятен, а потому удобен, с точки зрения написания кода. Чтобы писать программы на R, необязательно соблюдать четкую структуру – можно просто вводить последовательный набор команд, и этого будет вполне достаточно.

Язык R создавался специально для анализа данных, поэтому все конструкции синтаксиса достаточно емки и понятны. Python — более универсальный и многоцелевой язык, что, естественно, усложняет его понимание.

Среди достоинств языка R можно отметить следующие:

– Удобные и понятные языковые конструкции

– Базовые статистические методы реализованы в качестве стандартных функций, что значительно повышает скорость разработки.

– Есть несколько отличных пакетов для визуализации. Можно строить

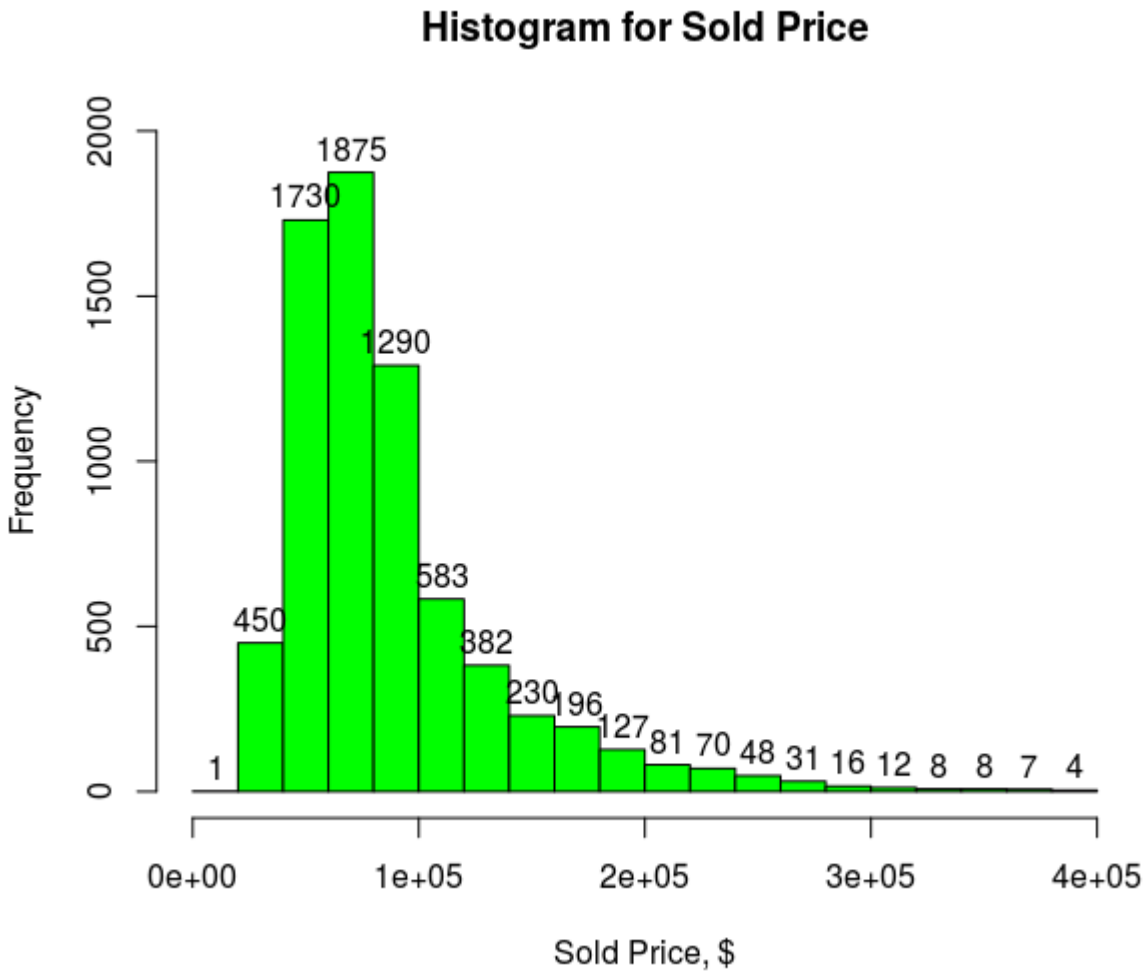

Рисунок 3.5 – Гистограмма стоимости продажи

и двумерную графику (диаграммы, боксплоты), а также и трехмерные модели. Результаты проведенной работы часто становятся значительно понятнее и выразительнее.

– Для R разработано огромное количество дополнительных пакетов.

Распределение цены объекта недвижимости в зависимости от его площади показана на рисунке 3.6. Можно заметить, что стоимость цены расчет с ростом площади, что ожидаемо, однако говорить о линейной зависимости не приходится. Коэффициент детерминации R^2 равен 0.69, что хоть и является достаточно высоким показателем, однако не позволяет говорить о сильной взаимосвязи цены и площади и точном прогнозе цены по одной лишь площади.

Построим множественную модель регрессии с учетом всех приведенных выше параметров.

Данные характеристики показывают нам, что такие параметры как общая площадь, жилая площадь, этаж, тип дома, год постройки, качество ре-

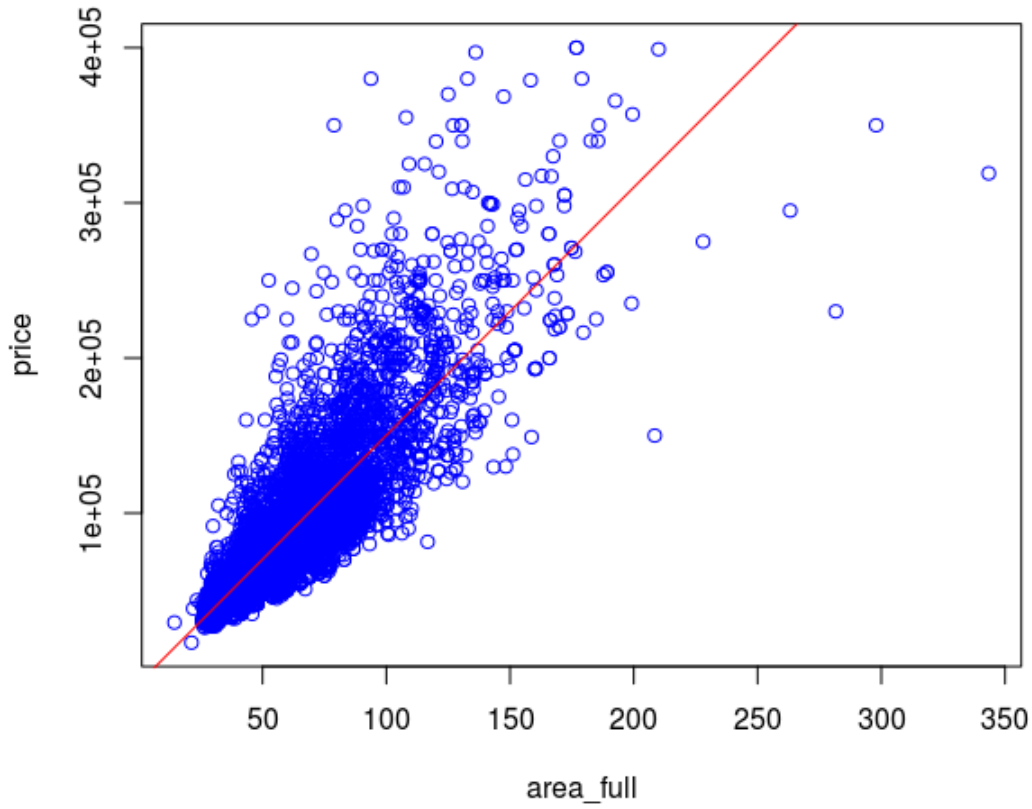

Рисунок 3.6 – Зависимость цены от площади

монта, район города являются значимыми и оказывают существенное влияние на формирование окончательной цены. Тип санузла не влияет на процесс формирования цены. Поэтому его можно исключить из модели.

Полученный коэффициент детерминации R^2 равен 0.79 оказался выше чем в модели парной регрессии, однако он все еще недостоточно велик чтобы точно формировать цену на объекты недвижимости. Далее модель стоимости строится отдельно по числу комнат, так как этот параметр имеет наибольшее значение после площади объекта недвижимости и этот параметр является дискретным.

Построенные отдельно модели по числу комнат показали схожие результаты и коэффициент детерминации R^2 равен 0.74. Полученная матрица корреляции представлена на рисунке 3.8. Исходя из данной матрицы можно сделать вывод, что на конечную цену существенное влияние оказывает район города, общая площадь и жилая площадь.

Для повышения результатов прогнозирования были построены отдельно модели по району, которые также показали схожие результаты, однако усредненный коэффициент детерминации R^2 составил 0.87. Таким образом

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    160349.78   28083.20   5.710 1.18e-08 ***
district_id       778.67      22.57  34.498  < 2e-16 ***
rooms_count     -4238.26     490.21  -8.646  < 2e-16 ***
floor             248.82      54.92   4.531 5.97e-06 ***
house_type_id    4151.37     247.43  16.778  < 2e-16 ***
area_full        1332.37      26.21  50.828  < 2e-16 ***
area_living       281.88      32.46   8.684  < 2e-16 ***
build_year       -101.10      14.05  -7.194 6.92e-13 ***
remont_type_id   4449.79     196.87  22.602  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22670 on 7140 degrees of freedom
Multiple R-squared:  0.787,     Adjusted R-squared:  0.7867
F-statistic:  3297 on 8 and 7140 DF,  p-value: < 2.2e-16
```

Рисунок 3.7 – Характеристики множественной модели регрессии

данная модель показывает лучшие результаты.

В таблице 3.2 приведено сравнение эффективности построенных моделей.

Таблица 3.2 – Сравнительная характеристика построенных моделей

Модель	Коэффициент детерминации R^2	Критерий Фишера F	Остаточная стандартная ошибка RSE
Парная регрессия(зависимость цены от площади)	0.69	16660	26900
Множественная модель	0.79	2930	22670
Отдельные модели по числу комнат	0.74	666	10990
Отдельные модели по району	0.87	209	8604

На рисунке 3.9 приведено сравнение прогнозируемой цены на недвижимиость с фактической. Можно видеть, что полученная модель позволяет достаточно точно прогнозировать цену на объект недвижимости. Исходя из

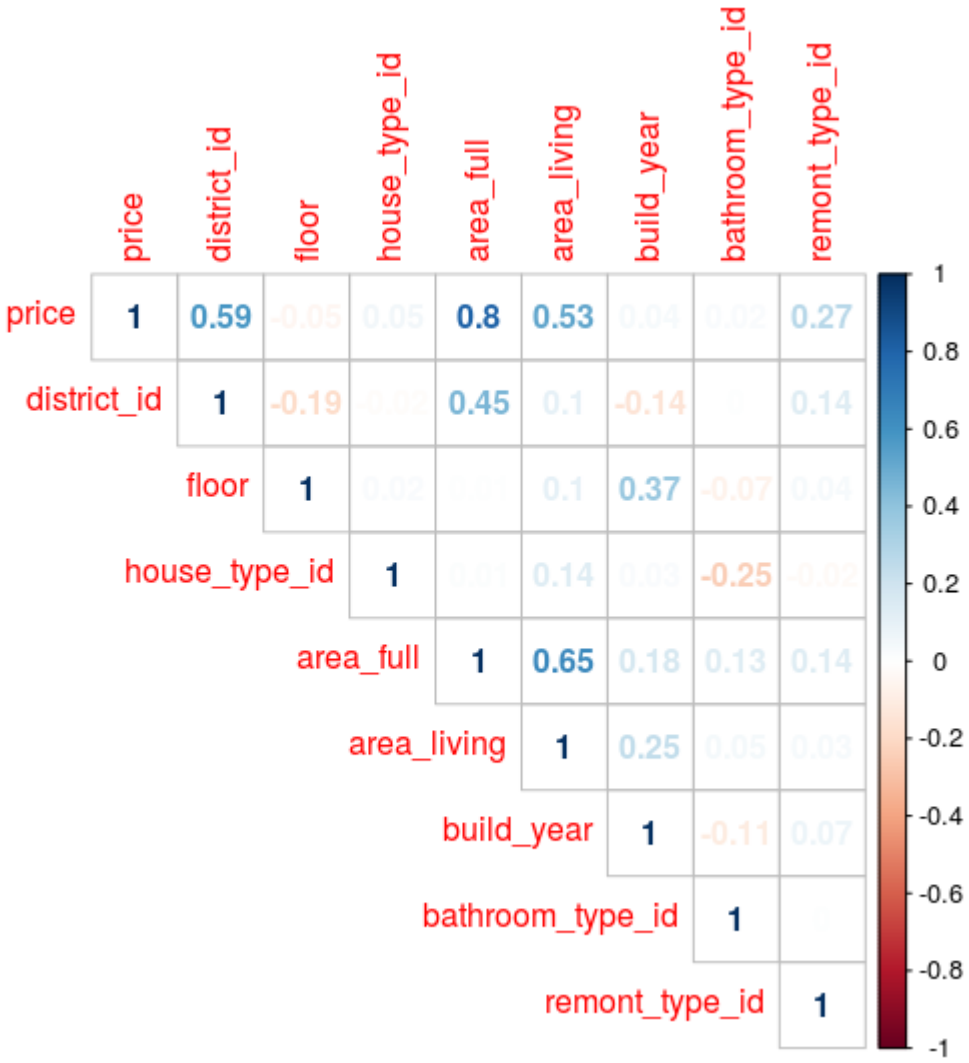

Рисунок 3.8 – Матрица корреляции

этого можно сделать вывод, что предложенная модель оказалась в целом эффективной и может быть использована для прогнозирования цены, однако результаты зависят от района продажи квартиры.

3.4 Анализ с использованием нейронных сетей

Для сравнения также было решено разработать методику оценки стоимости недвижимости с использованием нейронных сетей. Задача оценки недвижимости схематично представлена на рисунке 3.10

Для достижения цели необходимо выбрать факторы, влияющие на рыночную стоимость объектов недвижимости, подготовить выборку для обучения нейронной сети. Обучающая выборка построена для проектирования и обучения нейронной сети с учителем, поскольку такой тип нейронных сетей больше всего подходит для задач, когда имеется большой набор настоящих данных для обучения алгоритма. Исходя из сравнительного анализа несколь-

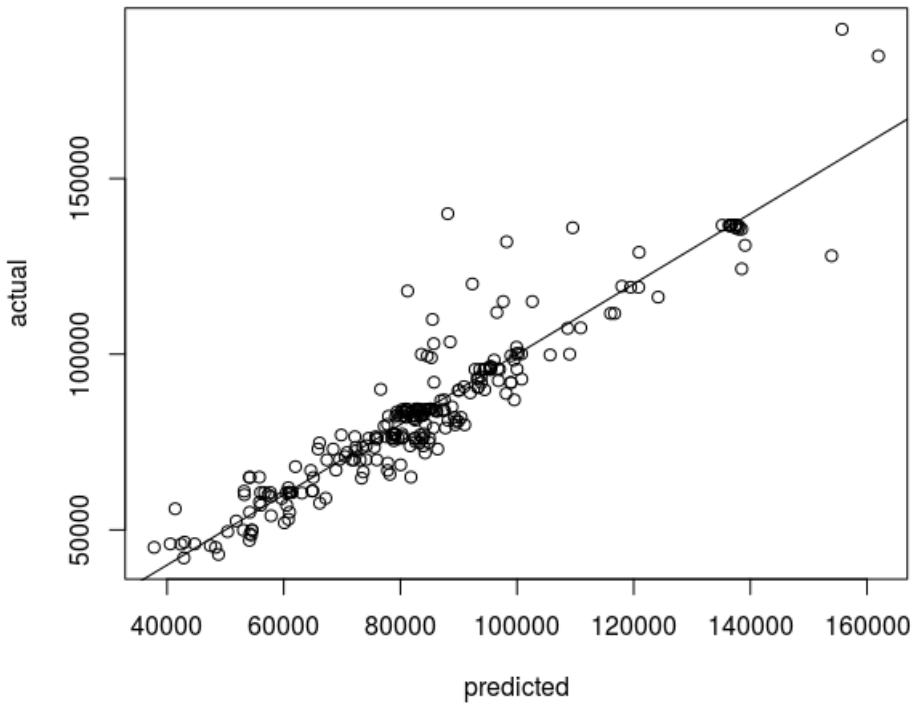

Рисунок 3.9 – Сравнение прогнозируемой цены с фактической

ких типов нейронных сетей с учителем, проведенного в статье, было решено использовать нейронную сеть многослойный персептрон с использованием метода обратного распространения ошибки. Многослойным персептроном называют нейронную сеть прямого распространения, где входной сигнал распространяется от слоя к слою в прямом направлении. В общем представлении такая нейронная сеть состоит из:

– множества входных узлов, образующих входной слой;
– одного или нескольких скрытых слоев вычислительных нейронов;
– одного выходного слоя нейронов.

Обощенная схема многослойного персептрона показана на рисунке 3.11

В качестве инструментального средства проектирования нейронной сети была выбрана STATISTICA Neural Networks. Для обучения многослойных персептронов в пакете STATISTICA Нейронные сети реализовано пять различных алгоритмов обучения. Это хорошо известный алгоритм обратного распространения, быстрые методы второго порядка – спуск по сопряженным градиентам и Левенберга–Маркара, а также методы быстрого распространения и «дельта–дельта с чертой» (представляющие собой вариации метода обратного распространения, которые в некоторых случаях работают быстрее).

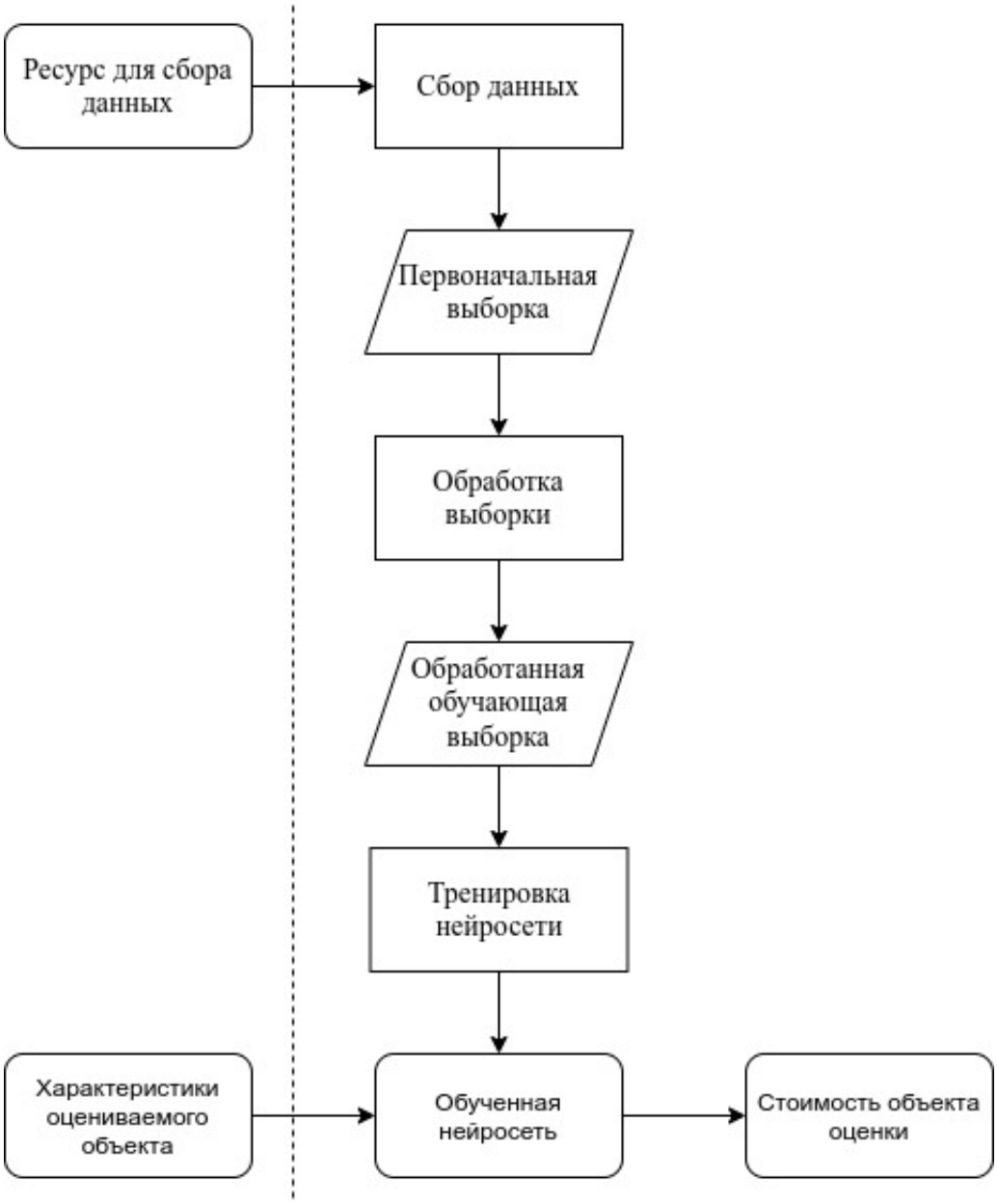

Рисунок 3.10 – Схема использования нейронных сетей для оценки стоимости недвижимости

Алгоритм обратного распространения ошибки является популярным алгоритмом обучения нейронных сетей с учителем. В основе идеи алгоритма лежит использование выходной ошибки нейронной сети для вычисления величин коррекции весов нейронов в скрытых слоях

$$E = \frac{1}{2} \sum_{i=1}^{k} (y - y')^2 \tag{3.1}$$

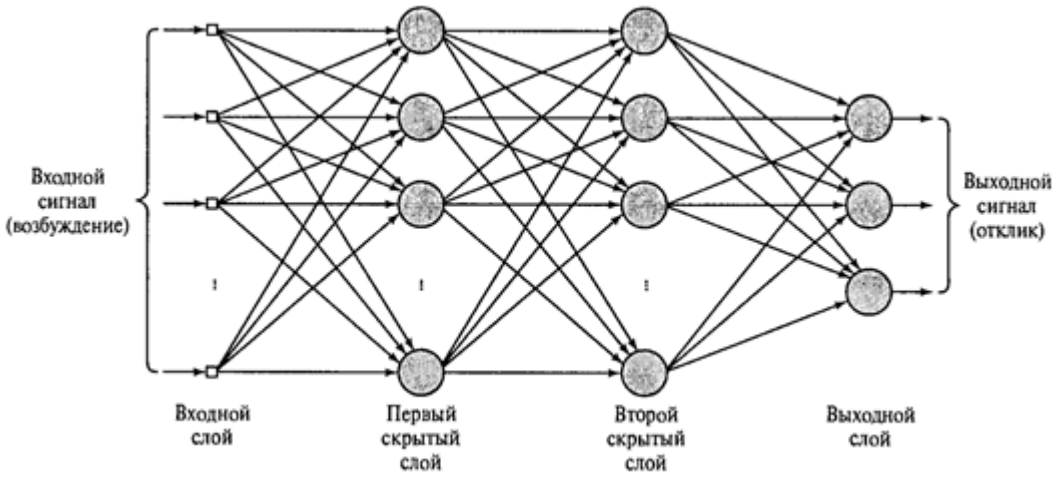

Рисунок 3.11 – Схема многослойного персептрона

где k — число выходных нейронов сети,
y — целевое значение,
y' — фактическое выходное значение.

Алгоритм является итеративным. На каждой итерации происходит прямой и обратный проходы. На прямом проходе входной вектор распространяется входов сети к ее выходам, в результате формируется выходной вектор, который соответствует фактическому состоянию весов. После вычисляется ошибка нейронной сети как разность между фактическим и целевым значениями. На обратном проходе эта ошибка распространяется от выхода сети к ее входам, и производится коррекция весов нейронов. Полученные данные дают возможность с достаточной точностью прогнозировать стоимость объектов недвижимости по заданным параметрам.

Для обучения нейронных сетей были также выбраны 3 модели с различными параметрами в качетсве входных данных. Сравнительная характеристика полученных нейронных сетей показана в таблице 3.3

Исходя из сравнения полученных нейронных сетей можно сделать вывод, что наилучшие результаты показывают отдельные модели по району. Однако результаты варьируются от района к району в зависимости от количества продаваемых объектов в данном районе, поэтому для отдельных районов целесообразнее использовать модель по районам, для остальных районов - общую модель.

Реализованные алгоритмы обучения представлены на рисунке 3.12. Кроме того, можно заметить, что в зависимости от алгоритма обучения и выбранной функции активации изменялась и конфигурация многослойного персептрона. Так, самой эффективной оказалась конфигурация с 8 нейроном

Таблица 3.3 – Сравнительная характеристика построенных моделей

Модель	Training performance	Test performance	Validation performance
Множественная модель	0.91	0.91	0.91
Отдельные модели по числу комнат	0.91	0.91	0.87
Отдельные модели по району	0.92	0.95	0.93

на первом скрытом слое, на втором – 5. На обучение отводилось 4000 объектов из обучающей выборки, что составляет 70% от общего числа объектов, на контроль и тестирование по 15% - по 600 объектов.

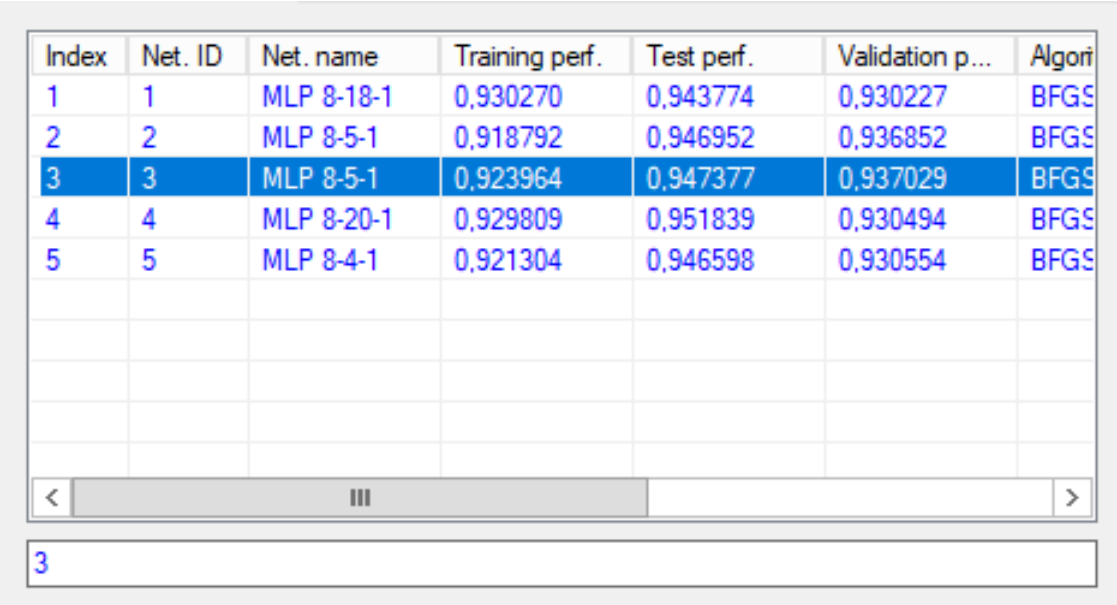

Рисунок 3.12 – Параметры качества реализованных алгоритмов обучения

Сравнение результатов ожидаемых с полученными представлено на рисунке 3.13. В первом столбце указана ожидаемая стоимость, во втором - полученная, в третьем - величина ошибки. Как можно заметить ошибка варьируется и может составлять как и небольшие значения так и значительные. Соотношение ожидаемых результатов с полученными показаны в виде графика на рисунке 3.14.

name	Target	3. MLP 8-5-1	3. MLP 8-5-1
248	37030,0	37052,4	22,38
959	60580,0	60609,2	29,20
518	40611,0	40569,1	41,93
813	80000,0	80054,9	54,92
245	37000,0	36941,3	58,66
688	55500,0	55437,9	62,15
741	59900,0	59963,3	63,34
640	51300,0	51235,6	64,42
983	64300,0	64366,6	66,61
593	47386,0	47315,9	70,11
168	32645,0	32743,3	98,26
268	37254,0	37367,0	112,98
1029	67960,0	68096,6	136,63
1093	80000,0	80154,4	154,35
694	55979,0	55818,5	160,45
623	50605,0	50776,9	171,94
646	51400,0	51210,7	189,27
1015	67249,0	67439,4	190,42
128	31390,0	31184,9	205,14
27	27768,0	27980,6	212,58
1094	80000,0	80224,9	224,89
296	38500,0	38728,4	228,38
230	36726,0	36969,7	243,70
246	37000,0	37248,5	248,46
313	39400,0	39148,5	251,46
648	51400,0	51143,4	256,62
46	29340,0	29072,3	267,69
556	44711,0	44427,1	283,87
629	51010,0	50711,9	298,13
252	37065,0	37375,4	310,35
940	59180,0	59492,1	312,07
613	50250,0	50562,4	312,36
550	44223,0	44542,5	319,49
30	28256,0	28578,3	322,26
125	31390,0	31059,1	330,93
270	37290,0	36952,1	337,92
1085	81200,0	81548,5	348,47
1086	81200,0	81548,5	348,47
665	53320,0	52938,6	381,40
126	31390,0	31007,2	382,81
243	36979,0	37375,4	396,35
129	31390,0	30989,5	400,50
703	56681,0	56273,5	407,50
523	40928,0	40514,3	413,73
188	34240,0	34663,4	423,41

Рисунок 3.13 – Сравнение полученных результатов с ожидаемыми

3.5 Сравнение результатов

На основании проведенных исследований можно утверждать, что применение нейронных сетей для прогнозирования стоимости объектов недвижимости более эффективно использования методов регрессионного анали-

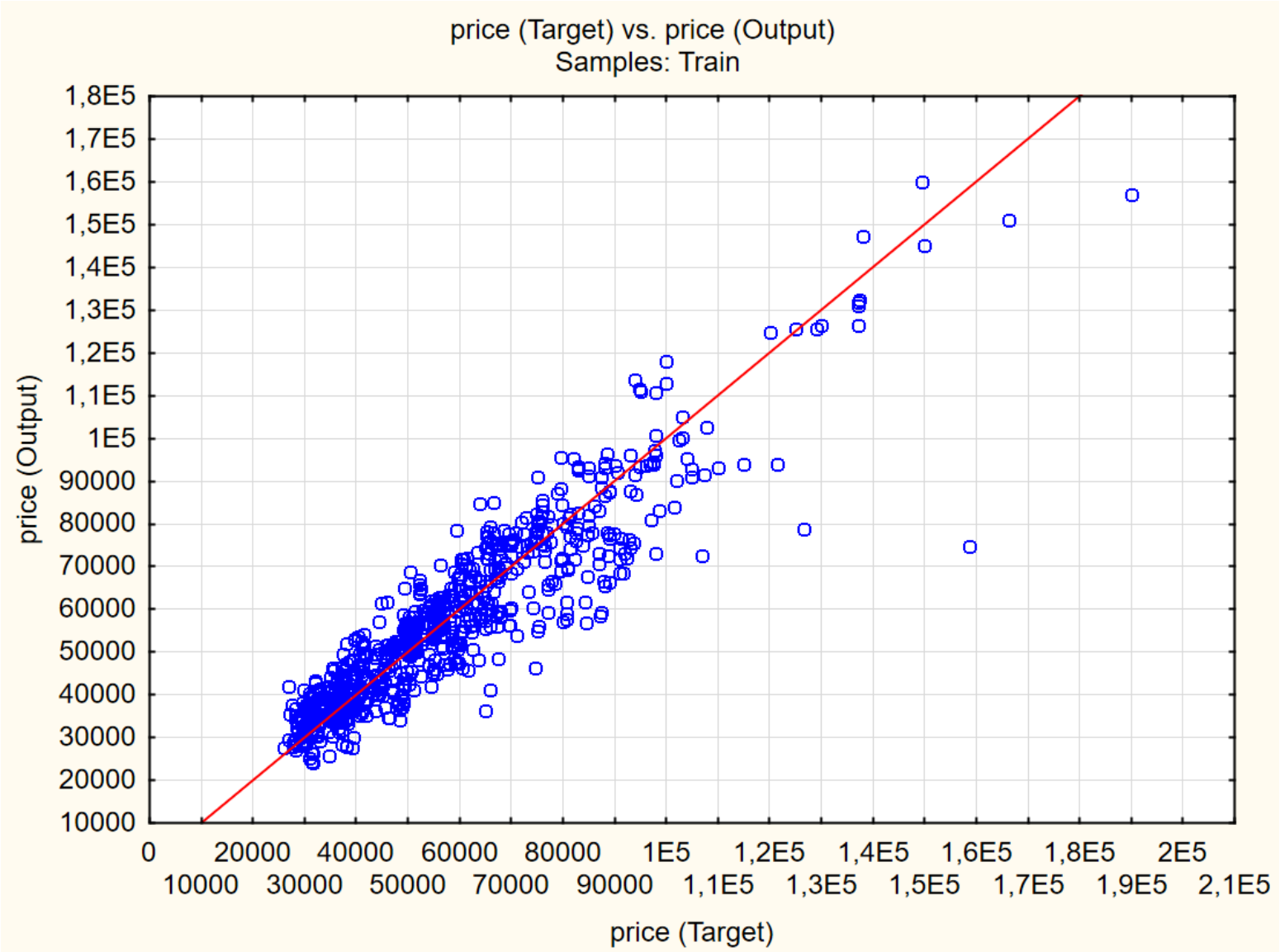

Рисунок 3.14 – Соотношение полученных результатов с ожидаемыми

за и может достаточно точно отражать рыночную стоимость недвижимости. Предложенные методы могут быть использованы продавцами для первичной оценки стоимости жилой недвижимости, а покупателями могут использоваться в качестве дополнительного источника информации, однако следует отметить, что ни одна из моделей не показывает точность свыше 95%.

ЗАКЛЮЧЕНИЕ

В ходе работы над магистерской диссертацией были изучены и проанализированы существующие способы и алгоритмы анализа данных. Были собраны данные о продающихся объектах недвижимости на рынке г. Минска на момент конца 2020 года. Всего было собрано данных о более чем 8 тысяч проданных объектов недвижимости. После был выполнен регрессионный анализ полученных данных и анализ с использованием нейронных сетей. Было проведено сравнение полученных результатов и сделаны соответствующие выводы об их эффективности и возможности реального использования.

На основании проведенных исследований можно утверждать, что применение нейронных сетей для прогнозирования стоимости объектов недвижимости более эффективно использования методов регрессионного анализа и может достаточно точно отражать рыночную стоимость недвижимости. Исходя из сравнения полученных результатов нейронных сетей можно сделать вывод, что наилучшие результаты показывают отдельные модели по району. Однако результаты варьируются от района к району в зависимости от количества продаваемых объектов в данном районе, поэтому для отдельных районов целесообразнее использовать модель по районам, для остальных районов - общую модель. Предложенные методы могут быть использованы продавцами для первичной оценки стоимости жилой недвижимости, а покупателями могут использоваться в качестве дополнительного источника информации. Данные модели можно улучшить путем сбора данных о проданных объектах недвижимости, а не только продаваемых в текущий момент времени, это увеличит объем выборки и повысит качество получаемых результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Специфика ценообразования на рынке жилья и факторы, влияющие на цену недвижимости [Электронный ресурс]. — Электронные данные. — Режим доступа: https://auditfin.com/fin/2009/2/Rodionova/Rodionova.pdf. — Дата доступа: 04.10.2020.

[2] Метод наименьших квадратов [Электронный ресурс]. — Электронные данные. — Режим доступа: https://wiki.loginom.ru/articles/least-squares-method.html. — Дата доступа: 04.10.2020.

[3] Коэффициент детерминации [Электронный ресурс]. — Электронные данные. — Режим доступа: https://wiki.loginom.ru/articles/coefficient-of-determination.html. — Дата доступа: 04.10.2020.

[4] Аппроксимация [Электронный ресурс]. — Электронные данные. — Режим доступа: https://wiki.loginom.ru/articles/approximation.html. — Дата доступа: 04.10.2020.

[5] Регрессионный анализ [Электронный ресурс]. — Электронные данные. — Режим доступа: https://wiki.loginom.ru/articles/regression-analysis.html. — Дата доступа: 04.10.2020.

[6] Нейорнные сети [Электронный ресурс]. — Электронные данные. — Режим доступа: https://habr.com/ru/post/312450/. — Дата доступа: 04.10.2020.

[7] Метод обратного распространения ошибки [Электронный ресурс]. — Электронные данные. — Режим доступа: https://otus.ru/nest/post/1592/. — Дата доступа: 04.10.2020.

[8] Карачун, С. В. Использование регрессионного анализа для оценки стоимости жилья / С. В. Карачун // Сборник работ 68-й научной научной конференции студентов и аспирантов БГУ, Минск. — 2011. — 5–8 Р.

[9] Е. А. Арефьева, Д . С. Костяев. Использование нейронных сетей для оценки рыночной стоимости недвижимости / Д . С. Костяев Е. А. Арефьева // Известия ТулГУ. — 2017. — no. 10. — 177–185 Р.

[10] Язык программирования R [Электронный ресурс]. — Электронные данные. — Режим доступа: https://www.r-project.org. — Дата доступа: 04.10.2020.

[11] Многослойный персептрон [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.aiportal.ru/articles/neuralnetworks/multi-perceptron.html. — Дата доступа: 04.10.2020.

[12] Алгоритм обучения RProp [Электронный ресурс]. — Электронные данные. — Режим доступа: https://basegroup.ru/community/articles/rprop. — Дата доступа: 04.10.2020.

Список публикаций соискателя

[1] Д. В. Голубко. Анализ цен на рынке недвижимости / Д. В. Голубко // IX Республиканская научно-практическая конференция «Вычислительные методы, модели и образовательные технологии»: сборник материалов. – Брест: БрГУ, 2020.

[2] Д. В. Голубко. Анализ цен на рынке недвижимости / Д. В. Голубко // Студенческий форум: электрон. научн. журн. 2021. № 2(138). URL: https://nauchforum.ru/journal/stud/138/84832.

ПРИЛОЖЕНИЕ А
Исходный код парсера и анализатора данных

```
# parser.rb

require 'watir'
require 'webdrivers'
require 'csv'
require 'pry'

class Parser

  def self.get_advertisements(out_file)
    start_time = Time.now
    browser = Watir::Browser.new :chrome, headless: true
    page = 0
    link = "https://realt.by/sale/flats/?search=eJwdijEOgCAQwF5zzh7mEAYHkX8QIBCNiiZi%2
        FL7AOnZovr5kkjODkDq2vUiX6osmOD3IpVKJRgOaQWDruTNPyO9d5rXOjiFKKy1xJOb94MnZIAKPIOUmBf8BzQgbzQ
        %3D%3D&page=#{page}"
    browser.goto link
    csv = CSV.read(out_file, :headers=>true)
    existing_ads = csv['ID']
    page_count = browser.elements(css: '.paging-list a')[-1].text.to_i
    CSV.open(out_file, 'a') do |writer|
      (0..page_count-1).each do |page_number|
        puts "Page Number: #{page_number}"
        link = "https://realt.by/sale/flats/?search=eJwdijEOgCAQwF5zzh7mEAYHkX8QIBCNiiZi%2
            FL7AOnZovr5kkjODkDq2vUiX6osmOD3IpVKJRgOaQWDruTNPyO9d5rXOjiFKKy1xJOb94MnZIAKPIOUmBf8BzQg
            %3D%3D&page=#{page_number}"
        browser.goto link
        browser.elements(css: '.listing-item', data_mode: '3').each do |ad|
          ad_id = ad.a(css: '.teaser-title').href.split('/')[-1]
          if !existing_ads.include?(ad_id)
            br2 = Watir::Browser.new :chrome, headless: true
            puts ad.a(css: '.teaser-title').href
            br2.goto ad.a(css: '.teaser-title').href
            price = br2.element(css: '.price-block').element(css: '.d-flex.align-items-
                center.fs-giant').text.gsub(" USD", '').gsub(' ', '').to_i
            next unless br2.element(css: '.table-params').elements(css: '.color-graydark')
                .select{|a| a.text == 'Район города'}.first
            district = br2.element(css: '.table-params').elements(css: '.color-graydark').
                select{|a| a.text == 'Район города'}.first.parent.children[1].children.
                last.text.gsub('"', '').to_s
            rooms_count = br2.elements(css: '.table-params').last.elements(css: '.color-
                graydark').select{|a| a.text == 'Комнат всегоразд/.'}.first.parent.
                children[1].text.split(' / ').first
            if br2.elements(css: '.table-params').last.elements(css: '.color-graydark').
                select{|a| a.text == 'Этаж / этажность'}.first
              floor = br2.elements(css: '.table-params').last.elements(css: '.color-
                  graydark').select{|a| a.text == 'Этаж / этажность'}.first.parent.children
                  [1].text.split(' / ').first
            else
              floor = 1
            end
```

```
        if br2.elements(css: '.table-params').last.elements(css: '.color-graydark').
            select{|a| a.text == 'Тип дома'}.first
          house_type = br2.elements(css: '.table-params').last.elements(css: '.color-
              graydark').select{|a| a.text == 'Тип дома'}.first.parent.children[1].text
        else
          house_type = 'панельный'
        end

        area_full = br2.elements(css: '.table-params').last.elements(css: '.color-
            graydark').select{|a| a.text == 'Площадь общаяжилаякухня//'}.first.parent.
            children[1].text.split(' / ').first
        area_living = br2.elements(css: '.table-params').last.elements(css: '.color-
            graydark').select{|a| a.text == 'Площадь общаяжилаякухня//'}.first.parent.
            children[1].text.split(' / ')[1]

        if br2.elements(css: '.table-params').last.elements(css: '.color-graydark').
            select{|a| a.text == 'Год постройки'}.first
          build_year = br2.elements(css: '.table-params').last.elements(css: '.color-
              graydark').select{|a| a.text == 'Год постройки'}.first.parent.children
              [1].text
        else
          build_year = '2020'
        end

        if br2.elements(css: '.table-params').last.elements(css: '.color-graydark').
            select{|a| a.text == 'Санузел/'}.first
          bathroom_type = br2.elements(css: '.table-params').last.elements(css: '.
              color-graydark').select{|a| a.text == 'Санузел/'}.first.parent.children
              [1].text
        else
          bathroom_type = 'раздельный'
        end

        if br2.elements(css: '.table-params').last.elements(css: '.color-graydark').
            select{|a| a.text == 'Ремонт'}.first
          remont = br2.elements(css: '.table-params').last.elements(css: '.color-
              graydark').select{|a| a.text == 'Ремонт'}.first.parent.children[1].text
        else
          remont = 'без отделки'
        end

        writer << [ad_id, price, district, rooms_count, floor, house_type, area_full,
            area_living, build_year, bathroom_type, remont]
        br2.close
      else
        puts "#{ad_id} is already in CSV"
      end
    end
    end_time = Time.now
    puts(convert_time(start_time, end_time))
  end
 end
 browser.close
end
```

```
  def self.convert_time(start_time, end_time)
    difference = end_time - start_time
    seconds = difference % 60
    difference = (difference - seconds) / 60
    minutes = difference % 60
    difference = (difference - minutes) / 60
    hours = difference % 24
    "Parsed page time: #{hours.to_i}:#{minutes.to_i}:#{seconds.to_i}"
  end

end

start_time = Time.now
puts "Start time: #{start_time}"
# CSV.open('properties.csv', 'a') do |csv|
# csv << ['ID', 'price', 'district', 'rooms_count', 'floor', 'house_type', 'area_full', '
    area_living', 'build_year', 'bathroom_type', 'remont']
# end
advertisements = Parser.get_advertisements('properties.csv')
end_time = Time.now
puts end_time
puts(Parser.convert_time(start_time, end_time))

# filter.rb

require 'csv'
require 'pry'

def median(array)
  sorted_array = array.sort
  count = sorted_array.count

  if sorted_array.count % 2 == 0
    first_half = (sorted_array[0...(count/2)])
    second_half = (sorted_array[(count/2)..-1])

    first_median = first_half[-1]
    second_median = second_half[0]

    true_median = ((first_median + second_median).to_f / 2.to_f)
    true_median
  else
    true_median = sorted_array[(count/2).floor]
    true_median
  end

  return true_median
end

properties = CSV.read('properties.csv')
headers = properties.shift
properties.delete_if{|p| p[1].to_i < 10000 || p[1].to_i > 400_000}

properties.map! do |p|
```

```
  p[4] = p[4].to_i
  p
end

properties.delete_if{|p| p[3].include? 'доли'}
properties.delete_if{|p| p[3].include? 'доля'}
properties.delete_if{|p| p[3].include? 'комната'}

properties.map! do |p|
  if p[3].to_s.include? 'Фактически'
    p[3] = p[3].scanФактически(/ \d/).first.gsub('Фактически ', '').to_i
  end
  p
end

properties.map! do |p|
  if p[3].to_s.include? 'Свободная'
    p[3] = p[3].scanСвободная(/ планировка\(\d/).first.gsub('Свободная планировка(', '').
        to_i
  end
  p
end

properties.map! do |p|
  p[3] = p[3].to_i
  p
end

properties.delete_if{|p| p[3] > 5}

properties.delete_if{|p| p[9] == '3 санузла.'}
properties.delete_if{|p| p[9] == '4 санузла.'}

headers.insert(10, 'bathroom_type_id')
properties.map! do |p|
  case p[9]
  when 'совмещенный'
    val = 1
  when 'раздельный'
    val = 2
  when '2 санузла.'
    val = 3
  end
  p.insert(10, val)
  p
end

properties.delete_if{|p| p[5] == 'кар'}
properties.delete_if{|p| p[5] == 'бревенчатый'}
properties.delete_if{|p| p[5] == 'блоккомнаты-'}
properties.delete_if{|p| p[5] == 'мк'}
properties.delete_if{|p| p[5] == 'силикатные блоки'}

headers.insert(6, 'house_type_id')
```

```
group = properties.group_by{|p| p[5]}
group.map do |type|
  median = median(type[1].map{|pr| pr[1].to_i})
  puts "#{type[0]} - #{median}"
end

properties.map! do |p|
  case p[5]
  when 'панельный'
    val = 1
  when 'монолитный'
    val = 2
  when 'каркасноблочный-'
    val = 3
  when 'кирпичный'
    val = 4
  end
  p.insert(6, val)
  p
end

headers.push('remont_type_id')
properties.delete_if{|p| p[12] == 'аварийное состояние'}
properties.delete_if{|p| p[12] == 'плохое состояние'}

group = properties.group_by{|p| p[12]}
group.map do |type|
  median = median(type[1].map{|pr| pr[1].to_i})
  puts "#{type[0]} - #{median}"
end

properties.map! do |p|
  case p[12]
  when 'удовлетворительный ремонт'
    val = 1
  when 'нормальный ремонт'
    val = 2
  when 'строительная отделка'
    val = 3
  when 'без отделки'
    val = 4
  when 'хороший ремонт'
    val = 5
  when 'отличный ремонт'
    val = 6
  when 'евроремонт'
    val = 7
  end
  p.push(val)
  p
end

group = properties.group_by{|p| p[2]}
districts_to_exclude = group.map{|type| {type: type[0], count: type[1].count}}.select{|
    type| type[:count] < 50}.map{|type| type[:type]}
```

```
properties.delete_if{|p| districts_to_exclude.include?(p[2])}

group = properties.group_by{|p| p[2]}
sorted_districts = group.map do |type|
  median = median(type[1].map{|pr| pr[1].to_i})
  puts "#{type[0]} - #{median}"
  {type: type[0], median: median}
end.sort_by{|type| type[:median]}.map{|type| type[:type]}

headers.insert(3, 'district_id')
properties.map! do |p|
  p.insert(3, sorted_districts.index(p[2])+1)
end

properties.delete_if{|pr| pr[8].to_f < 5 || pr[9].to_f < 5}

properties.map! do |p|
  p[5] = p[5].to_f
  p
end


puts properties.count
CSV.open('filtered_properties.csv', 'w+') do |writer|
  writer << headers
  properties.each{|p| writer << p}
end

districts = properties.map{|p| p[2]}.uniq
districts.each_with_index do |d, index|
  pr = properties.select{|p| p[2] == d}
  CSV.open("./districts/filtered_properties_#{index}.csv", 'w+') do |writer|
    writer << headers
    pr.each{|p| writer << p}
  end
end

room_1_properties = properties.select{|pr| pr[4].to_i == 1}
room_2_properties = properties.select{|pr| pr[4].to_i == 2}
room_3_properties = properties.select{|pr| pr[4].to_i == 3}
room_4_properties = properties.select{|pr| pr[4].to_i == 4}

CSV.open('1_room_properties.csv', 'w+') do |writer|
  writer << headers
  room_1_properties.each{|p| writer << p}
end

CSV.open('2_room_properties.csv', 'w+') do |writer|
  writer << headers
  room_2_properties.each{|p| writer << p}
end

CSV.open('3_room_properties.csv', 'w+') do |writer|
  writer << headers
```

```
  room_3_properties.each{|p| writer << p}
end

CSV.open('4_room_properties.csv', 'w+') do |writer|
  writer << headers
  room_4_properties.each{|p| writer << p}
end

room_1_properties_intervals_count = 1 + Math.log(room_1_properties.count, 2).truncate
interval_step = ((room_1_properties.map{|p| p[8].to_f}.max - room_1_properties.map{|p| p
    [8].to_f}.min)/room_1_properties_intervals_count).round

min_interval_value = room_1_properties.map{|p| p[8].to_f}.min
max_interval_value = room_1_properties.map{|p| p[8].to_f}.min + interval_step

room_1_properties_intervals_count.times do |index|
  properties = room_1_properties.select{|p| p[8].to_f >= min_interval_value && p[8].to_f
      <= max_interval_value}
  CSV.open("./prices/1_room_properties_#{index}.csv", 'w+') do |writer|
    writer << headers
    properties.each{|p| writer << p}
  end
  min_interval_value += interval_step
  max_interval_value += interval_step
end

#anylizer.r

setwd("/home/dmitry/Documents/regression/parser")
data_f = read.csv("filtered_properties.csv", header = TRUE)

data_f
h <- hist(data_f$area_full, main="Histogram for Full area", xlab="Full Area, sqm", col="
    green", ylim=c(0,3000))
text(h$mids,h$counts,labels=h$counts, adj=c(0.5, -0.5))

## bathrooms
mytable <- table(data_f$bathroom_type)
b <- barplot(mytable,
       main = "Types of bathrooms",
       ylab = "Bathrooms",
       names.arg = names(mytable),
       ylim=c(0,4500)
       )
text(b, mytable+100, mytable, font=2)


##bedrooms
mytable <- table(data_f$rooms_count)
b2 <- barplot(mytable,
            main = "Number of rooms",
            ylab = "Rooms",
            names.arg = names(mytable),
            ylim=c(0,3000), col="green"
)
```

```
text(b2, mytable+100, mytable, font=2)


##price
h <- hist(data_f$price, main="Histogram for Sold Price", xlab="Sold Price, $", col="green
    ", ylim=c(0,2000))
text(h$mids,h$counts,labels=h$counts, adj=c(0.5, -0.5))

## linear model
plot(price ~ area_full, data = data_f, col = "blue")
reg1 = lm(price ~ area_full, data=data_f)
reg1
summary(reg1)
abline(reg1, col="red")

plot(sold_price ~ bedrooms, data = data_f, col = "blue")
reg2 = lm(sold_price ~ bedrooms, data=data_f)
reg2
summary(reg2)
abline(reg2, col="red")


## miltiple model

regt = lm(price ~ district_id + rooms_count + floor + house_type_id + area_full +
    area_living + build_year + bathroom_type_id + remont_type_id, data = data_f)
summary(regt)

regt = lm(price ~ district_id + rooms_count + floor + house_type_id + area_full +
    area_living + build_year + remont_type_id, data = data_f)
summary(regt)

modified = data_f[, -c(1, 3, 7, 12, 14)]
M <- cor(modified)
library(corrplot)
corrplot(M, type = "upper", method = "number")


## multiple model rooms == 1
new_data_f = read.csv("1_room_properties.csv", header = TRUE)
regt = lm(price ~ district_id + floor + house_type_id + area_full + area_living +
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)

regt = lm(price ~ district_id + area_full + area_living + build_year + bathroom_type_id +
     remont_type_id, data = new_data_f)
summary(regt)

modified = new_data_f[, -c(1, 3, 5, 7, 12, 14)]
M <- cor(modified)
corrplot(M, type = "upper", method = "number")

## multiple model bedrooms == 2
new_data_f = read.csv("2_room_properties.csv", header = TRUE)
regt = lm(price ~ district_id + floor + house_type_id + area_full + area_living +
```

```
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)


## multiple model bedrooms == 3
new_data_f = read.csv("3_room_properties.csv", header = TRUE)
regt = lm(price ~ district_id + floor + house_type_id + area_full + area_living +
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)

## multiple model bedrooms == 4
new_data_f = read.csv("4_room_properties.csv", header = TRUE)
regt = lm(price ~ district_id + floor + house_type_id + area_full + area_living +
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)

## multiple model rooms == 1, with interval
new_data_f = read.csv("1_room_properties_1.csv", header = TRUE)
regt = lm(price ~ district_id + floor + house_type_id + area_full + area_living +
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)

regt = lm(price ~ district_id + area_full + bathroom_type_id + remont_type_id, data =
    new_data_f)
summary(regt)


## multiple model with district
setwd("/home/dmitry/Documents/regression/parser/districts")
new_data_f = read.csv("filtered_properties_9.csv", header = TRUE)
regt = lm(price ~ rooms_count + floor + house_type_id + area_full + area_living +
    build_year + bathroom_type_id + remont_type_id, data = new_data_f)
summary(regt)


modified = new_data_f[, -c(1, 3, 5, 7, 12, 14)]
M <- cor(modified)
corrplot(M, type = "upper", method = "number")

# predict
predicted_vals = predict(regt, new_data_f)
plot(predicted_vals, new_data_f$price,
     xlab="predicted",ylab="actual")
abline(a=0,b=1)

predicted_vals
new_data_f$price

precision(new_data_f$price, predicted_vals)
```