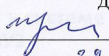


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой
д.п.н., профессор

 И.Г. Захарова

29 июля 2018г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

РАЗРАБОТКА СИСТЕМЫ ДЛЯ ОЦЕНКИ ЛИКВИДНОСТИ
НЕДВИЖИМОСТИ (КВАРТИР) В Г.ТЮМЕНИ С ПОМОЩЬЮ МЕТОДОВ
МАШИННОГО ОБУЧЕНИЯ

02.04.03.Математическое обеспечение и администрирование информационных
систем

Магистерская программа «Высокопроизводительные вычислительные
системы»

Выполнил работу
Студент 2 курса
очной формы обучения


(Подпись)

Глушак
Виктор
Юрьевич

Научный руководитель
к.п.н, доцент


(Подпись)

Плотоненко
Юрий
Анатольевич

Рецензент
к.т.н. Старший преподаватель
Кафедра программного обеспечения
ИМиКН ТюмГУ


(Подпись)

Глазкова
Анна
Валерьевна

Оглавление

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	4
§ 1.1. Понятие ликвидности.	4
§ 1.2. Обзор существующих решений.	4
ГЛАВА 2. ОПИСАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ	7
§ 2.1. Постановка задачи.....	7
§ 2.2. Требования к web-приложению	7
§ 2.3. Архитектура приложения	11
§ 2.4. Описание модели расчета стоимости квартиры.....	12
§ 2.5. Описание корректировки параметров	17
§ 2.6. Расчет цены квартиры после корректировки.....	21
ГЛАВА 3. ОПИСАНИЕ РАСЧЕТА ЛИКВИДНОСТИ	22
§ 3.1. Входные данные.	22
§ 3.2. Загрузка и очистка данных.....	22
§ 3.3. Расчет времени экспозиции и анализ результата	24
§ 3.4. Категоризация ликвидности.....	27
§ 3.5. Ликвидность по районам.	29
§ 3.6. Поиск закономерности проданных объектов методом Случайный лес.....	32
§ 3.7. Поиск категории ликвидности объектов.....	35
§ 3.8. Предсказание срока экспозиции объекта.....	38
§ 3.9. Библиотека Scikit-Learn	43
ЗАКЛЮЧЕНИЕ	45
СПИСОК ЛИТЕРАТУРЫ И ИСТОЧНИКОВ	46
ПРИЛОЖЕНИЕ №1	47
ПРИЛОЖЕНИЕ №2	48
ПРИЛОЖЕНИЕ №3	49
ПРИЛОЖЕНИЕ №4	50

ПРИЛОЖЕНИЕ №5	51
ПРИЛОЖЕНИЕ №6	53

ВВЕДЕНИЕ

Правильное предсказание ликвидности могло бы дать ценную информацию для продавцов и покупателей недвижимости и обеспечить информационную прозрачность между продавцами и покупателями. Например, продавец знал бы, сколько времени займет продажа квартиры по определенной цене, используя систему по прогнозированию ликвидности. Следовательно, продавец мог бы скорректировать цену, чтобы повлиять на ожидаемое время продажи. Если продавец будет знать какие признаки влияют на ликвидность ты выставив адекватную цену смог бы быстрее её продать. С другой стороны знания об этих характеристиках могут уберечь покупателя от необдуманной покупки завышенную по цене квартиру и позволит правильно оценить параметры недвижимости и защититься от манипуляций. Это в итоге может привести к развитию индустрии недвижимости в целом.

Особенности этой задачи в том, что выборка очень неоднородная. Объекты недвижимости, которые оценивались в разные моменты времени, могли сильно зависит от текущей ситуации на рынке недвижимости, и меняться во времени. Кроме того, можно использовать линейную модель, потому что, такие признаки, как наличие балкона или жилая площадь, или число комнат — каждая, дает какой-то аддитивный вклад в стоимость квартиры, то есть чем больше площадь, тем больше будет стоимость квартиры. Но тем не менее есть такие признаки, у которых явно этот параметр не дает линейный вклад в стоимость квартиры. Например расстояние до остановки. Чем оно больше, тем, ниже стоимость квартиры, но нужно учесть, насколько понижается стоимость квартиры с каждым километром от развязки.

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

§ 1.1. Понятие ликвидности.

Ликвидность — это характеристика активов, определяющая, насколько быстро они могут быть проданы по близкой к рыночной цене. Чем меньше времени требуется для продажи, тем выше ликвидность.

§ 1.2. Обзор существующих решений.

Попытка найти ликвидность объектов недвижимости через количество дней с момента размещения до продажи, описана в статье *Days on Market: Measuring Liquidity in Real Estate Markets* [1], китайских исследователей.

Статья посвящена *Days on Market (DOM)* - количество дней, в течение которых недвижимость находится на активном рынке до момента продажи, что является важным показателем рыночной ликвидности в сфере недвижимости. Авторы ставят своей целью разработать систему по прогнозированию этого показателя. Для этого собирают необходимые данные. В частности, авторы сначала извлекают ключевые особенности из нескольких типов гетерогенных реальных имущественных данных, таких как профили дома и гео-социальную информацию из населенных пунктов Китая. Затем, на основе этих особенностей, разрабатывают систему обучения на основе регрессии[4] и таким образом обучают модель прогнозировать показатель DOM.

Интуитивная идея для предсказания DOM является непосредственно обучение регрессионной модели с исторических записей транзакций. Чтобы убедиться в эффективности подхода автора помимо основного метода *Multi-task Linear Regression for DOM prediction (MLR-DOM)* выбрали ряд других стандартных методов для проверки: линейно регрессии (LR), Ридж (т. е. L2-регуляризатор норм), Лассо (т. е. LR с L1-регуляризатор норм), Линейная

регрессия по местоположению (LsLR), Дерево принятия решений (DT), Случайный лес (RF), Поддержка векторной регрессии (SVR).

Для оценки эффективности прогнозирования различных методов авторы выбрали две широко используемые метрики из общего регрессионного анализа, т. е. среднеквадратичную погрешность (rMSE) и средняя абсолютная ошибка (MAE), и нормализованное среднее квадратичное ошибки (nMSE).

В результате подход авторов MLR-DOM в котором они разбили выборку по районам, превосходит другие базовые методы по всем наборам данных с точки зрения всех оценочных метрик, что четко подтверждает эффективность многозадачной модели регрессии на основе обучения.

Так же среди параметров внесших наибольший вклад стоит выделить (по убываю вклада): Средний DOM недавно проданных домов рядом с текущим, площадь жилья, процент недавно проданных домов, итоговая цена.

В другой статье A Localized Model for Residential Property Valuation: Nearest Neighbor with Attribute Differences [3], говорится о недвижимости в Гонконге.

Статья посвящена получению стоимости от сделок с недвижимостью, наиболее близкой к реальной с учетом различий в структурных и вспомогательных атрибутах объектов для более точной оценки имущества. Автором предлагается метод оценки имущества, сочетающий в себе преимущества подход ближайшего соседа и измерение различий характеристик между собственностью и ее соседями, чтобы прийти к более точной оценке.

В данной работе расширен подход к моделированию стоимости жилья в первую очередь были представлены два шага: (I) использование подхода ближайшего соседа и (II) корректировка различий неполных атрибутов в пространственной зависимости. Предлагаемая авторами модель смягчает эту

неоднородность и расширяет подход ближайшего соседа в оценке жилых объектов в непосредственной близости друг от друга. Авторы предлагают простой и точный двухступенчатый инструмент.

Представленная модель учитывает параметры макроуровня, например, локационные, физические и экономические переменные, а также параметры микроуровня, например структурные и вспомогательные переменные. Эта модель решает проблему отсутствия данных при нечестных сделках благодаря аналогичным сделкам с соседним жильём и учитывает присущие друг другу незначительные различия между похожими соседями.

ГЛАВА 2. ОПИСАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ

§ 2.1. Постановка задачи

Дать возможность продавцу оценить ликвидность собственной квартиры, на рынке недвижимости исходя из ее характеристик, узнать возможную цену и вероятный срок продажи за эту цену. Для покупателя найти наиболее привлекательную цену на желаемую квартиру, узнать, насколько цена занижена или завышена в отличие от цен в целом по рынку.

Для этого разработать систему с веб интерфейсом и формой, позволяющей на вход подавать данные о квартире, на выходе получать оценку ликвидности, вероятную стоимость квартиры и срок продажи, в виде таблицы.

§ 2.2. Требования к web-приложению

Реализовать прототип системы с интерфейсом написанном на JavaScript, и Django (сервер Python) вместе с MySQL 5.7.

Система должна состоять из следующих основных модулей, отвечающих за свои направления и взаимодействующих друг с другом:

1) Модуль администратора, содержащий инструментарий по администрированию системы, ее компонентов, функций, процессов, инструментарий по добавлению новых уже проданных объектов недвижимости для обновления информации в системе и актуальности оценки ликвидности.

Включает в себя следующие подсистемы:

а. Подсистема администратора – содержащая инструментарий по администрированию ее компонентов, функций, процессов, инструментарий по добавлению новых данных об объектах недвижимости таких как (Улица, Дом, Тип квартиры, Кол-во этажей в доме, Этаж, Площадь квартиры общая,

Площадь квартиры жилая, Количество комнат, Тип дома, Серия дома, Год строительства дома и т.д.).

в. Подсистема резервного копирования и защиты от сбоев – отвечает за регулярное резервное копирование базы данных, предотвращение сбоев и оперативное восстановление работоспособности после критических сбоев.

2) Модуль пользователя, где пользователю предоставляется возможность вводить данные о своей квартире и получать результат, имеет веб-интерфейс, не требует регистрации.

3) Модуль математической модели, содержащий возможность расчета ликвидности по минимальным и расширенным параметрам. Не имеет интерфейса и работает на сервере.

Система должна быть активна 98% времени, остальное время должно быть предусмотрено на обновление системы и затрат времени на бекап базы данных. Для ее штатного обслуживания не должно требоваться больше 1 человека. При работе системы нужно стремиться к минимизации сбоев в работе, система должна содержать механизмы по автоматическому восстановлению функционирования в случае критического сбоя в работе ее компонентов.

Пользовательский интерфейс должен отвечать современным требованиям к эргономике и технической эстетике.

Внутренний пользовательский интерфейс системы должен позволять пользователю свободно ориентироваться в информационном и функциональном пространстве системы и удовлетворять следующим требованиям:

1. Однозначно понимаемое назначение названий необходимых для ввода характеристик объектов недвижимости.

2. Отсутствие англоязычных терминов в элементах интерфейса.
3. Группировка элементов интерфейса по функциональному признаку.
4. Минимизация вертикальной и горизонтальной прокрутки.
5. Четко сформулированные и понятные пользователю сообщения об ошибках.
6. Цветовая гамма, выдержанная в спокойных тонах, не раздражающих пользователя.
7. Сохранять идентичность отображения на большинстве современных ОС и Web-браузерах (Mozilla Firefox, начиная с версии 55.0 и до текущей версии; Opera, начиная с версии 9 и до текущей версии; Google Chrome начиная с версии 60 и до текущей версии).
8. Содержать исчерпывающий набор метаданных для эффективного индексирования поисковыми системами и корректного автоматического выбора кодировки.

Интерфейс пользователя:

Поля Web-формы, заполняемой пользователем (* - обязательные):

- 1) Категория жилья (эконом, комфорт, бизнес, элит)* - выбор из списка.
type select.
- 2) Количество комнат * - пансионат, общежитие, одн.квартира... type select.
- 3) Этаж* type number.
- 4) Материал стен* блочный, кирпичный, монолитный и т.д. type select.
- 5) Район* - выбор из списка. type select.
- 6) Год постройки* - type date.

7) Ремонт* - Черновая, улучшенная черновая, ремонт от строителей, требует ремонта, частичный ремонт, косметический, современный, по дизайн проекту. - type select.

8) Площадь* type number.

Расширенные настройки – по умолчанию свернуты если нажать то развернется список для более точной оценки. (^ - если нажаты «Расширенные настройки»)

10) Серия дома^ type select – Брежневка, Сталинка, Хрущевка... type select.

11) Кол-во этажей в доме* type number.

12) Железная дверь, счетчики, парковка, видеокамеры и т.д. (Прил. №6) type checkbox.

Кнопка «Рассчитать»

В результате нажатия кнопки «Рассчитать» будут выведена таблица рекомендованных цен продажи со сроком от одного месяца до года.

Должна быть кнопка «вывод на печать», позволяющая распечатать таблицу с характеристиками и расчетом. И кнопка сохранить с выбором желаемого формата pdf, *.xls.

В случае, если необходимые требования для предоставления услуги не исполнены, поля Web-формы заполнены некорректно или обязательные поля не заполнены эти поля должны подсвечиваться чтобы пользователь понял где он допустил ошибку.

При вводе данных используются справочники:

1) Районы Тюмени;

2) Серия дома исходя из выбранного типа дома;

§ 2.3. Архитектура приложения

На уровне бизнес логики (Рис.1) показаны возможные действия пользователей системы по ролям, на уровне приложения показана программная реализация, на уровне развертывания, технологии которые будут использоваться.

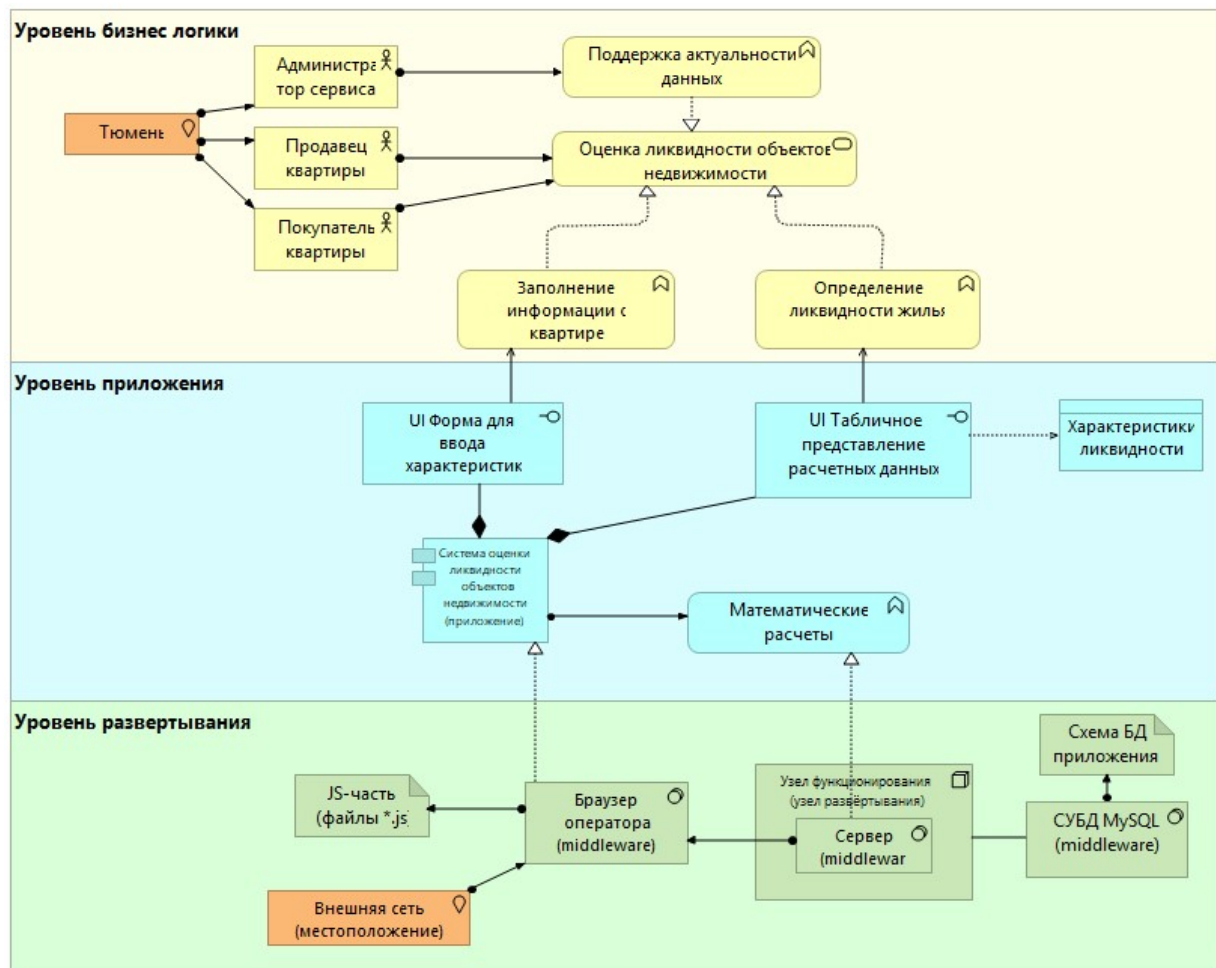


Рис.2.3.1 – Архитектура приложения выполнена в программе Archi на принципах Archimate

§ 2.4. Описание модели расчета стоимости квартиры

Напомним, что ликвидность это характеристика актива, определяющая, насколько быстро он может быть продан по близкой к рыночной цене.

Расчет стоимости ведется по методу подбора точных аналогов[5].

Ход работы алгоритма:

1) Алгоритм следует по порядку внутри каждого параметра (Таб.2.4.1).

Таблица 2.4.1

Параметры недвижимости

Номер	Параметр
1	Категория жилья
2	Количество комнат
3	Этаж
4	Материал стен
5	Район
6	Год постройки
7	Ремонт
8	Площадь
9	Серия

На каждом шаге выборка уменьшается.

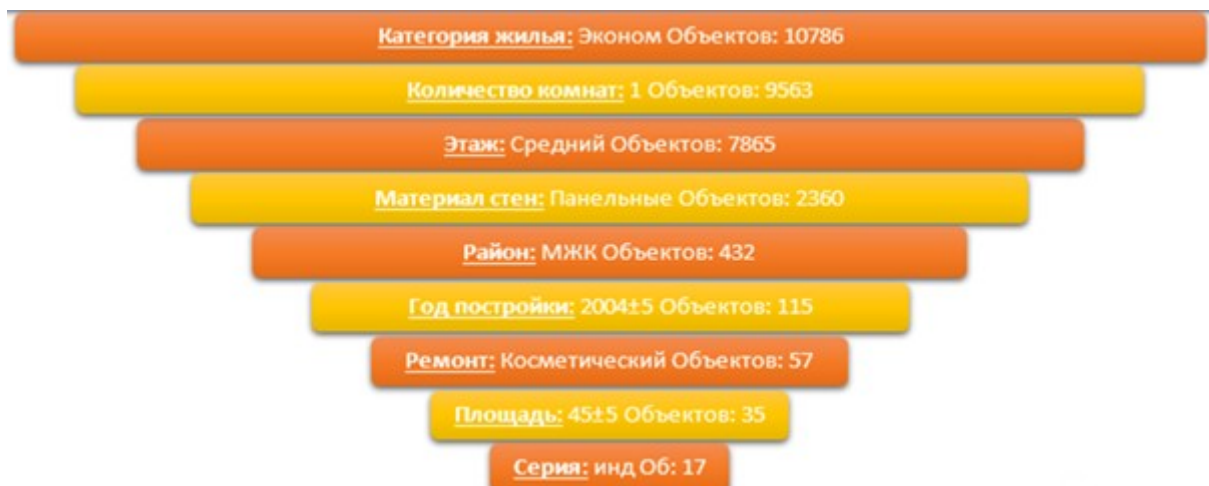


Рис.2.4.1 – Работа алгоритма подбора точных аналогов

2) Если на каком-то параметре не выполняется условие достаточности выборки (не менее 3 объектов отличных от анализируемого объекта), то шаг увеличивается на 1, и в данном параметре рассматривается уже большее объединение.

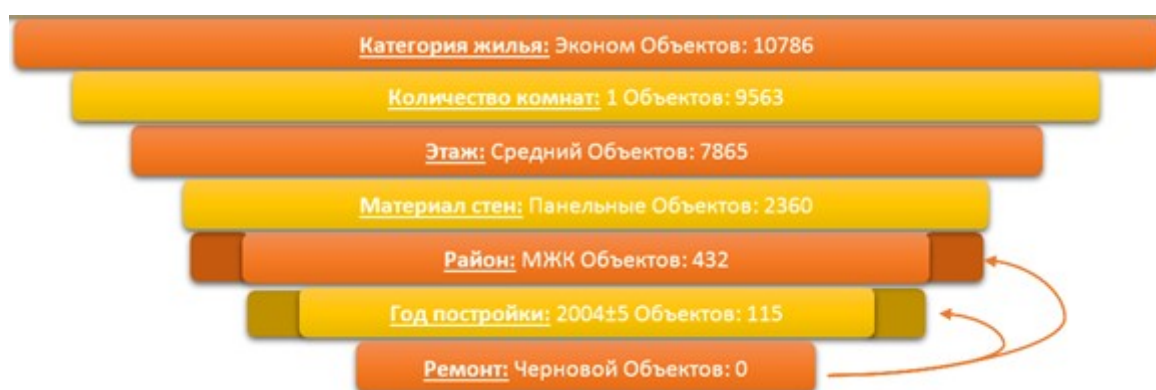


Рис.2.4.2 – Возврат на предыдущий шаг алгоритма подбора точных аналогов

3) Если снова не выполняется условие, то происходит переход к параметру выше (согласно нумерации из Таблицы 2.4.1), и уже здесь увеличивается шаг.

4) На пунктах ниже фильтрация начинается с 1-го шага и увеличивается на 1, только если условие не выполняется.

5) Шаг увеличивается только тогда, когда все пункты выше уже отфильтрованы по текущему максимальному шагу.

Пример работы алгоритма:

Таблица 2.4.2

Параметры недвижимости и шаги расширения выборки

Параметры	Шаги			
	1	2	3	4
Категория жилья	наш	наш	наш	все
Количество комнат	наш	наш	наш	все
Этаж	наш	наш	наш	все
Материал стен	наш	наш	наш	все
Район	наш	1-е объединение	2-е объединение	2-е объединение
Год постройки	1-е объединение	2-е объединение	3-е объединение	3-е объединение
Ремонт	наш	1-е объединение	2-е объединение	все
Площадь	1-е объединение	2-е объединение	2-е объединение	2-е объединение
Серия	наш	1-е объединение	2-е объединение	все

Выборка была отфильтрована по Району. Но при первом объединении по году постройки (1), объема выборки не хватило, поэтому перешли на 2-е объединение (2). Здесь также объем выборки был не достаточен. Далее алгоритм не может перейти к 3-му объединению, т.к. параметр выше (Район) еще не отфильтрован по 2-му шагу. Поэтому алгоритм возвращается на первое объединение по Районам (3). Уже здесь выборки достаточно, переходим к 1-му объединению по Году постройки (4). Здесь выборки также не достаточно, поэтому переходим ко 2-му объединению (5), такое расширение не помогает выборка остается маленькой. Т.к. все параметры выше уже отфильтрованы по 2-му шагу, то в Годе постройки мы можем перейти уже на 3-й шаг (6). Т.к. здесь выборки достаточно переходим на 1-й шаг по Ремонту (7) и идем дальше по алгоритму (8).



Рис.2.4.3 – Виды объединений

Числовые объединения:

Изначальный поиск ведется не по точному совпадению, а сразу по 1-му объединению. При фильтре по таким объединениям выборка расширяется путем вычитания/сложения числового значения из значения анализируемого объекта. Например, если площадь анализируемого объекта = 35м².

	Расширение	Площадь аналогов
1-й уровень объединений	+5 кв. м	30-40 кв. м
2-й уровень объединений	+10 кв. м	25-45 кв. м

Рис.2.4.4 – Пример объединения по площади

Нечисловые объединения:

Район анализируемого объекта	Центр: Исторический
Районы из 1-го объединения	Центр: Драмтеатр
Районы из 2-го объединения	Центр: Дом печати



Рис.2.4.5 – Объединение районов

Корректировки:

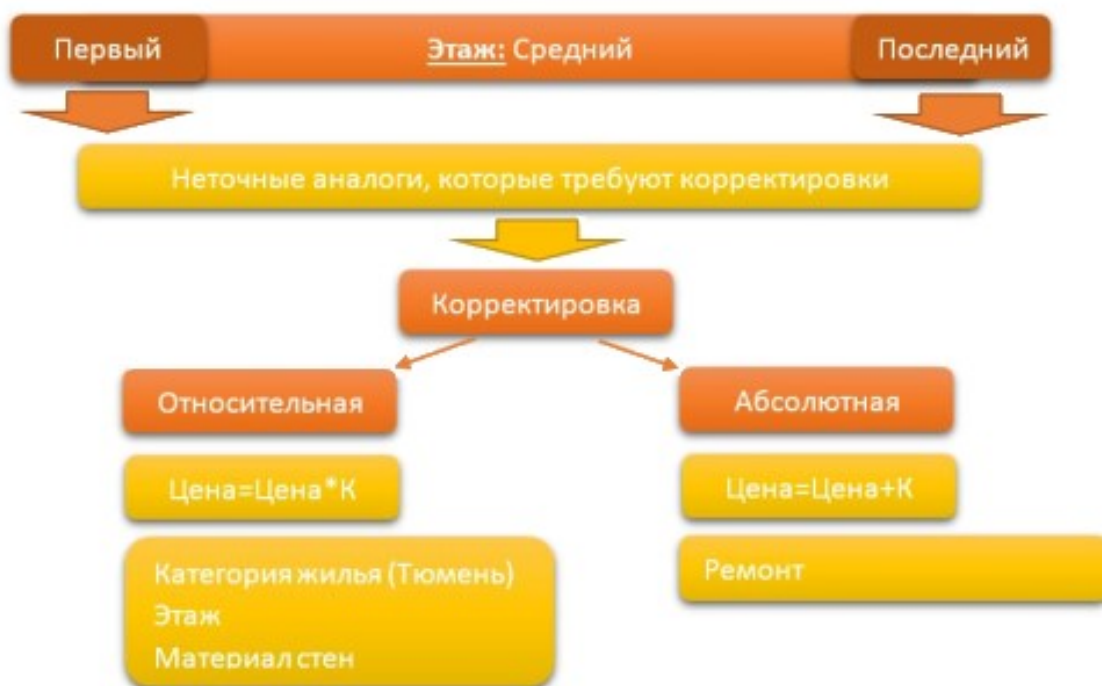


Рис.2.4.6 – Корректировка по этажу

Пример относительной корректировки:



Рис.2.4.7 – Относительная корректировка

Пример абсолютной корректировки:



Рис.2.4.8 – Абсолютная корректировка

§ 2.5. Описание корректировки параметров

Корректировка «Район»:

В выборке, попавшей на данный пункт, оставляем объекты с районом = району анализируемого объекта (если район анализируемого объекта входит в группу районов, то в выборке оставляем объекты со всеми районами из группы).

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с районом (группой районов) анализируемого объекта + районами из первого объединения из Приложения 4 .

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с районом (группой районов) анализируемого объекта + районами из первого объединения из Приложения 4 + районами из второго объединения из Приложения 4.



Рис.2.5.1 – Корректировка района

Для районов которые находятся вне Тюмени и дальше 35км от центра города (рис.2.5.2) стоимость не вычисляется кроме районов исключений.



Рис.2.5.2 – Окружность 35 км от центра Тюмени

Корректировка «Год постройки»:

В выборке, попавшей на данный пункт, оставляем объекты с годом постройки попадающим в диапазон [год анализируемого объекта - N; год анализируемого объекта + N], где N - значение для первого объединения из таблицы 2.5.1.

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с годом постройки попадающим в диапазон [год анализируемого объекта - N; год анализируемого объекта + N], где N - значение для второго объединения из таблицы 2.5.1.

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с годом постройки попадающим в диапазон [год анализируемого объекта - N; год анализируемого объекта + N], где N - значение для третьего объединения из таблицы 2.5.1.

Таблица 2.5.1

Объединение «Год постройки»

Первое объединение	Второе объединение	Третье объединение
+5 лет	+8 лет	+11 лет

Корректировка «Площадь»:

В выборке, попавшей на данный пункт, оставляем объекты с площадью попадающей в диапазон [площадь анализируемого объекта - N; площадь анализируемого объекта + N], где N - значение для первого объединения из Таблицы 2.5.2.

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с площадью попадающей в диапазон [площадь анализируемого объекта - N; площадь анализируемого объекта + N], где N - значение для второго объединения из Таблицы 2.5.2.

Объединение «Площадь»

Кол-во комнат	Первое объединение	Второе объединение
Квартиры с площадью $\leq 30 \text{ м}^2$	$\pm 3 \text{ м}^2$	$\pm 5 \text{ м}^2$
Однокомнатные	$\pm 5 \text{ м}^2$	$\pm 8 \text{ м}^2$
Двухкомнатные	$\pm 8 \text{ м}^2$	$\pm 10 \text{ м}^2$
Трехкомнатные	$\pm 10 \text{ м}^2$	$\pm 13 \text{ м}^2$
Многокомнатные	$\pm 15 \text{ м}^2$	$\pm 20 \text{ м}^2$

Корректировка «Серия»:

В выборке, попавшей на данный пункт, оставляем объекты с серией = серии анализируемого объекта.

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с серией анализируемого объекта + сериями из первого объединения из Приложения №5. Если серии анализируемого объекта в Приложении №5 нет или для нее не заполнено 1-е объединение, то в выборке оставляем объекты со всеми сериями.

При недостаточности выборки, возвращаем в выборку все объекты, отфильтрованные на данном шаге, и оставляем объекты с серией анализируемого объекта + сериями из первого объединения из Приложения №5 + сериями из второго объединения из Приложения №5. Если серии анализируемого объекта в Приложении №5 нет или для нее не заполнено 2-е объединение, то в выборке оставляем объекты со всеми сериями.

При недостаточности выборки возвращаем в выборку все объекты отфильтрованные на данном шаге.

§ 2.6. Расчет цены квартиры после корректировки

После всех необходимых корректировок можно приступить к расчету средней цены квадратного метра, стандартного отклонения и погрешности по следующей формуле:

$$X_{\text{ср.}} = \frac{X_1 + X_2 + X_3 + \dots + X_n}{n} \quad (1)$$

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - X_{\text{ср.}})^2}{n-1}} \quad (2)$$

$$\delta = \frac{2 \cdot \frac{S}{\sqrt{n-1}}}{X_{\text{ср.}}} \quad (3)$$

где, $X_1, X_2, X_3, \dots, X_n$ - удельные цены объектов, n - количество таких объектов.

Если погрешность $\delta > 15\%$, то алгоритм возвращается к последнему параметру (согласно порядку из Таблицы 2.4.2), шаг которого меньше текущего максимального. Если все параметры отфильтрованы по текущему максимальному шагу, то алгоритм возвращается на последний пункт и увеличивает на 1 максимальный шаг.

Если погрешность $\delta \leq 15\%$, то алгоритм переходит к расчету средневзвешенной цены:

ГЛАВА 3. ОПИСАНИЕ РАСЧЕТА ЛИКВИДНОСТИ

§ 3.1. Входные данные.

Изначально входные данные состоят из файла формата csv. Файл размером 1,2 Гб, содержит 370 551 строк и 308 столбцов. Строки это количество квартир которые продавались или были проданы в Тюмени начиная с 2007 года. Столбцы это необходимые для анализа признаки объектов, такие как площадь, этаж, цена продажи, район, улица и т.д.

§ 3.2. Загрузка и очистка данных.

Для загрузки и очистки данных использовались средства библиотеки pandas которая содержится в пакете Anaconda v.5.1.

```
1 import pandas as pd
2 from pandas import Series, DataFrame
3
4 data = pd.read_csv('objects.csv', sep = ",", nrows=500000,
5                   encoding='utf-8', error_bad_lines=False)
```

Рис.3.2.1 – Загрузка данных.

Так как для наших целей нет необходимости исследовать весь файл в связи с инфляцией цен на квартиры и низким качеством ранних данных, было принято решение взять объекты начиная с 2016 г. и по июнь 2017г. включительно. И которые не были удалены.

```
7 dataTyumen = data.loc[(data['date_create'] > '2015-12-31') &
8                      (data['status'] != 'deleted')]
```

Рис.3.2.2 – Отбор данных за определенный период.

Полученные данные содержат 43800 объектов. Из них проданных 4676, что составляет 10%. Далее были удалены все не представляющие интереса признаки объектов.

```

10 dataTyumen = dataTyumen.drop(columns=[
11     'about_price', 'action_description',
12     'allowed', 'apartment_letter', 'attach_cnt',
13     'banner', 'banner_1', 'banner_2', 'banner_3', 'banner_4',
14     'banner_presence', 'beds_count', 'build', 'builder_contract', 'buildf',
15     'cadaster_number', 'callstat_agents_id', 'call_status', 'change_color',
16     'children_playground', 'city_id', 'comment', 'commerce_agent',
17     'common_note', 'contract_cost', 'contract_raschets', 'coords', 'cords_side',
18     'cords_top',
19     'date_locked', 'deposit_source', 'description',
20     'direction', 'district_mk_id', 'doc', 'doc_date',
21     'enable_house',
22     'fee', 'fee_temp', 'fee_type', 'fee_type_temp', 'file_pd', 'fire_signal',
23     'gp_full', 'gp_link', 'gp_link_short', 'gp_number_new_developments_avito',
24     'gp_number_new_developments_cian', 'gp_number_new_developments_yandex',
25     'gp_short', 'guid',
26     'house_uni', 'house_uuid',
27     'installment',
28     'la', 'lo', 'landmark_id', 'last_repairs', 'layout_type', 'la_pan', 'lo',
29     'loggia', 'lo_pan',
30     'main_photo', 'map', 'max_action',
31     'name_alt', 'newhouses_contact', 'notes',
32     'old_price', 'ondeadline', 'on_map', 'own_note',
33     'parking_space_number', 'people_count', 'photo_ticket_params',
34     'place_rating', 'pond_distance', 'pond_distancef', 'pond_type',
35     'porch', 'price_changed', 'price_editor', 'probable_price',
36     'renovation', 'rise', 'rounded',
37     'set_banner', 'sold_price_ar', 'sold_without_us',
38     'tenants_info', 'timephotoallowed', 'time_photo', 'transferred_for_photo',
39     'unfinished',
40     'views',
41     'webcam', 'web_owner',
42     'young_family'])

```

Рис.3.2.3 – Удаление не нужных признаков.

Далее необходимо проверить данные на пустые значения, NaN и infinity и если необходимо очистить или заменить их.

```

44 import numpy as np
45
46 #проверяем есть ли NaN
47 dataTyumen.isnull().values.any()
48 #проверяем есть ли пустота
49 np.where(pd.isnull(dataTyumen))
50 np.where(dataTyumen.applymap(lambda x: x == ''))
51 #Проверка на ноль
52 zeroBuildingYear = dataTyumen[(dataTyumen.building_year == 0)]
53 #проверка на инфинити и другие значения
54 dataTyumen[dataTyumen.isin([np.nan, np.inf, -np.inf]).any(1)]

```

Рис.3.2.4 – Проверка и очистка данных.

§ 3.3. Расчет времени экспозиции и анализ результата.

Одним из ключевых параметров необходимых для расчета ликвидности является количество дней экспозиции – это количество дней от размещения до продажи квартиры, исходя из этих данных, далее будет рассчитана категория ликвидности объекта. Берем только все проданные объекты. Которых 4676.

```
56 dataTyumen['time_exposition'] = Series('', index=dataTyumen.index)
57
58 dataTyumen['sold_date'] = pd.to_datetime(dataTyumen['sold_date'])
59 dataTyumen['sold_date'] = dataTyumen['sold_date'].dt.date
60 dataTyumen['date_create'] = pd.to_datetime(dataTyumen['date_create'])
61 dataTyumen['date_create'] = dataTyumen['date_create'].dt.date
62 dataTyumen['time_exposition'] = dataTyumen['sold_date'] - dataTyumen['date_create']
63 dataTyumen['time_exposition'] = dataTyumen['time_exposition'] / np.timedelta64(1, 'D')
```

Рис.3.3.1 – Расчет времени экспозиции.

Создаем объект Series который будет столбцом кол-ва дней экспозиции. Приводим данные о продаже и создании объекта к временному формату, а время экспозиции к количеству дней. Далее сгруппируем полученные данные, по наибольшему количеству проданных объектов выведем первые 10 значений.

```
60 countDaysExposition = dataTyumen.groupby('time_exposition').count()['id'].sort_values(ascending=False)
61 print(countDaysExposition.head(10))
```

Рис.3.3.2 – Группировка по количеству проданных объектов.

```
time_exposition
43.0    42
91.0    41
39.0    40
42.0    39
65.0    38
52.0    38
49.0    38
29.0    37
85.0    36
56.0    36
Name: id, dtype: int64
```

Рис.3.3.3 Количество дней (левый столбец) и проданных объектов.

Для построения графика узнаем максимальное количество дней за которое проданся объект и максимальное количество объектов проданных за определенный период (рис. 2.3.3).

```
66 dataTyumen['time_exposition'].max()  
67 dataTyumen.groupby('time_exposition').count()['id'].max()
```

Рис.3.3.4 Максимальные значения.

Получаем что, максимальное количество дней 544 за которые был продан объект, и 42 объекта были проданы за 43 дня. Строим график.

```
65 import matplotlib.pyplot as plt  
66 from numpy.random import randn  
67  
68 groupExposition = dataTyumen.groupby('time_exposition')  
69  
70 x = groupExposition.count()['id'].index.get_level_values('time_exposition')  
71 y = groupExposition.count()['id']  
72  
73 maxX = dataTyumen['time_exposition'].max()  
74 maxY = groupExposition.count()['id'].max()+10  
75  
76 plt.plot(x, y, 'bo')  
77 plt.axis([0, 365, 0, maxY])  
78 z = randn(1)  
79 red_dot, = plt.plot(z, "bo", markersize=10)  
80 plt.legend([red_dot], ["Кол-во объектов проданных за кол-во дней"])  
81 plt.xlabel("Дни от размещения до продажи")  
82 plt.ylabel("Количество проданных объектов")  
83 plt.savefig('daysOnMarket.png')  
84 plt.show()
```

Рис.3.3.5 Строим график экспозиции.

На графике (График 3.3.1) видно, что чем дольше объект находится на продаже, тем меньше шансов его продать.

Путем расчетов удалось выяснить, что 45% всех проданных объектов, продаются в, первые 90 дней.

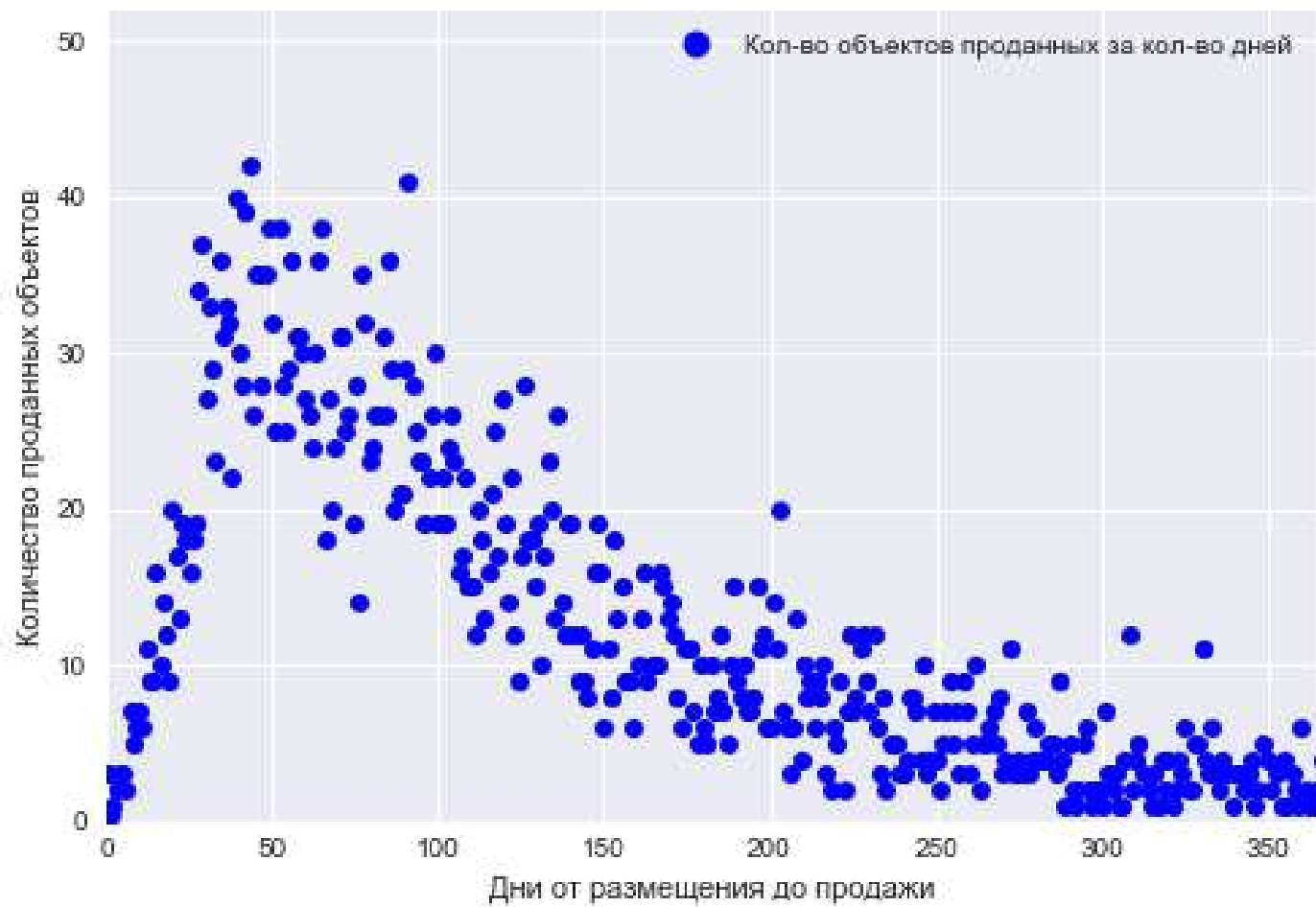


График.3.3.1 График экспозиции.

§ 3.4. Категоризация ликвидности.

Разбиваем объекты на 6 уровней ликвидности исходя из данных графика 3.3.1:

1. Сверхвысокая ликвидность - объект продается максимум за 20 дней;
2. Очень высокая - объект продается от 21 до 45 дней (+25 дней по сравнению с прошлым шагом);
3. Высокая - объект продается от 46 до 90 дней (+45 дней по сравнению с прошлым шагом);
4. Средняя - объект продается от 91 до 135 дней (+45 дней по сравнению с прошлым шагом);
5. Ниже средней - объект продается от 136 до 180 дней (+45 дней по сравнению с прошлым шагом);
6. Низкая - объект продается от 181 дней и более;

```
conditionUltraLicvidObjects = (dataTyumen['time_exposition']<21.0)
conditionVeryHighLicvidObjects = ((dataTyumen['time_exposition']>20.0)
& (dataTyumen['time_exposition']<46.0))
conditionHighLicvidObjects = ((dataTyumen['time_exposition']>45.0)
& (dataTyumen['time_exposition']<91.0))
conditionAverageLicvidObjects = ((dataTyumen['time_exposition']>90.0)
& (dataTyumen['time_exposition']<136.0))
conditionBelowAverageLicvidObjects = ((dataTyumen['time_exposition']>135.0)
& (dataTyumen['time_exposition']<181.0))
conditionLowLicvidObjects = ((dataTyumen['time_exposition']>180.0))
```

Рис.3.4.1 Категоризация ликвидности.

Рассчитаем количество объектов в каждой категории ликвидности.

```

106 fewer21 = (countDaysExposition.index.get_level_values('time_exposition')<21.0)
107 fewer46 = ((countDaysExposition.index.get_level_values('time_exposition')>20.0)
108 & (countDaysExposition.index.get_level_values('time_exposition')<46.0))
109 fewer91 = ((countDaysExposition.index.get_level_values('time_exposition')>45.0)
110 & (countDaysExposition.index.get_level_values('time_exposition')<91.0))
111 fewer136 = ((countDaysExposition.index.get_level_values('time_exposition')>90.0)
112 & (countDaysExposition.index.get_level_values('time_exposition')<136.0))
113 fewer181 = ((countDaysExposition.index.get_level_values('time_exposition')>135.0)
114 & (countDaysExposition.index.get_level_values('time_exposition')<181.0))
115 fewer271 = ((countDaysExposition.index.get_level_values('time_exposition')>135.0)
116 & (countDaysExposition.index.get_level_values('time_exposition')<181.0))
117 more271 = (countDaysExposition.index.get_level_values('time_exposition')>270.0)
118
119 #Сверхвысокая ликвидность - объект продается максимум за 20 дней
120 percentSellOnFirst20DaysUltra = countDaysExposition[(fewer21)].sum()/
121 dataTyumen['time_exposition'].count()
122 countSellOnFirst20DaysUltra = countDaysExposition[(fewer21)].sum()
123 #Очень высокая - объект продается от 21 до 45 дней (+25 дней по сравнению с прошл
124 percentSell21To45DaysVeryHigh = countDaysExposition[(fewer46)].sum()/
125 dataTyumen['time_exposition'].count()
126 countSell21To45DaysVeryHigh = countDaysExposition[(fewer46)].sum()
127 #Высокая - объект продается от 46 до 90 дней (+45 дней по сравнению с прошлым ша
128 percentSell46To90DaysHigh = countDaysExposition[(fewer91)].sum()/
129 dataTyumen['time_exposition'].count()
130 countSell46To90DaysHigh = countDaysExposition[(fewer91)].sum()
131 #Средняя - объект продается от 91 до 135 дней (+45 дней по сравнению с прошлым ш
132 percentSell91To135DaysAverage = countDaysExposition[(fewer136)].sum()/
133 dataTyumen['time_exposition'].count()
134 countSell91To135DaysAverage = countDaysExposition[(fewer136)].sum()
135 #Ниже средней - объект продается от 136 до 180 дней (+45 дней по сравнению с про
136 percentSell136To180DaysBelowAverage = countDaysExposition[(fewer181)].sum()/
137 dataTyumen['time_exposition'].count()
138 countSell136To180DaysBelowAverage = countDaysExposition[(fewer181)].sum()
139 #Низкая - объект продается от 181 до 270 дней (+90 дней по сравнению с прошлым ша
140 percentSell181To270DaysLow = countDaysExposition[(fewer271)].sum()/
141 dataTyumen['time_exposition'].count()
142 countSell181To270DaysLow = countDaysExposition[(fewer271)].sum()
143 #Очень низкая - объект продается от 271 дня и более
144 percentSell271AndMoreDaysVeryLow = countDaysExposition[(more271)].sum()/
145 dataTyumen['time_exposition'].count()
146 countSell271AndMoreDaysVeryLow = countDaysExposition[(more271)].sum()

```

Рис.3.4.2 Расчет количества объектов по ликвидности.

Количество объектов: 4676

Сверхвысокая ликвидность:

Процент объектов проданных за первые 20 дней: 0.03314798973481608

Количество объектов проданных за первые 20 дней: 155

Очень высоколиквидные:

Процент объектов проданных за 21-45 дней: 0.1490590248075278

Количество объектов проданных за 21-45 дней: 697

Высоко-ликвидные:

Процент объектов проданных за 46-90 дней: 0.2703165098374679

Количество объектов проданных за 46-90 дней: 1264

Средне-ликвидные:

Процент объектов проданных за 91-135 дней: 0.2219846022241232

Количество объектов проданных за 91-135 дней: 1038

Ниже среднего:

Процент объектов проданных за 136-180 дней: 0.08319076133447391

Количество объектов проданных за 136-180 дней: 389

Низко-ликвидные:

Процент объектов проданных за 181 дней и более: 0.24

Количество объектов проданных за 181 дней и более: 1130

§ 3.5. Ликвидность по районам.

Группируем проданные объекты выборки по районам.

```
148 countSellApartmentsInDistrict = dataTyumen.groupby('district')
149                                     .count()['id']
150                                     .sort_values(ascending=False)
```

Рис.3.5.1 Группировка по районам.

На рисунке 3.5.2 указаны районы с наибольшим количеством продаж. В средней в каждом районе продается 10% от выставленных на продажу квартир.

district	
Центр: КПД	336
МЖК	323
Центр: Драмтеатр	222
Дом Оборона	221
Лесобаза	194
Тюменский-3	179
Южный мкр	173
Мыс	161
Восточный-2	157
Тюменский-1	150
Тюменская слобода	146
Центр: Дом печати	145
Тюменский-2	139

Рис.3.5.2 Группировка по районам с наибольшим количеством продаж.

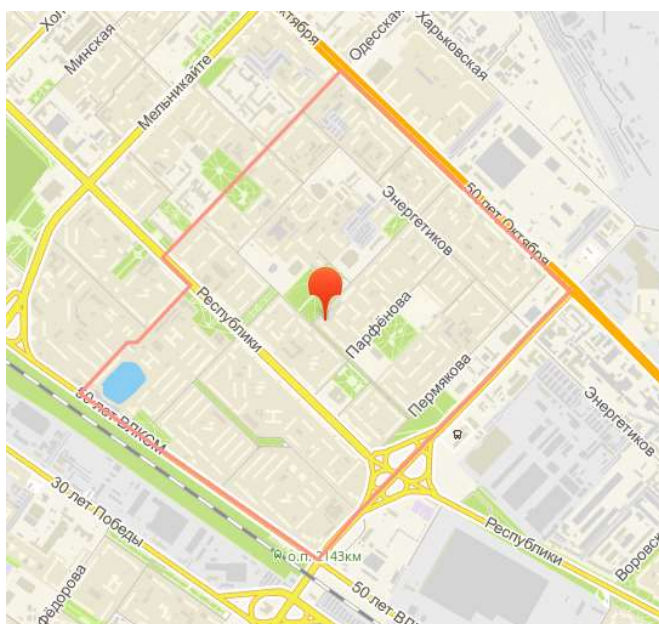


Рис.3.5.3 КПД на первом месте по количеству продаж.

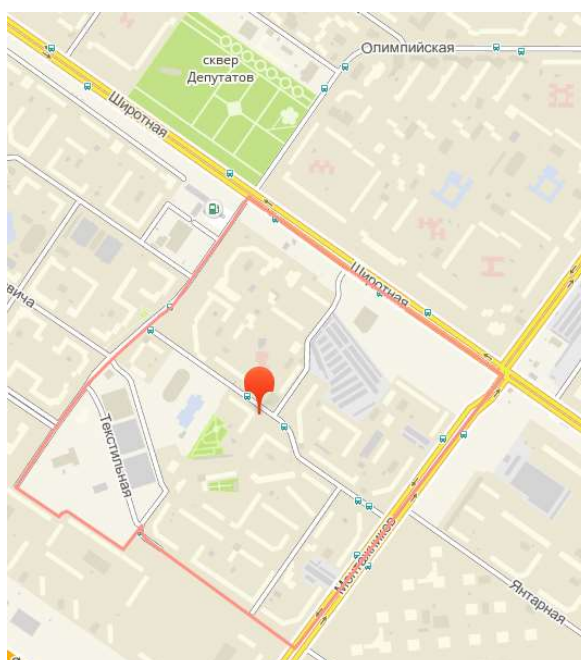


Рис.3.5.4 МЖК на втором месте по количеству продаж.

Количество объектов по районам проданных за первые 20 дней:		проданных за 91-135 дней:	
Центр: КПД	14	МЖК	68
Тюменский-1	13	Центр: КПД	66
МЖК	10	Дом Обороны	50
Тюменская слобода	10	Лесобаза	48
Антипино	8	Тюменский-3	40
Южный мкр	8	Центр: Драмтеатр	40
Тюменский-3	7	Центр: Дом печати	39
Тарманы	7	Южный мкр	37
Центр: Драмтеатр	6	Восточный-2	34
Тюменский-2	5	Мыс	32
проданных за 21-45 дней:		проданных за 136-180 дней:	
Центр: КПД	65	МЖК	24
МЖК	60	Центр: КПД	24
Мыс	34	Дом Обороны	24
Тюменская слобода	32	Центр: Драмтеатр	20
Тюменский-1	32	Лесобаза	19
Тюменский-3	30	Тюменский-3	16
Дом Обороны	30	Тюменский-2	15
Антипино	27	Восточный-2	14
Лесобаза	27	Мыс	13
Южный мкр	27	3-й Заречный мкр	13
проданных за 46-90 дней:		проданных за 181-270 дней:	
Центр: КПД	103	МЖК	43
МЖК	81	Центр: Драмтеатр	38
Центр: Драмтеатр	71	Центр: КПД	34
Тюменский-3	57	Дом Обороны	32
Лесобаза	53	Южный мкр	29
Дом Обороны	51	Центр: Дом печати	25
Восточный-2	47	Восточный-2	25
Антипино	47	Мыс	22
Тюменский-2	43	Лесобаза	22
Тюменский-1	41	Центр: Исторический	22
		проданных за 271 и более дней:	
		МЖК	37
		Дом Обороны	30
		Центр: КПД	30
		Восточный-2	25
		Южный мкр	25
		Лесобаза	24
		Мыс	22
		Центр: Драмтеатр	20
		Московский тракт	16
		Червишевский тракт	14

Рис.3.5.7 Группировка по ликвидности района.

§ 3.6. Поиск закономерности проданных объектов методом Случайный лес.

Добавим данным еще один столбец 'sold', продан или нет объект.

```

195 dataTyumen['sold'] = Series('', index=dataTyumen.index)
196 dataTyumen.loc[dataTyumen.status != 'sold', 'sold'] = 0
197 dataTyumen.loc[dataTyumen.status == 'sold', 'sold'] = 1

```

Рис.3.6.1 Добавление столбца 'sold'.

Формируем объект DataFrame из необходимых столбцов (полный список в приложении №1).

```

dataTyumen = dataTyumen[['sold', 'area_totalf', 'floor_num',
    'floors_cnt', 'rooms_cnt', 'district', 'street',
    'building_year', 'secondary', 'type', 'two_levels',
    'levels_count', 'balcon',
    'walls_id', 'child_place', 'closed', 'yard_opened',
    'yard_barrier', 'yard_gate', 'yard_security', 'parking',
    'subway_parking', 'semisubway_parking', 'garage_parking',
    'guest_parking', 'no_parking', 'elevator', 'cargo_elevator',
    'kolyasochnaya', 'communication', 'water_counters', 'hot_counters',
    'domofon', 'videodomofon', 'iron_door', 'safe_signal',
    'consjerg', 'security', 'social', 'flatsType', 'category',
    'old_elit', 'countryside', 'shop_among_house', 'trc',
    'sports_objects', 'policlinic', 'sto_azs', 'kindergarden',
    'school', 'city_center', 'bus_stop', 'cnt_transport_routes',
    'environmentally', 'non_residential', 'rampant', 'dog_yard',
    'bicycle_paths', 'storeroom', 'material_window', 'fire_alarm',
    'parking_bicycles', 'sports_ground', 'controls_heat',
    'electricity_meters', 'no_pan']]

```

Рис.3.6.2 Список столбцов для обучения объекта Случайный лес.

Подключаем библиотеку sklearn и формируем объект классификатор RandomForestClassifier.

```

from sklearn import ensemble, cross_validation, learning_curve, metrics

price_target = dataTyumen.sold.values
price_data = dataTyumen.iloc[:, 1:]

rf_classifier_low_depth = ensemble.RandomForestClassifier(n_estimators = 100,
    max_depth = 15,
    random_state = 1)

```

Рис.3.6.3 Объект классификатора Случайный лес.

В первую переменную sold_target записываем целевую переменную, во вторую данные для обучения. Объект будет иметь глубину 100 деревьев с глубиной 15.

```

train_sizes,
train_scores,
test_scores = learning_curve.learning_curve(rf_classifier_low_depth,
    sold_data,
    sold_target,
    train_sizes=np.arange(0.1,1., 0.2),
    cv=10,
    scoring='accuracy')

plt.grid(True)
plt.plot(train_sizes, train_scores.mean(axis = 1),
    'g-', marker='o', label='train')
plt.plot(train_sizes, test_scores.mean(axis = 1),
    'r-', marker='o', label='test')
plt.ylim((0.0, 1.05))
plt.legend(loc='lower right')

```

Рис.3.6.4 Обучение модели и построение графика.

С помощью метода `learning_curve` будут построены несколько моделей, мы получим оценку качества, на каждом объеме обучающей выборки. В результате получим размер обучающей выборки, оценка качества на обучение и тесте.

```
print(train_sizes)
print(train_scores.mean(axis = 1))
print(test_scores.mean(axis = 1))

[ 3941 11825 19709 27593 35477]
[0.98063943 0.92199577 0.87794916 0.85220165 0.8492629 ]
[0.7471849 0.76812185 0.79784973 0.81479184 0.82330937]
```

Рис.3.6.5 Результаты обучения.

По результатам видно, что на тестовой выборке мы получили качество обучения 82,3%, что является хорошим результатом.

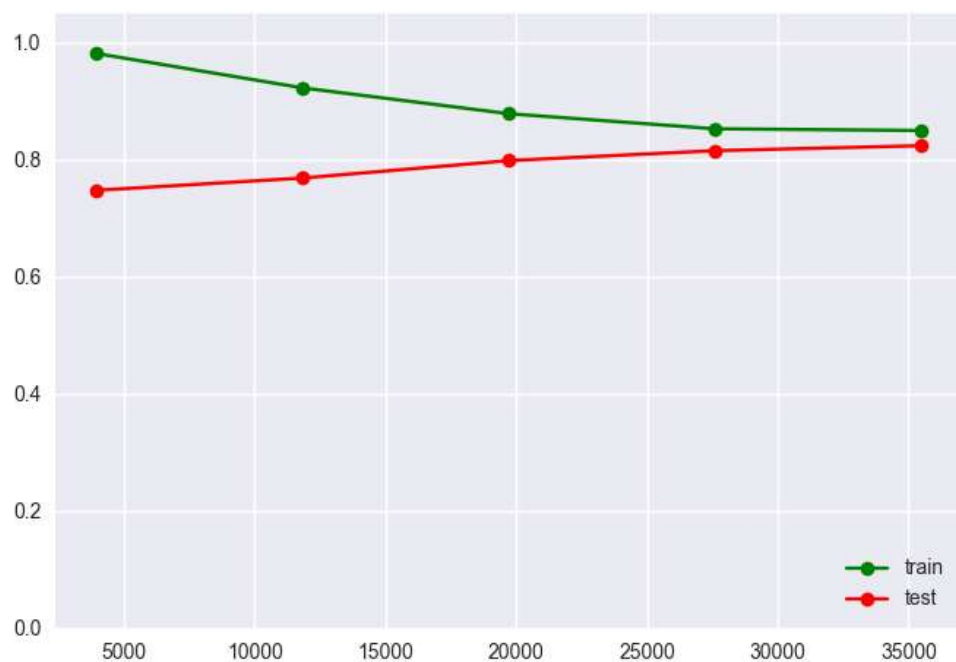


Рис.3.6.6 График обучения. (нижняя линия test)

§ 3.7. Поиск категории ликвидности объектов.

Отбираем необходимые данные из DataFrame.

```
dataTyumen = dataTyumen.loc[(dataTyumen['date_create'] > '2015-12-31') &
                             (dataTyumen['status'] != 'deleted')]
```

Рис.3.7.1 Отбор данных из общего DataFrame

Там где дата продажи равно null, это значит объект не был продан, заменяем на последнюю дату которая есть в данных.

```
dataTyumen.loc[dataTyumen.sold_date.isnull(), 'sold_date'] = '2017-11-28'
```

Рис.3.7.2 Заменяем null на дату

Присваиваем объектам метку в зависимости от, продан объект или нет.

```
dataReport990['sold'] = Series('', index=dataReport990.index)
dataReport990.loc[dataReport990.status != 'sold', 'sold'] = 0
dataReport990.loc[dataReport990.status == 'sold', 'sold'] = 1
```

Рис.3.7.3 Метка проданных объектов

Считаем время экспозиции (количество дней от размещения объекта до продажи).

```
dataTyumen['sold_date'] = pd.to_datetime(dataTyumen['sold_date'])
dataTyumen['sold_date'] = dataTyumen['sold_date'].dt.date
dataTyumen['date_create'] = pd.to_datetime(dataTyumen['date_create'])
dataTyumen['date_create'] = dataTyumen['date_create'].dt.date
dataTyumen['time_exposition'] = dataTyumen['sold_date'] - dataTyumen['date_create']
dataTyumen['time_exposition'] = dataTyumen['time_exposition'] / np.timedelta64(1, 'D')
```

Рис.3.7.4 Подсчет времени экспозиции реальных объектов

Для удачного выполнения классификации по категориям ликвидности увеличиваем выборку за счет непроданных объектов. Для этого берем все непроданные объекты которые продаются более 180 дней и присваиваем им последнюю шестую категорию ликвидности (которая характеризуется как низкая). Все объекты которые продаются меньше этого срока и не были проданы, удаляем. Получаем 29 382 объекта.

```
dataTyumen = dataTyumen.drop(dataTyumen[(dataTyumen.time_exposition < 181)
                                         & (dataTyumen.sold == 0)].index)
```

Рис.3.7.5 Удаление не проданных объектов

Создаем категории ликвидности по временным промежуткам.

```
conditionUltraLicvidObjects = (dataTyumen['time_exposition']<21.0)
conditionVeryHighLicvidObjects = ((dataTyumen['time_exposition']>20.0)
& (dataTyumen['time_exposition']<46.0))
conditionHighLicvidObjects = ((dataTyumen['time_exposition']>45.0)
& (dataTyumen['time_exposition']<91.0))
conditionAverageLicvidObjects = ((dataTyumen['time_exposition']>90.0)
& (dataTyumen['time_exposition']<136.0))
conditionBelowAverageLicvidObjects = ((dataTyumen['time_exposition']>135.0)
& (dataTyumen['time_exposition']<181.0))
conditionLowLicvidObjects = ((dataTyumen['time_exposition']>180.0))
```

Рис.3.7.6 Промежутки дней продажи для категорий ликвидности

Исходя из полученных дней, присваиваем проданным объектам категорию ликвидности (от первой до шестой) по проданным объектам, не проданные удаляем.

```
dataTyumen['liquidity_on_demand'] = Series('',index=dataTyumen.index)
dataTyumen.loc[conditionUltraLicvidObjects,'liquidity_on_demand'] = 1
dataTyumen.loc[conditionVeryHighLicvidObjects,'liquidity_on_demand'] = 2
dataTyumen.loc[conditionHighLicvidObjects,'liquidity_on_demand'] = 3
dataTyumen.loc[conditionAverageLicvidObjects,'liquidity_on_demand'] = 4
dataTyumen.loc[conditionBelowAverageLicvidObjects,'liquidity_on_demand'] = 5
dataTyumen.loc[conditionLowLicvidObjects,'liquidity_on_demand'] = 6
```

Рис.3.7.7 Категории ликвидности

Удаляем строки где район не известен.

```
dataTyumen = dataTyumen.drop(dataTyumen[(dataTyumen.district_id == -1)].index)
```

Рис.3.7.8 Удаление не известных районов.

Выбираем 56 признаков для анализа (Приложение № 6).

```
dataForLiquidityCategory = dataTyumen[
    ['liquidity_on_demand', 'area_totalf', 'floor_num',
    'district_id', 'floors_cnt', 'rooms_cnt', 'building_year', 'levels_count',
    'walls_id', 'child_place', 'closed', 'yard_opened', 'yard_barrier',
    'yard_gate', 'yard_security', 'parking', 'subway_parking', 'semisubway_parking',
    'garage_parking', 'guest_parking', 'no_parking', 'elevator', 'cargo_elevator',
    'kolyasochnaya', 'communication', 'water_counters', 'hot_counters',
    'domofon', 'videodomofon', 'iron_door', 'safe_signal', 'consjerg',
    'security', 'social', 'old_elit', 'countryside', 'bicycle_paths', 'storeroom',
    'fire_alarm', 'parking_bicycles', 'sports_ground', 'controls_heat',
    'electricity_meters', 'no_pan', 'shop_among_house', 'trc', 'sports_objects',
    'policlinic', 'sto_azs', 'kindergarden', 'school', 'city_center', 'bus_stop',
    'cnt_transport_routes', 'environmentally', 'material_window']]
```

Рис.3.7.9 Выбор признаков для анализа.

Импортируем RandomForestClassifier, обучаем модель.

```
from sklearn import ensemble, learning_curve

price_target = dataForLiquidityCategory.liquidity_on_demand.values
price_data = dataForLiquidityCategory.iloc[:, 1:]

rf_classifier_low_depth = ensemble.RandomForestClassifier(n_estimators = 100,
                                                         max_depth = 15,
                                                         random_state = 1)

train_sizes, train_scores, test_scores =
    learning_curve.learning_curve(rf_classifier_low_depth,
                                 price_data,
                                 price_target,
                                 train_sizes=np.arange(0.1,1., 0.2),
                                 cv=10,
                                 scoring='accuracy')
```

Рис.3.7.10 Обучение модели для классификации категорий недвижимости.

В результате получаем точность предсказывания категории ликвидности 81,2% Рис. 3.7.11.

```
[ 2644  7932 13220 18508 23796]
[0.93698941 0.87634897 0.86910741 0.86822995 0.86441839]
[0.80212175 0.8075681 0.80855494 0.81035922 0.81274187]
```

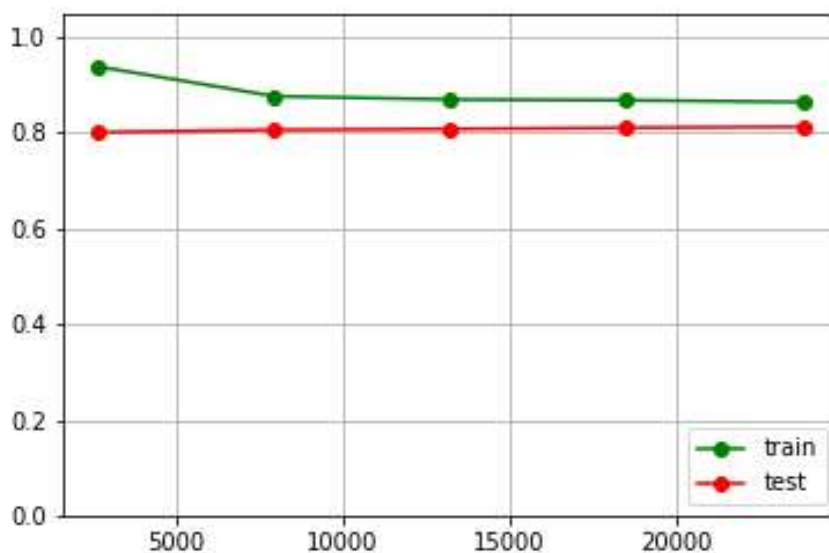


Рис.3.6.6 График обучения (нижняя линия test)

Равномерность линий говорит о том что большее увеличение выборки не улучшит данную модель.

§ 3.8. Предсказание срока экспозиции объекта.

При подготовке проводим аналогичные шаги описанные в § 3.7. и удаляем все непроданные объекты т.к. эти данные не подходят для наших целей. Получаем 7555 объектов.

```
dataTyumen = dataTyumen.drop(dataTyumen[(dataTyumen.sold == 0)].index)
```

Рис.3.8.1 Удаление не проданных объектов

Выбираем 57 признаков для анализа (Приложение № 6).

```
dataDaysForRandomForest = dataTyumen[
    ['time_exposition', 'liquidity_on_demand', 'area_totalf', 'floor_num',
     'district_id', 'floors_cnt', 'rooms_cnt', 'building_year', 'levels_count',
     'walls_id', 'child_place', 'closed', 'yard_opened', 'yard_barrier',
     'yard_gate', 'yard_security', 'parking', 'subway_parking', 'semisubway_parking',
     'garage_parking', 'guest_parking', 'no_parking', 'elevator', 'cargo_elevator',
     'kolyasochnaya', 'communication', 'water_counters', 'hot_counters',
     'domofon', 'videodomofon', 'iron_door', 'safe_signal', 'consjerg',
     'security', 'social', 'old_elit', 'countryside', 'bicycle_paths', 'storeroom',
     'fire_alarm', 'parking_bicycles', 'sports_ground', 'controls_heat',
     'electricity_meters', 'no_pan', 'shop_among_house', 'trc', 'sports_objects',
     'policlinic', 'sto_azs', 'kindergarden', 'school', 'city_center', 'bus_stop',
     'cnt_transport_routes', 'environmentally', 'material_window']]
```

Рис.3.8.2 Выбор признаков для анализа.

Импортируем RandomForestRegressor.

```
from sklearn.ensemble import RandomForestRegressor
```

Рис.3.8.3 Импорт RandomForestRegressor.

Перемешиваем данные для более точного анализа.

```
dataDaysForRandomForest = dataDaysForRandomForest.sample(frac=1).reset_index(drop=True)
```

Рис.3.8.4 Перемешиваем данные.

Создаем объект регрессор. Для обучения берем все данные сначала кроме последней тысячи. Для тестовой выборки берем последнюю тысячу. В labels записываем столбец значения которого нужно найти. В обучающей выборке соответственно удаляем.

```

regressor = RandomForestRegressor(random_state = 0, max_depth = 20, n_estimators = 150)

train_data = dataDaysForRandomForest.iloc[:-1000, :]
hold_out_test_data = dataDaysForRandomForest.iloc[-1000:, :]

train_labels = train_data['time_exposition'].values
test_labels = hold_out_test_data['time_exposition'].values

#возможно стоит дронуть целевые столбцы
train_data = train_data.drop(['time_exposition'], axis=1)
hold_out_test_data = hold_out_test_data.drop(['time_exposition'], axis=1)

```

Рис.3.8.5 Создание обучающей и тестовой выборки.

Разбиваем данные на бинарные, категориальные, и со значениями. Вешаем на данные булев тип чтобы привести их к одному типу.

```

test_data = hold_out_test_data;

binary_data_columns = ['closed', 'yard_barrier', 'yard_gate', 'parking',
    'semisubway_parking', 'garage_parking', 'guest_parking', 'cargo_elevator',
    'kolyasochnaya', 'water_counters', 'hot_counters', 'domofon',
    'videodomofon', 'iron_door', 'safe_signal', 'consjerg', 'security',
    'social', 'countryside', 'bicycle_paths', 'storeroom', 'parking_bicycles',
    'controls_heat', 'electricity_meters', 'no_pan', 'child_place', 'yard_opened',
    'yard_security', 'subway_parking', 'no_parking', 'elevator', 'communication',
    'old_elit', 'fire_alarm', 'sports_ground']

binary_data_indices = np.array([(column in binary_data_columns)
    for column in train_data.columns], dtype = bool)

categorical_data_columns = ['liquidity_on_demand', 'district_id', 'walls_id', 'material_window']
categorical_data_indices = np.array([(column in categorical_data_columns)
    for column in train_data.columns], dtype = bool)

numeric_data_columns = ['area_totalf', 'floors_cnt', 'floor_num', 'rooms_cnt',
    'building_year', 'shop_among_house', 'sports_objects',
    'polyclinic', 'levels_count', 'trc', 'sto_azs', 'kindergarden',
    'school', 'city_center', 'bus_stop', 'cnt_transport_routes',
    'environmentally']
numeric_data_indices = np.array([(column in numeric_data_columns)
    for column in train_data.columns], dtype = bool)

```

Рис.3.8.6 Разбивка данных и приведение к одному типу.

Импортируем pipeline. Первый шаг в нем это, обработка данных, второй построение модели.

```
from sklearn import pipeline, preprocessing

estimator = pipeline.Pipeline(steps = [
    ('feature_processing', pipeline.FeatureUnion(transformer_list = [
        #binary
        ('binary_variables_processing',
         preprocessing.FunctionTransformer(lambda data: data[:, binary_data_indices])),

        #numeric
        ('numeric_variables_processing', pipeline.Pipeline(steps = [
            ('selecting',
             preprocessing.FunctionTransformer(lambda data: data[:, numeric_data_indices])),
            ('scaling', preprocessing.StandardScaler(with_mean = 0, with_std = 1))
        ])),

        #categorical
        ('categorical_variables_processing', pipeline.Pipeline(steps = [
            ('selecting',
             preprocessing.FunctionTransformer(lambda data: data[:, categorical_data_indices])),
            ('hot_encoding',
             preprocessing.OneHotEncoder(handle_unknown = 'ignore'))
        ]))
    ])),
    ('model_fitting', regressor)
])
```

Рис.3.8.7 Обработка данных в pipeline.

Обучаем модель. Строим график (Рис.3.8.9).

```
estimator.fit(train_data, train_labels)
print(metrics.mean_absolute_error(test_labels, estimator.predict(test_data)))
print(test_labels[:10])
print(estimator.predict(test_data)[:10])

plt.subplot(1,2,2)
plt.grid(True)
plt.xlim(-20,750)
plt.ylim(-20,600)
plt.scatter(train_labels, estimator.predict(train_data), alpha=0.9, color = 'red')
plt.scatter(test_labels, estimator.predict(test_data), alpha=0.5, color = 'blue')
plt.title('random forest regressor model')
```

Рис.3.8.8 Построение графика.

На графике Рис.3.8.9-10 показано попадание тестовой выборки 1000 объектов (синие точки) в фактические значения 7555 (красные точки).

Точность предсказания +-15 дней. При перемешанной выборке. И +-5 дней, если взять последние значения.

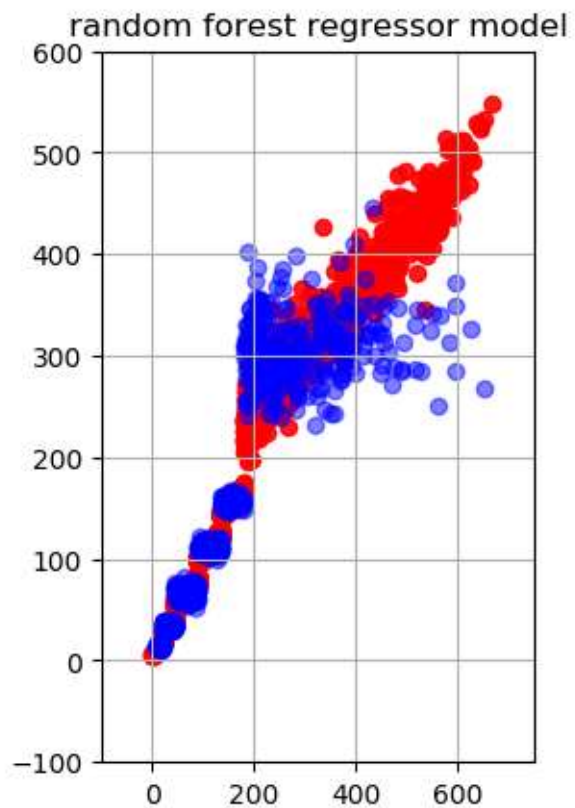


Рис.3.8.9 График попадания предсказанных значений (синие точки) в фактические значения (красные точки). (Перемешанные данные.) Точность +/-15 дней.

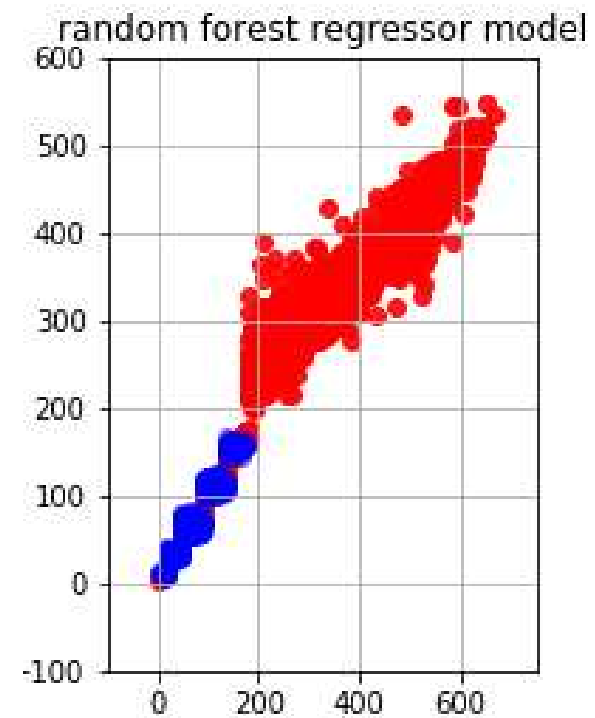


Рис.3.8.10 График попадания предсказанных значений (синие точки) в фактические значения (красные точки). (Последние 1000 значений, без перемешивания.) Точность +/-5 дней.

Таблица 3.8.1

Перемешанные данные. Точность средняя по всей выборке +/-15 дней.

Фактические значения, дней	Предсказанные значения, дней	Ошибка, дней
331	316	-15
134	112	-22
22	32	+10
124	109	-15
118	110	-8
68	58	-10
34	33	-1
98	118	+20
59	69	+10
167	158	-9

Таблица 3.8.2

Последние 1000 значений, без перемешивания. Точность средняя по всей выборке +/-5 дней.

Фактические значения, дней	Предсказанные значения, дней	Ошибка, дней
129	115	-14
140	153	+13
116	109	-7
83	60	-23
45	33	-12
174	156	-18
47	68	+21
80	65	-15
67	68	+1
123	115	-8

§ 3.9. Библиотека Scikit-Learn

Существует несколько библиотек языка Python с надежными реализациями широкого диапазона алгоритмов машинного обучения. Одна из самых известных — Scikit-Learn, пакет, предоставляющий эффективные версии множества распространенных алгоритмов. Пакет Scikit-Learn отличается аккуратный, единообразный и продвинутой API, а также удобная и всеохватывающая онлайн-документация.

Чаще всего использование API статистического оценивания библиотеки Scikit-Learn включает следующие шаги:

1. Выбор класса модели с помощью импорта соответствующего класса оценщика из библиотеки Scikit-Learn.
2. Выбор гиперпараметров модели путем создания экземпляра этого класса с соответствующими значениями.
3. Компоновка данных в матрицу признаков и целевой вектор в соответствии с описанным выше.
4. Обучение модели на своих данных посредством вызова метода `fit()` экземпляра модели.
5. Применение модели к новым данным:
 - в случае машинного обучения с учителем метки для неизвестных данных обычно предсказывают с помощью метода `predict()`;
 - в случае машинного обучения без учителя выполняется преобразование свойств данных или вывод их значений посредством методов `transform()` или `predict()`.

Согласно документации по API Scikit-Learn, основывается на следующих принципах:

- *единообразие* — интерфейс всех объектов идентичен и основан на ограниченном наборе методов, документация тоже единообразна;

- *контроль* — видимость всех задаваемых значений параметров как открытых атрибутов;
- ограниченная иерархия объектов — классы языка Python используются только
- для алгоритмов; наборы данных представлены в стандартных форматах (массивы NumPy, объекты DataFrame библиотеки Pandas, разреженные матрицы библиотеки SciPy), а для имен параметров используются стандартные строки языка Python;
- *объединение* — многие из задач машинного обучения можно выразить в виде последовательностей алгоритмов более низкого уровня, и библиотека Scikit-Learn пользуется этим при любой возможности;
- разумные значения по умолчанию — библиотека задает для необходимых моделей пользовательских параметров соответствующие значения по умолчанию.

ЗАКЛЮЧЕНИЕ

В данной работе был показан алгоритм оценки ликвидности объектов недвижимости, алгоритм нахождения стоимости объектов методом подбора аналогов.

Для нахождения ликвидности был использован Python [9] с библиотеками с открытым исходным кодом под лицензией BSD, такие как библиотеки для машинного обучения Pandas, NumPy, sklearn [10].

В процессе создания данной работы были изучены и реализованы методы машинного обучения такие как логистическая регрессия, ридж классификация, деревья решений, случайный лес [11]. В работу попал алгоритм случайны лес классификатор и случайный лес регрессор, так как он показал наименьший процент ошибки.

Был разработан прототип программы с веб-интерфейсом. Для создания прототипа использовался язык программирования JavaScript. JavaScript объектно-ориентированный сценарный язык программирования [6]. Наиболее широко применяется в браузерах как язык сценариев для придания интерактивности web-страницам. Django сервер, на языке Python [9]. MySQL 5.7 свободная реляционная система управления базами данных [8]. MySQL часто используется в качестве СУБД в разработке сайтов.

СПИСОК ЛИТЕРАТУРЫ И ИСТОЧНИКОВ

1. Hengshu Zhu1 ,Hui Xiong. Days on Market: Measuring Liquidity in Real Estate Markets , KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining San Francisco, California, USA — August 13 - 17, 2016
2. Simon K.C. Cheung. A Localized Model for Residential Property Valuation: Nearest Neighbor with Attribute Differences, International real estate review 2017 Vol. 20 No. 2: pp. 221 – 250
3. Raymond Brummelhuis, Zhongmin Luo. CDS Rate Construction Methods By Machine Learning Techniques, 2017
4. Материалы компании «Этажи».
5. <http://ries3.etagi.com>
6. Этан Браун. Изучаем JavaScript. 2017г.
7. Шелли Пауэрс.Изучаем Node. Переходим на сторону сервера. 2017г.
8. Сейед Тахагхогхи, Хью Е. Руководство по MySQL. Вильямс 2017г.
9. Майкл Доусон. Программируем на Python. 2017г.
10. Дж. Вандер Плас. Python для сложных задач. 2018г.
11. Coursera: <https://www.coursera.org/specializations/machine-learning-data-analysis>

ПРИЛОЖЕНИЕ №1**КОЭФФИЦИЕНТ КАТЕГОРИЙ ЖИЛЬЯ**

№	Категория жилья объекта	Категория жилья аналога	Коэффициент
1	Эконом	Комфорт	0,848
2	Эконом	Бизнес	0,683
3	Эконом	Элит	1
4	Эконом	Старый элитный фонд	0,925
5	Комфорт	Эконом	1,179
6	Комфорт	Бизнес	0,805
7	Комфорт	Элит	1
8	Комфорт	Старый элитный фонд	1,101
9	Бизнес	Эконом	1,471
10	Бизнес	Комфорт	1,247
11	Бизнес	Элит	1
12	Бизнес	Старый элитный фонд	1,451
13	Элит	Эконом	1
14	Элит	Комфорт	1
15	Элит	Бизнес	1
16	Элит	Старый элитный фонд	1
17	Старый элитный фонд	Эконом	1,111
18	Старый элитный фонд	Комфорт	0,910
19	Старый элитный фонд	Бизнес	0,689
20	Старый элитный фонд	Элит	1

ПРИЛОЖЕНИЕ №2**КОЭФФИЦИЕНТ ЭТАЖЕЙ.**

Кол-во комнат	Этаж	Коэффициент
1-комнатная	1-этаж	0,9481
2-комнатная	1-этаж	0,9643
3-комнатная	1-этаж	0,9769
3+-комнатная	1-этаж	1
1-комнатная	Последний этаж	1
2-комнатная	Последний этаж	1
3-комнатная	Последний этаж	1
3+-комнатная	Последний этаж	1
1-комнатная	Средний этаж	1,0519
2-комнатная	Средний этаж	1,0357
3-комнатная	Средний этаж	1,0231
3+-комнатная	Средний этаж	1

ПРИЛОЖЕНИЕ №3

КОЭФФИЦИЕНТ МАТЕРИАЛОВ СТЕН.

Тип стен	Материал стен	Коэффициент
Кирпичные	Кирпичные керамический Кирпичные облицовочные Кирпичные полнотелые Кирпичные силикатные Кирпичные	1
Панельные	Железобетонные панели Кирпично-панельные панели Панельные + кирпич	0,95
Блочные	Блоки железобетонные Блочные Блочные + кирпич Газобетонные блоки (Сибит) Керамзитобетонные блоки Крупноблочные Пенобетонные блоки Пеноблоки Силикатные блоки «Поревит» Шлакобетонные блоки Шлакоблочные	0,96
Монолитные	Кирпичные, монолитно-каркасные Монолитно-кирпичные Монолитные Сборно-монолитные	0,98

ПРИЛОЖЕНИЕ №4

ОБЪЕДИНЕНИЕ РАЙОНОВ.

Район	Объединение 1	Объединение 2
Маяк	Стрела	Московский тр. Червишев. тр.
Антипино	Лесной мкр.	
4 мкр.	5 мкр. 6 мкр.	3 мкр. Южный мкр.
5 мкр.	5 мкр. 6 мкр.	3 мкр. Южный мкр.
6 мкр.	4 мкр. 5 мкр.	Южный мкр.
Бабарынка	ДОК	
Ватутино	СМП	Югра
Восточный	1 мкр.	Войновка, Восточный-2
Войновка	Восточный, Восточный-2	
ДОК	Бабарынка	
Казарово	Березняковский п.	
Метелево	Воронино, Метелево	Бабарынка, Воронино, ДОК, Метелево
Мыс	Матмасы, Тарманы	
Матмасы, Тарманы	Мыс	
Стрела	Маяк, Червишевский тр.	Московский тр.
Югра	Ватутино, СМП	
Парфеново	Нефтяников	ММС
СМП	Ватутино, Югра	
ММС	Парфеново	
Восточный-2	Восточный-3	Войновка
Восточный-3	МЖК	Восточный, Восточный-2
1-й Заречный	2-й Заречный	3-й Заречный
2-й Заречный	1-й Заречный	3-й Заречный
3-й Заречный	1-й Заречный, 2-й Заречный	Европейский
Нефтяников	Парфеново	
Московский тр.	Червишевский тр.	Плеханово, Стрела
Южный мкр.	Червишевский тр.	6 мкр.
Червишевский тр.	Южный мкр.	Московский тр. , Стрела
Центр: Студгородок	Центр: КПД	
Центр: КПД	Центр: Студгородок	
Центр: Дом печати	Центр: Драмтеатр	Центр: Исторический
Центр: Драмтеатр	Центр: Дом печати	Центр: Исторический
Центр: Исторический	Центр: Драмтеатр	Центр: Дом печати
1 мкр.	2мкр., 3.мкр	Восточный
2 мкр.	1мкр., 3мкр.	Восточный
3 мкр.	1мкр., 2мкр.	4мкр., 5мкр.
Тюменский-1	Тюменский-2	Тюменский-3
Тюменский-2	Тюменский-1	Тюменский-3
Тюменский-3	Тюменский-1, Тюменский-2	
МЖК	Восточный-3	Тюменский-1, Тюменский-2
Европейский мкр.		3-й Заречный мкр.
Плеханово	Московский тр.	Тюменская слобода
Тюменская слобода	Плеханово	Ожогоино

ПРИЛОЖЕНИЕ №5

ОБЪЕДИНЕНИЕ СЕРИЙ ЗДАНИЙ.

Серия	Первое объединение	Второе объединение
лен	брежневка	хрущ
киев	стал	моск
ярослав	ярослав	болг сверд
болг	86 сверд ярослав	121 125 125-т
стал	киев	моск
школ		121 121-3т 121-7т 121-7тм 121-т 125-т Инд., керамзито-блочный
сверд	болг ярослав	121 125
121/14	121-3т 121-т 121/12	121-7т 121-7тм 125-т
трехлист	трехлист	сверд ярослав
125-т	121 125	121-3т 121-7т 121-7тм 121-т 121/12 121/14 Инд., керамзито-блочный
125	121 125-т	121-3т 121-7т 121-7тм 121-т 121/12 121/14 86 болг сверд ярослав
121-3т	121-7т 121-7тм 121-т Инд., керамзито-блочный	121/12 121/14 125-т
86	121 125	125-т болг Инд., керамзито-блочный

брежневка	хрущ	лен
хрущ	брежневка	лен
121	125 125-т	121-3т 121-7т 121-7тм 121-т 8б
121/12	121-3т 121-т 121/14	121-7т 121-7тм 125-т
121-7т	121-3т 121-7тм 121-т	121-т 121/12 121/14 125-т Инд., керамзито-блочный школ
121-т	121-3т 121-7т 121-7тм	121 121/12 121/14 125 125-т Инд., керамзито-блочный
турк	121 125 болг Грузинский проект	
121-7тм	121-3т 121-7т 121-т	121-т 121/12 121/14 125-т Инд., керамзито-блочный школ
ТУРА 20.12	инд	
Инд., керамзито-блочный	121-3т 121-7т 121-7тм 121-т 125-т	121 125 болг школ
Грузинский проект	турк	121 121-3т 121-т 125 125-т болг лен
обл	моск	121 121-3т 121-т 125 125-т киев
моск	обл	121 121-3т 121-т 125 125-т болг

ПРИЛОЖЕНИЕ №6

ПРИЗНАКИ, ПО КОТОРЫМ ОЦЕНИВАЕТСЯ ЛИКВИДНОСТЬ ОБЪЕКТОВ.

№	Название признака	Название атрибута в базе	№	Название признака	Название атрибута в базе
1	Кол-во фото	photo_cnt	44	Кол-во магазинов	shop_among
2	Кол-во туров	toure_cnt	45	ТРЦ	trc
3	Аукцион (0/1)	auction	46	Кол. спорт.объект.	sports_objects
4	Дата размещения	date_rise	47	Поликлиники	policlinic
5	Дата продажи	sold_date	48	Кол-во СТО, АЗС	sto_azs
6	Цена продажи	sold_price	49	Кол-во дет.садов	kindergaden
7	В ипотеке	in_mortgage	50	Кол-во школ	school
8	Цена за метр	price_ar	51	Км. до центра	city_center
9	Улица	street_intersection	52	Кол-во остановок	bus_stop
10	Дом	house	53	Кол-во маршрутов	cnt_transport_rates
11	Тип стен здания	walls_id	54	Экологичность	environmentally
12	Фасад здания	fasad_id	55	Нежилой (0/1)	non_residential
13	Дет. площадка (0/1)	child_place	56	Пандус (0/1)	rampant
14	Закрытый двор (0/1)	closed	57	Площад. собак (0/1)	dog_yard
15	Открытый двор (0/1)	yard_opened	58	Велодорожки (0/1)	bicycle_parths
16	Шлагбаум (0/1)	yard_barrier	59	Кладовая (0/1)	storeroom
17	Ворота (0/1)	yard_gate	60	Материал окон	material_window
18	Охрана во дворе (0/1)	yard_security	61	Парко. велосип.(0/1)	parking_bicycle
19	Парковка (0/1)	parking	62	Спорт.площ. (0/1)	sports_ground
20	Подземная парк. (0/1)	subway_parking	63	Управл. теплом (0/1)	controls_heat
21	Полуподз. парк. (0/1)	semisubway_park	64	Электр. счетч. (0/1)	electricity_meters
22	Гаражная парк. (0/1)	garage_parking	65	Номер этажа	floor_num
23	Лифт (0/1)	elevator	66	Кол-во этажей	floors_cnt
24	Грузовой лифт (0/1)	cargo-elevator	67	Кол-во комнат	room_cnt
25	Колясочная (0/1)	kolyasochnaya	68	Год здания	building_year
26	Коммуникации (0/1)	communication	69	Тип квартиры	type
27	Счетчики воды (0/1)	water_counters	70	Раздел сан.узел (0/1)	bathroom
28	Счетчики тепла (0/1)	hot_counters	71	Плита (элект, газ)	plate
29	Домофон (0/1)	domofon	72	Тип балкона	balcony
30	Видеонаблю. (0/1)	video_monitoring	73	Расположение дома	house_location
31	Железная дверь (0/1)	iron_door	74	Комфорт. кварт.	rooms_comfort
32	Пожарная сигнал. (0/1)	fire_signal	75	Район	district
33	Охранная сигнал. (0/1)	safe_signal	76	Цена	Price
34	Консьерж (0/1)	consiery	77	Видеодомофон	videodomofon
35	Охрана (0/1)	security	78	Гостевая парковка	guest_parking
36	Год постройки	deadline_y			
37	Дата добавления	date_add			
38	Категория жилья	category			
39	Итогов проц. цены	total_price			
40	Широта	la			
41	Долгота	lo			
42	Сельс. местнос. (0/1)	countryside			
43	Этажность	living_floors			