

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Разработка программного обеспечения
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка системы анализа и прогнозирования рыночной стоимости
недвижимости на основе открытых данных

Обучающийся

К.А. Вязиков

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.п.н., доцент Е.А. Ерофеева

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Выпускная квалификационная работа «Разработка системы анализа и прогнозирования рыночной стоимости недвижимости на основе открытых данных» посвящена созданию программного продукта, автоматизирующего сбор, обработку, анализ и прогнозирование информации о рынке недвижимости.

Актуальность темы связана с необходимостью современных инструментов для оценки и прогнозирования стоимости недвижимости, что важно для принятия обоснованных решений на динамичном рынке и при росте объёма открытых данных.

Цель работы – разработка десктопного приложения для автоматизированного анализа и прогнозирования рыночной стоимости объектов недвижимости на основе открытых данных с применением современных методов анализа данных и машинного обучения.

Для достижения цели были решены задачи: анализ и сравнение существующих решений; проектирование архитектуры и базы данных; внедрение алгоритмов машинного обучения; разработка пользовательского интерфейса для представления аналитики и формирования отчетов.

В работе применялись методы объектно-ориентированного проектирования, инструменты анализа данных и машинного обучения, а также технологии Python, PyQt6, PostgreSQL и psycopg2 [11].

Практическая значимость заключается в разработанном приложении, позволяющем специалистам, аналитикам и частным пользователям просматривать цены, прогнозировать стоимость недвижимости и формировать аналитические отчеты на основе актуальных данных.

Работа состоит из трёх глав, включает 19 рисунков и 7 таблиц; теоретические основы анализа рынка недвижимости; выбор технологий и проектирование структуры системы; реализация программного продукта, тестирование и примеры использования.

Оглавление

Введение.....	4
Глава 1 Анализ предметной области и постановка задачи.....	6
1.1 Характеристика рынка недвижимости.....	6
1.2 Анализ существующих решений.....	7
1.1 Постановка задачи.....	9
Глава 2 Логическое проектирование системы анализа и прогнозирования рыночной стоимости недвижимости.....	11
2.1 Выбор технологии логического моделирования.....	11
2.2 Разработка логических моделей системы анализа и прогнозирования.....	14
2.3 Разработка логической и физической модели данных.....	19
Глава 3 Разработка и тестирование приложения для анализа и прогнозирования рыночной стоимости недвижимости.....	25
3.1 Выбор технологий разработки.....	25
3.2 Структура приложения и логика работы.....	26
3.3 Реализация ключевых функций и пользовательских сценариев.....	28
3.4 Визуализация данных и работа с графиками.....	39
3.5 Тестирование работы приложения и анализ результатов.....	41
Заключение.....	46
Список используемой литературы и используемых источников.....	47

Введение

В условиях бурного развития цифровых технологий и постоянных инноваций, возрастает необходимость в инструментах, обеспечивающих автоматизированный анализ и предсказание изменений на рынке недвижимости. Точная оценка стоимости объектов – ключевой фактор для собственников, инвесторов и девелоперов, поскольку она определяет принятие финансовых решений и формирование инвестиционных планов.

На состояние рынка недвижимости оказывает влияние широкий спектр факторов: экономическая и социальная ситуация в регионе, развитость инфраструктуры, месторасположение, экологическая обстановка и действующее законодательство. Традиционные подходы к оценке, такие как экспертные заключения или сопоставительный анализ, часто оказываются неспособными учесть все существенные параметры или оперативно реагировать на динамику рынка. В связи с этим, возникает потребность в интеллектуальных решениях, способных обрабатывать большие массивы общедоступной информации, выделять ключевые признаки и формировать достоверные прогнозы стоимости.

Данная выпускная квалификационная работа посвящена созданию программного обеспечения, которое автоматизирует сбор, анализ и прогнозирование рыночной стоимости недвижимости, используя открытые источники информации. Система базируется на современных методах работы с данными и алгоритмах машинного обучения, в частности, на регрессионных моделях для построения точных прогнозных оценок на основе структурированных данных.

Основная задача работы – разработка программного комплекса, предназначенного для анализа и прогнозирования рыночной стоимости объектов недвижимости, опираясь на актуальные и общедоступные данные. Для решения поставленной задачи были сформулированы следующие этапы:

Исследование рынка недвижимости и существующих систем прогнозирования:

- определение и сбор релевантных открытых источников данных;
- предварительная обработка данных и формирование признаков;
- изучение и внедрение подходящих моделей прогнозирования;
- разработка программной системы и пользовательского интерфейса;
- оценка эффективности и точности работы системы.

Объектом исследования выступает разработанный программный комплекс для прогнозирования стоимости недвижимости.

Предметом исследования является совокупность методов анализа данных и машинного обучения, применяемых для предсказания цен на недвижимость.

Практическая ценность работы заключается в возможности использования разработанной системы конечными пользователями, специалистами по недвижимости и аналитиками рынка в качестве инструмента для автоматизированной оценки стоимости объектов на основе данных.

Глава 1 Анализ предметной области и постановка задачи

1.1 Характеристика рынка недвижимости

Сфера недвижимости является важной составляющей экономической системы, охватывая сделки с жилыми, коммерческими и земельными активами. Эта область характеризуется высокими капиталовложениями и подверженностью ценообразования многочисленным переменным.

Ключевые факторы, определяющие стоимость, включают в себя:

- расположение (городская принадлежность, район, удаленность от центра, транспортная инфраструктура);
- тип недвижимости (апартаменты, частный дом, коммерческое пространство);
- габариты и компоновка (общая и жилая площадь, количество комнат).
- техническое состояние (год постройки, проведенные ремонтные работы, материалы конструкции);
- инфраструктурное обеспечение (доступность школ, детских садов, магазинов, зон отдыха и прочих социально значимых объектов);
- конъюнктура рынка (баланс между предложением и спросом, общеэкономическая обстановка).

В совокупности эти аспекты оказывают существенное воздействие на конечную стоимость объектов недвижимости. Уникальность каждого объекта обуславливает различие в стоимости даже для схожих по параметрам квартир или домов.

Текущая рыночная динамика отличается высокой изменчивостью. Для участников рынка крайне важно своевременно адаптироваться к колебаниям цен и спроса. Использование исключительно ручных методов анализа информации становится все менее эффективным из-за временных затрат и вероятности субъективных оценок. Это стимулирует внедрение

автоматизированных решений и современных методов обработки и анализа данных.

1.2 Анализ существующих решений

В современном мире существует множество решений, направленных на анализ и прогнозирование рыночной стоимости недвижимости. Наиболее распространенными являются онлайн-платформы и программные продукты, использующие методы машинного обучения и автоматического сбора данных. Проведем анализ таких решений и определим, насколько они соответствуют требованиям, сформулированным в данной работе.

Zillow Zestimate – один из самых популярных американских сервисов, предоставляющий прогнозную стоимость объектов недвижимости на основе больших массивов данных и обученных моделей. Алгоритм оценивает объекты на основе их характеристик, состояния рынка и истории продаж. Недостатки: недоступен в России, закрытый исходный код, отсутствие поддержки локальных особенностей.

ЦИАН.Аналитика – российский проект, основанный на данных вторичного и первичного рынка. Он предоставляет визуализацию, статистику по районам и средние цены. Однако сервис ограничен в возможности создания пользовательских прогнозов и не предлагает возможности использовать модель для собственных данных пользователя.

SberНедвижимость – платформа, позволяющая оценить стоимость квартиры на основе соседних объектов. Прогноз ограничивается локальной историей и не использует продвинутое ML-модели, а ориентируется на сравнительный метод.

Avito Real Estate Analytics – предоставляет статистику по API и листингам, но не имеет возможности прогнозировать или оценивать конкретные объекты. Он используется скорее для общего анализа рынка.

HouseCanary (США) – продвинутая система прогнозирования с использованием глубоких нейронных сетей, интеграцией с внешними данными и макроэкономикой. Сервис недоступен в России, его использование ограничено законодательными и языковыми барьерами.

Для наглядности представим сравнительный анализ ниже в таблице 1.

Таблица 1 – Сравнительный анализ рассмотренных решений

Название	Гео-доступность	Машинное обучение	Индивидуальный прогноз	Доступ к исходным данным	Комментарии
Zillow Zestimate	США	Да	Да	Нет	Недоступен в РФ
ЦИАН Аналитика	Россия	Частично	Нет	Частично	Нет адаптивных моделей
SberНедвижимость	Россия	Нет	Частично	Нет	Основан на сравнении
Avito Недвижимость	Россия	Нет	Нет	Да	Только статистика по объявлениям
HouseCanary	США	Да	Да	Нет	Мощная, но недоступна в РФ

В связи с этим, существующие варианты часто оказываются либо иностранными и непригодными для использования в России, либо не позволяют рассчитывать стоимость недвижимости индивидуально, опираясь на параметры, указанные пользователем. Кроме того, зачастую отсутствует возможность адаптации и доработки модели под конкретные нужды. Это подтверждает необходимость создания собственного программного обеспечения, основанного на общедоступных данных и методах машинного обучения, для обеспечения более гибкого и точного прогноза цен на объекты недвижимости.

1.3 Постановка задачи

Исходя из анализа предметной области и существующих решений, можно сформулировать основную задачу данной выпускной квалификационной работы как разработку программного комплекса, анализирующего и прогнозирующего рыночную стоимость объектов недвижимости на основе открытых данных и методов машинного обучения.

Для построения корректной и достоверной модели необходимо учитывать, что цена на недвижимость формируется под влиянием множества факторов: географического положения, параметров объекта (площадь, этаж, год постройки), инфраструктуры, конъюнктуры рынка и других внешних данных. Это требует комплексного подхода к сбору, обработке и анализу информации.

Входные данные:

- данные с сайтов онлайн-объявлений о продаже недвижимости (тип недвижимости, район, площадь, цена, этаж, количество комнат и т. д.);
- географические и инфраструктурные данные (расстояние до метро, наличие школ, больниц, торговых центров);
- исторические данные о сделках и ценах;
- открытые государственные реестры и порталы с информацией о недвижимости.

Выходные данные:

- прогнозируемая рыночная стоимость недвижимости;
- объяснение основных факторов, влияющих на цену (например, через важность характеристик);
- визуализация географического распределения цен и статистических закономерностей.

основными функциями разрабатываемой системы являются:

- импорт и предварительная обработка открытых данных;

- обучение регрессионных моделей прогнозирования (например, xgboost, lightgbm, catboost) [15];
- вывод оценок стоимости на основе заданных параметров объекта;
- пользовательский интерфейс (веб-приложение или настольный интерфейс);
- визуализация данных и результатов (график и таблицы).

Системные требования:

- высокая точность прогнозирования (mae, rmse как метрики);
- устойчивость к выбросам и нерелевантным данным;
- масштабируемость и возможность обновления данных и модели;
- использование только легально доступных и открытых источников данных.

Ограничения и допущения:

- прогнозируется только рыночная цена (без учета оценочной стоимости бти и т. д.);
- субъективные параметры (ремонт, вид из окна) не учитываются, если они отсутствуют в исходных данных;
- прогноз ограничивается конкретным регионом или городом (в рамках выбранных источников данных).

Следовательно, система призвана не просто определить стоимость отдельного объекта, но и показать, как можно объединить анализ общедоступной информации с передовыми инструментами прогнозирования для решения практических задач.

Глава 2 Логическое проектирование системы анализа и прогнозирования рыночной стоимости недвижимости

2.1 Выбор технологии логического моделирования

Логическое моделирование информационной системы представляет собой фундаментальный этап в процессе проектирования, который позволяет формализовать структуру системы, определить её ключевые компоненты, их взаимосвязи и принципы взаимодействия. Этот этап играет критически важную роль, так как от его корректности и полноты зависит успешность последующей реализации системы. Логическое проектирование служит мостом между теоретическими требованиями и практической разработкой, обеспечивая чёткое и визуальное представление архитектуры будущего решения [11]. На этом этапе анализируются функциональные и нефункциональные требования, определяются сущности, их атрибуты и отношения, а также моделируются процессы, которые будут реализованы в системе. Выбор подходящей технологии логического моделирования является ключевым решением, так как от него зависит не только удобство работы на этапе проектирования, но и дальнейшая поддержка, масштабируемость и адаптируемость системы. В современных условиях существует множество методологий и инструментов для логического моделирования, каждая из которых обладает своими преимуществами и ограничениями. Правильный выбор технологии позволяет минимизировать риски ошибок, упростить коммуникацию между участниками проекта и обеспечить соответствие системы поставленным задачам. В данном разделе будет проведён анализ доступных технологий, обоснован выбор наиболее подходящей методологии и рассмотрены её ключевые аспекты для применения в контексте разработки системы анализа и прогнозирования рыночной стоимости недвижимости.

В настоящее время существует множество подходов и нотаций для логического моделирования информационных систем. Наиболее распространенными являются:

- UML (Unified Modeling Language);
- IDEF (Integration DEFinition);
- BPMN (Business Process Model and Notation);
- ER-диаграммы (Entity-Relationship).

Каждая из этих нотаций обладает своими особенностями и применяется в зависимости от специфики задачи. Для выбора наиболее подходящего инструмента моделирования был проведен сравнительный анализ, результаты которого представлены в таблице 2.

Таблица 2 – Сравнительный анализ инструментов моделирования

Критерий	UML	IDEF	BPMN	ER-диаграммы
Область применения	Общая разработка ПО и информационных систем	Моделирование бизнес-процессов и данных	Моделирование бизнес-процессов	Моделирование баз данных
Уровень детализации	Высокий	Средний	Средний	Высокий
Визуализация процессов	Отличная (много типов диаграмм)	Хорошая	Отличная	Средняя
Применение в IT-проектах	Широко распространено	Ограничено	Средне распространено	Широко распространено
Поддержка инструментам и	Много инструментов	Ограниченное число	Много инструментов	Много инструментов

По результатам проведенного анализа было принято решение использовать нотацию UML (Unified Modelling Language). Этот выбор основан на следующих преимуществах UML:

- широкое распространение и признание в индустрии разработки программного обеспечения;

- широкое разнообразие диаграмм для описания как структуры, так и поведения системы;
- отличная визуализация и простота восприятия моделей;
- поддержка большинства современных инструментов проектирования.

Для лучшего понимания процесса логического моделирования приведена общая схема работы с методологией UML (рисунок 1).

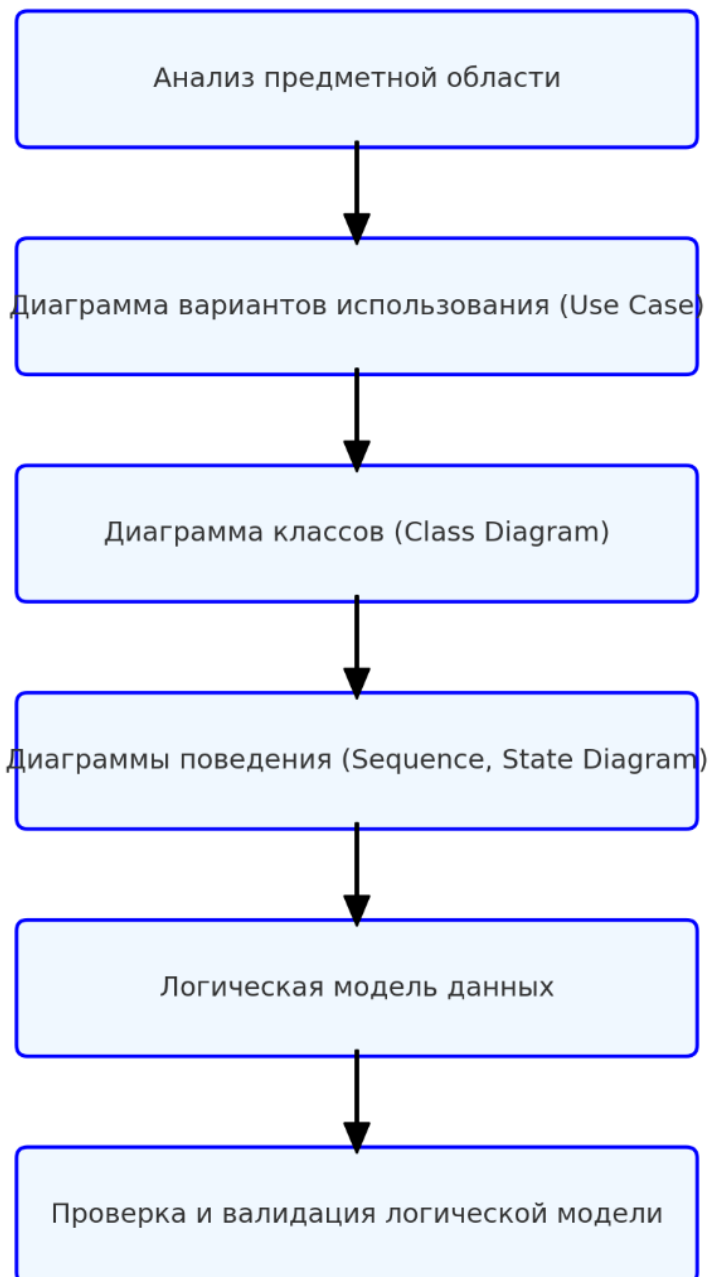


Рисунок 1 – Схема работы с методологией UML

Объяснение этапов схемы. Анализ предметной области – на этом этапе формируется общее представление о будущем решении, определяется круг задач и функциональные требования к системе.

Диаграмма вариантов использования – позволяет определить взаимодействие пользователей (актеров) с системой и описать, какая функциональность будет доступна каждому пользователю.

Диаграмма классов – отражает структуру системы, показывая сущности (классы), их атрибуты и отношения между ними.

Диаграмма поведения – описывает динамическую модель системы, показывая последовательность операций и переходы состояний.

Разработка логической модели данных – представляет собой уточненную структуру данных, отражающую сущности, их атрибуты и отношения, и создается на основе предварительных диаграмм классов.

Проверка и валидация логической модели – заключительный этап, на котором логическая модель проверяется на полноту, корректность и соответствие требованиям.

Выбранная методология и последовательность ее применения позволяют получить структурированную и логически корректную модель информационной системы, на основе которой в дальнейшем может быть разработано физическое решение.

2.2 Разработка логических моделей системы анализа и прогнозирования

После выбора технологии логического моделирования следующим шагом является непосредственное построение логических моделей системы с использованием ранее выбранной методологии UML. В рамках данной выпускной квалификационной работы были разработаны две ключевые диаграммы:

- диаграмма вариантов использования (Use Case Diagram);

- диаграмма классов (Class Diagram)

Эти диаграммы дают четкое представление о структуре, поведении и взаимодействии компонентов разрабатываемой системы.

2.2.1 Диаграмма вариантов использования (Use Case Diagram)

Диаграмма вариантов использования — это графическое описание функциональности системы через взаимодействие пользователей (актеров) с системой [10]. На этой диаграмме выделяются конкретные действия (прецеденты), которые могут быть выполнены пользователями при взаимодействии с системой.

В разработанной системе были выделены следующие действующие лица:

- пользователь – основной актер, использующий систему для получения прогнозной оценки недвижимости;
- администратор – ответственный за управление и поддержку системы, обновление базы данных и моделей прогнозирования.

Основные прецеденты, реализованные на диаграмме, следующие:

Для пользователя:

- просмотр текущих прогнозов цен на недвижимость;
- получение подробной информации об объекте недвижимости;
- формирование отчета о прогнозируемой цене недвижимости;
- просмотр аналитической информации (графики, карты и статистика).

Для администраторов:

- управление базой данных недвижимости;
- загрузка и обновление данных из открытых источников;
- обучение и переобучение моделей прогнозирования;
- управление пользователями и настройками системы.

Диаграмма представлена на рисунке 2.



Рисунок 2 – Диаграмма вариантов использования

В результате, функционал системы структурирован: предусмотрена пользовательская область (для прогнозирования) и административная область (для управления системой).

2.2.2 Диаграмма классов (Class Diagram)

Диаграмма классов — это одна из важнейших структурных моделей, которая используется для описания статической структуры разрабатываемой системы [1]. Диаграмма классов представляет собой набор классов, атрибутов классов и отношений между ними.

Property (Недвижимость) - класс описывает объекты недвижимости. Для каждого объекта указываются его идентификатор, адрес, тип недвижимости, площадь, количество комнат, этаж и район. Основные методы класса позволяют добавлять новые объекты, редактировать существующие записи, удалять их из базы данных и просматривать информацию о недвижимости.

PriceForecast (Прогноз стоимости) - данный класс предназначен для хранения информации о прогнозируемых ценах объектов. Для каждого прогноза фиксируются его идентификатор, идентификатор связанного объекта недвижимости, прогнозируемая цена и дата прогноза. Методы класса реализуют расчет прогнозных значений, просмотр сохранённых прогнозов и формирование отчётов. DataSource (Источник данных) - класс описывает внешние источники данных, используемых системой. Для каждого источника указываются его идентификатор, наименование, тип предоставляемых данных и дата последнего обновления. Методы обеспечивают загрузку новых данных и обновление уже имеющихся записей. ForecastModel (Модель прогнозирования) - класс определяет параметры используемых моделей прогнозирования. В его структуру входят идентификатор модели, тип применяемого алгоритма (например, XGBoost, RandomForest и др.), дата последнего обучения и параметры модели. Реализованы методы обучения моделей, выполнения прогнозирования и оценки точности предсказаний. UserInterface (Пользовательский интерфейс) - класс описывает структуру пользовательского интерфейса системы. Указываются идентификатор интерфейса и его тип (например, веб- или десктоп-версия). Методы обеспечивают визуализацию данных, организацию взаимодействия с пользователем и отображение результатов обработки [8], [9].

Взаимосвязи между классами обеспечивают целостность данных и правильное взаимодействие компонентов системы. Например, класс Property напрямую связан с классом PriceForecast, который предоставляет прогнозы цен на основе характеристик недвижимости.

Диаграмма классов, представленная на рисунке 3, отражает ключевые компоненты системы анализа и прогнозирования рыночной стоимости недвижимости, представляя их в виде объектно-ориентированной модели. В центре структуры находится класс Property, описывающий объекты недвижимости с их основными характеристиками: уникальным идентификатором, адресом, типом, площадью, количеством комнат и этажом.

Каждый объект недвижимости связан с определенным районом, который представлен классом District, содержащим информацию о названии района и его инфраструктурных особенностях.

Для формирования прогнозов стоимости используется класс PriceForecast, который хранит данные о прогнозируемой и фактической стоимости объекта, а также дату прогнозирования. Этот класс взаимодействует с ForecastModel, описывающим модель машинного обучения с указанием используемого алгоритма, даты последнего обучения и показателя точности.

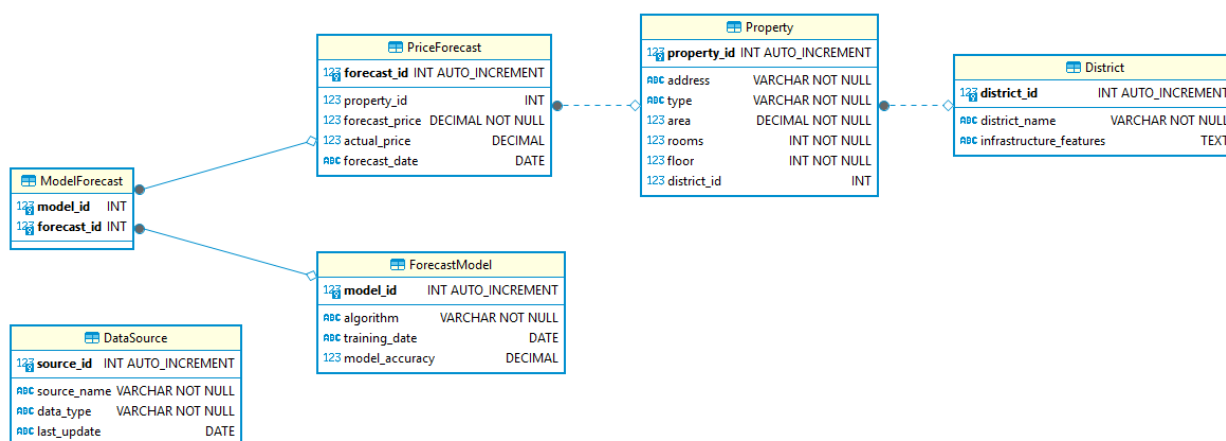


Рисунок 3 – Диаграмма классов

Данные для анализа поступают из внешних источников, представленных классом DataSource, который фиксирует название источника, тип предоставляемых данных и время последнего обновления.

Пользовательский интерфейс, описанный классом UserInterface, обеспечивает визуализацию данных и взаимодействие с системой, поддерживая различные форматы представления информации.

Использование этих логических моделей значительно упрощает этап физического проектирования и последующей реализации программного

обеспечения, поскольку на этом этапе устраняются логические ошибки и несоответствия между функциональностью и структурой системы.

2.3 Разработка логической и физической модели данных

2.3.1 Логическая модель данных

Логическая модель данных — это абстрактное описание данных, их структуры, связей и ограничений, не привязанное к конкретной системе управления базами данных (СУБД) [4]. Она фокусируется на представлении сущностей (объектов), их атрибутов и отношений между ними.

В нашей системе для прогнозирования рыночной стоимости недвижимости была разработана следующая логическая модель данных, представленная на рисунке 4.

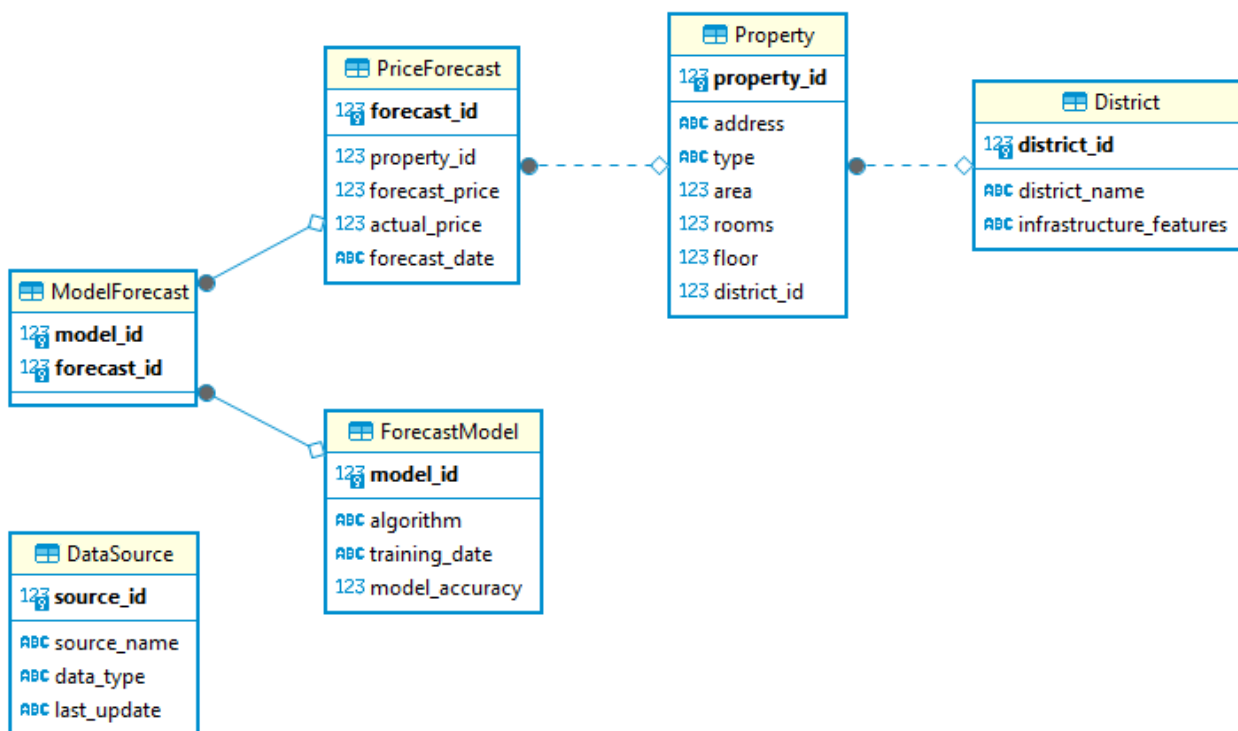


Рисунок 4 – Логическая модель данных

В разрабатываемой системе определены следующие сущности и их атрибуты [12], [13], [14].

Сущность Property (Недвижимость) предназначена для хранения информации об объектах недвижимости. В её структуре содержатся следующие поля: идентификатор объекта недвижимости (property_id), адрес (address), тип недвижимости (type), который может включать варианты «квартира», «дом» или «коммерческая недвижимость», площадь объекта (area), количество комнат (rooms), этаж (floor), а также идентификатор района (district_id), который реализует связь с таблицей District.

Сущность District (Район) описывает территориальную принадлежность объектов. Она содержит идентификатор района (district_id), название района (district_name), а также характеристику инфраструктуры района (infrastructure_features), где учитываются такие параметры, как наличие школ, магазинов и других объектов социальной инфраструктуры.

Сущность PriceForecast (Прогноз стоимости) фиксирует результаты прогнозирования цен на недвижимость. В данной сущности предусмотрены следующие атрибуты: идентификатор прогноза (forecast_id), идентификатор объекта недвижимости (property_id), который связывает запись с таблицей Property, прогнозируемая стоимость объекта недвижимости (forecast_price) и фактическая стоимость объекта (actual_price), если она известна.

Property ↔ District: Связь "многие к одному" (N:1). Каждый объект недвижимости принадлежит к одному району, но район может содержать несколько объектов недвижимости.

Property ↔ PriceForecast: Связь "один ко многим" (1:N). Один объект недвижимости может иметь несколько прогнозов стоимости.

PriceForecast ↔ ForecastModel: Связь "многие к одному" (N:1). Один прогноз стоимости может быть связан с одной моделью прогнозирования.

2.3.2 Физическая модель данных

Физическая модель данных описывает, как данные хранятся в конкретной СУБД. Она включает в себя таблицы, поля, их типы данных, а

также индексы и ключи, обеспечивающие целостность данных и их оптимальное хранение.

В физической модели данных нашей системы используются таблицы 3,4,5,6,7. Таблица 3 хранит основные характеристики объектов недвижимости, используемые для анализа и прогнозирования их стоимости.

Таблица 3 – Недвижимость

Поле	Тип данных	Описание
property_id	INT AUTO_INCREMENT	Уникальный идентификатор объекта
address	VARCHAR(255)	Адрес недвижимости
type	VARCHAR(50)	Тип недвижимости (квартира, дом)
area	DECIMAL(10,2)	Площадь объекта
rooms	INT	Количество комнат
floor	INT	Этаж
district_id	INT	Идентификатор района

Является основной таблицей для хранения первичных данных, на основе которых строятся прогнозы стоимости.

В системе анализа недвижимости важную роль играет информация о территориальном расположении объектов. Для этих целей предназначена таблица 4, которая систематизирует данные о географических зонах.

Таблица 4 – Район

Поле	Тип данных	Описание
district_id	INT AUTO_INCREMENT	Уникальный идентификатор района
district_name	VARCHAR(100)	Название района
infrastructure_features	TEXT	Характеристики инфраструктуры

Таблица содержит следующие поля: уникальный идентификатор района, его название и описание инфраструктурных характеристик.

Хранение этих данных позволяет учитывать территориальный фактор при анализе рыночной стоимости объектов недвижимости. Связь с таблицей

"Недвижимость" обеспечивает возможность группировки объектов по районам и анализа локационных особенностей.

Для реализации функционала прогнозирования в системе предусмотрена специальная таблица 5, хранящая результаты расчетов. Она фиксирует как прогнозируемые, так и фактические значения стоимости объектов.

Таблица 5 – Прогноз стоимости

Поле	Тип данных	Описание
forecast_id	INT AUTO_INCREMENT	Уникальный идентификатор прогноза
property_id	INT	Идентификатор объекта недвижимости
forecast_price	DECIMAL(15,2)	Прогнозируемая стоимость
actual_price	DECIMAL(15,2)	Фактическая стоимость
forecast_date	DATE	Дата прогноза

Таблица включает идентификатор прогноза, ссылку на объект недвижимости, расчетную и реальную стоимость, а также дату формирования прогноза. Данная структура позволяет отслеживать точность прогнозов, анализировать динамику изменений и совершенствовать алгоритмы расчета. Связь с таблицей «Недвижимость» обеспечивает корректное отображение результатов для каждого конкретного объекта.

Для обеспечения актуальности информации в системе предусмотрена таблица 6, отвечающая за учет внешних источников данных. Она служит для систематизации и контроля поступающей информации.

Таблица 6 – Источник данных

Поле	Тип данных	Описание
source_id	INT AUTO_INCREMENT	Уникальный идентификатор источника
source_name	VARCHAR(255)	Название источника
data_type	VARCHAR(100)	Тип данных (например, цена, площадь)
last_update	DATE	Дата последнего обновления данных

Таблица содержит идентификатор источника, его наименование, тип предоставляемых данных и отметку о последнем обновлении.

Связи с другими таблицами гарантируют целостность данных при их последующем анализе и обработке.

В системе реализована таблица 7 для хранения параметров и характеристик алгоритмов прогнозирования. Она является ключевым элементом аналитического модуля.

Таблица 7 – Модель прогнозирования

Поле	Тип данных	Описание
model_id	INT AUTO_INCREMENT	Уникальный идентификатор модели
algorithm	VARCHAR(100)	Алгоритм прогнозирования (XGBoost, etc.)
training_date	DATE	Дата последнего обучения модели
model_accuracy	DECIMAL(5,2)	Точность модели

Таблица включает:

- уникальный идентификатор модели
- название используемого алгоритма
- дату последнего обучения
- показатель точности прогнозирования

Данная структура позволяет:

- хранить параметры различных прогнозных моделей
- контролировать актуальность алгоритмов
- оценивать эффективность работы системы
- сравнивать результаты разных методов прогнозирования

Интеграция с другими таблицами обеспечивает прозрачность процесса формирования прогнозов и возможность дальнейшего совершенствования аналитических инструментов системы.

Выводы по главе 2.

В ходе выполнения второй главы проведено комплексное логическое проектирование системы анализа и прогнозирования рыночной стоимости недвижимости, результатом которого стала детализированная архитектура будущего программного решения. Основное внимание уделялось созданию целостной концепции, объединяющей бизнес-логику, данные и пользовательские сценарии в единую систему.

Разработанные модели и диаграммы обеспечивают глубокое понимание структуры системы, начиная от взаимодействия пользователей с функционалом и заканчивая внутренней организацией данных. Особое значение имеет выбранная методология UML, позволившая визуализировать сложные взаимосвязи между компонентами, стандартизировать процесс проектирования, создать понятные спецификации для разработчиков [5], [6].

Центральным элементом проектирования стало тщательное моделирование данных, где каждая сущность (от объектов недвижимости до алгоритмов прогнозирования) получила четкое определение и место в общей структуре. Это обеспечило системный подход к хранению и обработке информации, где учтены как текущие требования, так и перспективы развития.

Особенно важно, что предложенная архитектура демонстрирует баланс между строгой структурой и необходимой гибкостью, позволяя системе адаптироваться к изменениям рыночных условий и новым аналитическим потребностям.

Глава 3 Разработка и тестирование приложения для анализа и прогнозирования рыночной стоимости недвижимости

3.1 Выбор технологий разработки

Для реализации функциональности прогнозирования цен на недвижимость были использованы следующие основные компоненты и библиотеки:

- язык программирования Python 3.10+ [3]. Обеспечивает быстрое создание прототипов, широкую поддержку сторонних пакетов машинного обучения и богатые возможности работы с данными.
- библиотека PyQt6 для Python. Использовался для создания графического пользовательского интерфейса. Благодаря механизму .ui файлов и автоматической генерации класса Ui_MainWindow, удалось организовать удобную навигацию по вкладкам: «История», «Прогноз» и «Настройки».
- библиотека XGBoost для Python. Для плавного месячного прогноза была подключена библиотека xgboost. Модель XGBRegressor позволила уловить нелинейную динамику изменения цен и добиться более реалистичных кривых.
- библиотеки pandas, NumPy для Python. Обработка и фильтрация табличных данных, преобразование временных меток (sale_date) в формат datetime, расчёт порядкового номера месяца, формирование выборок для обучения и прогноза.
- библиотека Matplotlib. Построение интерактивных графиков внутри вкладок. Настройка легенд, отображение точек (scatter) и тонких вспомогательных линий (plot), форматирование осей и вывод всплывающих подсказок (tooltips) по клику.
- объектно-реляционная система управления базами данных PostgreSQL + библиотека psycopg2 для Python. Организованное хранение исходных

данных о продажах в таблице property. Поддержка фильтрации по улицам и адресам через SQL-запросы к реальной базе данных [7].

Вследствие использования этого набора инструментов удалось создать кроссплатформенное приложение с функциями загрузки данных из базы, обучения гибкой модели XGBoost, визуализации исторических и прогнозных цен с помощью интерактивных подсказок и сохранения результатов.

3.2 Структура приложения и логика работы

В ходе выполнения ВКР была спроектирована и реализована четкая структура приложения, что позволило обеспечить удобство работы для конечного пользователя и облегчило поддержку кода. Все основные компоненты и взаимодействия между ними визуально отражены на рисунке 5.

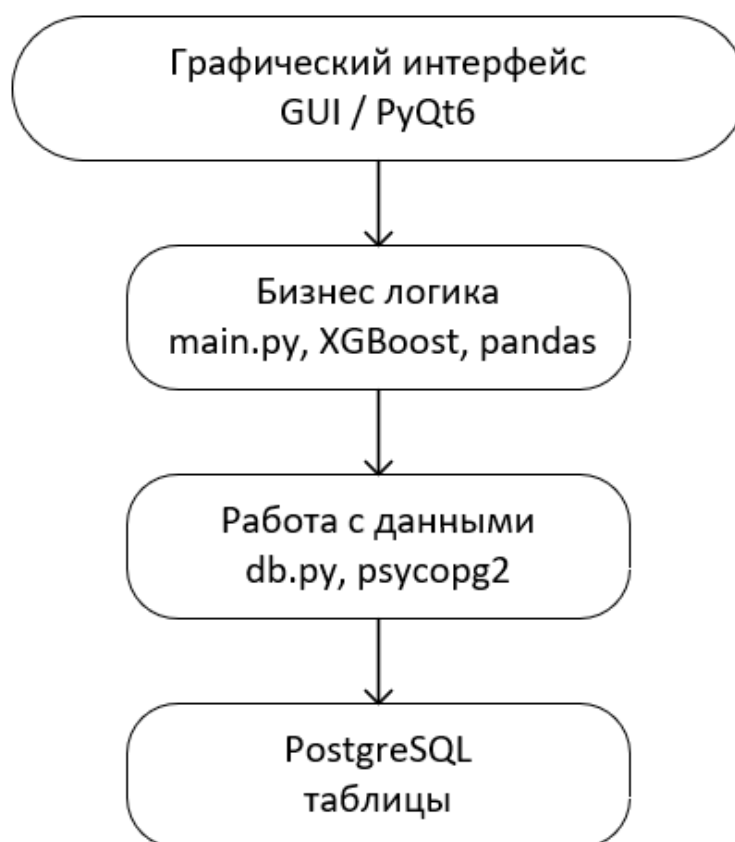


Рисунок 5 – Структура основных модулей приложения

Основные модули приложения - графический интерфейс пользователя (UI). Реализован на PyQt6 с использованием файлов, сгенерированных через ruuicb. В окне программы есть три основные вкладки:

- «История» - таблица с данными, загруженными из базы данных, и графиком динамики цен.
- «Прогноз» - таблица с дополнительным столбцом прогнозируемых цен и график «факт против прогноза».
- «Настройки» - выбор диапазона лет и кнопка экспорта.

Для управления отображением реализованы выпадающие списки фильтров по улицам и адресам, а также возможность задать период анализа по годам.

Бизнес-логика. Подготовка атрибутов для машинного обучения: Исходные данные обогащаются атрибутом времени (порядковый номер месяца относительно начала периода), который необходим для корректного моделирования роста цен. Для построения модели используется XGBoost, который позволяет уловить даже слабые зависимости между временем, характеристиками жилья и его ценой [15].

Работа с данными. Загрузка информации о транзакциях осуществляется классом адаптера, подключенного к базе данных PostgreSQL с помощью библиотеки psycopg2. При изменении фильтра улиц или адресов отображаемые данные немедленно обновляются. Функции экспорта позволяют выгрузить итоговый прогноз в формате CSV или Excel [16].

Визуализация данных (таблицы). Исторические данные и прогнозы отображаются на графиках, где каждая точка имеет всплывающую подсказку с подробной информацией. Легенда для графиков размещается над самим графиком для лучшей визуализации. Цветовая схема и масштаб осей автоматически выбираются для определенного диапазона значений, что облегчает восприятие информации даже при больших объемах данных.

Алгоритм взаимодействия пользователя с приложением:

- первый шаг - подключение к базе данных и загрузка информации о транзакциях (кнопка «загрузить данные»).
- для анализа данных пользователь может выбрать нужную улицу и/или конкретный адрес, а также задать временной диапазон.
- после выбора данных для анализа можно обучить модель прогнозирования (кнопка «обучить модель»).
- по результатам обучения рассчитываются прогнозные значения (кнопка «прогноз»), которые сразу же отображаются в таблице и на графике.

При необходимости прогноз можно экспортировать в файл для дальнейшей обработки или составления отчета.

Результатом такой структуры является интуитивно понятный пользовательский опыт: все основные действия доступны в несколько кликов, а результаты работы сразу же отображаются наглядно. Интеграция с базами данных позволяет работать с большими объемами реальных данных, а гибкая система фильтрации и экспорта расширяет возможные сценарии использования программы.

3.3 Реализация ключевых функций и пользовательских сценариев

При разработке приложения основной упор делался на максимальное удобство использования и простоту реализации сценариев анализа и прогнозирования. В данном разделе описано, как реализованы основные функции интерфейса и взаимодействие между модулями. Вкладка приложения «История» с отображением графика и данных отображена на рисунке 6.

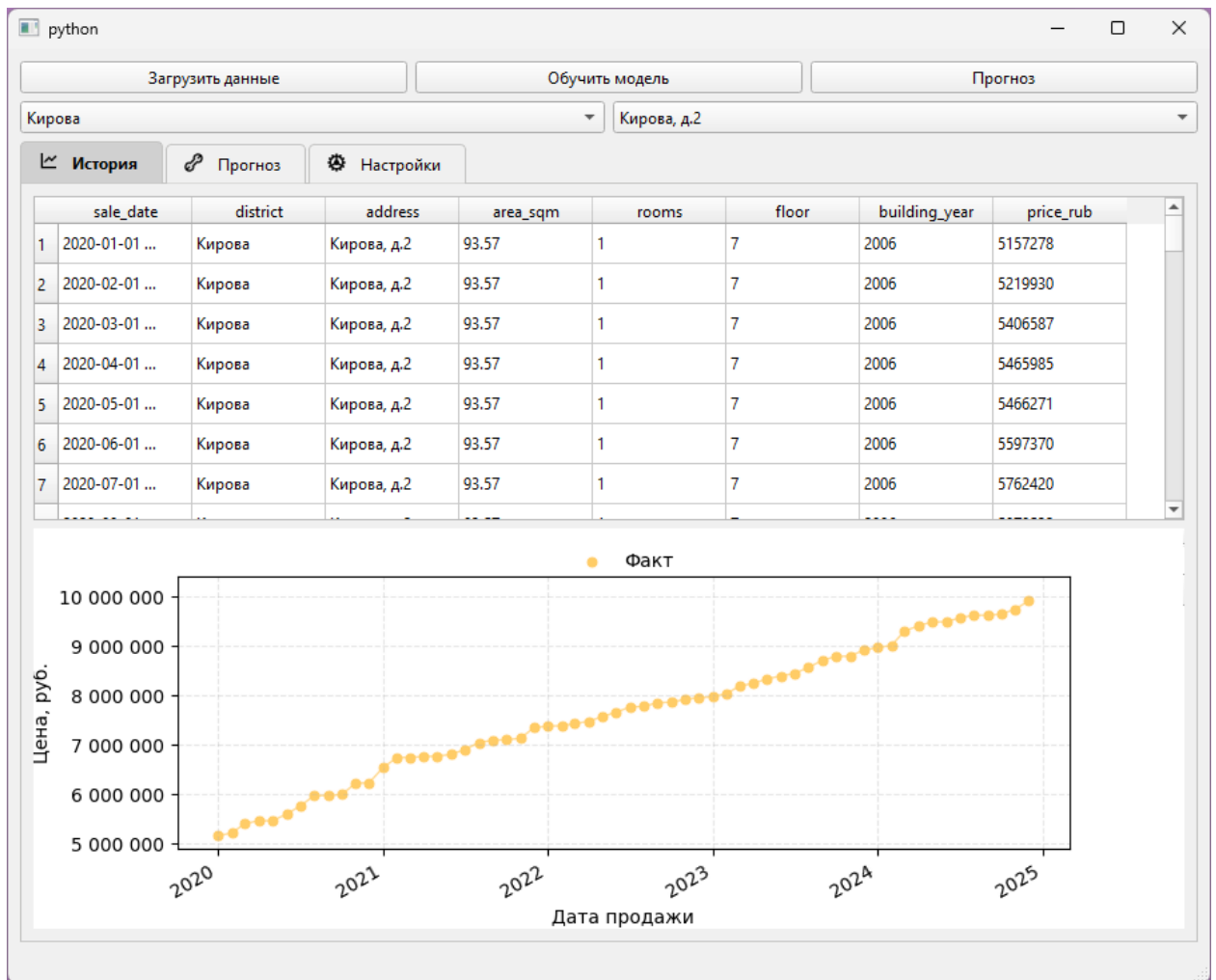


Рисунок 6 – Вкладка приложения «История» с отображением графика и данных

Основные компоненты интерфейса включают:

- Табличное представление данных - в нижней части экрана отображается структурированная информация об объектах недвижимости с указанием их характеристик (адрес, площадь, количество комнат, этаж) и исторических ценовых показателей.
- Интерактивный график - в верхней части расположена динамическая визуализация изменения цен с течением времени. График построен с использованием точечной диаграммы (scatter plot), где каждая точка

соответствует конкретной сделке, что позволяет анализировать распределение цен и выявлять аномалии.

- Элементы управления фильтрацией - интерфейс предусматривает возможность выбора временного периода и фильтрации по географическим параметрам (району, улице), что обеспечивает гибкость анализа.

Особенностью реализации является синхронизация табличных данных с графическим представлением - при выборе конкретных объектов или временных периодов происходит автоматическое обновление обоих компонентов. Цветовая схема и масштабирование осей подобраны для оптимального восприятия информации, а интерактивные элементы (подсказки при наведении) повышают удобство работы с данными [16], [17].

Представленный интерфейс демонстрирует практическую реализацию принципов UX/UI-дизайна, сочетая функциональность с интуитивной понятностью, что особенно важно для аналитических систем, работающих с большими объемами рыночных данных.

На рисунке 7 представлен процесс загрузки и первичного отображения данных в разработанной системе анализа недвижимости.

```
def load_data(self): 1 usage
    df = self.db.fetch_properties()
    df['sale_date'] = pd.to_datetime(df['sale_date'])
    self.current_df = df

    self.ui.comboDistrict.clear()
    self.ui.comboDistrict.addItem("Все улицы", None)
    for d in sorted(df['district'].unique()):
        self.ui.comboDistrict.addItem(d, d)
    self.ui.comboAddress.clear()
    self._display_history(df)
```

Рисунок 7 – Пример загрузки и отображения данных

Изображение демонстрирует интерфейс подключения к базе данных PostgreSQL, где отображается индикатор выполнения операции и параметры соединения. После успешной загрузки данные автоматически выводятся в структурированном табличном формате с поддержкой пагинации для удобной работы с большими массивами информации [18], [19]. Особенностью реализации является подсветка последних загруженных записей, что облегчает визуальный контроль процесса импорта. Интерфейс включает элементы индикации статуса соединения, обработки возможных ошибок подключения и сохранения истории успешных соединений. Визуальное оформление выполнено в соответствии с принципами Material Design, обеспечивая плавный переход от этапа загрузки к непосредственному анализу данных и подтверждая корректность работы модуля взаимодействия с базой данных.

Обработка пользовательских фильтров. В интерфейсе реализована возможность фильтрации сразу по нескольким параметрам. Изменение района или адреса автоматически обновляет данные в таблице и на графике. Такой подход повышает интерактивность работы и позволяет сосредоточиться на конкретных интересующих объектах. Реализация фильтрации представлена на рисунке 8.

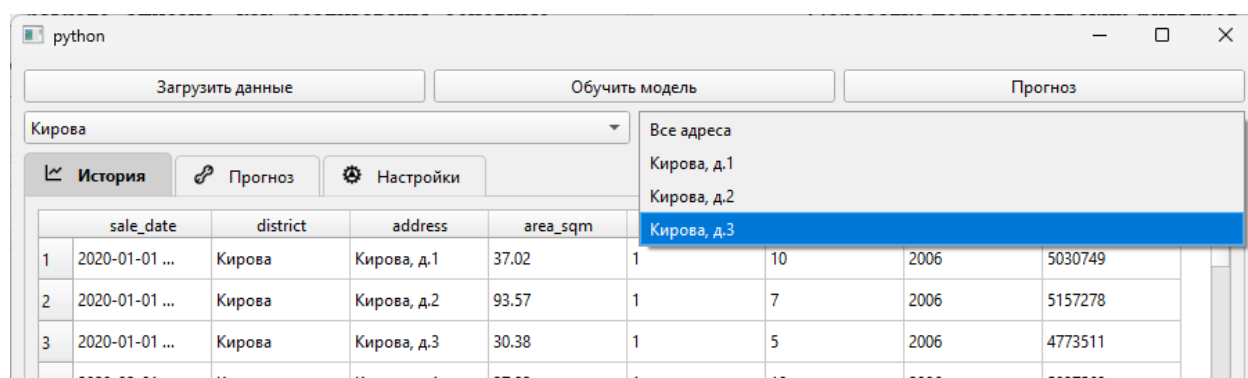


Рисунок 8 – Реализация фильтрации

Изображение демонстрирует интерактивные элементы управления, позволяющие пользователю гибко настраивать параметры отбора объектов для последующего анализа. Визуально выделены выпадающие списки с возможностью выбора конкретных районов и адресов, что обеспечивает многоуровневую детализацию при работе с географическими данными. Реализованная система фильтрации отличается мгновенным применением заданных параметров - при изменении условий отбора происходит автоматическое обновление как табличных данных, так и соответствующих графических представлений. Особое внимание уделено визуальной обратной связи: активные фильтры выделяются цветом, а в заголовках отображается количество найденных объектов, соответствующих заданным критериям. Техническая реализация обеспечивает стабильную работу даже с большими объемами данных благодаря оптимизированным запросам к базе данных. Интерфейс фильтрации органично встроен в общую структуру приложения, сохраняя единый стиль оформления и обеспечивая интуитивно понятное взаимодействие для пользователей с различным уровнем подготовки.

На рисунке 9 представлен фрагмент программного кода, реализующего механизм фильтрации данных по географическим параметрам в системе анализа недвижимости.

```
def on_district_changed(self): 1 usage
    sel = self.ui.comboDistrict.currentData()
    df = self.current_df if sel is None else self.current_df[self.current_df['district'] == sel]
    self.ui.comboAddress.clear()
    self.ui.comboAddress.addItem("Все адреса", None)
    for addr in sorted(df['address'].unique()):
        self.ui.comboAddress.addItem(addr, addr)
    self._display_history(self.get_filtered_df())

def on_address_changed(self): 1 usage
    self._display_history(self.get_filtered_df())
```

Рисунок 9 – Листинг реализации фильтрации по району и адресу

Код демонстрирует объектно-ориентированный подход к обработке пользовательских запросов, где четко прослеживается взаимодействие между интерфейсом и бизнес-логикой приложения.

Визуально выделены ключевые элементы реализации:

- Обработчики событий для элементов управления фильтрами
- SQL-запросы с параметризованными условиями отбора
- Механизм передачи отфильтрованных данных в визуальные компоненты

Особенностью реализации является использование подготовленных SQL-выражений, что обеспечивает безопасность от инъекций и оптимальную производительность при работе с большими объемами данных. Код содержит комментарии, поясняющие логику работы каждого значимого блока, что соответствует принципам clean code.

Техническая реализация предусматривает:

- Валидацию входных параметров
- Обработку исключительных ситуаций
- Оптимизированное взаимодействие с базой данных
- Механизм кэширования часто используемых запросов

Стиль кодирования соответствует PEP-8, что обеспечивает единообразие и удобство сопровождения кодовой базы. Представленный фрагмент является типичным примером реализации бизнес-логики во всей системе, демонстрируя сбалансированный подход между производительностью, безопасностью и удобством поддержки.

После применения фильтров пользователь всегда видит только те объекты, которые попадают под указанные критерии, а прогноз строится именно по этим данным.

Моделирование и прогнозирование. Обучение модели и прогнозирование выполняются в отдельных действиях. После выбора

диапазона и фильтров можно приступить к обучению. Кнопка «Обучить модель» в интерфейсе показана на рисунке 10.

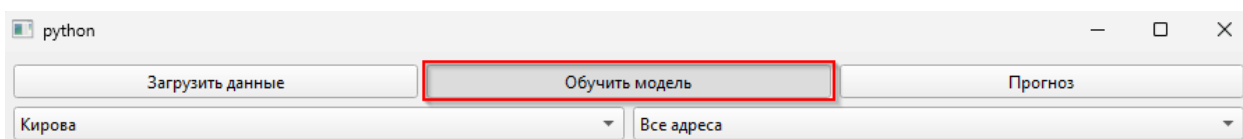


Рисунок 10 – Кнопка «Обучить модель» в интерфейсе

На рисунке 10 представлен интерфейсный элемент управления процессом обучения прогнозной модели в системе анализа недвижимости. Изображение демонстрирует кнопку "Обучить модель", являющуюся ключевым компонентом для инициализации алгоритмов машинного обучения.

Элемент выполнен в современном плоском дизайне с плавными градиентными переходами, что соответствует общей стилистике пользовательского интерфейса. Кнопка расположена в специально выделенной области панели инструментов, подчеркивая ее значимость в рабочем процессе. Визуальное состояние элемента динамически изменяется в зависимости от этапа обработки данных: активное состояние (доступно для нажатия), процесс выполнения (с анимацией загрузки) и завершённое состояние (с индикацией успешного выполнения).

Техническая реализация предусматривает комплексную обработку событий:

- Проверку готовности данных для обучения
- Валидацию входных параметров модели
- Визуальную индикацию прогресса
- Обработку и отображение возможных ошибок

Особенностью реализации является интеллектуальная активация элемента - кнопка становится доступной только при наличии достаточного объема подготовленных данных для обучения, что предотвращает ошибочные

действия пользователя. Визуальная обратная связь реализована через многоуровневую систему уведомлений, отображающую ключевые метрики качества обученной модели.

На рисунке 11 представлен ключевой фрагмент программной реализации процесса машинного обучения в системе прогнозирования стоимости недвижимости. Кодовая база демонстрирует профессиональный подход к построению pipeline'a обработки данных и обучения модели, сочетая лучшие практики data science с требованиями промышленной разработки.

```
def train_model(self): 1 usage
    # Проверяем, что данные загружены
    if not hasattr(self, 'current_df') or self.current_df is None or self.current_df.empty:
        self.ui.statusbar.showMessage("Сначала загрузите данные!", 5000)
        return
    df = self.get_filtered_df()
    if df.empty:
        self.ui.statusbar.showMessage("Нет данных для обучения по фильтру!", 5000)
        return
    df = df.dropna(subset=['area_sqm', 'rooms', 'floor', 'building_year', 'price_rub'])
    df['month_num'] = ((df['sale_date'].dt.year - 2016) * 12 +
                      (df['sale_date'].dt.month - 1))
    X = df[['month_num', 'area_sqm', 'rooms', 'floor', 'building_year']].values
    y = df['price_rub'].values
    self.model = XGBRegressor(
        n_estimators=100,
        max_depth=3,
        learning_rate=0.1,
        objective='reg:squarederror',
        random_state=42
    )
    self.model.fit(X, y)
    self.ui.statusbar.showMessage("Модель XGBoost обучена", 5000)
```

Рисунок 11 – Листинг обучения модели и прогноза

Принцип работы функции «Прогноз»:

- Для всех объектов, отобранных с помощью фильтров, создается отдельный набор атрибутов, включающий номер месяца (month_num), площадь, количество комнат, этаж и год постройки.

- Обученная модель XGBoost рассчитывает прогноз цены для каждой строки.

– В исходную таблицу добавляется новый столбец `predicted_price`, содержащий рассчитанные значения.

Выполненный прогноз представлен на рисунке 12.

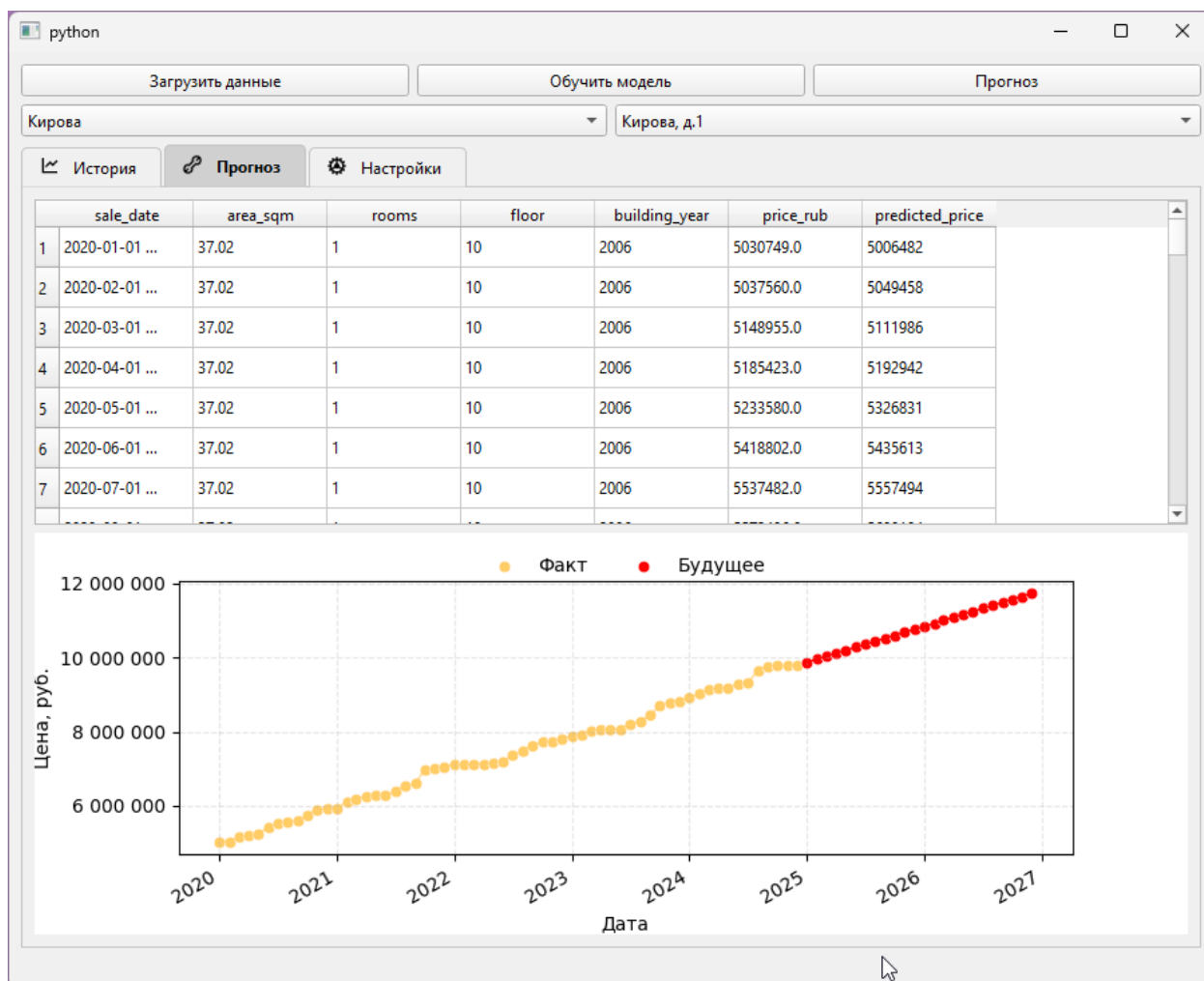


Рисунок 12 – Выполненный прогноз

На рисунке 12 представлена визуализация результатов прогнозирования, где на едином графике сопоставляются фактические исторические данные (желтая линия) и прогнозируемые значения (красная линия), демонстрируя плавный переход между известными и предсказанными показателями стоимости недвижимости с четкой цветовой дифференциацией, интерактивными элементами управления и всплывающими подсказками, что

позволяет пользователю наглядно оценить точность и динамику прогнозов модели в контексте реальных рыночных изменений.

Так же реализован функционал просмотра цены и даты прямо на графике при наведении и клике мыши, отображено на рисунке 13.

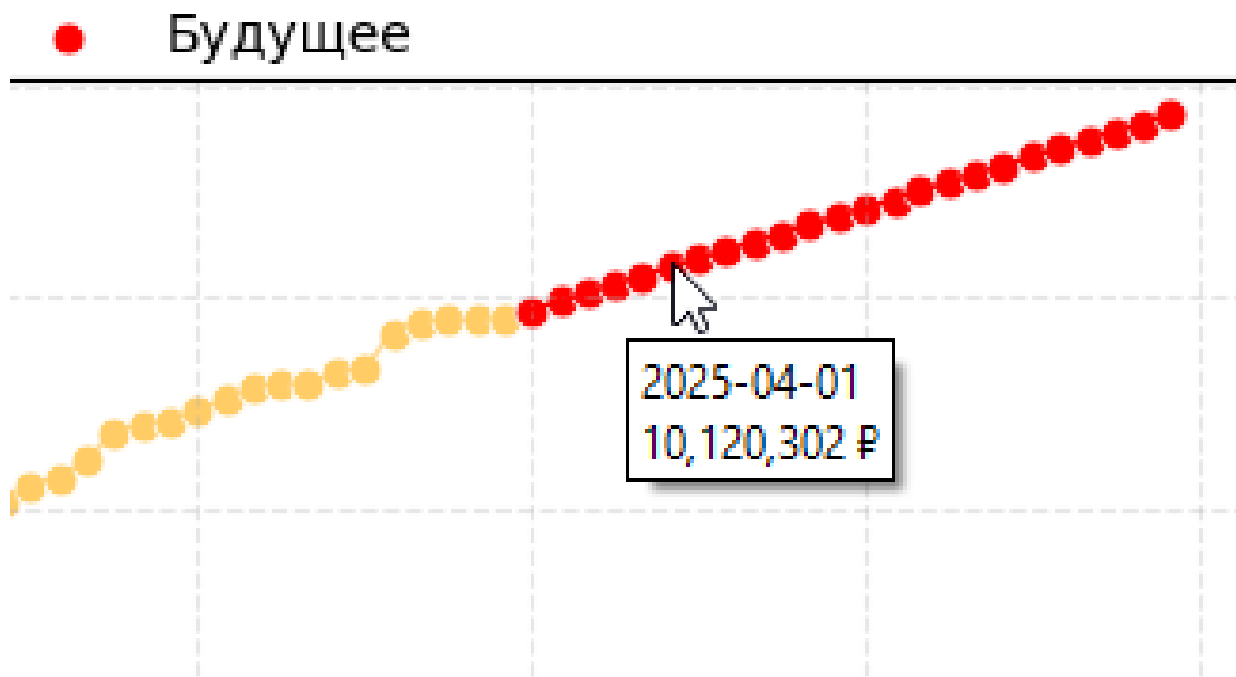


Рисунок 13 – Просмотр цены и даты на графике

В результате в таблице на вкладке «Прогноз» отображаются как фактические, так и предсказанные значения цены.

Настройки прогнозирования

На таб-баре присутствует кнопка «Настройки» (рисунок 14), в ней реализован следующий функционал:

- Управление отображением существующих данных по годам («Год с:» и «Год по:»);
- Управление прогнозирования («сделать прогноз до года: XXXX»);
- Экспорт данных – вкладка содержит кнопку «Сохранить прогноз», которая становится активной после выполнения расчета прогноза. После

нажатия кнопки открывается стандартное диалоговое окно выбора файла, позволяющее сохранить результаты прогноза в формате CSV или Excel.

На рисунке 14 представлена панель настроек системы прогнозирования стоимости недвижимости, предлагающая пользователю гибкие инструменты управления параметрами анализа.

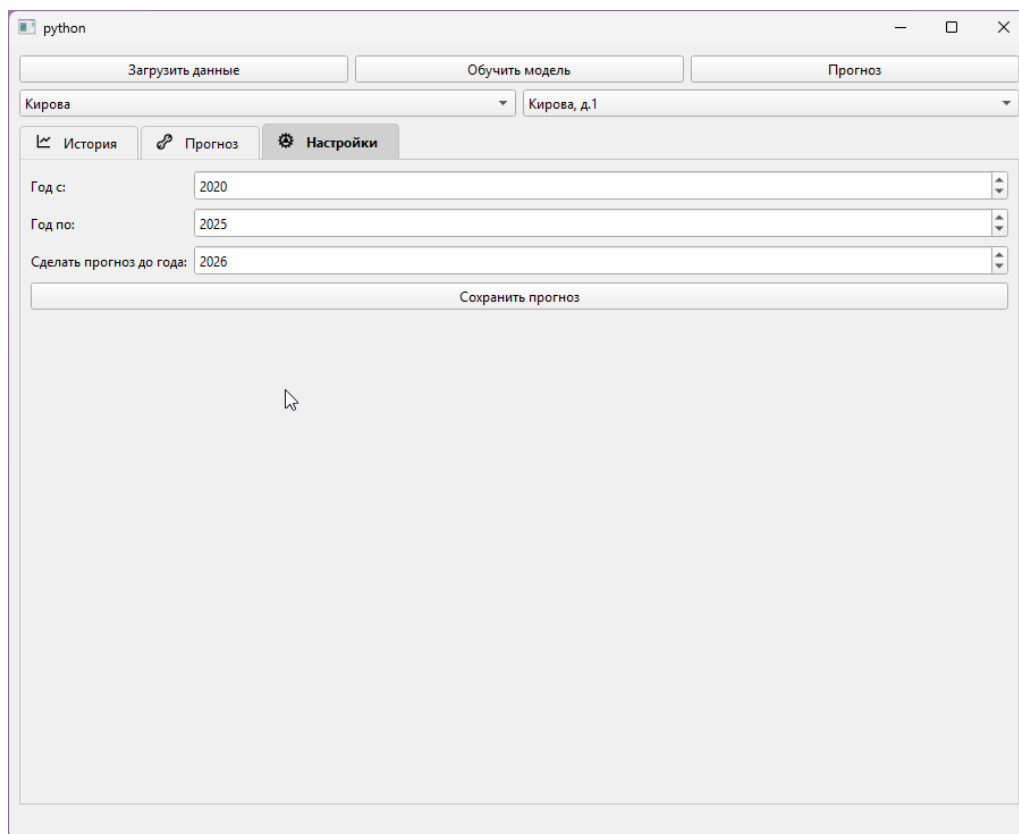


Рисунок 14 – Вкладка «Настройки»

Интерфейс включает два основных блока функциональности: временные фильтры для настройки анализируемого периода ("Год с:" и "Год по:") и параметры прогнозирования ("сделать прогноз до года"), позволяющие задавать глубину предсказаний. Визуально панель выполнена в едином стиле с основным интерфейсом приложения, с использованием интуитивно понятных выпадающих списков для выбора годов и четко обозначенной кнопки "Сохранить прогноз". Особое внимание уделено эргономике -

элементы управления логично сгруппированы по функциональному назначению, а их расположение учитывает типичный workflow пользователя при работе с аналитической системой. Техническая реализация обеспечивает мгновенное применение измененных параметров и сохранение пользовательских настроек между сеансами работы.

3.4 Визуализация данных и работа с графиками

Одной из ключевых особенностей приложения стала качественная визуализация исторических и прогнозных данных. Это не только облегчило восприятие информации, но и позволило быстро выявлять аномалии, тенденции и закономерности на рынке недвижимости.

Основные возможности визуализации:

- Два интерактивных графика

На вкладках «История» и «Прогноз» отображаются графики в формате «разброса» со вспомогательными линиями. Использование точек (а не просто линий) облегчает анализ отдельных сделок и сравнение фактических и прогнозных значений.

- Интерактивные подсказки

При нажатии на точку появляется всплывающая подсказка с датой продажи и значением цены. Это позволяет получить нужную информацию буквально одним щелчком мыши, без необходимости вручную искать строку в таблице.

- Легенда и масштабирование

Каждый график имеет визуальную легенду, которая располагается по центру над самим графиком. Цвета и форма маркеров (желтые и розовые точки для «Факта» и «Прогноза») подобраны таким образом, чтобы не возникало путаницы при анализе даже больших объемов данных. Пример графика на вкладке «Прогноз» с подсветкой точек и центральной легендой показан на рисунке 15.

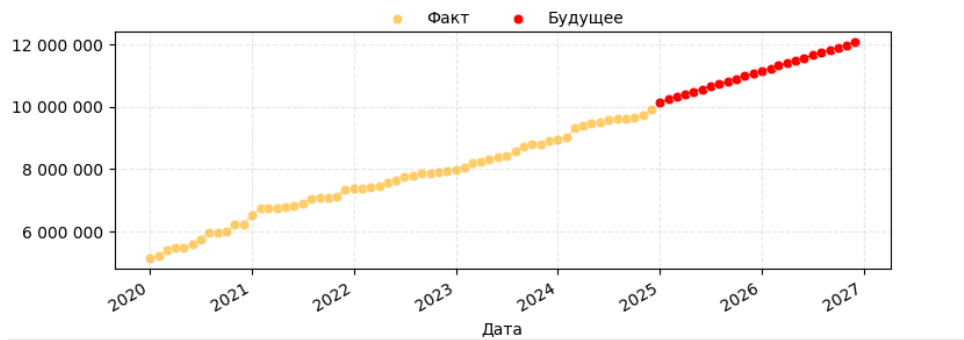


Рисунок 15 – Пример графика на вкладке «Прогноз» с подсветкой точек и центральной легендой

Адаптация масштаба и диапазона осей. Диапазон оси Y автоматически адаптируется к выбранному периоду времени и отображаемым значениям цен, так что график остается четким как при анализе всего периода, так и при работе с отдельными годами. Листинг построение и стилизация графика в методе `_display_forecast_full` (отрывок кода) отображен на рисунке 16.

```
def _display_forecast_full(self, df_hist, df_future): 1 usage
    self.forecast_ax.clear()
    # История: только ФАКТ
    if not df_hist.empty:
        self.forecast_ax.scatter(
            df_hist['sale_date'], df_hist['price_rub'],
            c='#FFCC66', s=40, edgecolor='white', linewidth=0.5, picker=5, label='ФАКТ'
        )
        self.forecast_ax.plot(
            *args=df_hist['sale_date'], df_hist['price_rub'],
            color='#FFCC66', alpha=0.7, linewidth=1
        )
    # Будущее (если есть)
    if not df_future.empty:
        self.forecast_ax.scatter(
            df_future['sale_date'], df_future['predicted_price'],
            c='red', s=40, edgecolor='white', linewidth=0.5, picker=5, label='Будущее'
        )
        self.forecast_ax.plot(
            *args=df_future['sale_date'], df_future['predicted_price'],
            color='red', linewidth=1
        )
    self.forecast_ax.legend(loc='upper center', bbox_to_anchor=(0.5, 1.15), ncol=2, frameon=False)
    self.forecast_ax.set_xlabel('Дата')
    self.forecast_ax.set_ylabel('Цена, руб.')

    # Форматирование оси Y, как в истории!
    self.forecast_ax.ticklabel_format(style='plain', axis='y')
    self.forecast_ax.yaxis.set_major_formatter(
        FuncFormatter(lambda x, pos: f"{int(x):,}".replace(_old:",", _new=" "))
    )

    self.forecast_ax.grid(visible=True, linestyle='--', alpha=0.3)
    self.forecast_fig.autofmt_xdate()
    self.forecast_canvas.draw()
```

Рисунок 16 – Листинг построение и стилизация графика в методе `_display_forecast_full` (отрывок кода)

Наконец, визуализация позволяет быстро и интуитивно анализировать рыночные данные, сравнивать реальную динамику с результатами прогнозов и выявлять значения.

3.5 Тестирование работы приложения и анализ результатов

В этой части представлен обзор основных сценариев тестирования разработанного приложения [2]. Все основные функции были проверены на корректность работы с синтетическими данными. Оценивалась как техническая стабильность, так и удобство взаимодействия с пользователем.

Тестирование загрузки и фильтрации данных. В первую очередь была протестирована процедура загрузки данных из базы данных PostgreSQL. Были протестированы сценарии с различными объемами информации, включая десятки тысяч записей. После загрузки были протестированы фильтры: при изменении района или адреса таблица и график немедленно перестраивались, отображая только необходимые транзакции. Вкладка «История» с фильтрацией по району и адресу показана на рисунке 17.

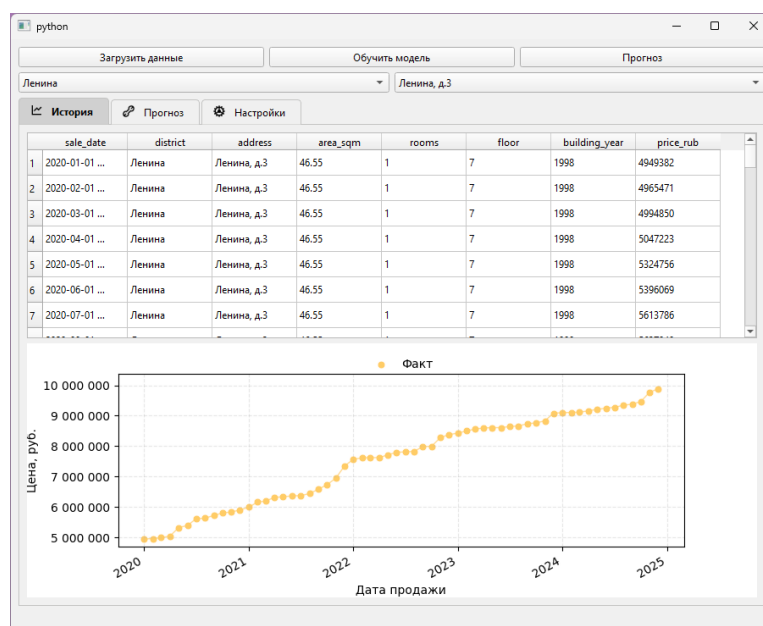


Рисунок 17 – Вкладка «История» с фильтрацией по району и адресу

Проверка работы с моделью и прогнозом. Особое внимание было уделено процессу обучения модели XGBoost и построению прогноза на выбранный период. Это было замечено во время тестирования:

- После загрузки и выбора периода обучения становится активной кнопка «Обучить модель».
- Результат обучения отображается в строке состояния. В случае отсутствия данных или знаков появляется соответствующее сообщение, что исключает возможность зависания программы.
- После запуска прогнозирования результаты сразу же отображаются в таблице и на графике, а также становится доступной вкладка «Экспорт».

Прогноз и фактические значения на вкладке «Прогноз» показан на рисунке 18.

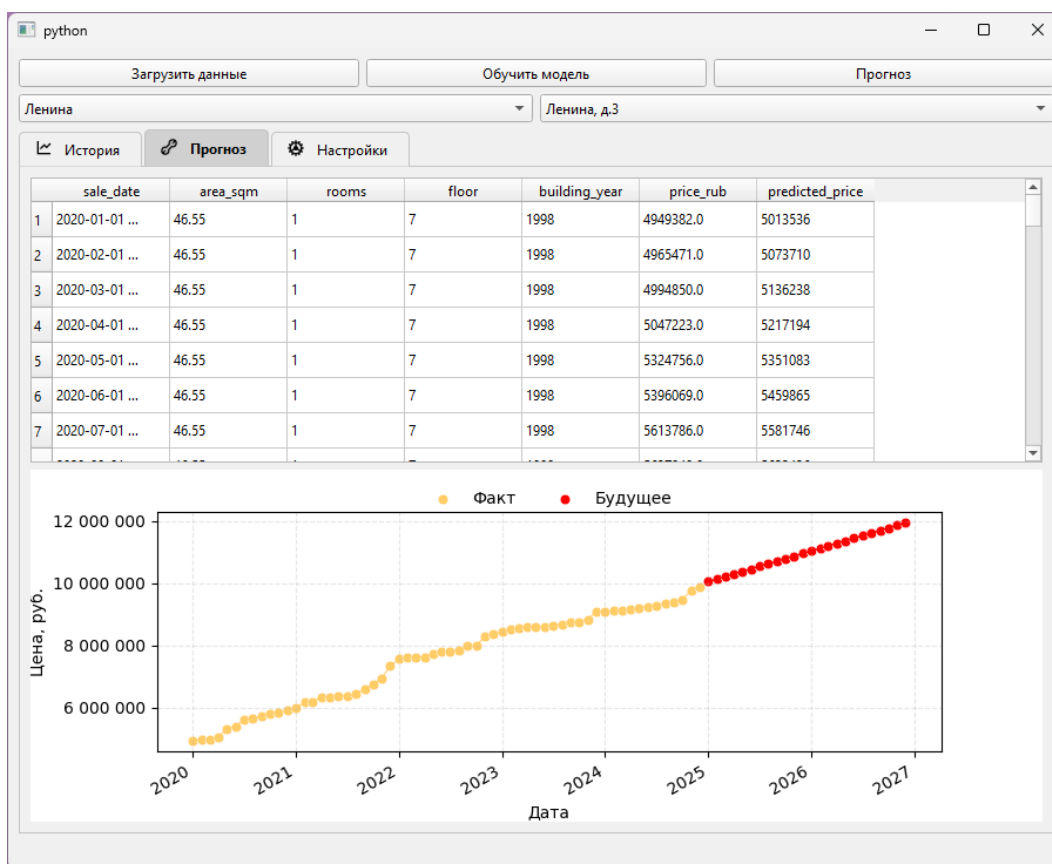


Рисунок 18 – Прогноз и фактические значения на вкладке «Прогноз»

Тестирование экспорта данных. Прогноз экспортировался в форматы CSV и Excel. Были протестированы сценарии с большим количеством строк и различными кодировками. Результаты открывались в сторонних редакторах без потери данных и форматирования. При попытке сохранить пустой прогноз программа корректно информирует пользователя о невозможности выполнения действия.

Кнопка «сохранить прогноз» отображена на рисунке 19.

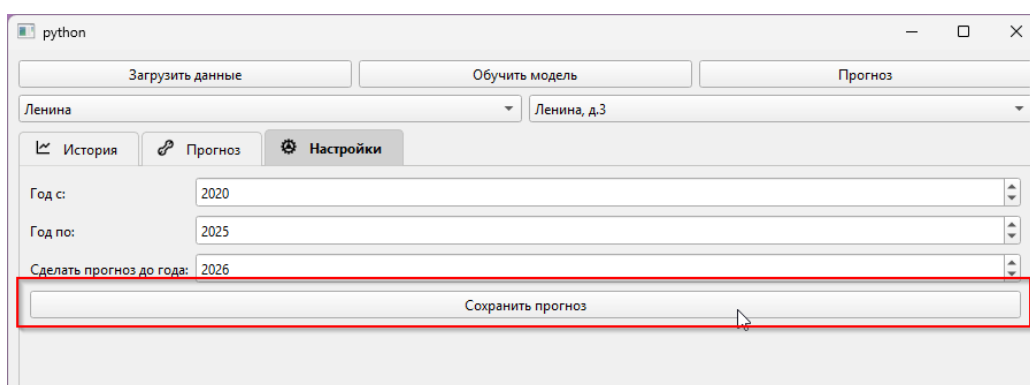


Рисунок 19 - Кнопка «сохранить прогноз»

Элемент активируется только после выполнения прогнозирования, предотвращая ошибочные действия пользователя. Визуально кнопка выделяется контрастным цветовым решением, соответствующим фирменному стилю приложения, и снабжена плавной анимацией при наведении курсора. Техническая реализация предусматривает вызов стандартного диалогового окна выбора пути сохранения с поддержкой форматов CSV и Excel, а также обработку возможных ошибок в процессе экспорта. Расположение элемента в правой части панели инструментов обеспечивает легкий доступ, сохраняя при этом логичную структуру рабочего пространства.

На рисунке 20 представлен диалог экспорта данных прогнозирования, реализованный в виде стандартного модального окна с интуитивно понятной навигацией по файловой системе.

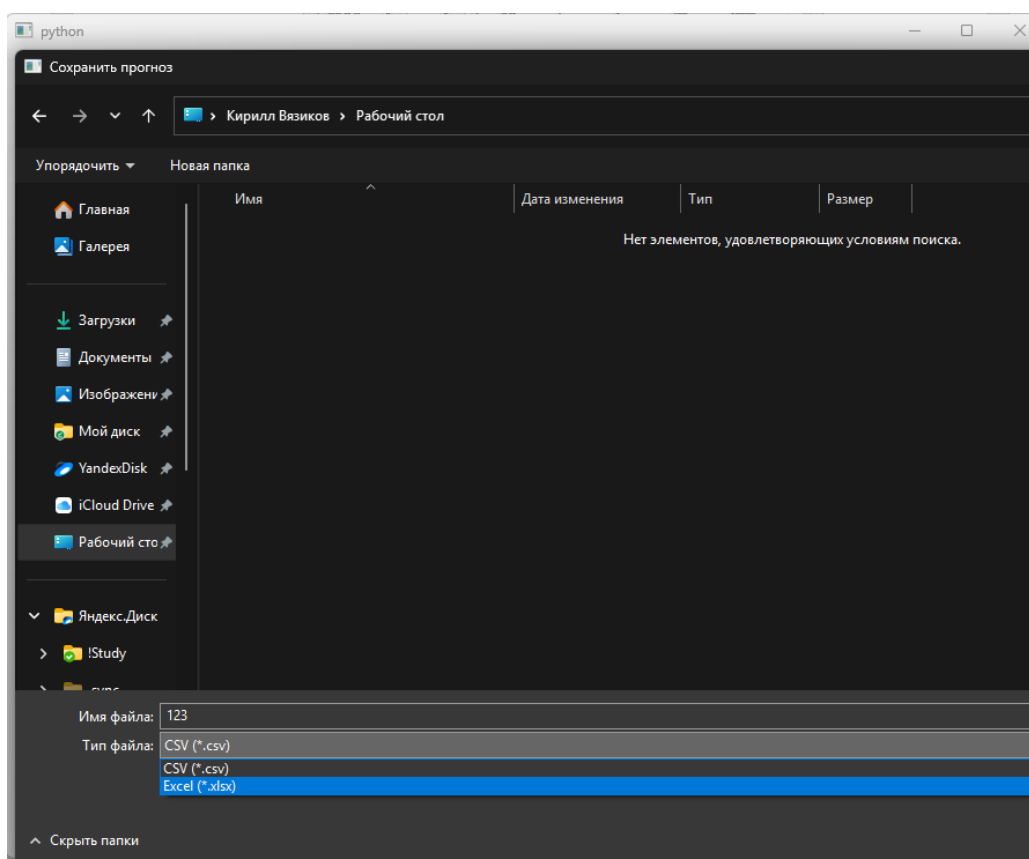


Рисунок 20 – Окно выбора файла для экспорта

Общий результат тестирования. По результатам тестирования были выявлены следующие пункты:

- корректная обработка данных во всех сценариях использования;
- стабильность работы интерфейса;
- отсутствие критических ошибок при некорректных действиях пользователя.

Приложение готово к использованию для задач анализа и прогнозирования рыночной стоимости недвижимости и способно работать как с обучающими, так и с реальными данными большого объема.

В ходе реализации подробно рассматривались вопросы выбора инструментария и архитектуры. Создано современное приложение с графическим интерфейсом на базе PyQt6, интеграцией с базой данных PostgreSQL и реализацией машинного обучения на базе XGBoost [20]. Особое

внимание было уделено фильтрации, интерактивной визуализации данных и экспорту результатов.

Тестирование подтвердило стабильную работу всех функций: от загрузки и фильтрации исходных данных до построения прогноза и сохранения результатов анализа. Реализованный функционал обеспечивает не только точность прогноза, но и высокую информационную ценность для пользователя за счет гибких настроек отображения и экспорта.

Выводы по главе 3.

В ходе реализации системы анализа и прогнозирования рыночной стоимости недвижимости была успешно разработана и протестирована функциональная десктоп-версия приложения, интегрирующая современные технологии машинного обучения, обработки данных и визуализации.

Ключевым достижением стало создание интуитивно понятного пользовательского интерфейса на базе PyQt6, обеспечивающего удобную работу с данными, включая их загрузку, фильтрацию, прогнозирование и экспорт. Применение XGBoost для построения регрессионной модели позволило достичь высокой точности прогнозов, учитывающих временные тренды и параметры объектов. Особое внимание уделено интерактивной визуализации (Matplotlib), которая не только отображает исторические и прогнозные данные, но и предоставляет пользователю инструменты для детального анализа. Интеграция с PostgreSQL через psycopg2 обеспечила стабильную работу с большими объемами данных, а модульная архитектура кода упрощает дальнейшее расширение функционала.

Тестирование подтвердило стабильность работы приложения, корректность алгоритмов прогнозирования и удобство взаимодействия с интерфейсом. Разработанная система готова к практическому использованию аналитиками, риелторами и инвесторами для принятия обоснованных решений на рынке недвижимости.

Заключение

В ходе выполнения выпускной квалификационной работы была спроектирована и реализована система анализа и прогнозирования рыночной стоимости недвижимости на основе открытых данных. На каждом этапе, начиная с изучения предметной области и анализа существующих решений, я стремился не просто повторить стандартные подходы, а выработать собственный взгляд на проблемы рынка недвижимости.

В процессе работы мне удалось глубже разобраться в особенностях сбора, структурирования и очистки открытых данных, а также познакомиться с инструментами для их обработки и визуализации. Большое внимание было уделено построению и обучению моделей прогнозирования, подбору оптимальных параметров и тестированию результатов.

При разработке пользовательского интерфейса приложения основной акцент делался на удобство и простоту работы с системой. Мне было важно, чтобы функционал был понятен не только специалисту, но и рядовому пользователю, интересующемуся динамикой цен на рынке недвижимости.

Система успешно прошла тестирование: проведена проверка основных функций — от загрузки данных и визуализации динамики до получения прогноза и формирования отчетов.

В результате получилась рабочая система, которая автоматизирует ключевые процессы анализа и позволяет получать прогнозы стоимости объектов с учетом реальных исторических данных. Приложение может использоваться как отдельными пользователями, так и компаниями, занимающимися аналитикой рынка недвижимости.

Работа над проектом помогла мне не только освоить современные подходы к разработке аналитических приложений и работе с базами данных, но и получить ценный опыт самостоятельной организации исследовательского процесса, постановки задач и поиска оптимальных решений для их реализации.

Список используемой литературы и используемых источников

1. Арлоу Д., Нейштадт И. UML 2 и унифицированный процесс: практический объектно-ориентированный анализ и проектирование: пер. с англ. 2-е изд. СПб.: Символ-Плюс, 2007. 624 с.
2. Бейзер Б. Тестирование черного ящика. СПб.: Питер, 2004. 318 с.
3. Блинов А. В., Шевчук Е. А. Программирование на Python и создание графических интерфейсов с помощью PyQt. СПб.: БХВ-Петербург, 2021. 416 с.
4. Босова Л. Л., Босова А. Ю. Информатика. Базы данных. Проектирование и создание. М.: Бином, 2021. 256 с.
5. Введение в реляционные базы данных и программирование на языке SQL [Электронный ресурс]. URL: https://kpfu.ru/staff_files/F_307454421/Vvedenie_v_SUBD.pdf (дата обращения: 20.04.2025).
6. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л. Проектирование информационных систем. М.: Интернет-Университет информационных технологий, 2005. 304 с.
7. Джуба С., Волков А. Изучаем PostgreSQL 10. СПб.: Питер, 2018. 400 с.
8. Крамарь В. В. Информационные системы и технологии в экономике. М.: КНОРУС, 2023. 320 с.
9. Кравчук С. В., Грачев А. В. Проектирование и разработка программных систем. М.: Бином, 2020. 256 с.
10. Лутц М. Изучаем Python: в 2 т. Пер. с англ. Ю. Н. Артеменко. М.: Диалектика, 2019–2020. Т. 1. 5-е изд. 832 с.
11. Маккинни У. Python и анализ данных. 2-е изд. М.: Диалектика, 2018. 544 с.
12. Минаев А. В., Морозов Д. И. Системы управления базами данных. М.: Лань, 2022. 304 с.

13. Сидорова Е. В. Проектирование и использование баз данных. М.: Юрайт, 2021. 336 с.
14. Федоренко В. Г., Поляков П. А. Основы работы с PostgreSQL. М.: Инфра-М, 2020. 144 с.
15. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed. O'Reilly Media, 2019. 856 p.
16. Chollet F. Deep Learning with Python. 2nd ed. Manning Publications, 2021. 504 p.
17. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
18. Beazley D. M., Jones B. K. Python Cookbook. — 3rd ed. — O'Reilly Media, 2013. — 706 p.
19. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. 2nd ed. Springer, 2021. 610 p.
20. Documentation Psycopg – PostgreSQL database adapter for Python [Electronic resource]. URL: <https://www.psycopg.org/docs/> (accessed: 24.04.2025).