ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А.Н. Тихонова

Рожин Андрей Константинович, группа БИТ 212

Оценка аренды квартиры

Курсовой проект
по дисциплине «Алгоритмизация и программирование»

студента образовательной программы бакалавриата
«Инфокоммуникационные технологии и системы связи»

Студент _____ А.К. Рожин

Научный руководитель
к.т.н., доцент
И. В. Назаров

_____

Москва 2022

Аннотация

Разработана система полного цикла оценивания стоимости аренды квартиры в городе Москве. В частности - создан парсер данных, скрипт предобработки, написаны алгоритмы машинного обучения (без использования готовых решений от сторонних разработчиков), исследованы зависимости в данных, проведен EDA, обучена и полностью готова к использованию система машинного обучения.

Благодаря возможностям API Telegram, создана оболочка взаимодействия конечного пользователя со всеми вышеперечисленными компонентами системы.

В открытом доступе существуют похожие решения, но они не отвечают высоким стандартам удобства использования, точности прогнозов и масштабируемости системы.

Annotation

A system of a full cycle assessment of the cost of renting an apartment in Moscow has been developed. In particular, a data parser was created, a preprocessing script, machine learning algorithms were written (without using ready-made solutions from third-party developers), data dependencies were investigated, EDA was conducted, a machine learning system was trained and fully ready for use.

Thanks to the capabilities of the Telegram API, a shell of end-user interaction with all of the above system components has been created.

There are similar solutions in the public domain, but they do not meet the high standards of usability, accuracy of forecasts and scalability of the system.

Содержание

I. Обоснование выбора языка программирования и средств разработки

Проект разрабатывался на языке *Python 3.8.8* [1]. Он позволяет встраивать в себя программы написанные на языке C/C++, упрощая интерфейс взаимодействия с ними. Также, это основный язык для написания оболочки взаимодействия с API Telegram и возможность взаимодействия с ядром *Jupyter Notebook* [2], что не менее важно в задачах анализа данных.

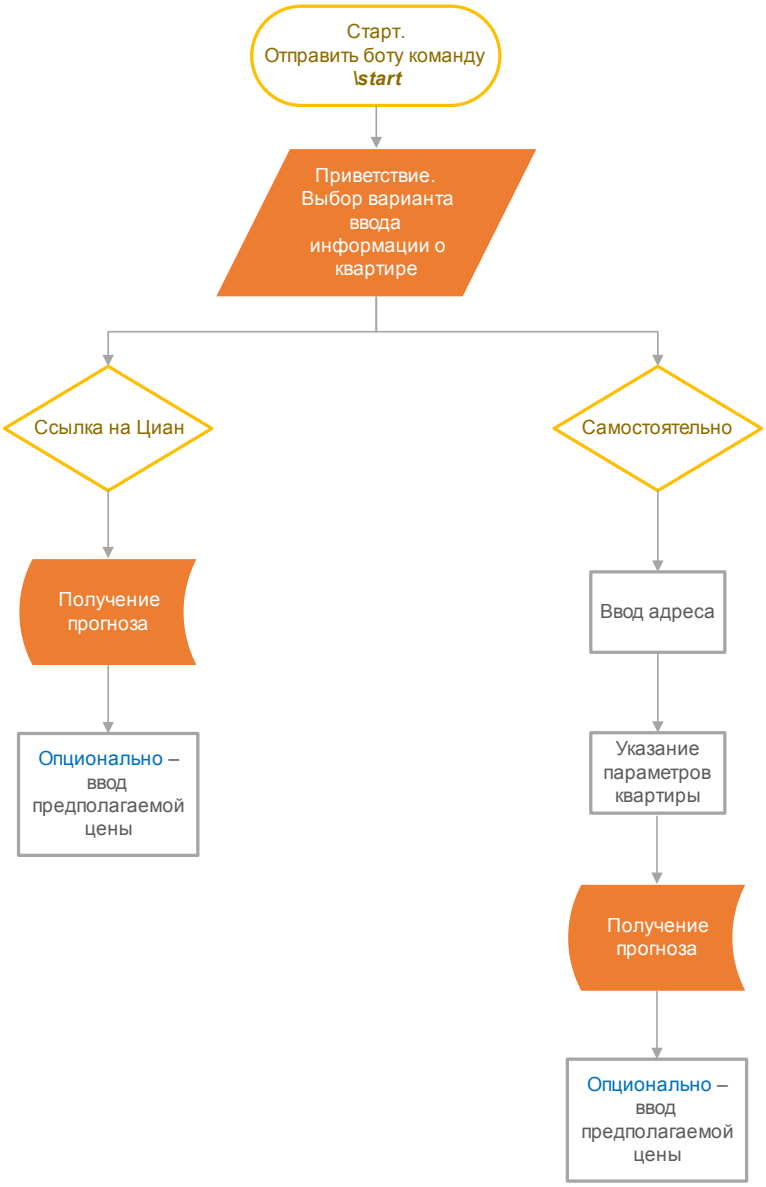
Основой всего проекта стала библиотека *NumPy* [3] — проект с открытым исходным кодом, который упрощает работу с массивами и включает в себя сложные операции линейной алгебры.

Для хранения информации о квартирах был выбран *Microsoft Excel* [4] — простой и удобный интерфейс. В рамках моего проекта достаточно использования такой базы данных.

Весь проект был разработан в интерактивной среде разработке *IDE PyCharm 2021* [5], которая обладает широким функционалом возможностей. Помимо этого, эта среда полностью настроена под язык *Python*.

Для контроля версий использовался *Git* [6]. Для кроссплатформености — возможности запуска программы на любом компьютере — использовался *Docker* [7].

II. Описание сценария



(a)

Рис. 1: Сценарий

На Рис. 1 представлен сценарий взаимодействия пользователя с программой. Ввод предполагаемой цены от пользователя — опциональная возможность, которую можно пропустить.

III. Спецификация интерфейса

(a) (b)

Рис. 2: Элементы взаимодействия

На Рис.2 (a) мы видим диалоговое окно программы, описание бота и кнопку *Начать*, которая отправит программе сообщение *start* и запустится сценарий, показанный на Рис.1. Запустить этот сценарий можно иначе — на Рис.2 (b) показано *Меню команд*, нажав на которое можно в ручную отправить стартовое сообщение.

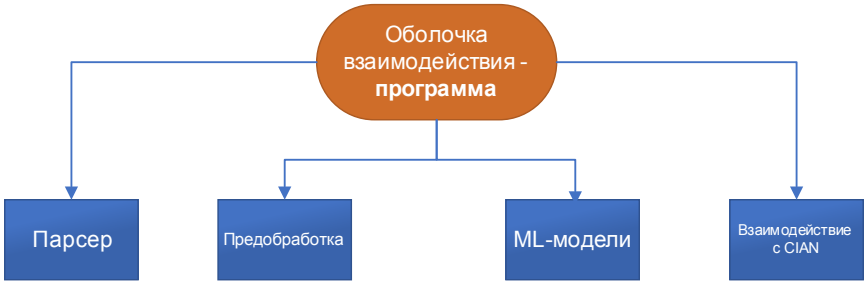
(a) (b)

Рис. 3: Клавиатуры

На Рис.3 показаны 2 типа клавиатур, которые будет использовать пользователь помимо встроенной в его систему. У них есть свои внутренние названия — *инлайн и дефолтная* клавиатуры. С точки зрения пользователя они отличаются только местонахождением на экране — одна находится в самом диалоге, другая там же, где встроенная в систему клавиатура.

IV. Структура программы

Для удобства, декомпозируем программу на 4 части, которые будут подсистемами одной большой системы — оболочки взаимодействия.

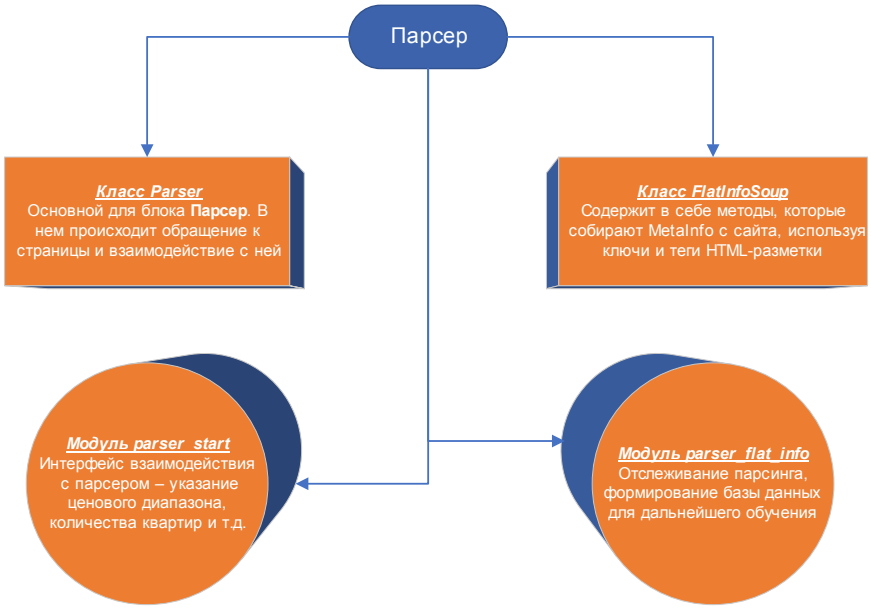


(a)

Рис. 4: Основные блоки программы

i. Парсер

Парсер — скрипт для сбора информации о квартире. Он написан на двух фреймворках — *Selenium* [8] и *BeatifulSoup4* [9]. Первый отвечает за запуск браузера и сбор HTML разметки, второй за преобразование этой разметки в удобный для обработки вид.



(a)

Рис. 5: Структура парсера

ii. Предобработка

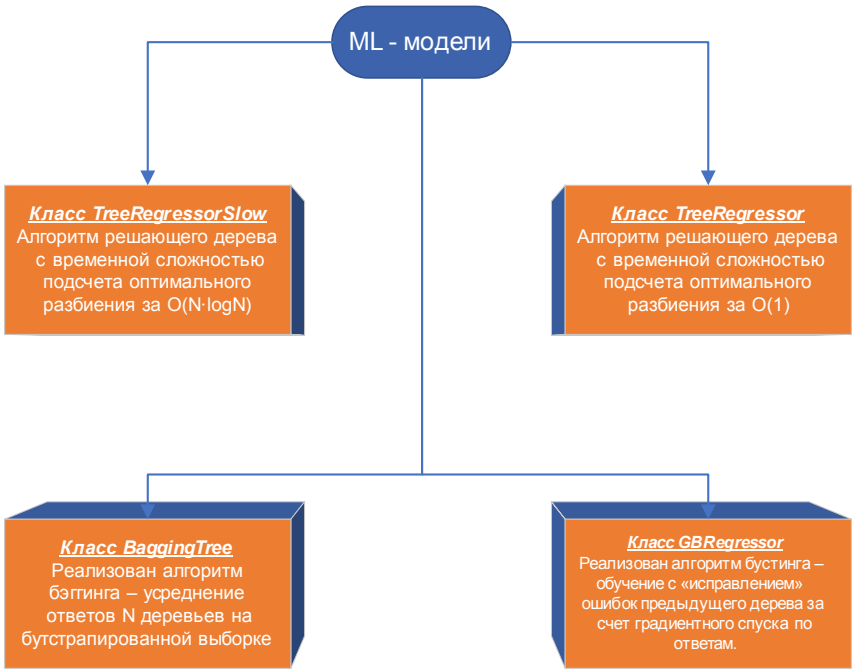
Предобработка данных — один из самых важных этапов построения пайплайна обучения. Все модели машинного обучения работают только с числами, они не способны определять номер дома, район и так далее. Как следствие, после сбора информации преобазовать ее в вид понятный компьютеру — убрать названия домов, улиц и многое другое.

(a)

Рис. 6: Структура предобработки

iii. ML-модели

Как известно, в этом проекте не использовались реализации алгоритмов из библиотек, например, из *SkLearn*. Решающие деревья и их ансамбли писались в ручную. Ниже представлена схема взаимодействия с реализацией алгоритмов машинного обучения.



(a)

Рис. 7: Структура моделей машинного обучения

На Рис.7 представлена схема интерфейса взаимодействия с кодом, написанным на *Cython* [10] — библиотеки, которая компилирует код написанный на *Python* в код на *C* с использованием макросов. Уместить всю структуру блока *ML-модели* довольно сложно, поэтому была выбрана такая реализация блок-схемы.

iv. Взаимодействие с CIAN

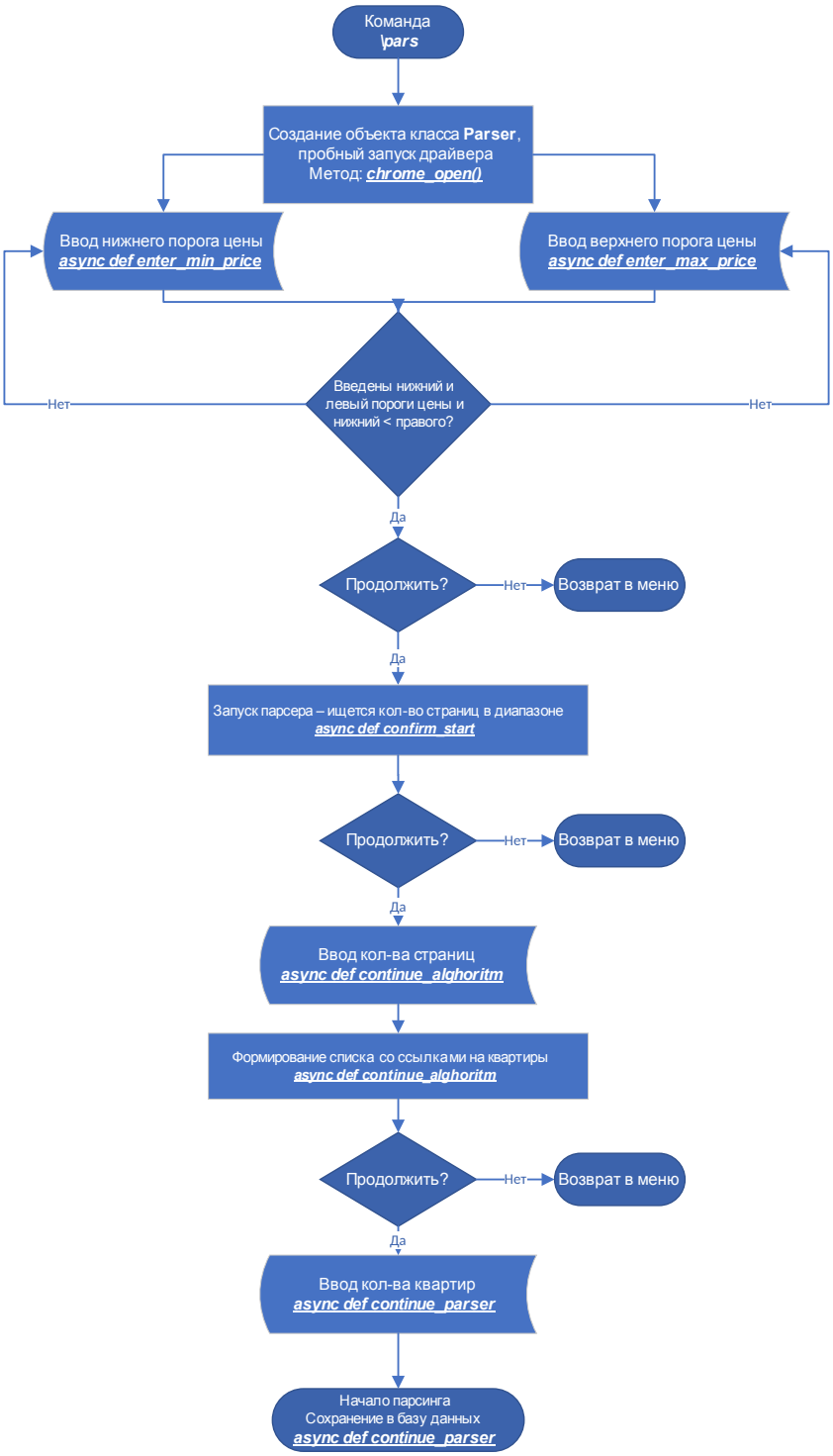
Одной из особенностей программы является *возможность оценить аренду в 2 клика*. Пользователю непотребуется самостоятельно вводить параметры квартиры, будет достаточно прислать ссылку на квартиру с сайта CIAN. Я вынес эту возможность в отдельный блок, так как не смотря на то, что работает она на тех же компонентах — парсер, предобработка, модели машинного обучения, ее реализация потребовала значительного изменения этих блоков.

(a)

Рис. 8: Структура взаимодействия с CIAN

V. Описание алгоритма

Алгоритм формирования БД для обучения.



(a)

Рис. 9: Парсинг

Библиографический список

[1] Python release Python 3.8.8 // Python URL:
https://www.python.org/downloads/release/python-388/

[2] Jupyter Notebook Home // JN URL:
https://jupyter.org/

[3] NumPy Get Started // NP URL:
https://numpy.org/

[4] Microsoft Excel Official // ME URL:
https://www.microsoft.com/ru-ru/microsoft-365/excel

[5] PyCharm The Complete Package // PC URL:
https://www.jetbrains.com/pycharm/

[6] GIT –distributed-is-the-new-centralized // GIT URL:
https://git-scm.com/

[7] Docker Home // DH URL:
https://www.docker.com/

[8] Selenium Chrome Python // SCP URL:
https://selenium-python.readthedocs.io/

[9] BeatifulSoup4 Python // BS URL:
https://pypi.org/project/beautifulsoup4/

[10] Cython C-Extensions for Python // C URL:
https://cython.org/