# Predicting Rental Prices for Real Estate

This document provides step-by-step instructions for setting up the project, running the code, and obtaining rental price predictions. Follow these instructions to configure the environment, execute the program, and analyze model performance.

## 1. Project Setup

### 1.1. Downloading the Project Code

* **Download the Archive or Clone the Repository:**
  + Download the archive containing the source code or clone the repository (this option will be added to the repository shortly).
  + Ensure that all files (source code, configuration, models, databases) are located in the correct folders.

### 1.2. Creating a Virtual Environment

* **Navigate to the Project Root Folder:**
  + For example, if your project folder is located at:

...\Final\_project

* **Create the Virtual Environment:**

python -m venv venv

* **Activate the Virtual Environment:**
  + **For Windows (PowerShell):**

.\venv\Scripts\Activate

* + **For Unix/Linux/Mac:**

source venv/bin/activate

### 1.3. Installing Dependencies

* **Install the Required Libraries:**
  + Run the following command to install all required packages from the requirements.txt file:

pip install -r requirements.txt

* **Verify the Dependencies:**
  + Ensure that the requirements.txt file includes the following dependencies:
    - pandas
    - numpy==1.26.4
    - matplotlib
    - scikit-learn
    - xgboost
    - lightgbm
    - catboost==1.2.7
    - tensorflow
    - joblib
    - seaborn
    - shap
    - flask
    - tabulate
    - shap

**Note:** The package shap appears twice. Please verify if this repetition is intentional or if adjustments are needed.

### 1.4. Configuring Parameters

* **Folder Structure:**
  + If needed, rename the folders:
    - data – should contain the raw database data.
    - src – should contain all the source code.
  + All code files must be placed in the src folder, except for mail.py, config.py, and requirements.txt, which should remain in the project’s root directory.
* **Outlier Removal Settings:**
  + Open the config.py file to choose the outlier filtering method.

**Method 1 (Default – Quantile Filtering):**

* + In the file cleaned\_data\_v3.py, quantile filtering is set by default (lines 113-116):
    - **Price:** 3rd and 98th percentiles.
    - **Area:** 15th and 98th percentiles.
  + Listings with values outside these percentiles will be removed. You may adjust these threshold percentages (between 0 and 100) as needed.

**Method 2 (Threshold Values):**

* + In the config.py file, adjust the threshold values on lines 19-23:
    - MIN\_PRICE, MAX\_PRICE, MIN\_SQUARE, MAX\_SQUARE.
  + Modify the price and area thresholds as necessary.
  + Additionally, in cleaned\_data\_v3.py, uncomment lines 104-108 and comment out lines 113-116 to switch from quantile filtering.

## 2. Running the Program

### 2.1. Starting the Main Script

* **Run the Main Script:**
  + From the project root directory, execute:

python main.py

* **City Selection Prompt:**
  + The program will display:

Choose a city (Almaty, Astana, Shymkent):

* + Enter the desired city (e.g., Astana) and press Enter.

### 2.2. Choosing the Mode of Operation

* **Model Usage Prompt:**
  + After selecting the city, you will see:

Use the last saved model? (y/n):

* + Enter:
    - n if you wish to retrain the models from scratch.
    - y if you already have trained models and want to proceed directly to making predictions.

### 2.3. If Retraining Mode Is Selected (n)

* **Data Cleaning and Preprocessing:**
  + The program will clean data from SQLite files, merge tables, and save the cleaned data to a new database.
  + It will then load and process the data (e.g., adding new features, scaling, etc.).
  + The preprocessing pipeline will be saved for future predictions.
* **Training Models:**
  + The program will start training regression models, including:
    - Ridge, ElasticNet, Random Forest, Gradient Boosting, XGBoost, LightGBM, CatBoost, and a neural network.
  + It will output performance metrics (MAE for the training set, test set, and cross-validation).
  + All models are saved in a dedicated folder for the selected city. Additionally, a ranking file based on MAE is created in an automatically generated models folder (each city gets its own subfolder).
* **Visualization of Results:**
  + The program will generate plots (e.g., scatter plots, error histograms, neural network loss curves) to help analyze model quality.
* **Launching Interactive Prediction:**
  + After training and visualization, the program automatically launches the price prediction module (direct\_predict.py).
* **Training Classification Models (Optional):**
  + In addition to regression models, classification models are also trained to separate listings into high-priced and low-priced categories (this option is under development).

### 2.4. Interactive Prediction (Available in Both Modes)

The interactive prediction module works in both modes:

* In retraining mode (n), the above steps will run, and then the prediction module starts.
* In saved model mode (y), the training stages are skipped, and the prediction module starts immediately (provided that saved training data exists).
* **Entering Apartment Parameters:**
  + You will be prompted to enter the following:
    - **Apartment Area (in m²):** Enter a number (e.g., 60) and press Enter.
    - **Number of Rooms:** Enter an integer (e.g., 2) and press Enter.
* **Selecting the Date Mode:**
  + Two options will be offered:
    - **Specify a Specific Date:**
      * Enter the date in the format YYYY-MM-DD (e.g., 2025-04-15).
    - **Specify a Range of Months for the Current Year:**
      * Enter a range of months (e.g., 4-12 or 4 12). The program will then make a prediction for the 15th day of each month within this range.
* **Selecting Models for Prediction:**
  + A list of available models with their MAE rankings will be displayed.
  + Enter the model number or multiple model numbers separated by commas (e.g., 2,3). If multiple models are selected, the arithmetic mean of the predictions will be calculated.
* **Displaying and Saving Results:**
  + The program will display a table with predictions containing the following details: date, number of rooms, area, city, and the predicted price.
  + The results will also be saved to a CSV file in the predictions folder (this folder will be automatically created if it does not already exist).
* **Repeat Prediction:**
  + The program will ask:

vbnet

Копировать

Would you like to make another prediction? (y/n):

* + Enter y to make another prediction, or n to exit the program.

### 2.5. Using Saved Models Mode (y)

* If you select y to use saved models, the program will bypass the training steps and immediately launch the interactive prediction module (as described above).

## 3. Final Steps

* **Verifying the Results:**
  + After the program completes its run, check the CSV files in the predictions folder to review the predictions.
  + Models and preprocessing pipelines are saved in the folders specified in config.py (e.g., in models/<city>).
* **Additional Setup:**
  + If you need to fetch new rental price data, refer to the other project available at <https://github.com/andprov/krisha.kz>. This repository contains a detailed document outlining the steps for downloading, configuring, and running the parser.

This sequence of steps will enable you to properly configure your environment, run the program, obtain rental price predictions, and assess the quality of the models.

1. **Подготовка проекта**
2. **Скачивание кода проекта:**

Скачайте архив с исходным кодом или клонируйте репозиторий с проектом *(опция находится в процессе добавления на репозиторий)*. Убедитесь, что все файлы *(исходный код, конфигурация, модели, базы данных)* находятся в корректных папках.

1. **Создание виртуального окружения:**

* Перейдите в корневую папку проекта в терминале, в моем случае это:

***…\Final\_project***

* Создайте виртуальное окружение командой:

***python -m venv venv***

* Активируйте виртуальное окружение:

Для Windows (PowerShell):

***.\venv\Scripts\Activate***

Для Unix/Linux/Mac:

***source venv/bin/activate***

1. **Установка зависимостей:**

* Установите все необходимые библиотеки из файла requirements.txt командой:

***pip install -r requirements.txt***

Проверьте заблаговременно файл requirements.txt – зависимости должны быть такими:

pandas

numpy==1.26.4

matplotlib

scikit-learn

xgboost

lightgbm

catboost==1.2.7

tensorflow

joblib

seaborn

shap

flask

tabulate

shap

1. **Настройка параметров:**

В данное время все настроено и не требует вмешательства, но для исследования есть возможность изменения конфигурации. Измените название папок *(data – c исходными данными баз данных и src – расположение всего кода)* если при скачивании эти папки определены не верно либо их нет. Все файлы кода должны быть помещены в папку src, за исключением файлов mail.py, config.py и requirements.txt которые должны быть расположены в корневом каталоге проекта.

Проверьте файл config.py. Определите каким способом будет отсекаться выбросы.

***1 способ*** (стоит по умолчанию) в файле cleaned\_data\_v3.py установлена квантильная фильтрация *(строки 113-116: Цена: 3-й и 98-й процентиль. Площадь: 15-й и 98-й процентиль, то есть ниже и выше этих показателей объявления буду отсекаться).* При необходимости можете их смещать от 0 до 100 в большую или меньшую сторону, в зависимости от задачи.

***2 способ*** в файле config.py изменить пороговые значения *(строки 19-23 - MIN\_PRICE, MAX\_PRICE, MIN\_SQUARE, MAX\_SQUARE)*. При необходимости можете смещать показатели цены и площади в большую или меньшую сторону, в зависимости от задачи. Кроме того раскомментируйте в файле cleaned\_data\_v3.py строки 104-108 и наоборот закомментируйте строки *113-116* с квантильной фильтрацией.

1. **Запуск программы**
2. **Запуск основного скрипта:**

* В терминале, находясь в корневой папке проекта, запустите:

***python main.py***

* Программа выведет запрос: **"Выберите город (Алматы, Астана, Шымкент):"**

Введите нужный город (например, "Астана") и нажмите Enter.

1. **Выбор режима работы:**
   * После выбора города программа спросит:

**"Использовать последнюю сохранённую модель? (y/n):"**

* если вы хотите переобучить модели с нуля, введите n.
* если у вас уже обученные модели и хотите сразу сделать прогноз, введите y.

1. **Если выбран режим переобучения (n):**
   * **Очистка и предобработка данных:**

* программа выполнит очистку данных из SQLite файлов, объединит таблицы и сохранит очищенные данные в новую базу.
* затем данные будут загружены и обработаны *(добавление новых признаков, масштабирование и т.д.)*.
* пайплайн предобработки будет сохранён для будущих прогнозов.
  + **Обучение моделей:**

- запустится процесс обучения регрессионных моделей *(Ridge, ElasticNet, Random Forest, Gradient Boosting, XGBoost, LightGBM, CatBoost и нейронной сети)*.

- вывод метрик *(MAE для обучающей, тестовой выборок и кросс-валидации)*.

- все модели сохраняются в отдельной папке для выбранного города, а также создаётся файл с ранжированием по метрике MAE *(папка models создается автоматически c с распределением обученных моделей для каждого города в отдельной папке)*.

* + **Визуализация результатов:**
* построение графиков *(scatter-плоты, гистограммы ошибок, кривые потерь нейронной сети и т.д.)* для анализа качества моделей.
  + **Запуск интерактивного прогнозирования:**
* после завершения обучения и визуализации, программа автоматически запускает модуль прогнозирования цен - direct\_predict.py.
  + **Обучение классификационных моделей:**
* дополнительно обучаются модели классификации для разделения объявлений на высоко- и низкоценовые *(опция дорабатывается)*.

1. **Интерактивное прогнозирование** *(работает в обоих режимах – в режиме нового обучения (при вводе n) отработают все ранее указанные этапы и после этого запускается модуль прогнозирования, при выборе работы с заранее сохраненными моделями (при вводе y) минуя все этапы обучения сразу начинает работать модуль прогнозирования, в случае наличия заранее сохраненных данных об обучении)***:**
   * **Ввод параметров квартиры,** программа попросит ввести:

**- Площадь квартиры (в м²):** введите число (например, 60) и нажмите Enter.

**- Количество комнат:** введите целое число (например, 2) и нажмите Enter.

* + **Выбор режима даты, п**рограмма предложит два режима:

**1 - Указать конкретную дату:** Если выберете этот режим, введите дату в формате ГГГГ-ММ-ДД *(например, 2025-04-15)*.

**2 - Указать диапазон месяцев текущего года:** Если выберете этот режим, введите диапазон месяцев *(например, 4-12 или 4 12)*, после чего для каждого месяца будет сделан прогноз на 15-е число месяца.

* + **Выбор моделей для прогнозирования:**

- на экране будет выведен список доступных моделей с их рейтингом по MAE. Введите номер модели, либо номера модеей через запятую *(например, 2,3)* и нажмите Enter *(в случае выбора нескольких моделей будет использовано среднеарифметическое значение цены)*.

* + **Отображение и сохранение результатов:**

- программа выведет таблицу с прогнозами *(дата, количество комнат, площадь, город, прогнозируемая цена)* и сохранит результаты в CSV-файл в папке predictions *(папка также создается автоматически, если ее еще нет)*.

* + **Повторный прогноз,** программа спросит: **"Хотите сделать ещё один прогноз? (y/n):"**

- если ответ y, процесс ввода повторяется.

- если ответ n, программа завершает работу.

1. **Если выбран режим использования сохранённой модели (y):**
   * Программа сразу переходит в интерактивный режим прогнозирования (алгоритм указан выше).
2. **Заключительные шаги**

* **Проверка результатов:**

После завершения работы проверьте сохранённые CSV-файлы в папке predictions для просмотра прогнозов.

Модели и пайплайны сохранены в папках, указанных в config.py (например, в папке models/<город>).

* **Дополнительная настройка:**

Если необходимо получить новые данные о ценах на аренду жилья, обращаемся к другому коду (<https://github.com/andprov/krisha.kz>), в котором есть документ подробно описывающий шаги для скачивания, настройки и запуска парсинга.

Эта последовательность шагов позволит пользователю правильно настроить окружение, запустить программу и получить прогнозы аренды жилья, а также оценить качество работы моделей.