**Отчёт о проделанной работе**

Файлы для просмотра конечного результата находятся в папке финал.

Промежуточные результаты находятся в папках версия n.

В папке финал также есть файл с отчётом и README.md.

1. Версия 1:

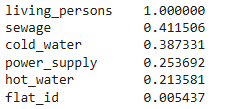
Данные не были предобработаны. В качестве стандарта и модели для последующих сравнений была выбрана линейная регрессия (в качестве наиболее изученной, легко интерпретируемой и часто используемой) со значением метрики R2 около 0.18 (где 0 – модель везде ошиблась, 1 – идеальные предсказания. Такая метрика была выбрана для возможности отображения точности в процентах (0.18 = 18% точности) для лёгкого сравнения качества моделей.

1. Версия 2:

Была произведена предобработка данных: избавление от NaN-значений в поле sewage (объём водоотведения): т.к. известно, что это сумма холодной и горячей воды, заменено на сумму полей cold\_water и hot\_water. NaN-значения убраны для более высокой интерпретируемости работы модели и возможности обучения случайного леса и градиентного бустинга (они не обрабатывают такие значения).



Из матрицы корреляций можно заметить, что с целевой переменной больше всего коррелирует водоотведение. Номер квартиры же вообще никак не влияет.



Были выбраны три модели для сравнения: линейная регрессия, случайный лес и градиентный бустинг. Выбраны именно эти модели, как наиболее часто использующиеся в машинном обучении и выдающие самые качественные результаты.

Для них подбирались гиперпараметры с помощью GridSearchCV.

Точность модели вычислялась методами среднеквадратичной ошибки и R2:

Линейная регрессия – MSE 1.37, R2  0.18

Случайный лес – MSE 1.09, R2 0.35

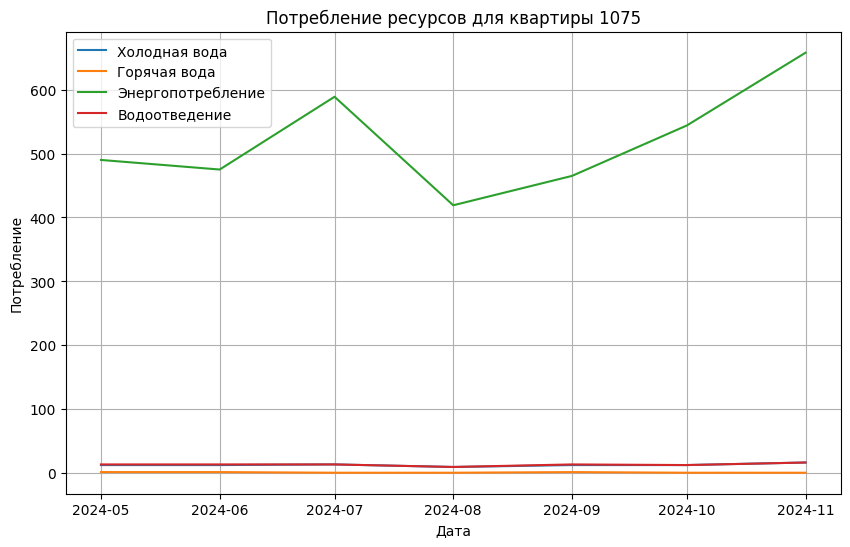
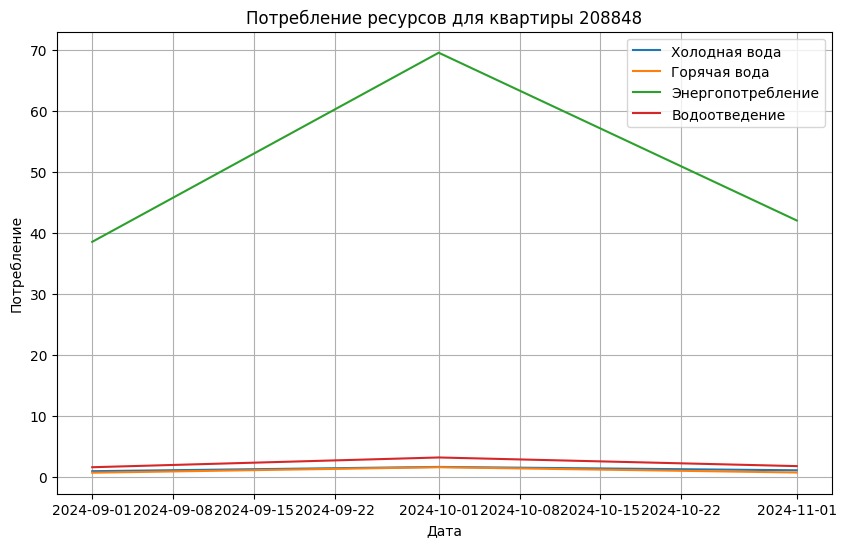
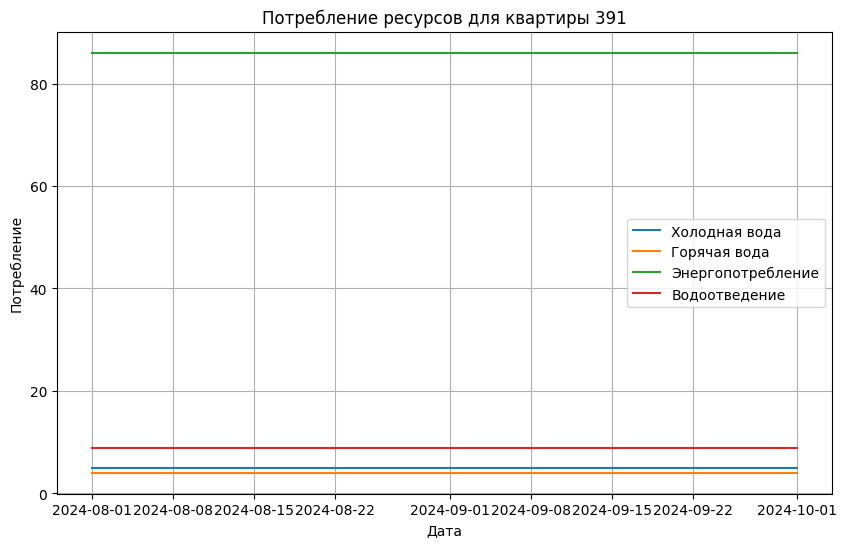
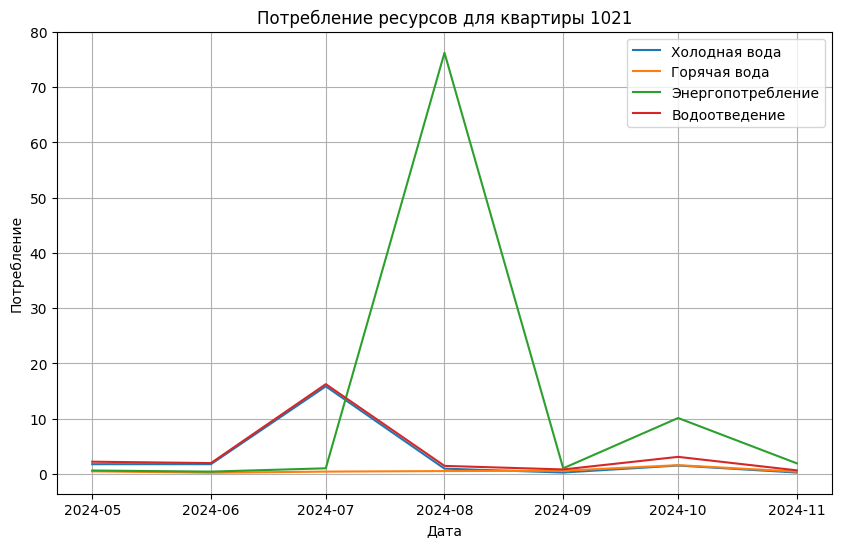
Градиентный бустинг – MSE 1.07, R2  0.36

В результате сравнения, моделью для сабмита и последующего подбора гиперпараметров был выбран градиентный бустинг. В основе градиентного бустинга лежит принцип построения нескольких моделей, исправляющих ошибки предыдущей.

1. Версия 3:

Был проанализирован временной ряд (поквартирно). Исходя из анализа, сделаны следующие выводы:

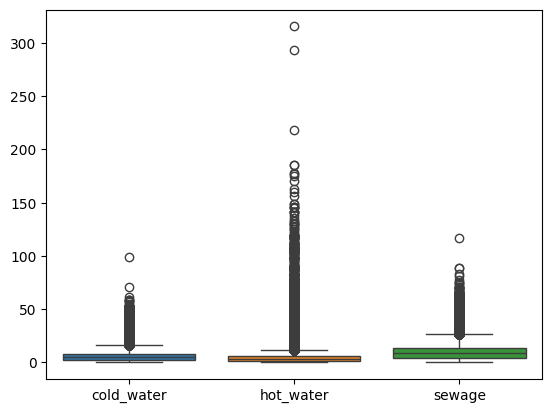
* У большого количества данных линейное потребление ресурсов (не меняется в зависимости от взятого периода)
* Потребление горячей и холодной воды меняется незначительно в зависимости от периода
* Потребление электроэнергии может меняться достаточно значительно и хаотично в зависимости от периода
* Т.к. водоотведение является суммой потребления холодной и горячей воды, напрямую от них зависит (не берется в анализ)



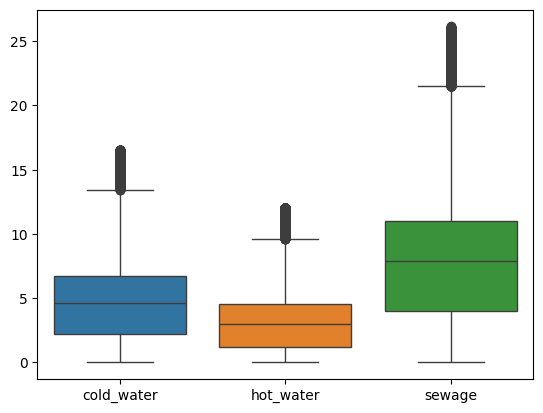
1. Версия 4

В связи с низким значением метрики (менее 50%) с помощью моделей машинного обучения на scikit-learn, было принято решение попробовать другие модели и более глубокую предобработку данных.

С помощью графика boxplot были обнаружены выбросы.



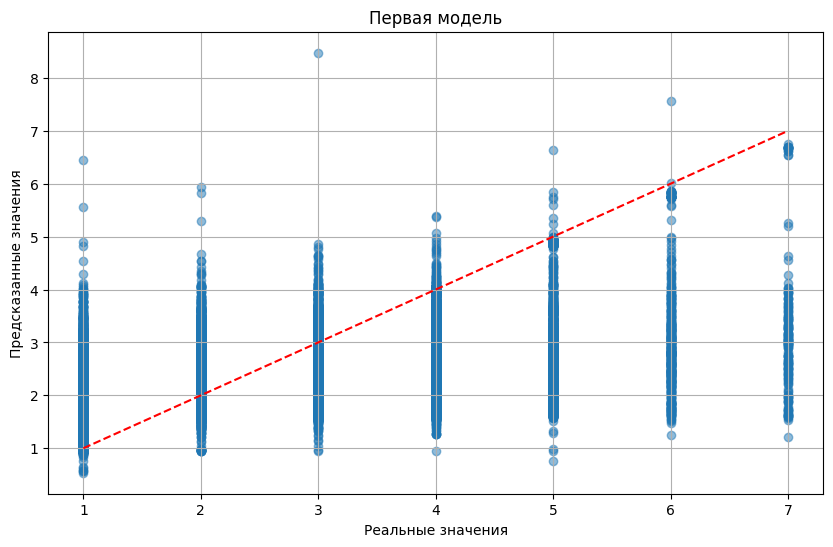
После избавления от значительного количества выбросов:



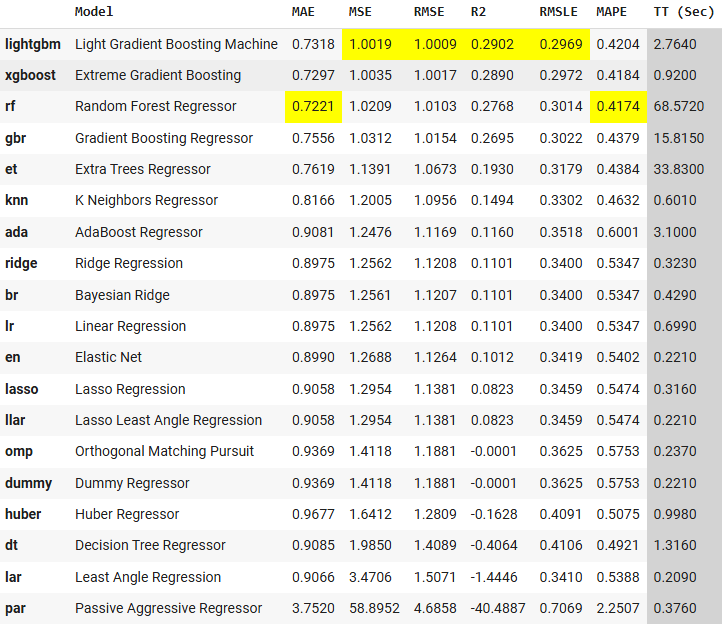
Для моделей первым делом я использовала нейронную сеть на фреймворке TensorFlow для регрессии.

Первая версия модели состояла всего из трёх слоёв , 64-32-1 нейронов. Использовалась функция активации ReLU (Rectified Linear Unit, все значения ниже 0 становятся 0) и оптимизатор Адам (как стандартный и наиболее распространённый). Обучалась на 50 эпохах.

Первая версия модели – R2 0.31



В поисках лучшей метрики, было принято решение использовать AutoML-фреймворк (pycaret) для быстрого развертывания моделей и поиска наилучшей метрики.



Затем был использован AutoML с предварительной генерацией новых признаков и игнорированием периода и номера квартиры. Исходя из результатов, итоговой моделью была выбрана Extra Trees Regressor.



Был загружен result.csv, веса модели (best\_model.pkl), файл с решением через TF и первое решение через AutoML в файл Волга2024 AutoML.ipynb.

1. Финальная версия:

Для упрощения тестов модели, была создана версия .ipynb для создания сабмитов с загрузкой весов.

В финальном варианте присутствуют файлы README.md с кратким пояснением к проекту, отчёт (данный файл), веса модели (final\_model.pkl), файл с решением (Volga.ipynb).

Т.к. мы считаем данные за ноябрь, мы везде проставляем именно ноябрь (01.11.2024)

В финальном варианте предоставлено решение через генерацию новых фичей и AutoML.

Также были искоренены дублирования квартир и указание периода в result.csv (в версии (исправленное))