ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
«Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии.»

Тохчуков Данил Андреевич
317 группа ВМК МГУ

Оглавление

1 Введение

Основная цель данной работы - реализовать случайный лес и градиентный бустинг на базе деревьев и проанализировать как влияют параметры моделей на их качество. Заодно модели можно сравнить, что мы и сделаем в этой работе. Исследования будем проводить на датасете данных о продажах недвижимости **House Sales in King County, USA** Параметры моделей, которые мы будем анализировать:

1. количество деревьев в ансамблях – `n_estimators`
2. размерность подвыборки признаков для дерева – `feature_subsample_size`
3. максимальная глубина дерева – `max_depth`
4. скорость обучения (только для градиентного бустинга) – `learning_rate`

Далее требуется создать веб-сервер, презентующий реализованные модели. Все исходные файлы будут в репозитории github: **репозиторий**.

2 Предобработка данных

Данные представляют из себя csv-таблицу с 21 колонкой. Требутся предсказать цену дома по его параметрам, записанным в колонки таблицы Из этих колонок можно сразу исключить колонку „id", ведь она не влияет на цену дома. Интересная колонка: „date". Возможно от даты цена на товары может сильно меняться, проверим это:

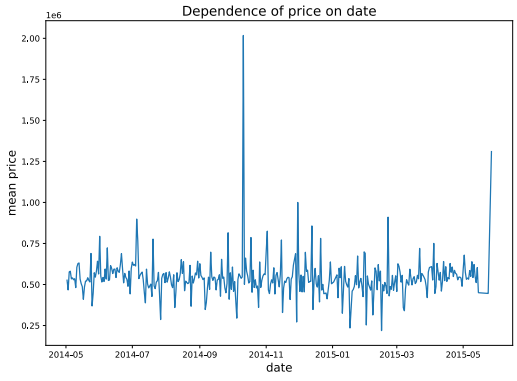


Рис. 1: Цена в уникальный день

Отлично! Периодов повышения или понижения цены нет, есть только скачки в некоторых датах – удалим эту колонку (в каждую дату мы считали среднюю цену дома в этот день, всего в данных 372 уникальные даты).

Далее мы разобъём высю выборку на обучающую (0.8 от всей выборки) и отложенную (0.2 от всей выборки).

3 Случайный Лес

Рассмотрим параметры для случайного леса, о которых мы говорили в введении. Будем изучать метрику **RMSE** и **время работы алгоритма** в зависимости от параметров.

3.1 Количество деревьев в ансамбле

Будем перебирать параметр в пределе 30 деревьев.

- RMSE: (Приложение 1)
- Время обучения: (Приложение 2)

Видим, что при увеличении количества деревьев, качество улучшается, но при увеличении количества деревьев после достижения некоторого количества деревьев (для каждой сложности модели это количество своё)– качество выходит на плато, и уже не имеет смысла увеличивать количество деревьев. Время обучаения закономерно увеличивается при увеличения количества деревьев, это связано с тем, что мы делаем больше итераций в алгоритме. Отметим, что чем больше сложность деревьев, тем дольше обучается модель.

Далее будем брать 20 деревьев в случайном лесе, чтобы точно выйти на плато для модели любой сложности.

3.2 Размерность подвыборки признаков для дерева

- RMSE: (Приложение 3)
- Время обучения: (Приложение 4)

Будем перебирать параметр `feature_subsample_size` RMSE убывает при небольших размерах признакового пространства (< 6), затем качество выходит на плато. То есть параметр $\lfloor \frac{n}{3} \rfloor = 6$ (у нас 18 признаков) в нашем случае оптимален (обычно таким его берут для задач регрессии) Ну и при увеличении размерности признакового пространства закономерно увеличивается время обучения модели, поэтому нужно брать параметр минимально возможным.

3.3 Глубина

- RMSE: (Приложение 5)
- Время обучения: (Приложение 6)

RMSE случайного леса экспоненциально убывает с ростом сложности деревьев, так и должно было быть, ведь чем сложнее модель тем больше зависимостей она может выявить. Однако последней точкой нашего графика отрисовано качество модели, у которой сложность деревьев не ограничена. Как мы видим, RMSE такой модели стало немного больше, чем у предыдущих моделей. Сложность модели стала слишком большой и переобучилась, поэтому RMSE стало больше. Также отметим, что время обучения случайного леса с увеличением сложности линейно увеличивается и достигает своего пика в модели с неограниченной сложностью.

4 Градиентный Бустинг

Теперь будем рассматривать параметры для Градиентного Бустинга. Также рассматриваем метрику **RMSE** и **время работы алгоритма**.

4.1 Количество деревьев в ансамбле

Будем перебирать параметр в пределе 1000 деревьев.

- RMSE: (Приложение 7)
- Время обучения: (Приложение 8)

RMSE убывает экспоненциально, причём с большой скоростью, поэтому рядом приведён ещё один график в логарифмических шкалах. Отсюда видно, что чем больше деревьев, тем больше качество Однако время обучения модели линейно возрастает с ростом числа деревьев в модели, поэтому стоит ограничиться небольшим числом деревьев, но так, чтобы модель была сравнима по качеству с моделью с больших числом дереьев (то есть найти оптимальный параметр учитывая качество и время обучения). Такой точкой является 400 деревьев – на графиках она отмечена красной точкой – после неё, если прибавить целых 600 деревьев, качество сильно не поменяется, а время обучения почти 14 секунд.

Получается, что чем больше деревьев в градиентном бустинге, тем точнее он может настроиться на обучающую выборку, но время обучения стремительно возрастает, поэтому нужно выбирать что важнее.

4.2 Размерность подвыборки признаков для дерева

- RMSE: (Приложение 9)
- Время обучения: (Приложение 10)

Время обучения закономерно возрастает при увеличения размерности признакового пространства. Качество же скачет. Вызвано это тем, что признаки по своей сути неоднородны, какие-то сильнее влияют на таргет(предсказание), какие-то меньше. Но всё равно, все „провалы" RMSE (на нескольких запусках) находятся рядом с $\lfloor \frac{n}{3} \rfloor == 6$, поэтому дальше мы будем брать параметр именно таким

4.3 Глубина

- RMSE: (Приложение 11)
- Время обучения: (Приложение 12)

Лучший depth оказался равным 3-м. Чтобы строить более качественную модель, нужны более простые модели, потому что так мы снижаем корреляцию между этими моделями, а значит мы уменьшаем разброс. Время обучения также линейно увеличивается с ростом глубины деревьев. На модели с неограниченной глубиной деревьев – пик времени обучения.

4.4 Скорость обучения

Будем подбирать лучший `learning_rate`

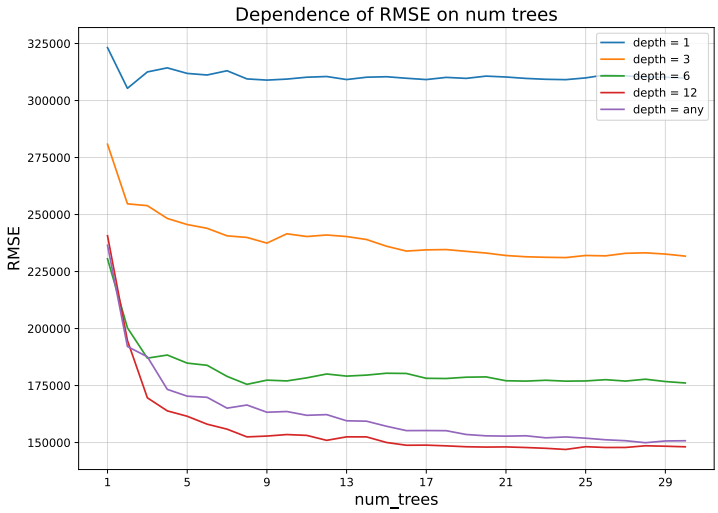
- RMSE on logspace: (Приложение 13)
- RMSE near the 0.1: (Приложение 14)
- Время обучения: (Приложение 15)

Сначала начнём подбирать `learning_rate` значениями вида: $10^z, z \in \mathbb{Z}$. Результат подбора на первом графике. Получилось, что где-то рядом с 0.1 находится оптимум. Дальше, на втором графике, мы уточняем эту оценку параметра. Наиболее оптимальный парметр оказался равным 0.3 – в этой точке наибольшее качество, и рядом с ней меньше всего скачков качества.
Время обучения, чисто теоретически, не должно сильно зависеть от `learning_rate` потому что этот параметр не влияет на число итераций, а на обучение решающих деревьев влияет очень мало. Параметр `learning_rate` > 1 не был учтён, потому что когда градиентный бустинг обучается, он уже подбирает оптимальный коэффициент для нового дерева, чтобы добавить его в ансамбль, и если этот коэффициент увеличить по модулю, то мы каждый раз будем (вероятнее всего) перепрыгивать оптимум, и нет никакой гарантии, что метод с такими параметрами сойдётся.

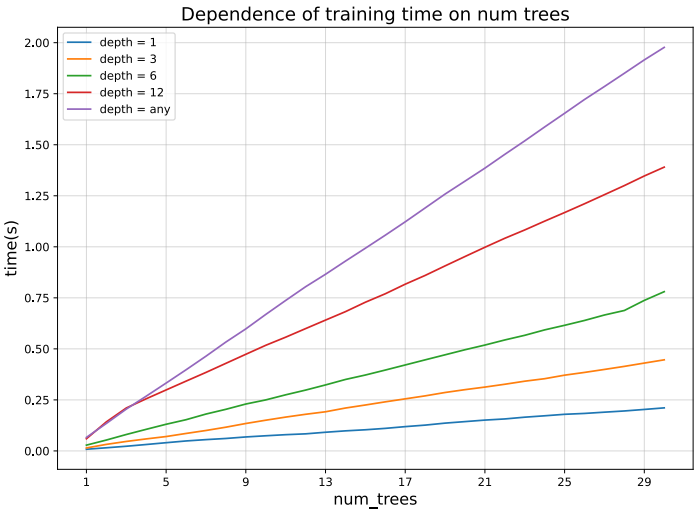
5 Вывод

Ансамблевые методы отлично дополняют базовые алгоритмы: случайный лес уменьшает разброс базовых алгоритмов, а градиентный бустинг ещё и уменьшает сдвиг (bias). В работе удалось найти оптимальные параметры для каждой модели, с помощью чего получилось достигнуть качества RMSE = 140000. И Случайный Лес и Градиентный Бустинг хороши, но эксперименты показали, что Градиентный бустинг добивается лучшего качества, причём мы узнали, что чем больше мощностей у компьютера, тем большее качество можно достигнуть, потмоу что на наших графиках, RMSE не достигла „плато”.

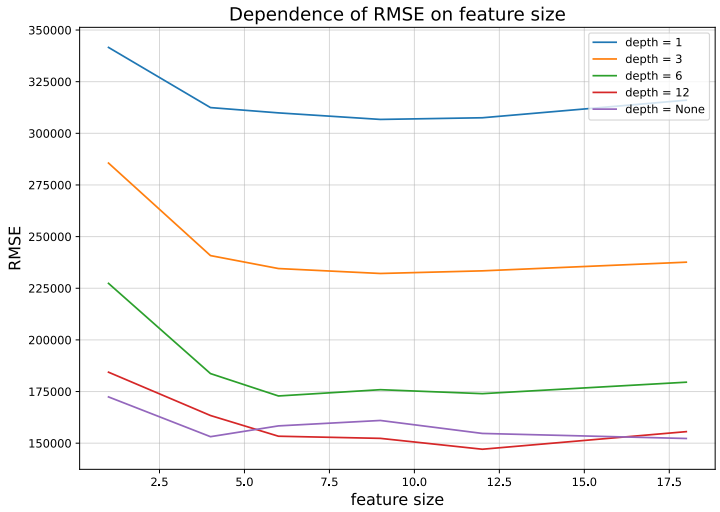
6 Приложения



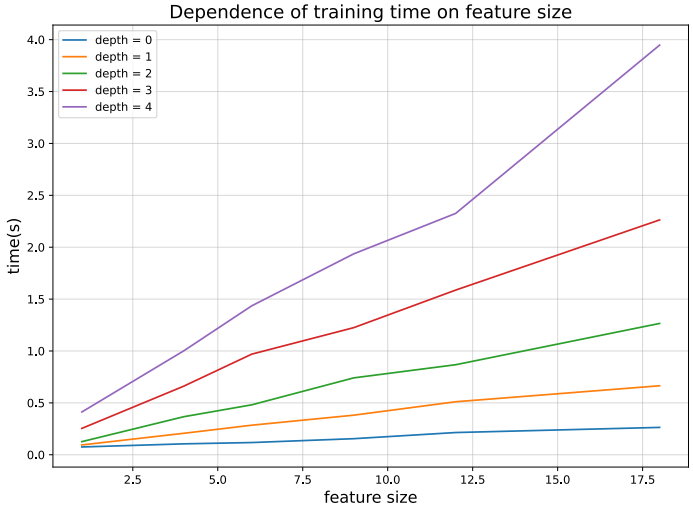
Прил. 6.1: Случайный лес



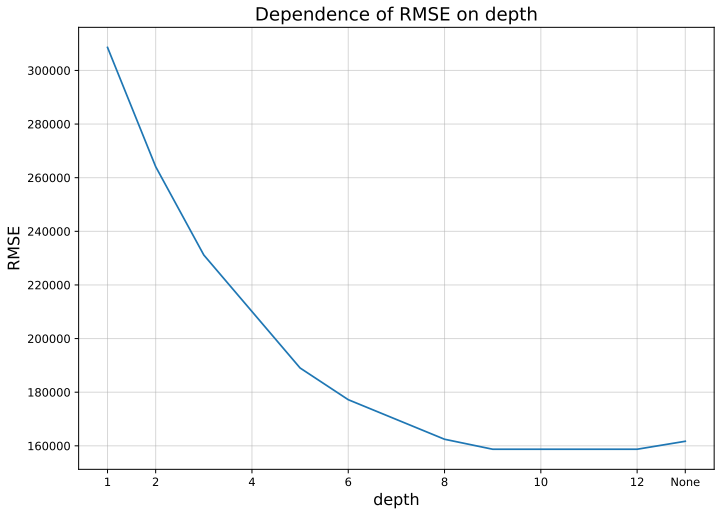
Прил. 6.2: Случайный Лес



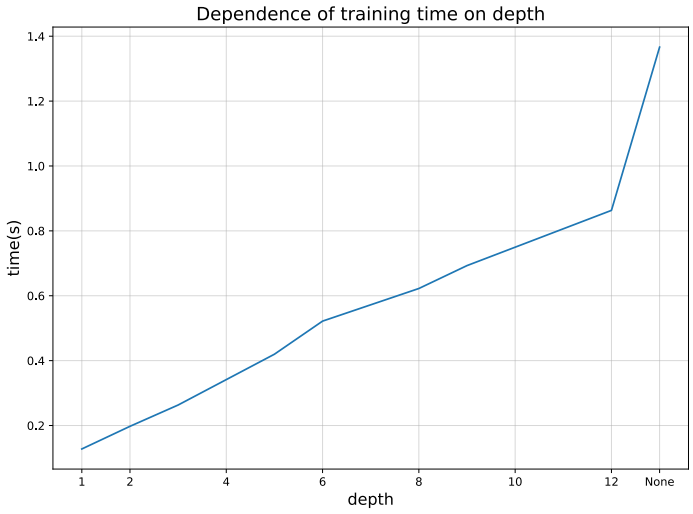
Прил. 6.3: Случайный Лес



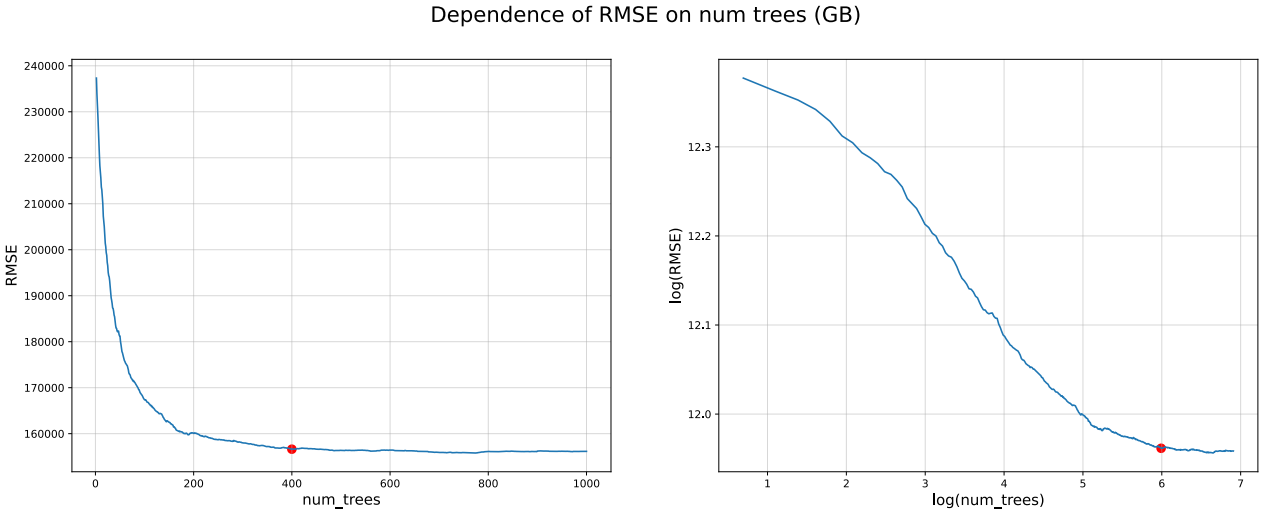
Прил. 6.4: Случайный Лес



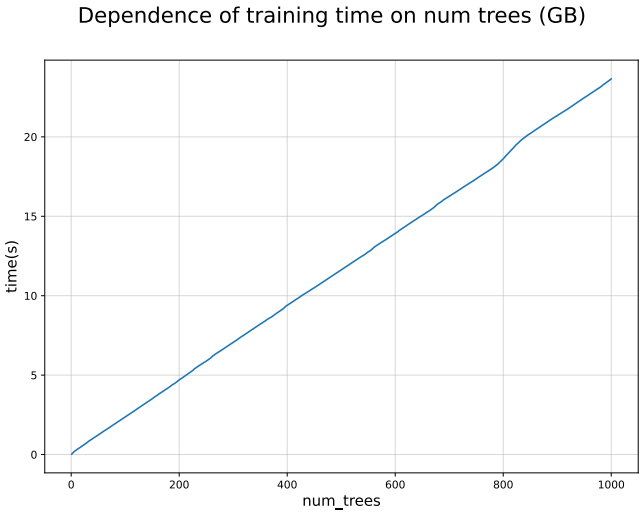
Прил. 6.5: Случайный Лес



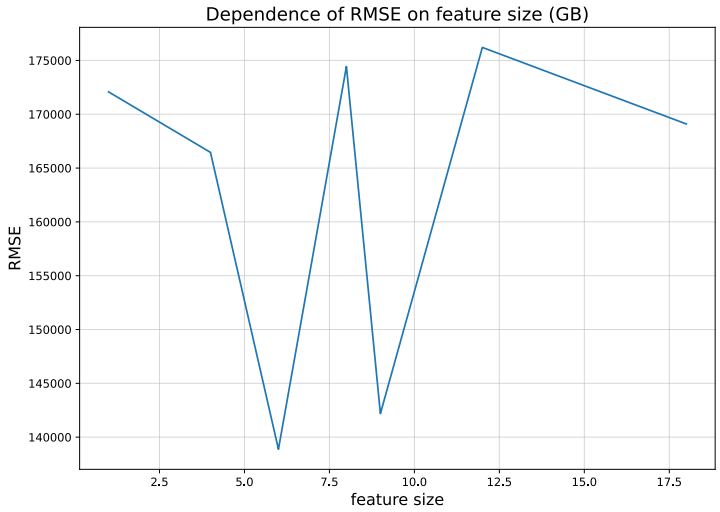
Прил. 6.6: Случайный Лес



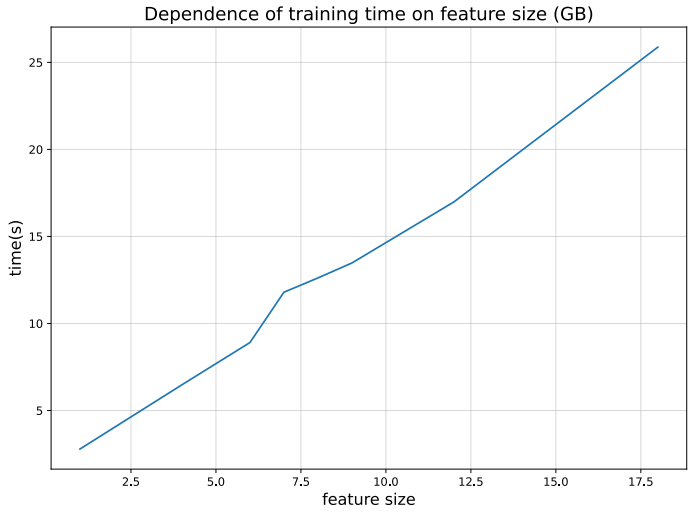
Прил. 6.7: Градиентный Бустинг



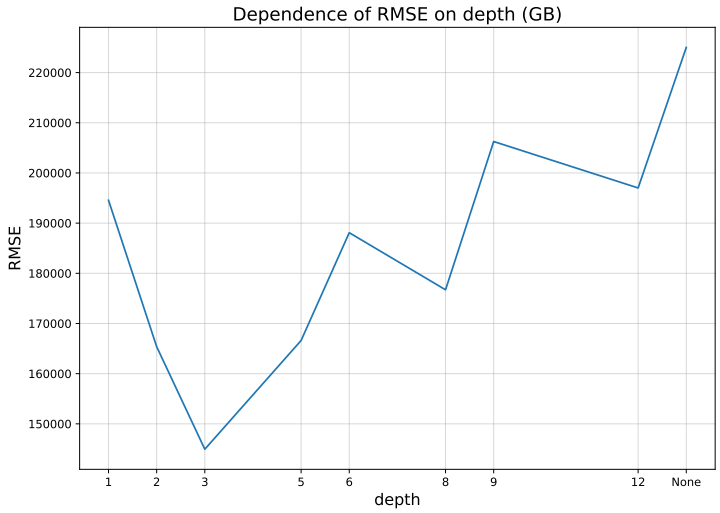
Прил. 6.8: Градиентный Бустинг



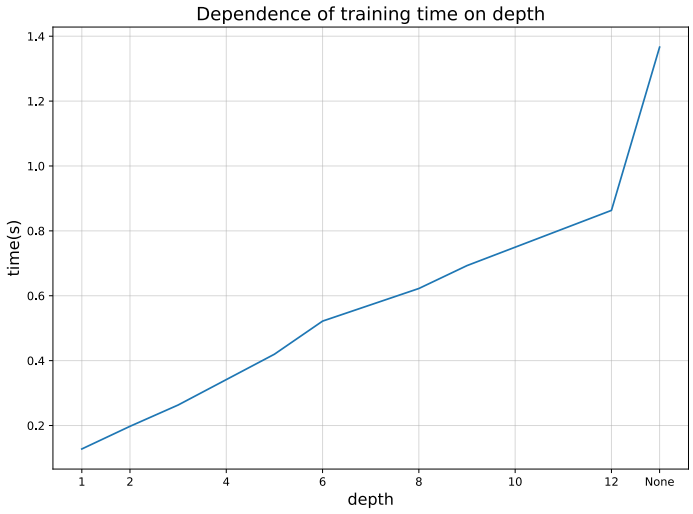
Прил. 6.9: Градиентный Бустинг



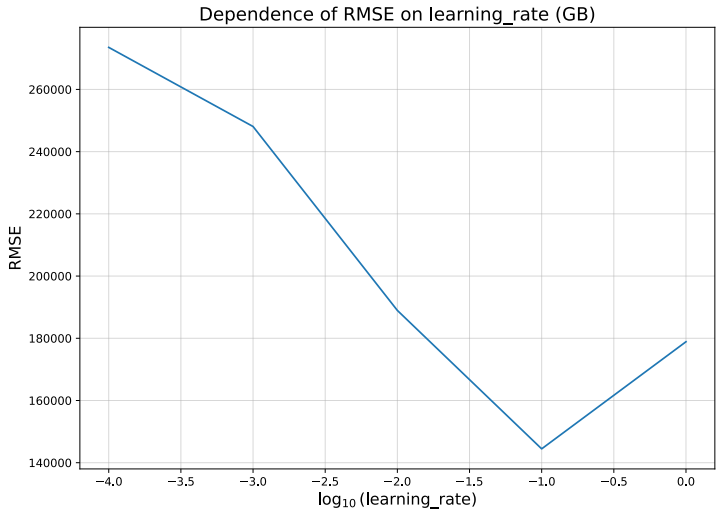
Прил. 6.10: Градиентный Бустинг



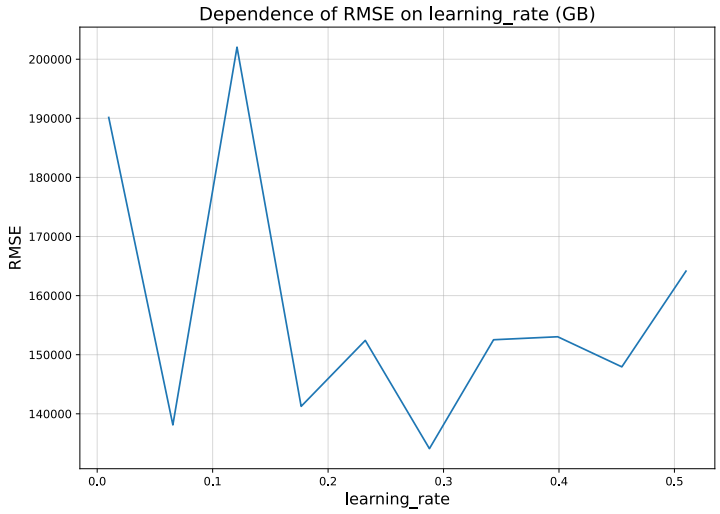
Прил. 6.11: Градиентный Бустинг



Прил. 6.12: Градиентный Бустинг



Прил. 6.13: Градиентный Бустинг



Прил. 6.14: Градиентный Бустинг

Прил. 6.15: Градиентный Бустинг