Отчет по заданию №3: Композиции алгоритмов для решения задачи регрессии

Васильев Руслан ВМК МГУ, 317 группа

26 декабря 2020 г.

Содержание

1 Введение

В заключительном практическом задании предлагается реализовать композиции алгоритмов машинного обучения и провести эксперименты, а также спроектировать веб-сервис для взаимодействия с моделью. Весь проект доступен в репозитории[1]. Данный отчет иллюстрирует результаты экспериментов с моделями на датасете данных о продажи недвижимости.

2 Постановка задачи

Итак, рассматривается задача регрессии с метрикой качества RMSE:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}},$$

где N — размер выборки, y_i — истинное значение целевой переменной на i-м объекте, $\hat{y}_i$ — предсказанное.

Для решения реализованы две модели, представляющие собой ансамбли решающих деревьев: случайный лес и градиентный бустинг. Исследование алгоритмов включает в себя измерение функции ошибки и времени работы при варьировании гиперпараметров (порядок экспериментов соответствует стандартной настройке данных моделей).

3 Эксперименты

В модели добавлен дополнительный параметр — `random_state`. С его помощью инициализируется генератор псевдослучайных чисел (PCG64). «Случайность» нужна для бэггинга (случайный лес) и формирования подвыборки признаков (размерность которой регулируется `feature_subsample_size`. Для воспроизводимости во всех исследовавшихся моделях используется сид, равный нулю. Кроме ошибки (RMSE), проводится измерение времени работы. Причем на графиках изображено усредненное время: каждая модель запускается трижды.

3.1 Предобработка данных

Исходные данные о недвижимости были разделены на обучение (80%) и контроль (20%, она же валидационная выборка). И здесь сразу учитывается особенность задачи. Хотя в задании отсутствует описание признаков и целевой переменной, можно с уверенностью предположить, что столбец `date` связан со временем поступления данных (даты имеют небольшой диапазон 2014–2015, монотонно возрастают, дублируются, следуют сразу за `ID`, а столбцы `build_year` и `renovation_year` с ними не связаны). По этой причине было бы некорректно перемешать выборку перед разделением на обучение и контроль — из-за утечки такая стратегия может дать

[1] https://github.com/artnitolog/mmf_prac_2020_task_3

ложную оценку качества моделей и привести к неправильным выводам. В качестве валидационной выборки берутся последние 20% данных, соответствующие хронологическому порядку по столбцу `date`.

3.2 Случайный лес

3.2.1 Количество деревьев

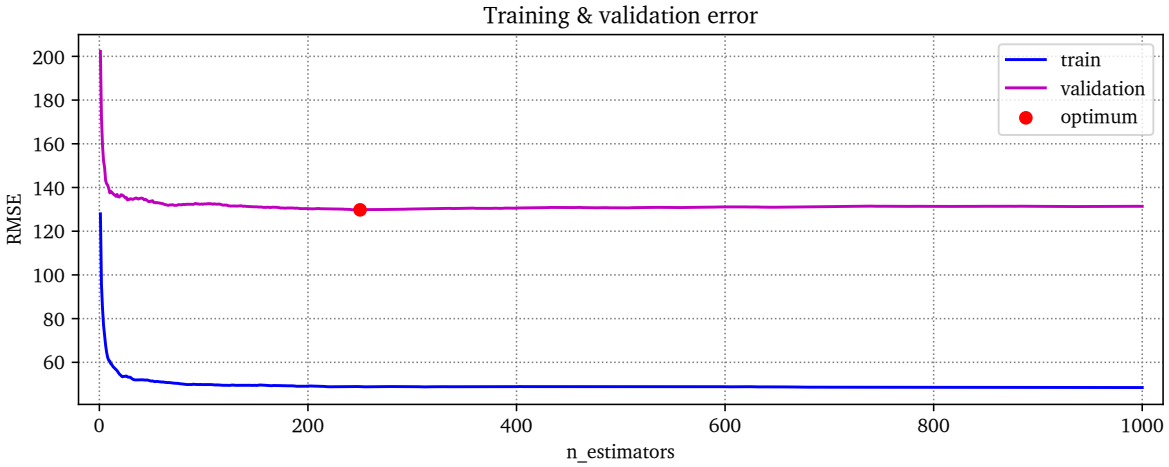

Рис. 1: Зависимость RMSE от количества деревьев в случайном лесе

Количество деревьев в случайном лесе регулирует число алгоритмов, по которому проводится ансамблирование (усреднение). На рис. 1 можно видеть, что с ростом числа деревьев ошибка практически монотонно убывает на обучении выборке. Тем не менее на контроле по достижении оптимального числа базовых алгоритмов функционал затем немного увеличивается (переобучение), а затем выходит на асимптоту. Для нашей задачи нам оказалось достаточно взять 250 деревьев. Что

Рис. 2: Зависимость времени обучения случайного леса от числа деревьев

касается времени работы, то понятно, что оно должно линейно зависеть от числа деревьев в лесе. Для более честной оценки обучим с нуля несколько моделей с

разным количеством деревьев, результаты приведены на рис. 2. И действительно, время обучения растет линейно.

3.2.2 Размерность подвыборки признаков для одного дерева

В случайном лесе данный параметр может сильно повлиять на качество предсказания. В задаче регрессии обычно берут либо все признаки, либо треть от их числа. Именно последний вариант оказался выигрышным в нашей задаче (рис. 3). На обучении, как и следовало ожидать, ошибка монотонно убывает с увеличением числа признаков.

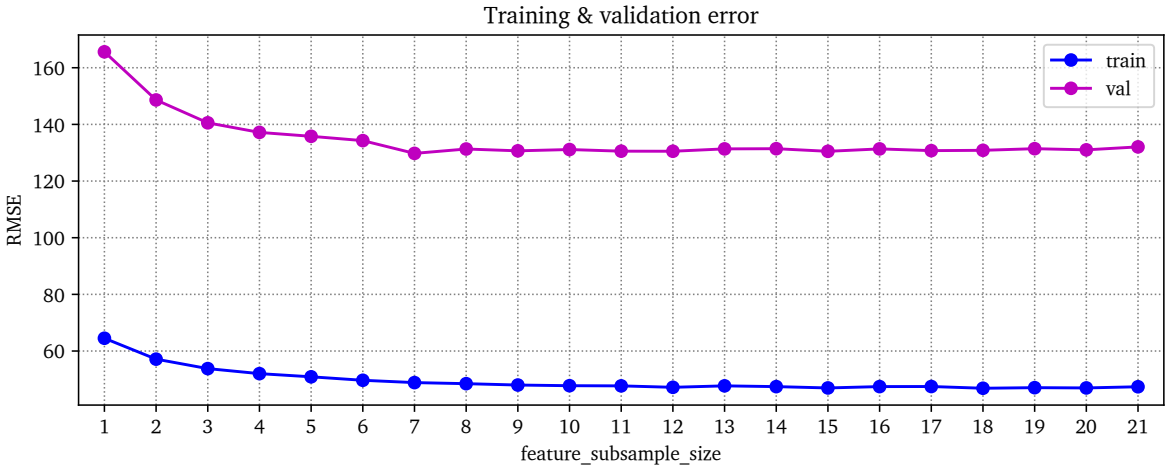

Рис. 3: Зависимость RMSE от максимального числа признаков (для одного дерева в случайном лесе)

Что касается времени работы (рис. 4) — с ростом размерности подвыборки признаков время растет — относительно линейно.

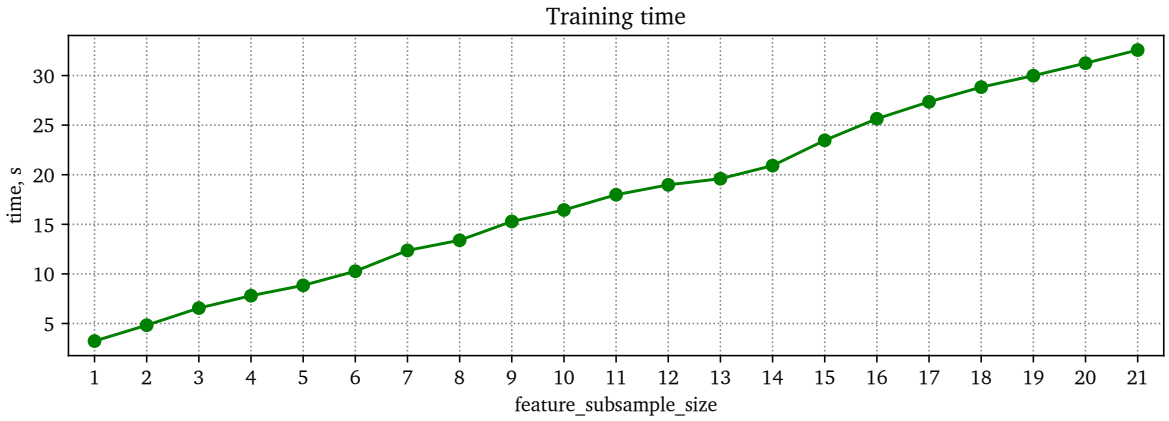

Рис. 4: Зависимость времени обучения случайного леса от максимального числа признаков (для одного дерева)

3.2.3 Глубина дерева

Случайный лес обычно состоит из глубоких переобученных деревьев. И наша задача не стала исключением. рис. 5 показывает, что лучшее качество регрессии достигается на деревьях без ограничений на глубину.

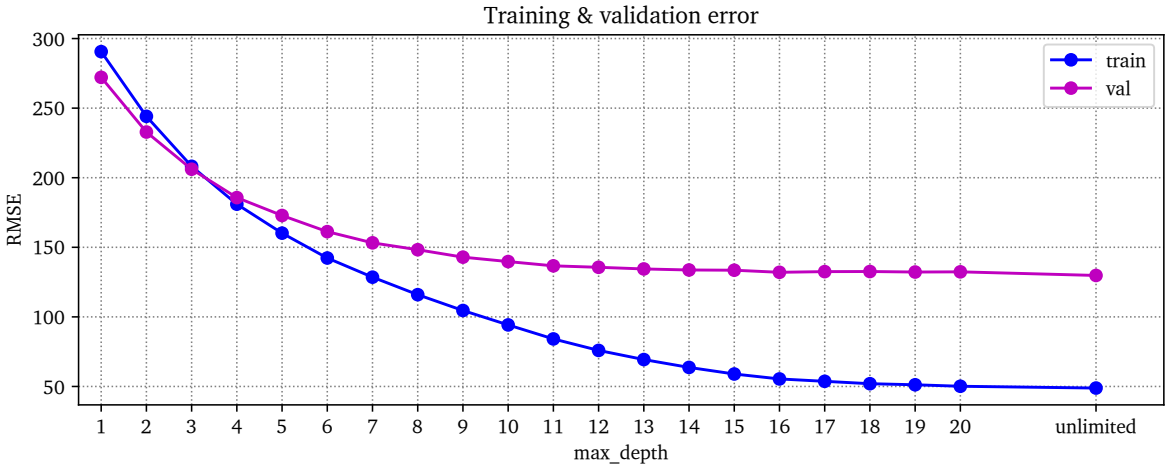

Рис. 5: Зависимость RMSE от глубины одного дерева в случайном лесе

Возможно, при большем объеме выборки или более высокой размерности признакового пространства ограничение на глубину имело бы смысл с точки зрения времени обучения. Но рис. 6 показывает, что в нашей задаче, ограничив глубину, на достаточном уровне (примерно > 13), особого выигрыша в производительности не будет.

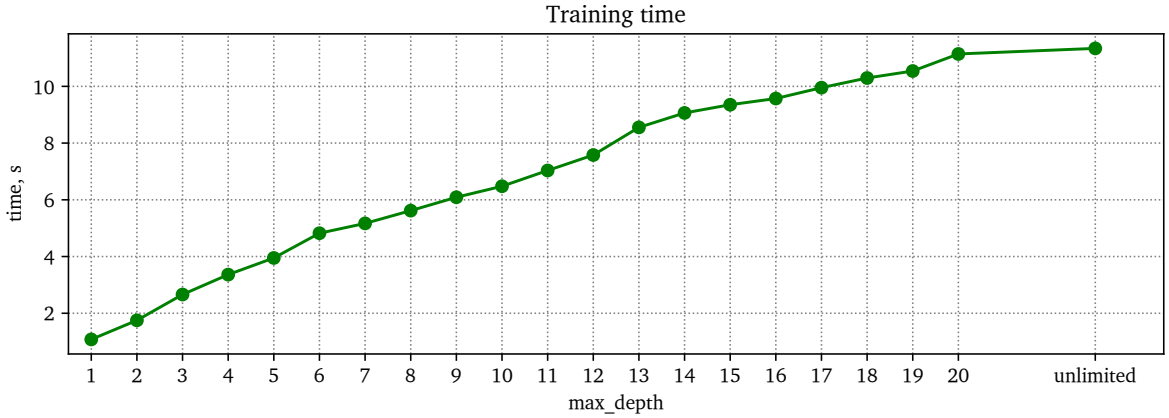

Рис. 6: Зависимость времени обучения случайного леса от глубины одного дерева

3.3 Градиентный бустинг

3.3.1 Количество деревьев и темп обучения

В отличие от случайного леса, в градиентном бустинге базовые алгоритмы не являются независимыми — каждый следующий исправляет ошибки предыдущих.

Поэтому при настройке гиперпараметров количество деревьев не подбирается отдельно, а рассматривается в паре с темпом обучения. Рассмотрим зависимость RMSE на обучающей и контрольной выборках (рис. 7).

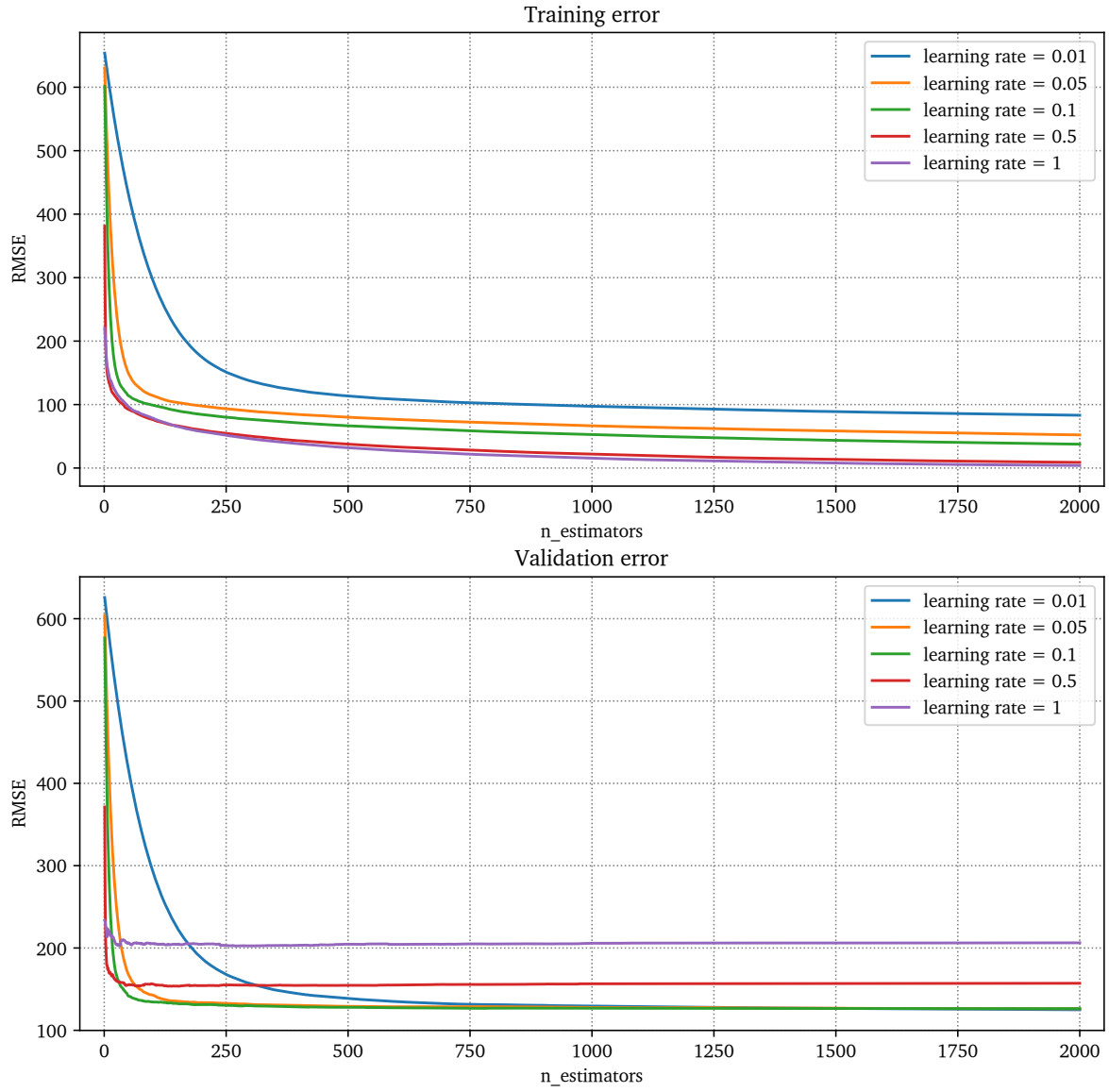

Рис. 7: Зависимость RMSE от числа деревьев и темпа обучения в градиентном бустинге

На обучении (рис. 7) ошибка с ростом числа деревьев монотонно стремится к нулю. Но на тестовой выборке в градиентном бустинге обычно монотонной зависимости нет. При высоком темпе обучения, как видим, качество на валидации действительно начинает ухудшаться с некоторого момента. Но при «умеренном» значении `learning_rate` ошибка, пусть крайне медленно, но продолжает уменьшаться даже после тысячи деревьев. Такое поведение связано с использованием классической реализации бустинга и особенностями задачи.

Если на обучающей выборке ошибка уменьшается с ростом темпа обучения, то на валидации зависимость обратная (на большом числе деревьев). Но для даль-

нейшего проведения экспериментов особого смысла в 2000 деревьев и темпе 0.01 нет — разница в качестве незначительная. Поэтому дальше для следующих пунктов оставим 400 деревьев с `learning_rate` = 0.1. Время работы в нашей реализации по-прежнему линейно растет с числом базовых алгоритмов, не не зависит от темпа обучения[2]. Результат подверждается графиком рис. 8.

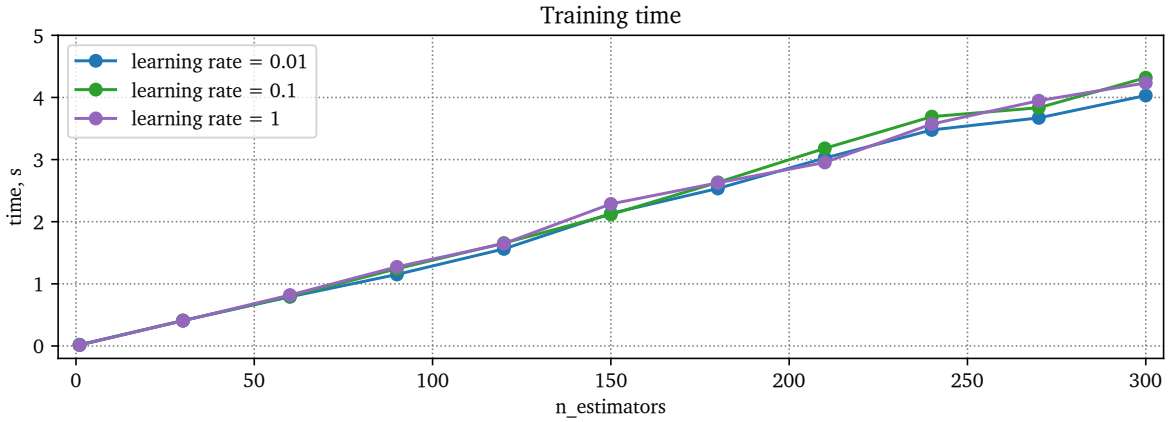

Рис. 8: Зависимость времени обучения градиентного бустинга от числа деревьев и темпа

Итак, рис. 8 показывает, что в нашей реализации бустинга время обучения растет линейно с числом деревьев и не зависит от темпа обучения. Но если сравнить со случайным лесом (рис. 2), то при бустинге обучение происходит быстрее. Но почему? Причина заключается в разных подходах к настройке глубины для бустинга и леса.

3.3.2 Глубина дерева

В отличие от случайного леса, в градиентном бустинге обычно используются неглубокие деревья. Причину можно проиллюстрировать рис. 9. На обучающей бустинг глубоких деревьев очень быстро переобучается, при отсутствии ограничений потери почти нулевые. И такое обучение приводит к некачественной работе на тестовой (в нашем случае валидационной) выборке. Тем не менее проблем с подбором нужной глубины у нас не возникло: видно, что 5 — оптимальное значение. Время работы алгоритма (рис. 10) заметно увеличивается с ростом глубины, хотя, не считая полное отсутствие ограничений, зависимость на рассмотренных значениях можно считать линейной.

3.3.3 Размерность подвыборки признаков для дерева

В данном гиперпараметре снова проявляется отличие от случайного леса. На рис. 3 виден оптимум (на контроле) и стационарное значение RMSE поведение с дальнейшим ростом числа признаков. На рис. 11 ситуация нестабильная. Но на большом числе признаков ошибка больше, чем на «среднем». Значение 12 кажется

[2]Хотя время могло бы уменьшаться с ростом темпа обучения, если добавить критерий останова при отсутствии улучшения качества на валидации.

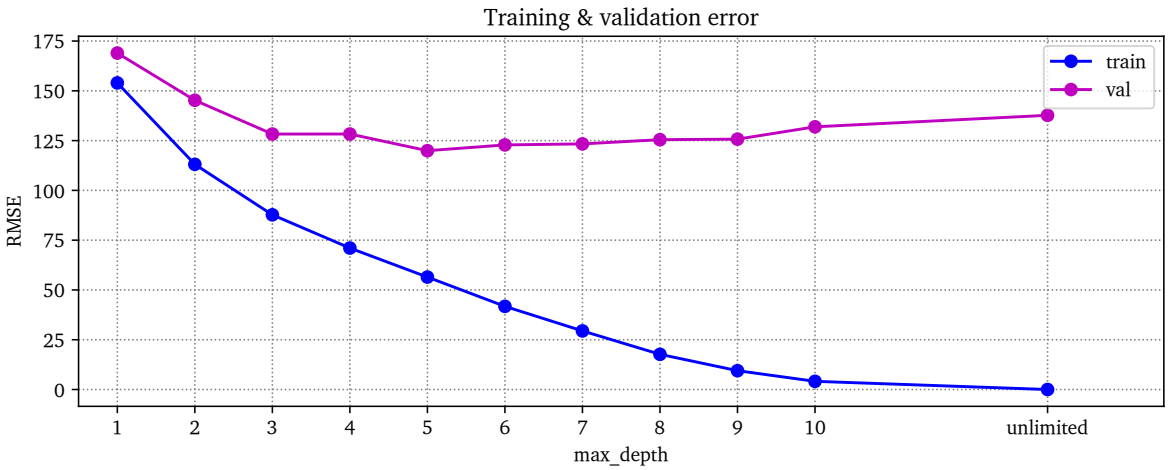

Рис. 9: Зависимость RMSE от глубины одного дерева в градиентном бустинге

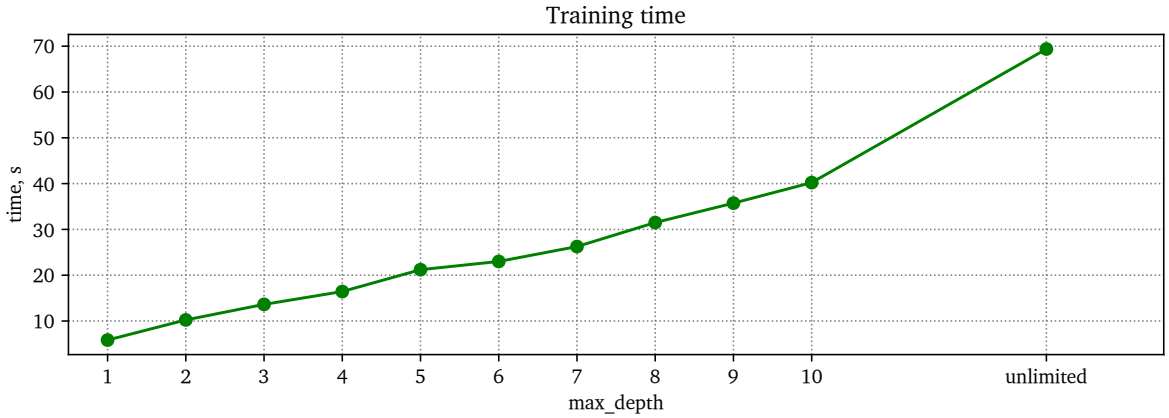

Рис. 10: Зависимость времени обучения градиентного бустинга от глубины одного дерева

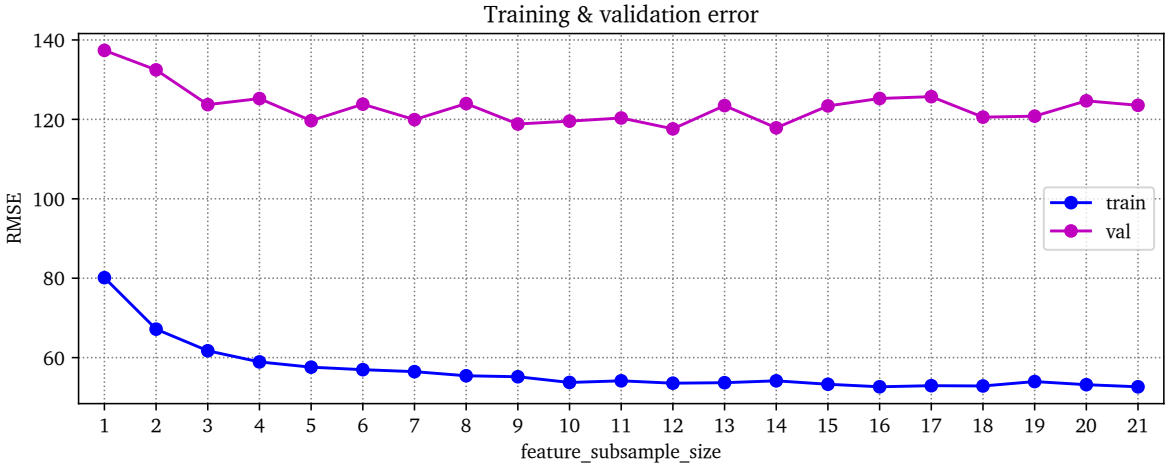

Рис. 11: Зависимость RMSE от максимального числа признаков (для одного дерева в градиентном бустинге)

оптимальным (по валидации), но для более тщательного отбора следовало бы рассмотреть графики он числа деревьев.

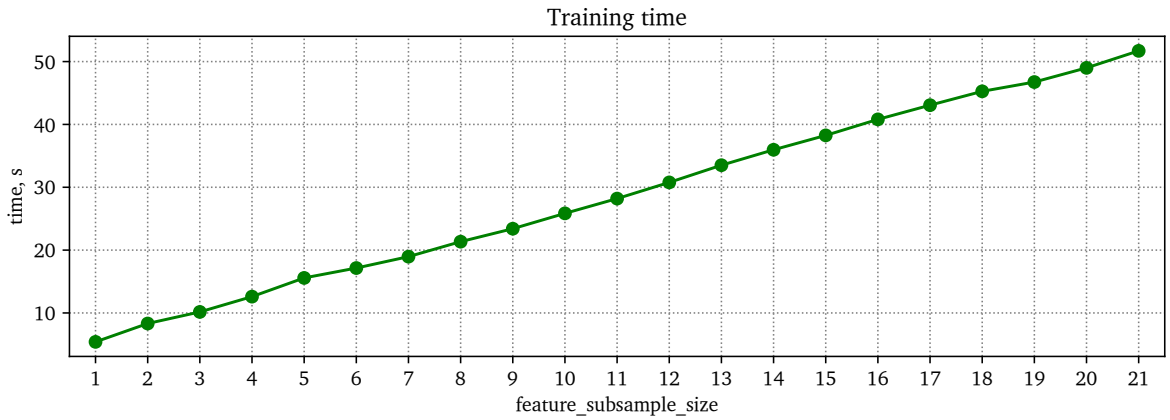

Рис. 12: Зависимость времени обучения градиентного бустинга от максимального числа признаков (для одного дерева)

Время работы можем видеть на рис. 12: здесь достаточно точно прослеживается линейная зависимость.

4 Заключение

По итогам экспериментов можно обучить алгоритмы с лучшими гиперпараметрами (из исследованных, все конкретные значения, обучение и построение графиков указаны в jupyter-ноутбуке, расположенном в репозитории). Для нашей задачи градиентный бустинг, на контроле RMSE < 118, оказался предпочтительней случайного леса с RMSE > 130.

Рис. 13: Ошибки итоговых моделей

В небольшой серии экспериментов мы сравнили влияние гиперпараметров на случайный лес и бустинг над деревьями — отличия проявляются и в диапазоне оптимальных значениях, и в стратегии настройки, и в значимости для качества модели.