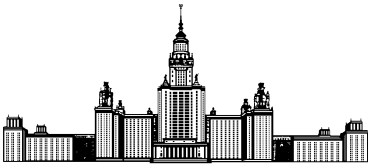
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

ОТЧЁТ ПО ЗАДАНИЮ №3

«Композиции алгоритмов для решения задачи регрессии»

Выполнил:
студент 3 курса 317 группы
Суглобов Кирилл Алексеевич

Москва, 2021

Содержание

Введение

В практическом задании требовалось:

1. Реализовать композиции алгоритмов машинного обучения: случайный лес (RF) и градиентный бустинг над решающими деревьями (GBM)
2. Провести эксперименты с данными моделями
3. Реализовать веб-сервер для удобного взаимодействия с моделями
4. При этом качественно вести проект в Git-репозитории

Выполненное задание доступно в Git-репозитории[1].
Этот отчёт сопровождает пункт 2 задания – проведение экспериментов.

Постановка задачи

Рассматривается задача регрессии с метрикой RMSE:

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

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

Для решения потставленной задачи реализованы две модели – ансамбли решающих деревьев: случайный лес (RF) и градиентный бустинг (GBM). Требовалось

[1] https://github.com/ksuglobov/mmf_ensemble_learning_fall_2021

эмпирическим путём исследовать функцию потерь и время обучения каждого из алгоритмов при варьировании гиперпараметров (последовательно в порядке важности гиперпараметров) [1].

Эксперименты

Особенности реализации моделей:

- В каждую модель добавлен параметр `random_state` – сид, который по умолчанию во обеих моделях = 0, для воспроизводимости экспериментов.
- Указав специальные параметры в функции обучения `return_train_loss` и `return_val_loss` (для этого в функцию обучения ещё надо передавать валидационную выборку), можно возвращать из неё массив значений функции потерь на обучающей и на валидационной выборках, соответственно, на всех количествах деревьев в ансамбле – до `n_estimators`. Это используется для построения графиков функций потерь при переборе количества деревьев: для поиска оптимального `n_estimators` `n_estimators`.
- Время обучения моделей усреднялось по трём независимым итерациям, для большей точности (уменьшения дисперсии) графиков.

Предобработка данных

Данные представляют собой таблицу с информацией о недвижимости. Из неё был извлечён столбец с целевой переменной `price` и определён в соответствующую целевую переменную. Далее в таблице был исправлен формат столбца `date` (на нужный и удобный `datetime`). Этот столбец, по всей видимости, отвечает за дату добавления информации в таблицу: все даты 2014-2015 годов и не связаны со столбцами `build_year` и `renovation_year`, отвечающими за год постройки и год реновации дома, соответственно. Сосредоточенные рядом даты и 2014, 2015 годы могут дать ложную корреляцию с целевой переменной, так что во избежание ложной оценки из столбца `date` были сделаны 3 другие, отвечающие за день (`day`), месяц (`month`) и день недели (`dayofweek`) добавления данных в таблицу. После предобработанная выборка, исключая оригинальный столбец `date`, была перемешана и разделена на обучающую и на отложенную выборки в соотношении 80/20, соответственно. Таким образом, в предобработанной обучающей выборке оказался 21 признак.

Случайный лес

Зависимость RMSE от количества деревьев в RF

В случайном лесе производится усреднение решающих деревьев, независимых (по возможности) базовых алгоритмов. Было проведено обучение модели, возвращающее функции потерь на обучении и контроле на всех деревьях от 1 до 1000. На рис. 1 видно, что на обучении потери монотонно уменьшаются и что на контроле

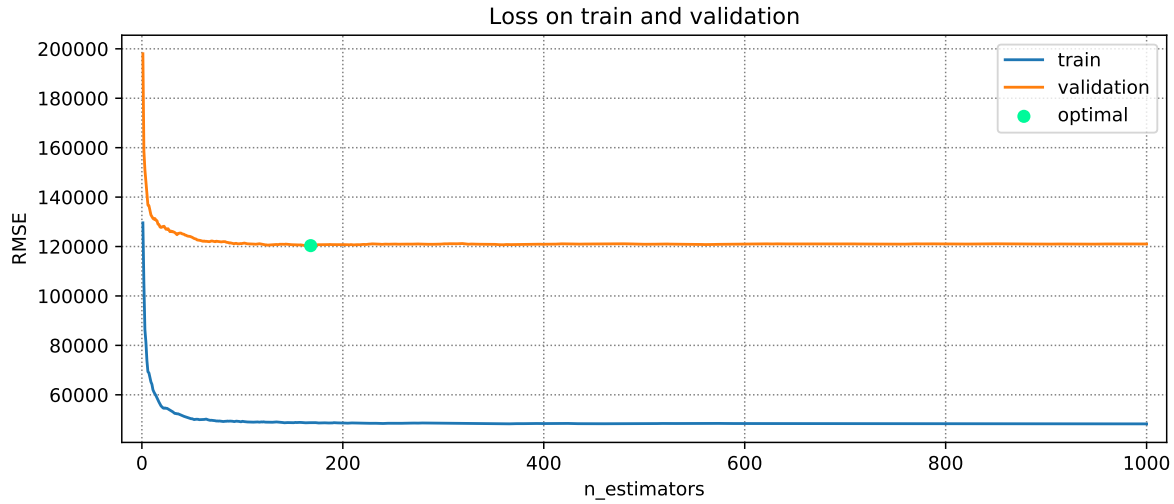


Рис. 1: Зависимость RMSE от количества деревьев в RF

потери уменьшаются до достижения оптимального числа деревьев, а после асимптотически возрастают из-за переобучения. Оптимальное количество деревьев в алгоритме при всех прочих гиперпараметрах, установленных по умолчанию, = 168. Время обучения случайного леса линейно зависит от количества деревьев,

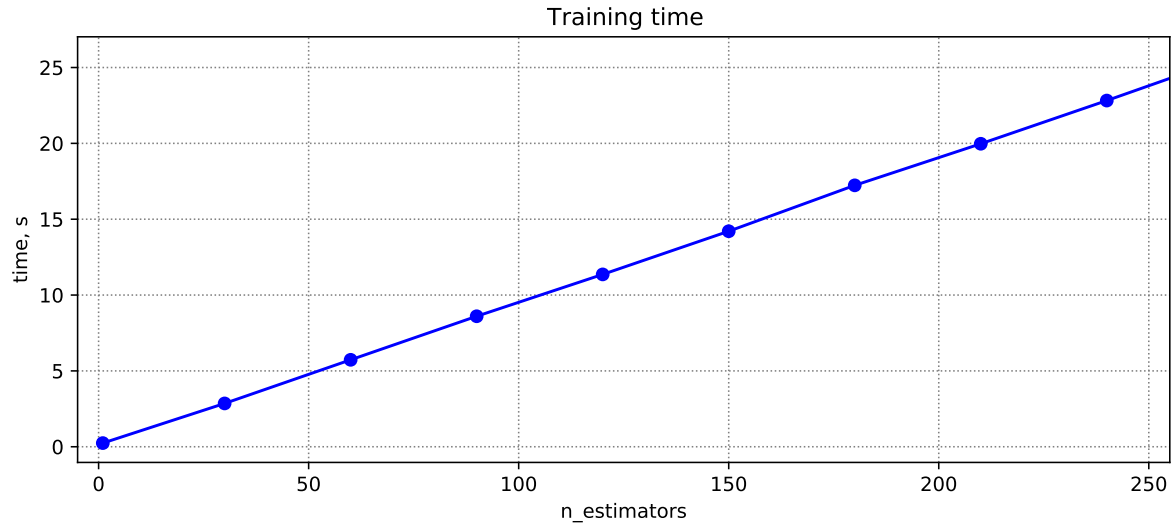


Рис. 2: Зависимость времени обучения RF от числа деревьев

это очевидно и было показано на рис. 2, где перебор количества деревьев вёлся от 1 до 250.

Размерность подвыборки признаков для одного дерева

qЭтот параметр второй по важности после количества деревьев в случайном лесе. В задаче регрессии по умолчанию задают этот параметр как треть от числа всех признаков, либо все признаки выборки. Но в данной задаче оптимальная размерность подвыборки признаков для одного дерева оказалась = 12. На графике (рис. 3) видно, что на обучении ошибка монотонно убывает с увеличением числа

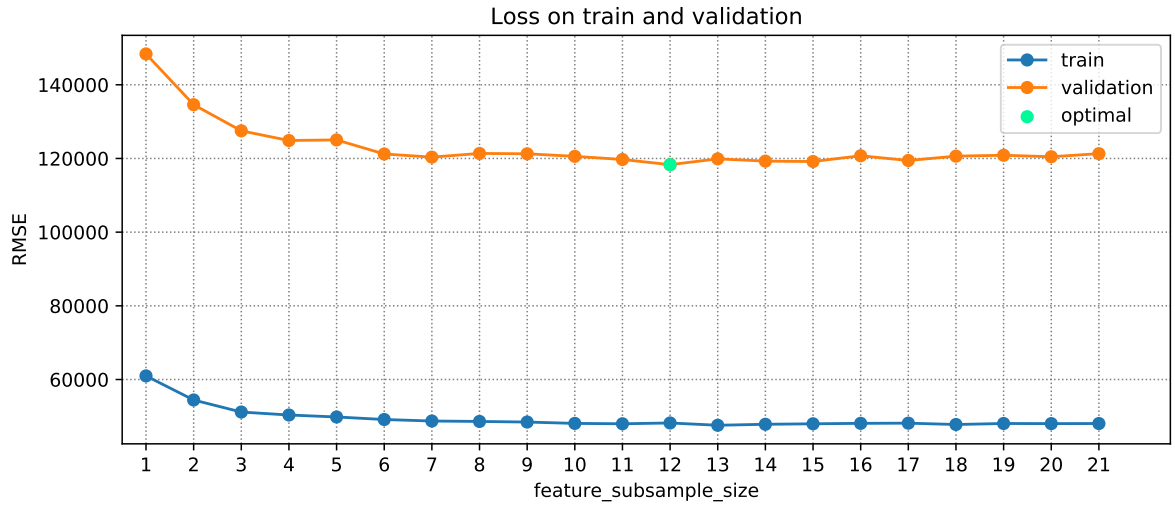


Рис. 3: Зависимость RMSE от максимального числа признаков для одного дерева в RF

признаков для одного дерева. А на контроле – сначала убывает, а после немного колеблется, но оптимум ясно виден. А график (рис. 4) говорит о линейной за-

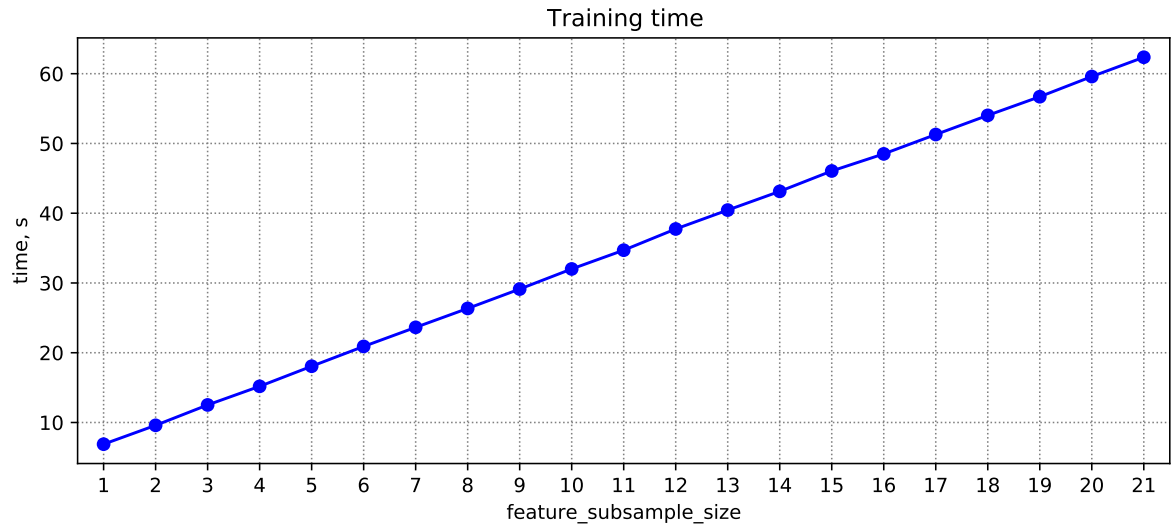


Рис. 4: Зависимость времени обучения RF от максимального числа признаков для одного дерева

висимости времени обучения RF от максимального числа признаков для одного дерева.

Максимальная глубина дерева

Обычно деревья в RF глубокие и переобученные [2], и график рис. 5 подтверждает, что наименьшие потери достигается на деревьях без ограничений на глубину. При этом «достаточно хорошее» качество достигается при глубине в 18 деревьев.

Можно ограничивать глубину деревьев с позиции уменьшения времени обучения, при этом увеличив выборку или число признаков для уменьшения потерь

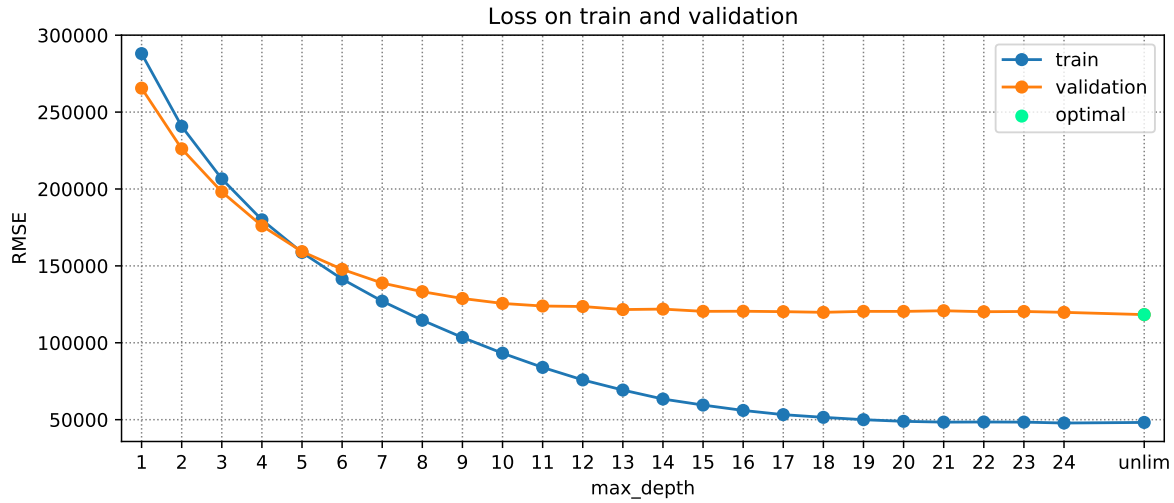


Рис. 5: Зависимость RMSE от максимально1 глубины одного дерева в RF

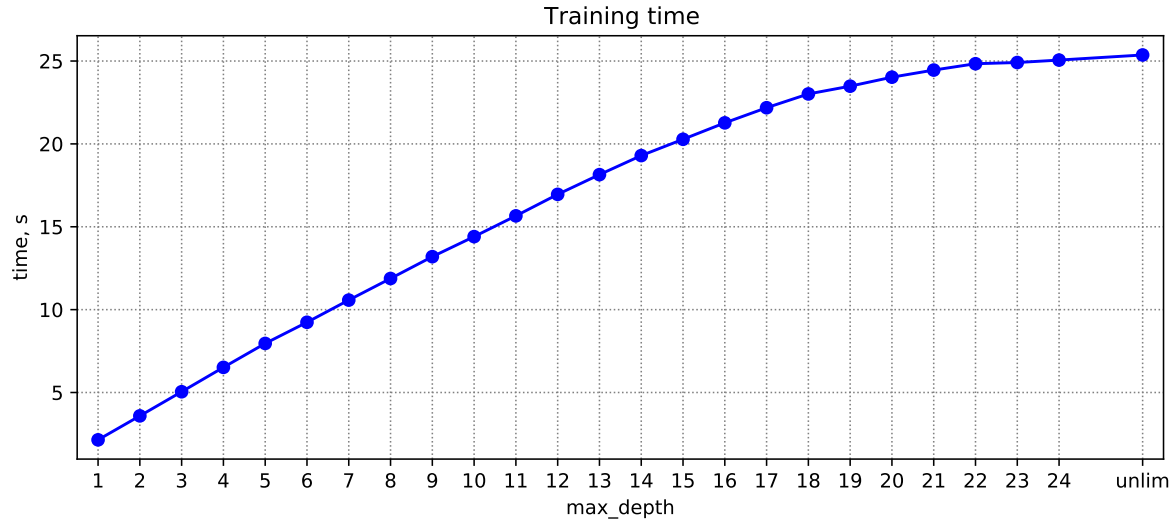


Рис. 6: Зависимость времени обучения RF от максимальной глубины одного дерева

качества. Но из графика рис. 6 ясно, что, даже ограничив глубину дерева на уровне = 18 (с «достаточно хорошим» качеством), прирост производительности совсем небольшой.

Градиентный бустинг

Количество деревьев в ансамбле и темп обучения

В градиентном бустинге базовые алгоритмы (решающие деревья) не являются независимыми: каждый следующий алгоритм исправляет ошибки предыдущих с соответствующим темпом обучения `learning_rate` (каждый новый алгоритм добавляется в композицию с коэффициентом $\alpha \cdot$ `lerning_rate`). Значит, зависящие друг от друга гиперпараметры – количество деревьев и темп обучения – нужно подбирать вместе. Например, смотреть зависимость RMSE на обучающей и контрольной выборках (рис. 7) от количества деревьев, перебирая темп обучения.

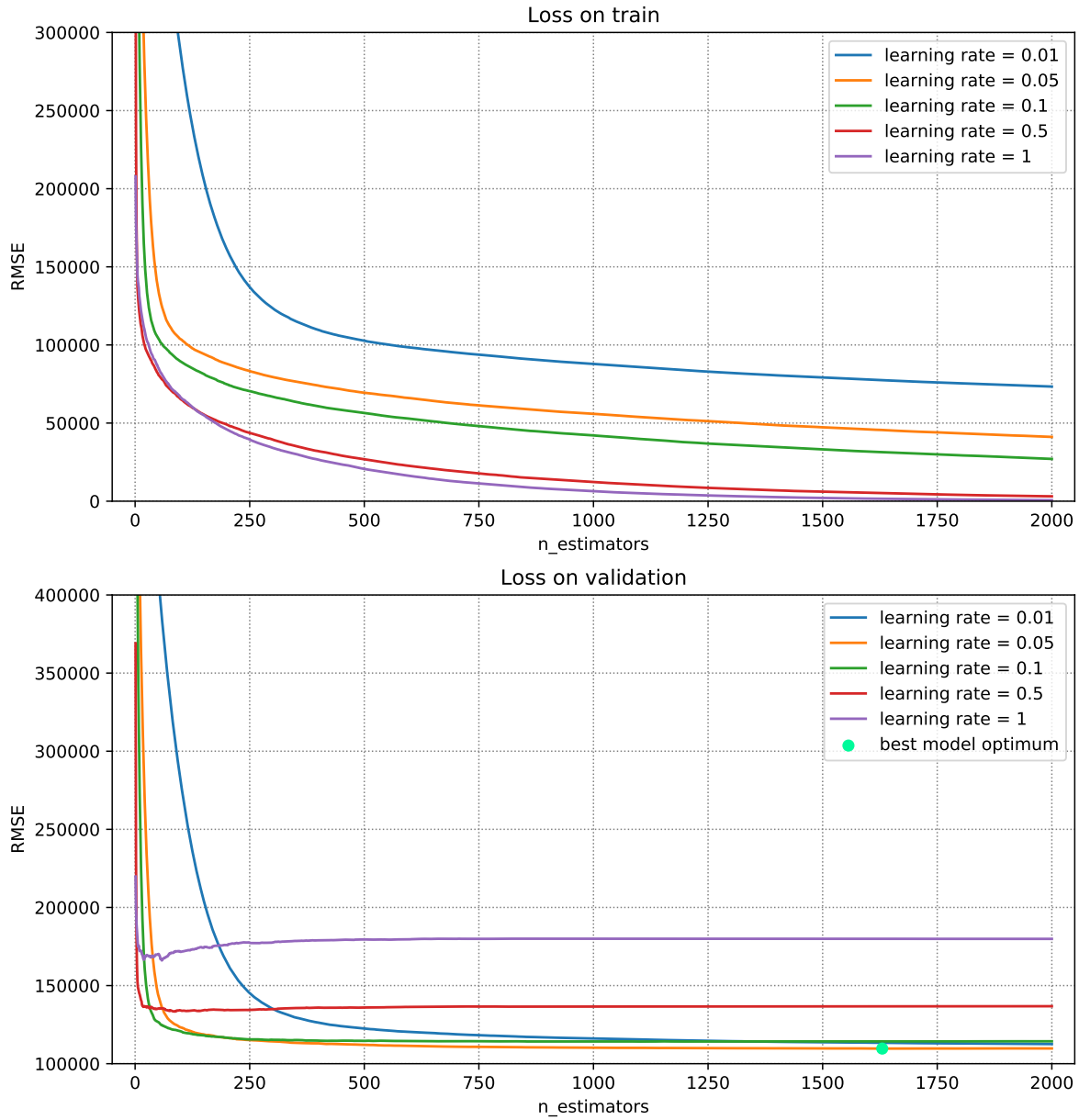


Рис. 7: Зависимость RMSE от числа деревьев и темпа обучения в GBM

По графику (рис. 7) видно, что ошибка на обучении монотонно убывает и стремится к нулю с ростом количества деревьев при любом темпе. А на валидации в градиентном бустинге поведение функции потерь другое: нет монотонности и переобучение с какого-то значения деревьев при большом темпе обучения и монотонное асимптотическое убывание функции потерь до меньших потерь (но тоже до определённого значения деревьев, а потом переобучение) при малом темпе, как видно по (рис. 7). Оптимум – `n_estimators` = 1629 (из экспериментов) и `learning_rate` = 0.05. Но это много деревьев, так что из соображений экономии времени обучения, можно взять `n_estimators` = 750 с очень близким, «достаточным» качеством. Ниже будет показана целесообразность сокращения числа деревьев.

Как видно по рис. 8, все графики (по графику на каждый темп) времени обу-

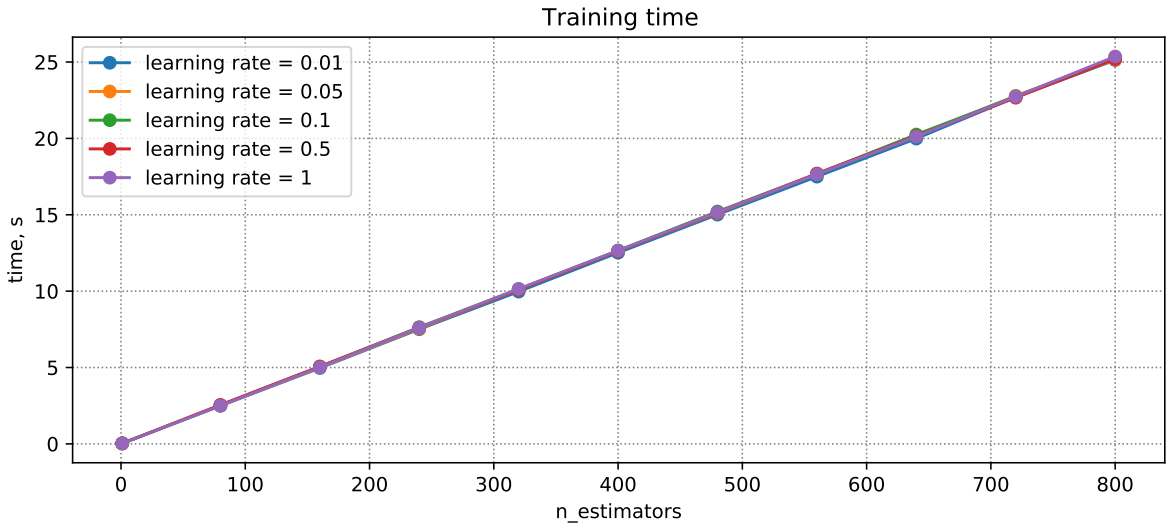


Рис. 8: Зависимость времени обучения GBM от числа деревьев и темпа

чения линейны и почти полностью накладываются. И при данном масштабе это показывает, что: время обучения градиентного бустинга линейно зависит от количества деревьев в ансамбле и не зависит от темпа обучения. Значит, выгодно почти не потерять в качестве, но выиграть более, чем в два раза, время обучения – взять 750 деревьев вместо 1629. Также, обучение GBM быстрее, чем у RF из-за разной настройки глубины в этих моделях.

Таким образом, оптимальные параметры: `n_estimators` = 750, `learning_rate` = 0.05.

Максимальная глубина дерева

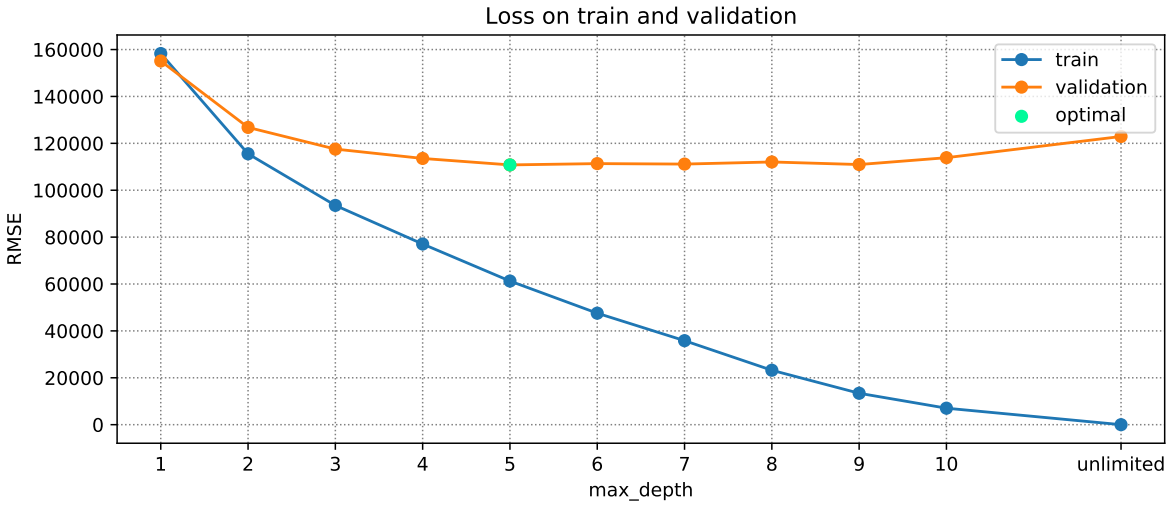


Рис. 9: Зависимость RMSE от максимальной глубины одного дерева в GBM

В градиентном бустинге используются неглубокие деревья [1], потому что на глубоких деревьях эта модель быстро переобучается. Как видно по графику рис. 9, на обучении у GBM быстро уменьшаются потери вплоть до ≈ 0 на неограниченных

деревьях. Тем временем, на валидации видно, что алгоритм переобучился. График показывает, что оптимальное значение максимальной глубины деревьев = 5.

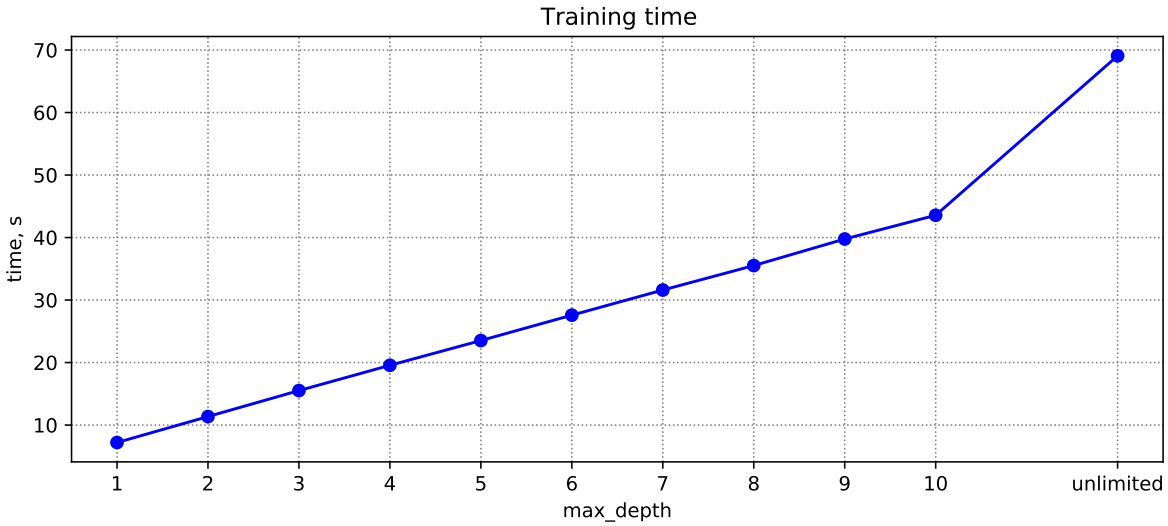


Рис. 10: Зависимость времени обучения GBM от максимальной глубины одного дерева

На (рис. 10) показано, что время обучения увеличивается с ростом глубины линейно и быстро при ограничении на максимальную глубину. Только время обучения неограниченных деревьев увеличивается скачком.

Размерность подвыборки признаков для одного дерева

Рис. 11: Зависимость RMSE от максимального числа признаков для одного дерева в GBM

В градиентном бустинге, в отличие от случайного леса, наоборот: размерность подвыборки признаков для одного дерева менее важена, чем глубина дерева. К тому же видно различие градиентного бустинга рис. 11 со случайным лесом рис. 3

на этом эксперименте. Графики на обучении и на валидации у GBM менее стабильные, чем у RF, более «дёрганные»: на большом числе признаков ошибка больше, чем при оптимуме, примерно $\frac{3}{4}$ от всех признаков: оптимальная размерность подвыборки признаков для одного дерева = 16.

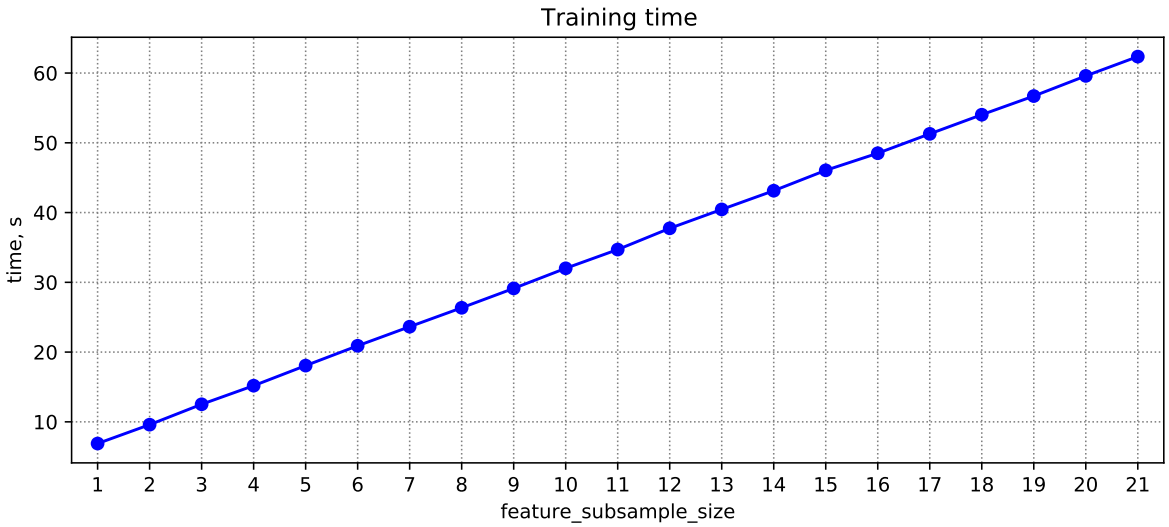


Рис. 12: Зависимость времени обучения GBM от максимального числа признаков для одного дерева

На рис. 12 явно линейная зависимость времени обучения GBM от числа признаков для одного дерева.

Заключение

На рис. 13 представлены потери итоговых моделей.

Таким образом, получили, что при всех лучших подобранных параметрах, кроме `n_estimators`, в моделях существуют `n_estimators`, которые ещё лучше:

- 100 вместо 168 в RF
- 1536 вместо 750 в GBM (750 мы брали оптимальным с точки зрения более быстрого обучения при сравнительно небольшом проигрыше в качестве)

Пояснение:

1. Сначала были найдены оптимальные `n_estimators` для RF и для GBM при их прочих **стандартных (значения по умолчанию)** параметрах.
2. Далее были подобраны оптимальные значения остальных параметров (последовательно, в порядке их важности).
3. Заново были найдены `n_estimators` для RF и для GBM при оптимально подобранных в пункте 2 параметрах.

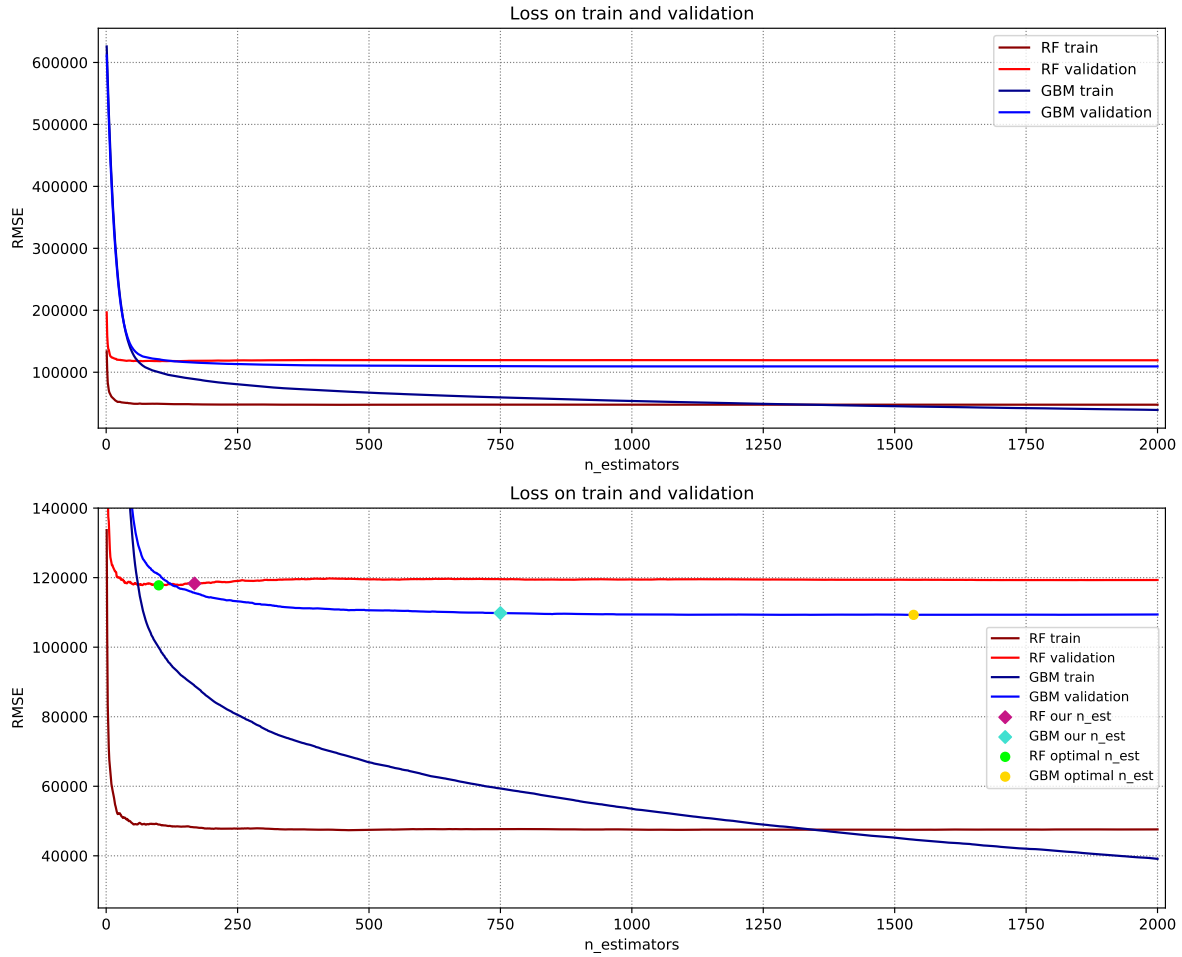


Рис. 13: Зависимость RMSE итоговых моделей RF и GBM на обучении и валидации

Но всё же брать 1536 деревьев в GBM – это более чем в два раза увеличить время работы алгоритма по сравнению с теми же 750-ю деревьями при улучшении качества всего на $\left(1 - \frac{109296.299}{109819.220}\right) \cdot 100\% \approx 0.48\%$.

То есть в production гораздо лучше взять модель, которая работает в два раза быстрее development-версии, но чуть (или даже не чуть) хуже качеством. Вроде бы так и делают. А судя по графику потерь GBM на валидации, как раз примерно с 750-ти деревьев и начинается **достаточный** уровень качества модели.

Тогда остановимся на том, что:

- У RF оптимальное число деревьев = 100 и для development-версии, и для production-версии.
- У GBM оптимальное число деревьев = 1536 для development-версии и = 750 для production-версии.

Прочие оптимальные параметры одинаковы и не изменялись.

Таким образом, за небольшое количество экспериментов было исследовано влияние гиперпараметров на случайный лес и на градиентный бустинг над решающими деревьями. Отличаются стратагии настройки моделей и значимости гиперпараметров, поэтому отличаются и оптимальные значения.

Экспериментальным путём были подобраны оптимальные гиперпараметры моделей случайного леса и градиентного бустинга для нашей задачи:

- Лучшие development-параметры:
  - RF (*RMSE* $\approx$ 117999):
    * `n_estimators` = 100
    * `features_subsample_size` = 12
    * `max_depth` = *None*
  - GBM (*RMSE* $\approx$ 109296):
    * `n_estimators` = 1536
    * `learning_rate` = 0.05
    * `max_depth` = 5
    * `features_subsample_size` = 16
- Лучшие production-параметры:
  - RF (*RMSE* $\approx$ 117999):
    * `n_estimators` = 100
    * `features_subsample_size` = 12
    * `max_depth` = *None*
  - GBM (*RMSE* $\approx$ 109819):
    * `n_estimators` = 750
    * `learning_rate` = 0.05
    * `max_depth` = 5
    * `features_subsample_size` = 16

References

[1] Evgeniy Sokolov. *Machine Learning, HSE FCS, seminar 10*. URL: https://github.com/esokolov/ml-course-hse/blob/master/2019-fall/seminars/sem10-gbm.ipynb.

[2] Alexander Dyakonov. *Random Forest*. URL: https://dyakonov.org/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/.