Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по практическому заданию №3

«Композиции алгоритмов для решения задачи регрессии»

По курсу «Практикум на ЭВМ»

Выполнил:
студент 317 группы
Дьяков И. А.

Москва
2023

Содержание

1 Постановка задачи

Данное задание было направлено на изучение алгоритмов композиций. Были рассмотрены такие подходы к ансамблированию, как случайный лес и градиентный бустинг. Кроме того, задание было направлено на освоение систем контроля версий и создание собственного веб-интерфейса. Весь код должен был располагаться в приватном репозитории на сайте github.com.
Таким образом, данное задание представляет собой создание полноценного программного продукта, который затем необходимо выложить в открытый доступ на платформе dockerhub.

Основными задачами работы являлись:

1. Написание собственных реализаций методов ансамблирования **случайный лес** и **градиентный бустинг** на языке `Python`.
2. Проведение экспериментов с реализованными моделями на данных о продажах недвижимости **House Sales in King County, USA**. Написание отчета о проведенных экспериментах.
3. Написание веб-сервера, реализующего следующий функционал:
   - Создание новой модели с указанием типа и гиперпараметров
   - Обучение модели на произвольном датасете в формате `.csv`
   - Возможность передачи валидационного датасета в формате `.csv`
   - Просмотр информации о созданной модели, включая гиперпараметры модели, датасет, на котором она обучалась, значения функции потерь на обучающей и валидационной выборках
   - Выполнение предсказаний с использованием ранее обученной модели на датасете в формате `.csv`
   - Создание docker-контейнера для развертки веб-сервера

2 Описание экспериментов

Все эксперименты проводились на датасете **House Sales in King County, USA**, ссылка.

2.1 Предобработка данных

Предобработка данных включала в себя следующие шаги:

1. удаление уникальных id недвижимости для предотвращения переобучения модели на данных, не связанных с предсказывающейся величиной
2. преобразование даты в секунды для работы с этим полем как с числом

Далее выборка разбивалась на обучающую и контрольную подвыборки, в соотношении 4 к 1.

2.2 Исследование алгоритма случайный лес

Изучалась зависимость **RMSE** на отложенной выборке и время работы алгоритма в зависимости от следующих гиперпараметров:

- количество деревьев в ансамбле
- размерность подмножества признаков для одного дерева
- максимальная глубина дерева

Графики зависимости функции потерь и времени обучения модели от гиперпараметров на каждой итерации приведены в приложении А.

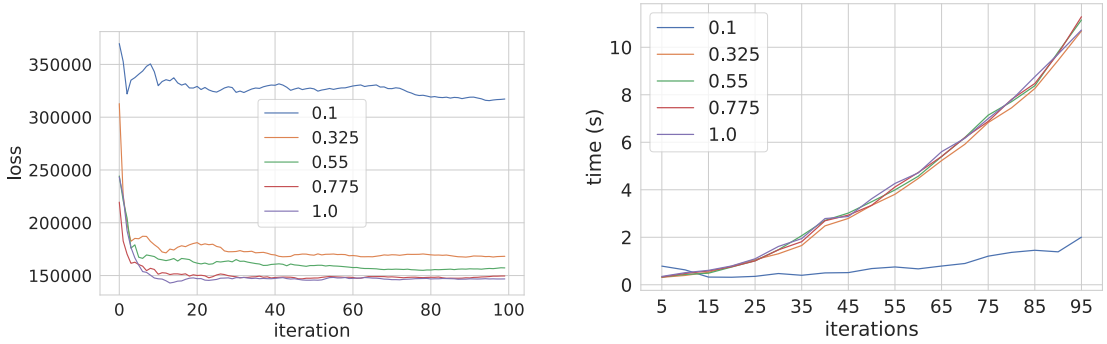
В целом на графиках прослеживается стабилизация функции потерь при увеличении количества деревьев в ансамбле. Так как предсказания деревьев усредняются, с каждой итерацией предсказание отдельного дерева вносит все меньше вклада в предсказание ансамбля. Благодаря беггингу и семплированию признаков, каждое дерево получается не похожим на предыдущие, что ведет к сглаживанию выбросов среди предсказаний и более стабильным предсказаниям.

При низкой глубине дерева и большом подмножестве признаков обучающие подвыборки становятся больше похожи друг на друга. Это ведет к увеличению ковариации в предсказаниях простейших моделей и увеличению стабилизировавшегося значения функции потерь в сравнении с выбором меньших подмножеств признаков.

При росте глубины деревьев в ансамбле модель начинает сходиться быстрее, однако при слишком большой глубине отдельные деревья переобучаются, что ведет к более долгому процессу стабилизации функции потерь и более длительному обучению до получения стабильной модели. Таким образом, время обучения получается оптимальным при выборе не слишком маленькой, но и не

слишком большой максимальной глубины дерева.

Стоит, однако, отметить, что, как было написано выше, при достаточном количестве простейших моделей ансамбль с большой глубиной деревьев стабилизируется за счет усреднения результатов переобученных деревьев и может демонстрировать лучшее качество по сравнению с деревьями ограниченной глубины. Для проверки данного утверждения было изучено поведение случайного леса в случае, когда глубина деревьев не ограничена (см. рис. 1).



(a) Зависимость функции потерь от размерности подвыборки признаков для одного дерева

(b) Зависимость времени обучения от размерности подвыборки признаков для одного дерева

Рис. 1: Время обучения и качество модели при максимальной глубине дерева

Представленные графики подтверждают, что при увеличении глубины деревьев в случайном лесу его качество только возрастает. Также заметим, что при отсутствии ограничения на глубину дерева становится выгодно брать большие подмножества признаков, так как достаточно глубокие деревья способны выделить из них больше полезной информации по сравнению с неглубокими деревьями. При этом качество модели не падает, так как беггинга хватает для создания достаточно разных простейших моделей.

Сравнивая данные графики с графиками, представленными в приложении А, можно сделать вывод, что при малой глубине деревьев размер подмножества признаков для одного дерева незначительно влияет на производительность. Для ансамблей с ограниченной глубиной дерева видна зависимость производительности от глубины дерева.
При неограниченной глубине деревьев, уменьшение подмножества признаков позволяет сильно сократить время обучения моделей, но, как уже было отмечено, качество модели при слишком низком размере подвыборки обучающих признаков получается значительно хуже, чем при больших размерах подвыборки.

2.3 Исследование алгоритма градиентный бустинг

Изучалась зависимость **RMSE** на отложенной выборке и время работы алгоритма в зависимости от следующих гиперпараметров:

- количество деревьев в ансамбле
- размерность подмножества признаков для одного дерева
- максимальная глубина дерева
- выбранный `learning_rate` (каждый новый алгоритм добавляется в ансамбль с коэффициентом $\alpha \cdot$ `learning_rate`)

Графики зависимости функции потерь и времени обучения модели от гиперпараметров на каждой итерации приведены в приложении Б.

Так как каждое новое дерево в ансамбле исправляет ошибки предыдущего дерева, график функции потерь для метода градиентного бустинга отличается от графика функции потерь для метода случайного леса — так как каждое решение прибавляется к результату, функция потерь для градиентного бустинга при правильно подобранном параметре `learning_rate` равномерно уменьшается, в то время как у графика случайного леса можно наблюдать колебания функции потерь около асимптотического значения.

При слишком низком размере подмножества признаков наблюдается недообучение ансамбля. Это объясняется тем, что тех признаков, которые видит каждое дерево по отдельности, недостаточно, чтобы осознать связь между признаками в датасете. При большом размере подмножества признаков ансамбль может начать переобучаться при более низком `learning_rate`, чем при меньших размерах подмножества признаков.

При слишком низких значениях параметра `learning_rate` ансамбль слишком медленно приближается к решению, что приводит к большим затратам по времени и/или низкому качеству итогового ансамбля. С ростом параметра наблюдается уменьшение времени сходимости модели, однако при слишком больших значениях данного параметра устанавливающееся качество ансамбля начинает падать.

Увеличение глубины дерева ведет к улучшению качества ансамбля, но при этом скорость обучения модели быстро растет (в ~ 2 раза при увеличении глубины с 7 до 10). При неограниченной глубине дерева (см. приложение В) мы наблюдаем худшее качество, чем при ограниченной глубине. Таким образом, в отличие от случайного леса, градиентный бустинг склонен к переобучению при увеличении глубины дерева.

Отметим также интересный результат для неограниченной глубины дерева — при довольно большом подмножестве признаков, единичном `learning_rate`

и неограниченной глубине дерева, первая модель уже довольно хорошо приближает обучающую выборку (много слагаемых в функции потерь обращаются в ноль), в результате чего обучение происходит значительно быстрее, чем при других показателях `learning_rate`. Это происходит потому, что при вычислении шага градиентного спуска параметр α является результатом вспомогательной задачи оптимизации

$$\alpha_t = \arg\min_{\alpha>0} \sum_{i=1}^{l} \mathbb{L}(a_{t-1,i} + \alpha b_t(x_i), y_i)$$

где a_{t-1} — приближение решения на шаге $t-1$, а b_t — базовый алгоритм, обученный на шаге t. То есть при шаге, равном α, функция потерь будет наименьшей. При ограниченной глубине дерева этого не происходит по понятной причине недообучения слишком примитивной базовой модели.

3 Заключение

В заключение хочется отметить, что оба метода ансамблирования, рассмотренные в отчете, подходят для решения широкого класса задач. Несмотря на то, что интуитивно деревья — модели для решения задач классификации, они без проблем справляются с рассмотренной в отчете задачей регрессии.
Ансамблирование — мощный подход для улучшения качества работы примитивного алгоритма, при этом оно интуитивно понятно и просто для реализации.

Список литературы

[1] *Воронцов К. В.* Линейные ансамбли. — http://www.machinelearning.ru/wiki/images/3/3a/Voron-ML-Compositions1-slides.pdf — 2021.

[2] *Воронцов К. В.* Продвинутые методы ансамблирования. — http://www.machinelearning.ru/wiki/images/2/21/Voron-ML-Compositions-slides2.pdf — 2021.

[3] учебник по машинному обучению ШАД. — https://education.yandex.ru/handbook/ml

Приложение А

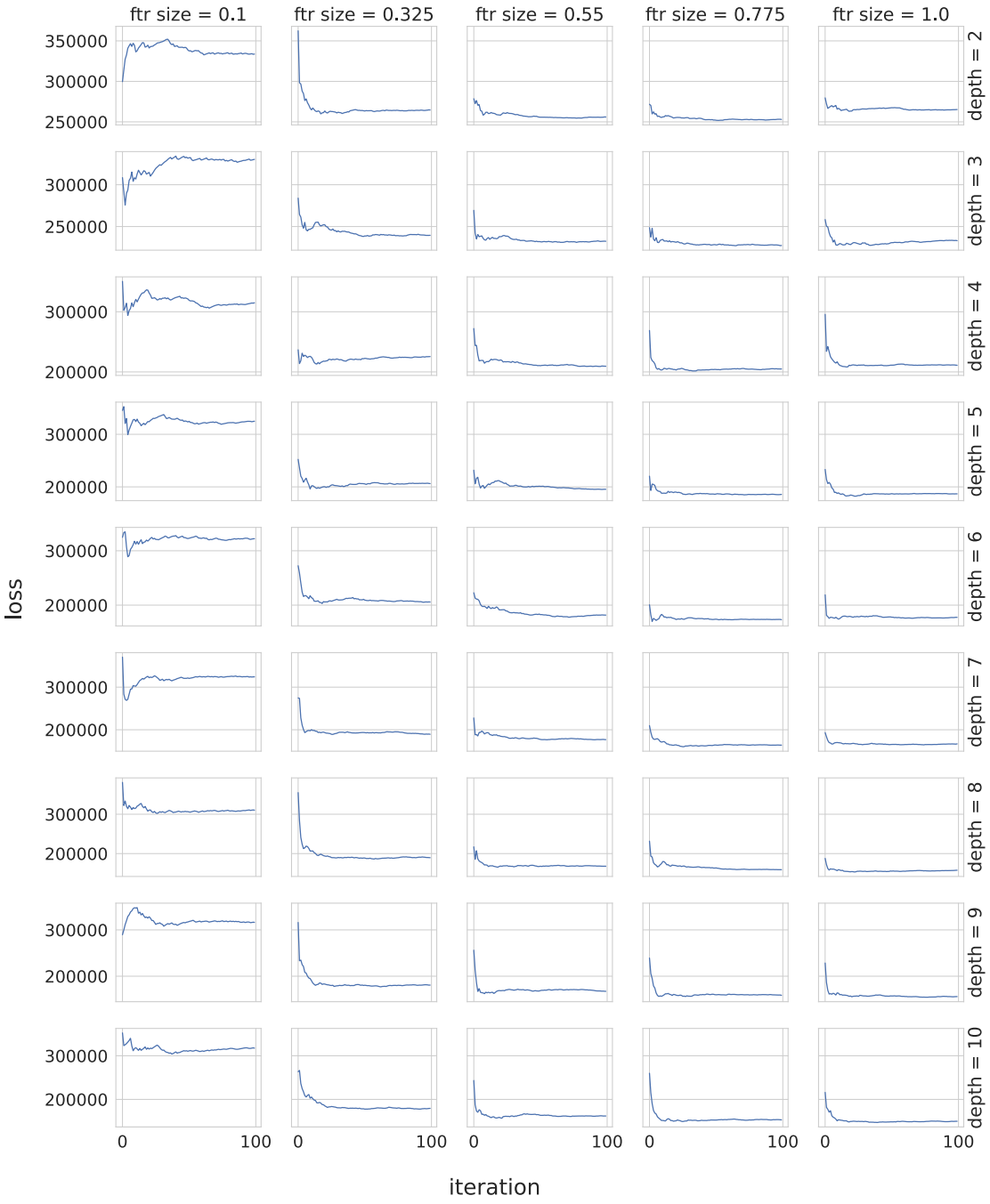


Рис. 2: Зависимость значения функции потерь модели случайный лес от гиперпараметров

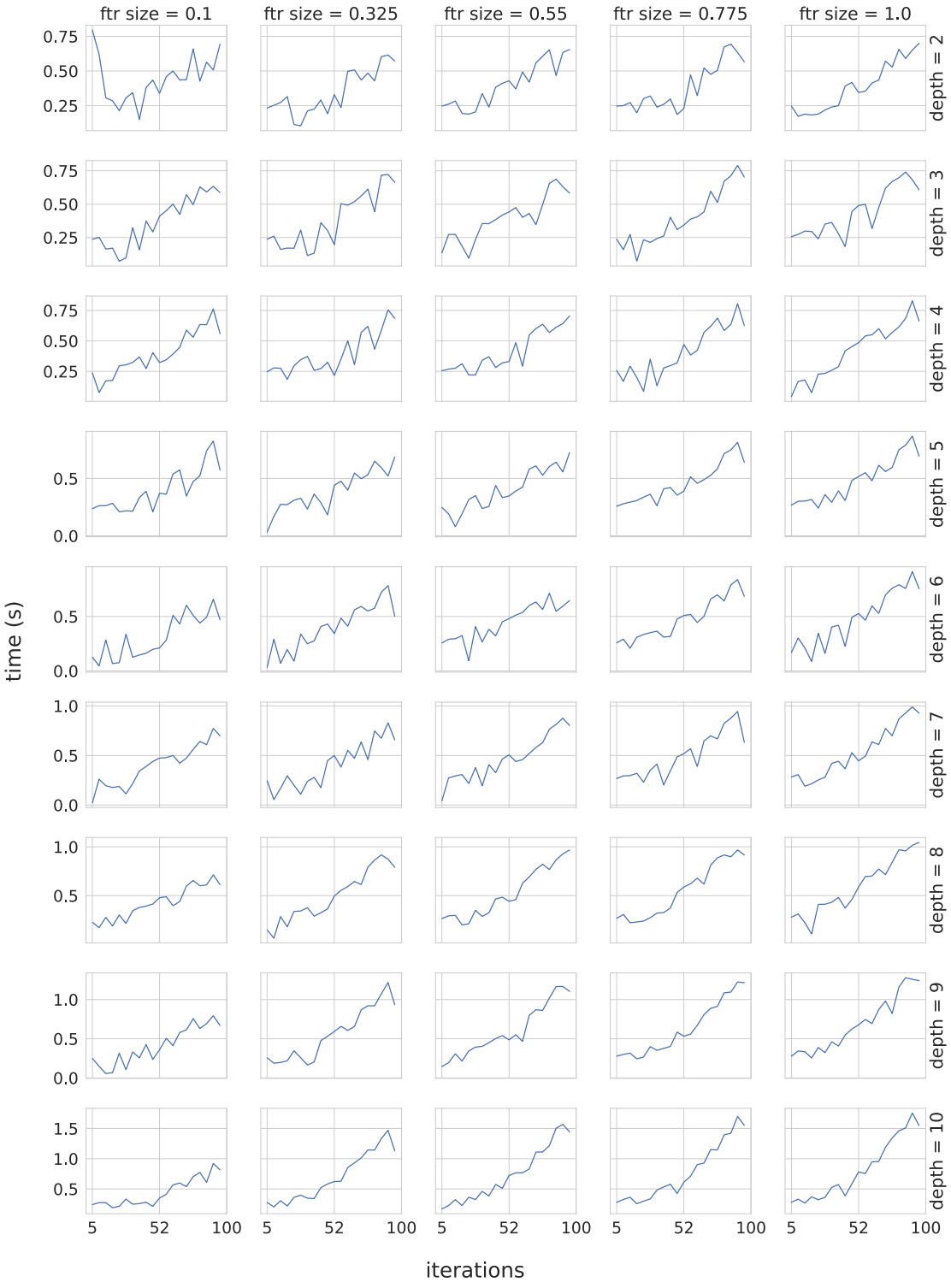


Рис. 3: Зависимость времени обучения модели случайный лес от гиперпараметров

Приложение Б

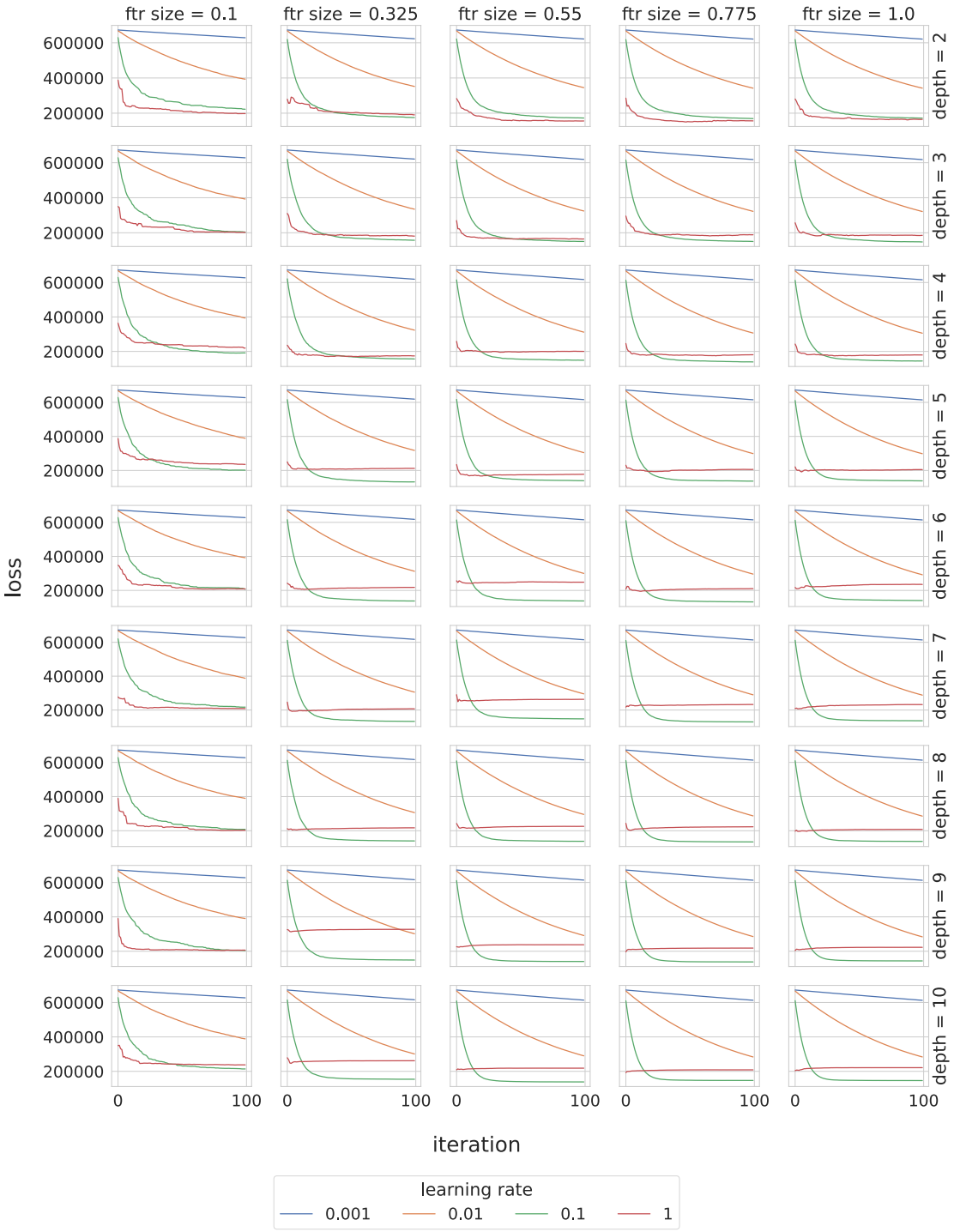


Рис. 4: Зависимость значения функции потерь модели градиентный бустинг от гиперпараметров

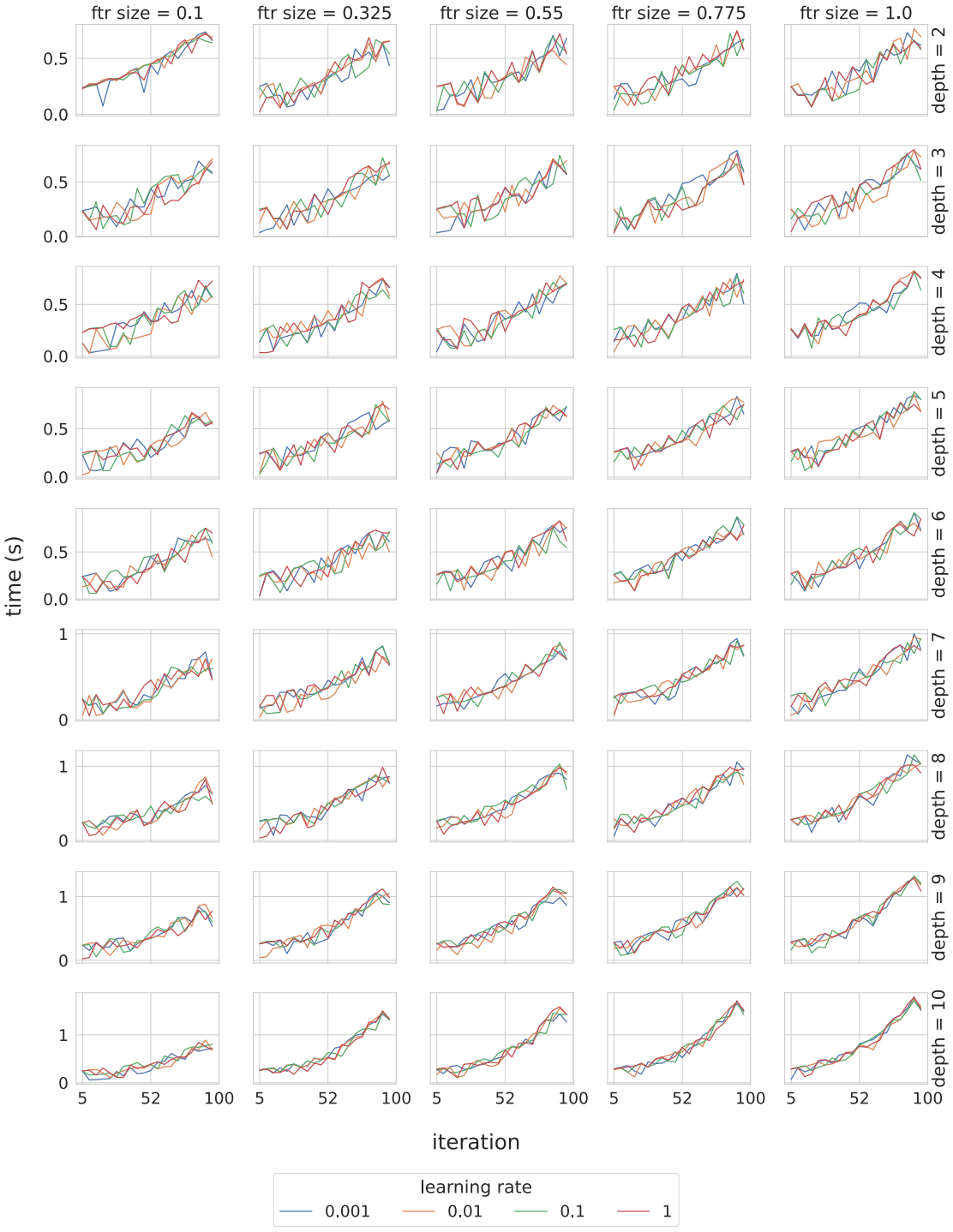


Рис. 5: Зависимость времени обучения модели градиентный бустинг от гиперпараметров

Приложение B

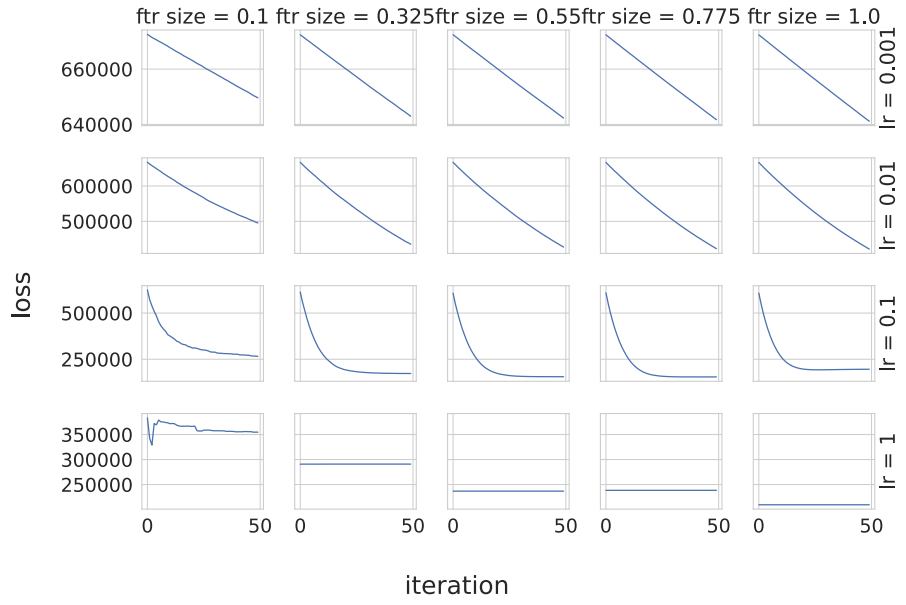


Рис. 6: Зависимость времени обучения модели градиентный бустинг от гиперпараметров без ограничения на глубину дерева

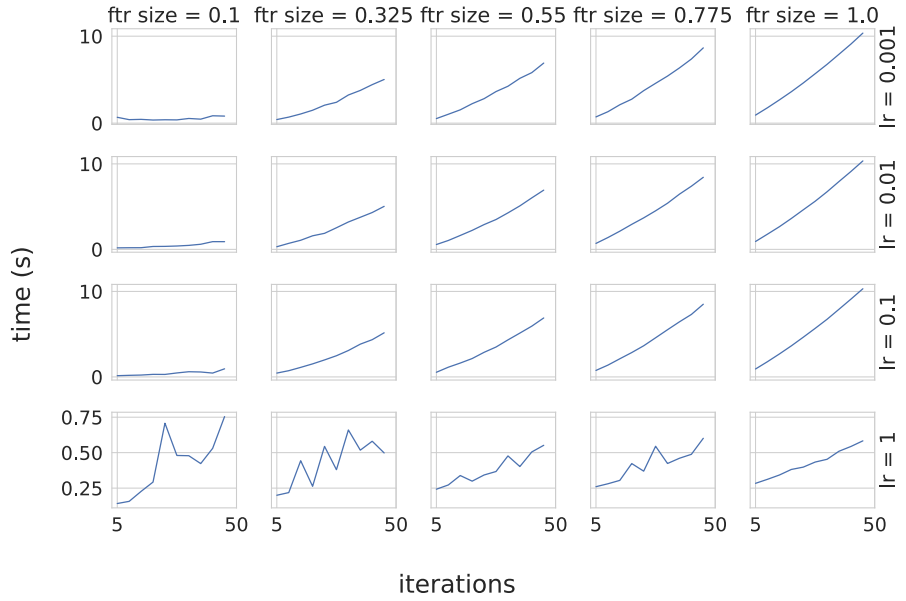


Рис. 7: Зависимость времени обучения модели градиентный бустинг от гиперпараметров без ограничения на глубину дерева