# Отчет о выполненной работе «Композиции алгоритмов для решения задачи регрессии»

Федоров Илья Сергеевич
курс «Практикум на ЭВМ» ММП ВМК МГУ

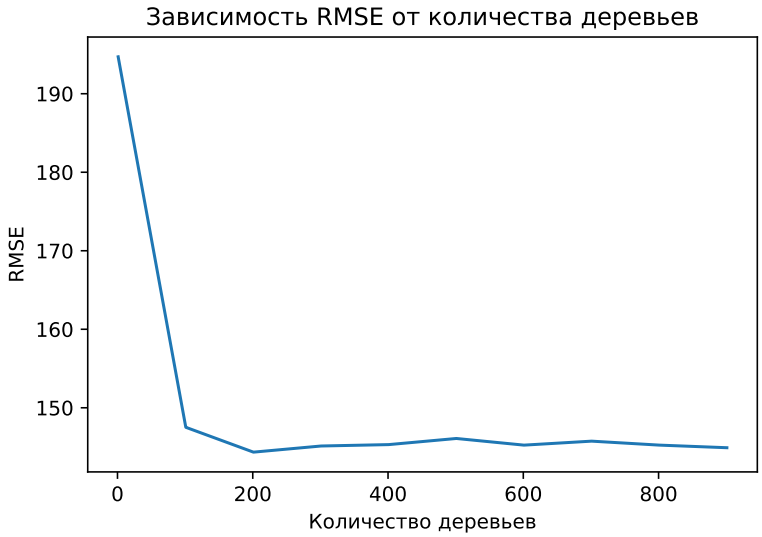
28 декабря 2019 г.

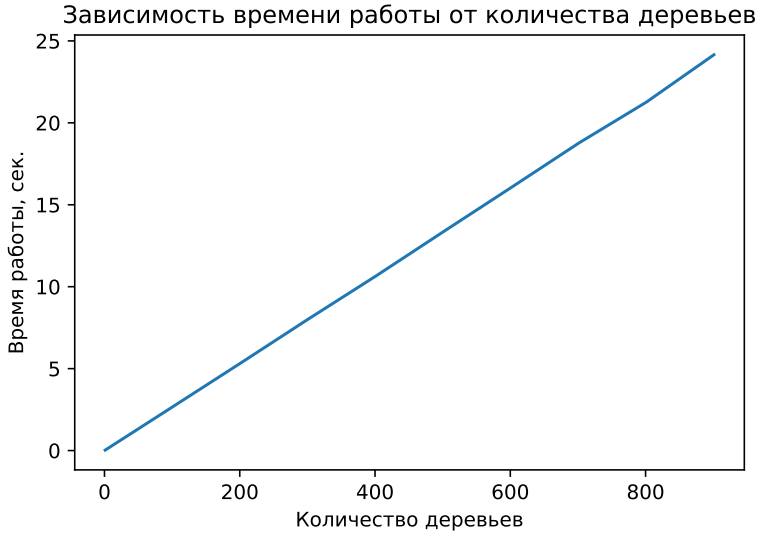
## Постановка задачи

В задании нужно было реализовать два алгоритма для решения задачи регрессии: случайный лес и градиентный бустинг, применить их для решения задачи прогнозирования цены на дом в зависимости от его характеристик и исследовать различные зависимости, связанные с этими алгоритмами, в зависимости от их гиперпараметров.

## Случайный лес

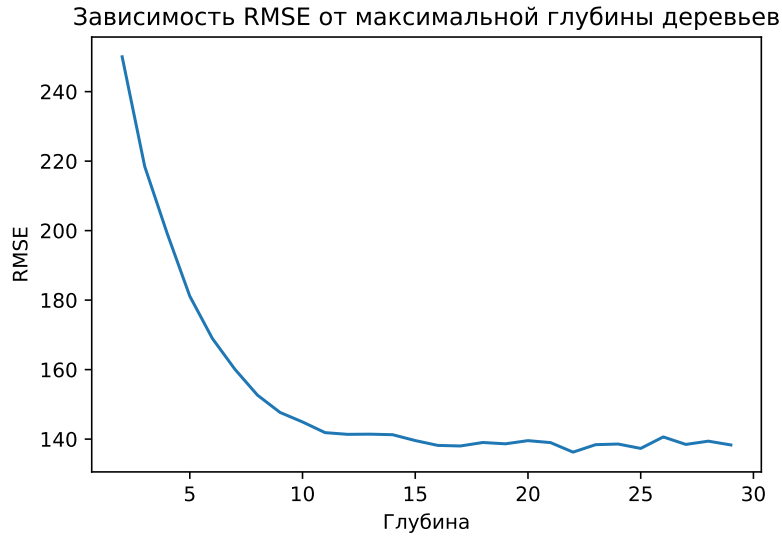
Исследуем качество работы модели в зависимости от числа деревьев. Как известно, случайный лес не переобучается при увеличении количества деревьев, а разброс модели в свою очередь уменьшается. Здесь и далее в качестве функции ошибки будем использовать RMSE. Максимальную глубину выберем равной 10. Количество признаков, из которых выбирается оптимальное разбиение в каждой вершине, по умолчанию будет иметь рекомендованное для задачи регрессии значение $\frac{n}{3}$, где $n$ - количество признаков в данных. Поскольку у нас деревья строятся не параллельно (в отличие от sklearn), то это занимает достаточно много времени. В связи с этим, будем перебирать гиперпараметр с достаточно большим шагом (добавлять по 100 деревьев).

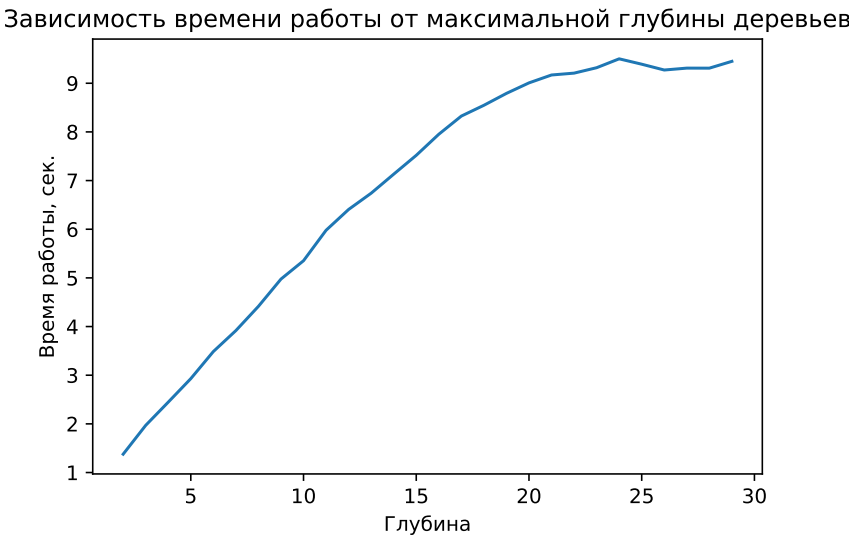



Как видим, при числе деревьев $\geq 200$, ошибка перестает значительно меняться и выходит на плато. В связи с этим, в дальнейших экспериментах будем брать 200 деревьев. Полученные результаты согласуются с теоретическими предположениями: при ещё большем увеличении количества деревьев, ошибка будет незначительно уменьшаться, поскольку будет уменьшаться разброс модели. Время работы изменяется линейным образом.

Далее, посмотрим на зависимость RMSE от максимальной глубины деревьев, входящих в случайный лес. Как известно, этот гиперпараметр отвечает за сложность модели: чем большую глубину мы позволим иметь решающим деревьям, тем более гибкой будет наша модель. Чем более модель гибкая, тем более сложные закономерности она может распознавать в данных, однако она стася более переобученной. И наоборот. Посмотрим на результаты эксперимента:



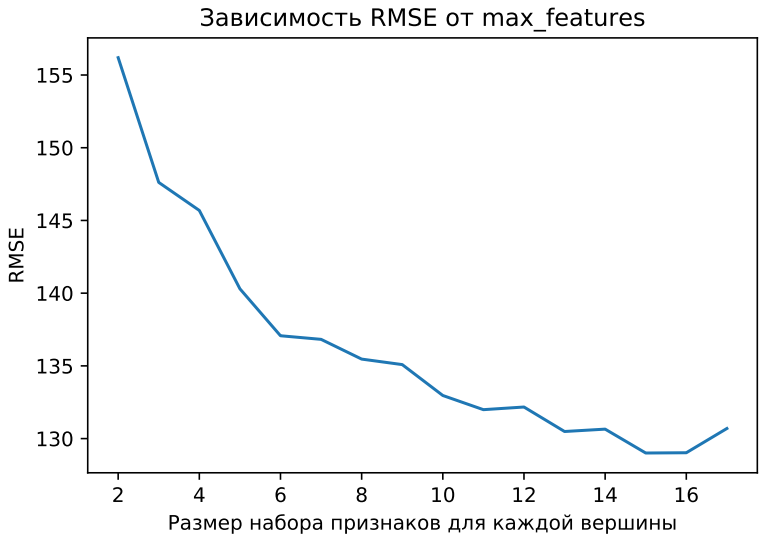

| **RMSE** |
|---|
| 136.25 |
| 137.31 |
| 138.02 |
| 138.19 |
| 138.32 |
| 138.40 |
| 138.49 |

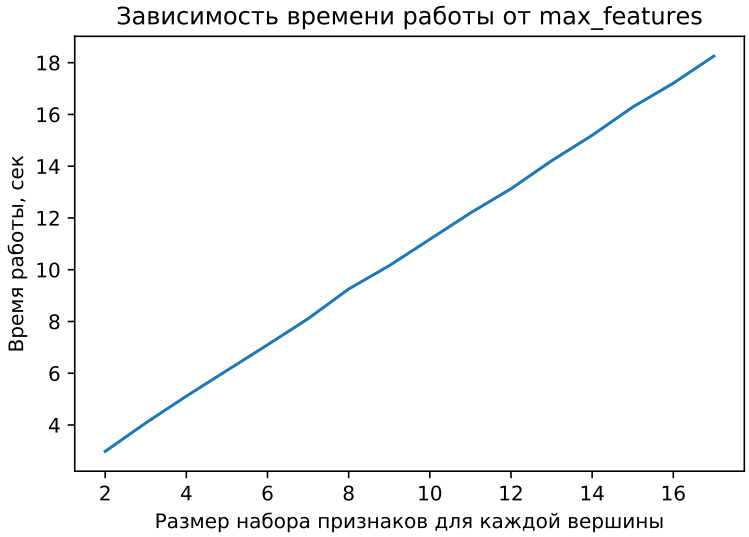
Таблица 1: Первые 7 наименьших показателей MSE для эксперимента с max_depth

Попробуем также обучить случайный лес, использующий деревья без ограничения на глубину. Получим значение RMSE, равное 138.46. Как видим, для слишком маленьких значений глубины, наша модель не будет являться достаточно гибкой, чтобы обнаружить сложные закономерности в данных. При глубине около 15, модель выходит на плато. Однако, как мы видим из численных показателей RMSE для моделей

с ограничением глубины и без ограничения, эта зависимость не является монотонной. Стоит вспомнить, что чем выше больше глубина деревьев, тем больше наша модель переобучается. Если отключить максимальную глубину, то деревья будут очень переобученным, что (в нашем случае немного) уменьшает качество работы. Время работы алгоритма сначала изменяется достаточно линейно, а потом начинает работать сублинейно.

Исследуем теперь зависимость качества модели в зависимости от размера набора признаков, по которому выбирается оптимальное разбиение в вершинах деревьев. Число деревьев будет равным 200, а максимальная глубина 15.



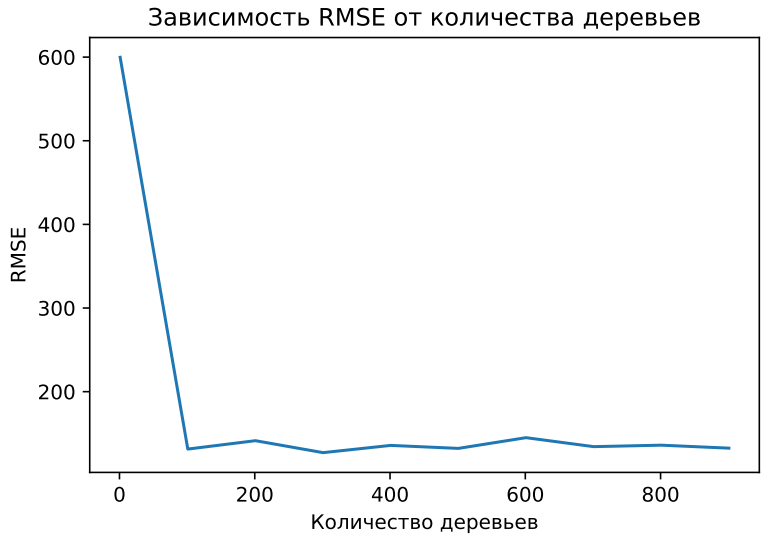



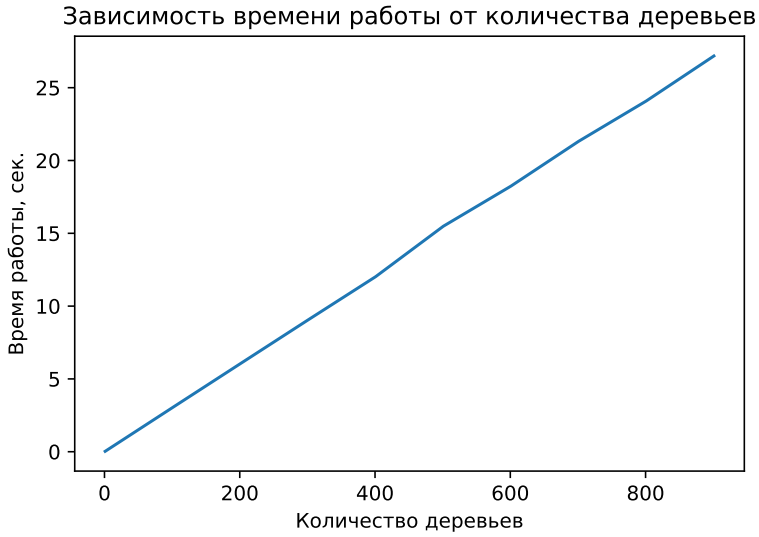

Как видим, имеется тренд к уменьшению ошибки при увеличении исследуемого гиперпараметра. Однако, при его значении, равном числу признаков в данных, ошибка увеличивается. Это можно объяснить тем, что в этом случае модели становятся

менее рандомизированными, а значит более скореллированными (а, как известно, для уменьшения разброса с помощью композиции алгоритмов, модели должны быть как можно более независимы), поскольку теперь в их рандомизации участвует лишь бутстрапированная выборка (по сути, в этом случае мы получаем бэггинг над случайными деревьями, а не случайный лес). Время работы меняется линейно.

## Градиентный бустинг

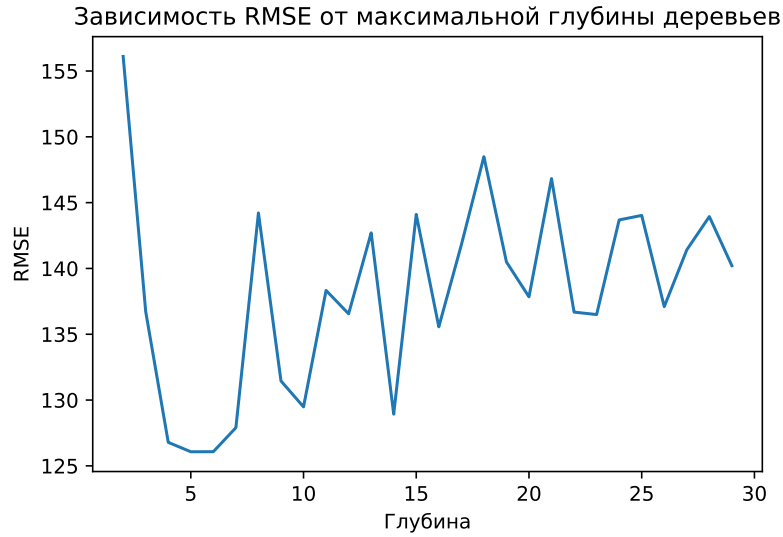
Для начала, рассмотрим зависимость RMSE от количества деревьев в бустинге. Поскольку градиентный бустинг на каждом шаге пытается исправить ошибку предыдущих алгоритмов, эта модель переобучается при достаточно большом количестве деревьев. Выберем максимальную глубину равной 10, max_features по умолчанию $\frac{n}{3}$, где $n$ - число признаков. learning_rate $= 0.1$. Получим следующий график:

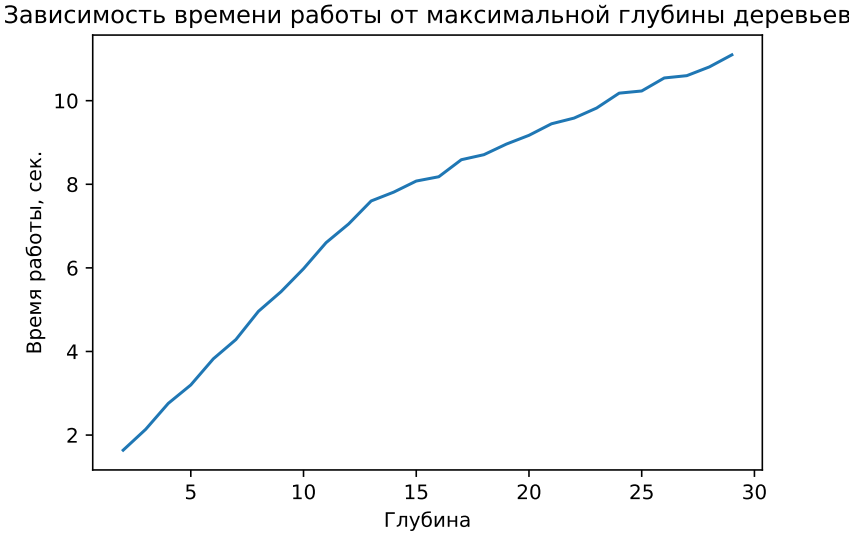


На этом графике не видно указанной выше особенности градиентного бустинга. Это связано с тем, что мы «не доверяем» каждому отдельному алгоритму, умножая его предсказания на learning_rate (равный в данном эксперименте 0.1). Однако, если мы начнем существенно увеличивать число деревьев, то начнем наблюдать существенное увеличение ошибки на отложенной выборке, что будет свидетельствовать о переобучении. Например, при количестве деревьев, равным 5000, ошибка на контрольной выборке повысится до 137 (для числа деревьев порядка 1000 ошибка приблизительно равна 132). Если же мы ещё больше увеличим число деревьев, а также увеличим learning_rate до 1.0 (будем полностью доверять каждому алгоритму), то ошибка на контрольной выборке станет равной 200. Время работы алгоритма увеличивается линейно.

Рассмотрим теперь зависимость RMSE от глубины деревьев.

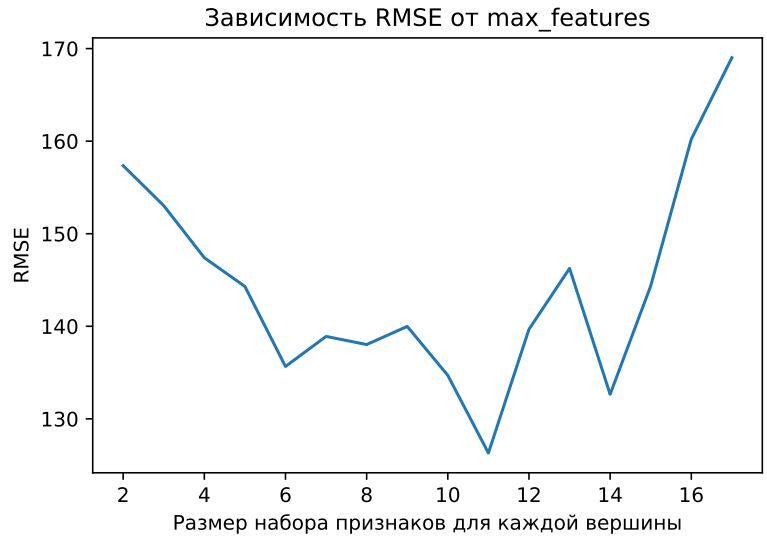


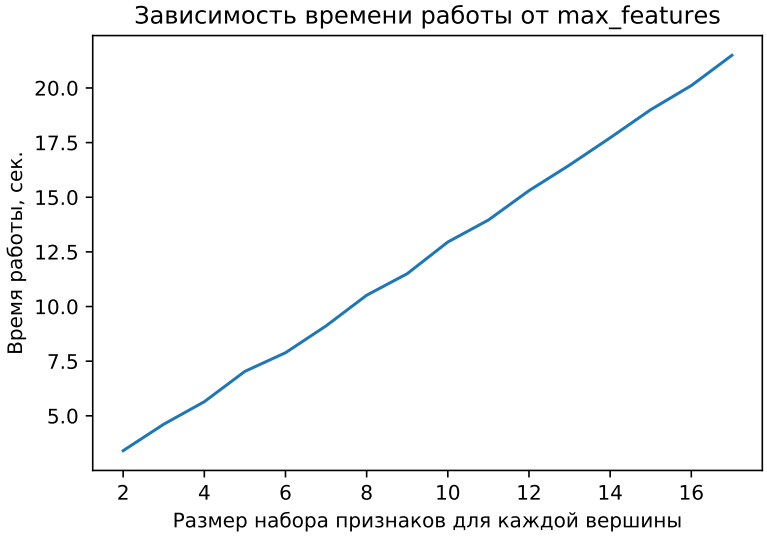




Можем наблюдать некоторую странную, почти случайную, зависимость изменения

RMSE при увеличении глубины. Сложно сказать, почему она получилась такой. Время работы кусочно-линейное.

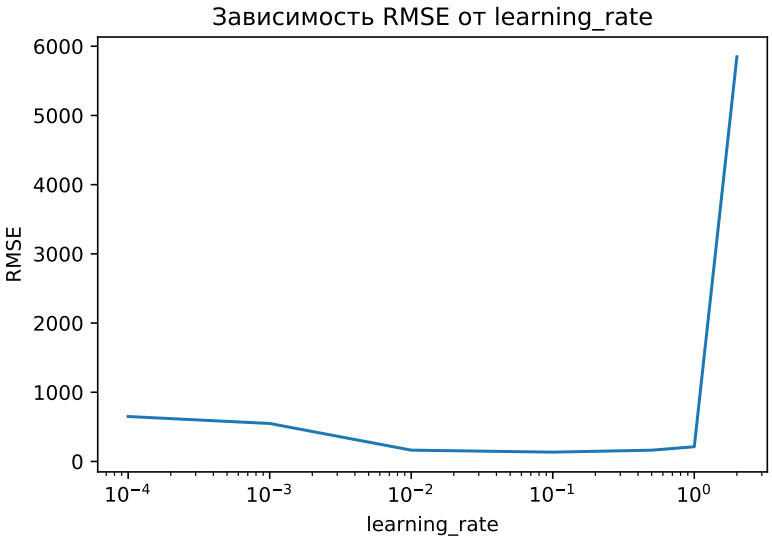
Посмотрим теперь на зависимость RMSE от размера набора признаков в каждой вершине решающих деревьев, участвующих в бустинге.

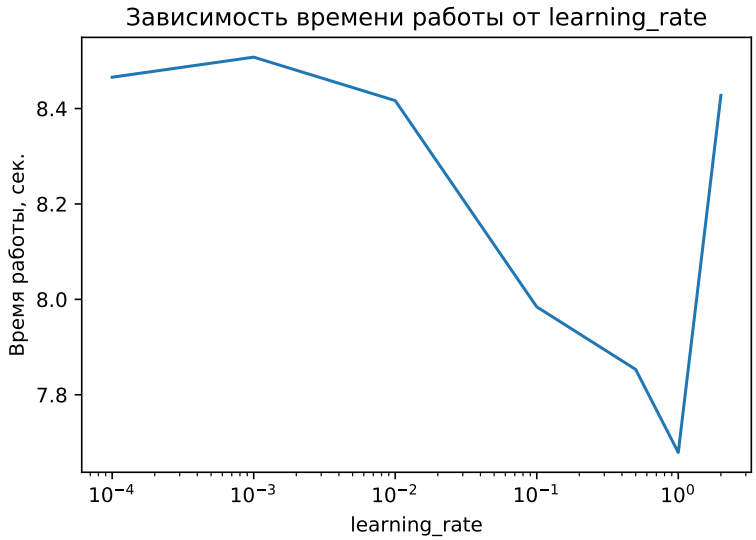



Зависимость, опять же, получилась достаточно сложной, но менее случайной, чем для глубины. Видим, что существует выраженный минимум при 11 признаках. Время работы алгоритма изменяется линейно.

Наконец, посмотрим, как влияет на качество модели параметр learning_rate. Мы уже видели выше, что этот параметр отвечает за то, насколько мы доверяем базовым алгоритмам. Он необходим, чтобы уменьшить переобучение модели (как мы знаем, бустинг переобучается при большом количестве деревьев).




В данном эксперименте параметр перебирался по сетке [0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0]. Как видим из графиков, при слишком маленьких значения learning_rate, модель не успевает достичь точки оптимума: требуется использовать больше деревьев. Лучшие результаты получаются при использовании значений порядка 0.1. При использовании learning_rate = 1.0, модель всё ещё ведет себя адекватно, однако если брать большие значения (например 2.0), то ошибка катострофически возрастает (модель начинает перескакивать точку оптимума, расходится). Время работы колеблется случайным образом в районе 8 секунд. Это можно объяснить тем, что при разных learning_rate получаются разные градиенты, и, соответственно, разные деревья, поэтому эти результаты можно считать несущественным. С небольшими погрешностями можно считать, что время работы в зависимости от learning_rate константно.

# Выводы

В данном отчете были представлены результаты экспериментов с моделями случайного леса и градиентого бустинга для решения задачи регрессии. Были представлены графики зависимостей функции ошибки и времени работы алгоритма от различных параметров этих модели. На практике подтвердились теоретически полученные результаты: случайный лес не обучается при увеличении числа базовых алгоритмов, а градиентный бустинг переобучается. Также было установлено, что если параметр градиентного бустина learning_rate превышает 1.0, то модель может расходиться.