

Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии.

Мелихов Дмитрий Александрович

18 декабря 2022 г.

Аннотация

В данной работе рассматриваются модели ансамблирования: случайный лес, градиентный бустинг, на примере датасета [House Sales in King County, USA](#). Исследуется зависимость сходимости и качества методов от гиперпараметров. Также описывается веб-приложение для создания, обучения моделей.

Введение

Рассматривается датасет "House Sales in King County, USA". Он состоит 21 613 домов с 22 признаками и ценой в долларах. Признаки: "date", "bedrooms", "bathrooms", "sqft_living", "sqft_lot", "floors", "waterfront", "view", "condition", "grade", "sqft_above", "sqft_basement", "yr_built", "yr_renovated", "zipcode", "lat", "long", "sqft_living15", "sqft_lot15". А также цена "price". К ним применяются случайный лес и градиентный бустинг.

Предобработка

Выборка была разделена на тренировочную и тестовую в отношении 7:3. Из даты "date" выделены год, месяц и день. К новым признакам, "zipcode" применён OneHotEncoder.

Эксперименты

1 Тестирование

1.1 Описание

В этой секции проверяется работа программы для модельных данных из 10 000 элементов с 15 признаками, 10 из них информативные. Сравнивается собственная реализация и алгоритмы из sklearn. Параметры: n_ensembles=100, max_depth=5, feature_subsample_size= $\frac{1}{3}$.

1.2 Результаты

В результате эксперимента получается RMSE:

sklearn Random Forest: 93.5; **ensembles.py Random Forest:** 94.1

sklearn Gradient Boosting: 28.8; **ensembles.py Gradient Boosting:** 44

Заметная разница в RMSE. По рисунку 1 можно заметить, что алгоритмы обучаются на модельных данных и сильно не переобучаются.

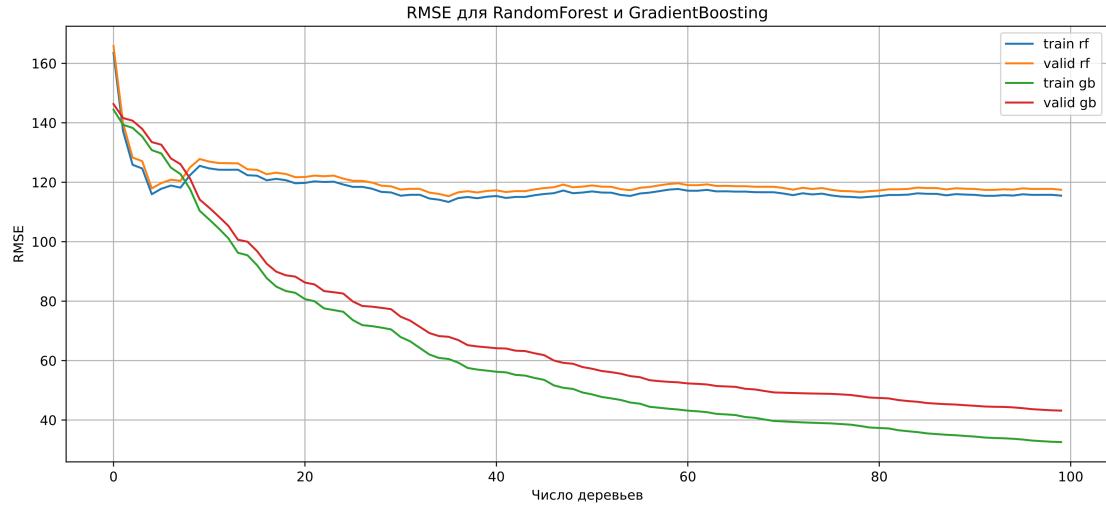


Рис. 1: Зависимость RMSE от итерации для модельных данных

1.3 Выводы

Алгоритмы работают, но реализация градиентного бустинга отличается от реализации в sklearn.

2 Случайный лес

2.1 Описание

Эксперимент проводится на изначальных данных. Перебираются параметры:

$$n_estimators \in \{1, \dots, 100\}, max_depth \in \{1, 5, 15, \infty\}, feature_subsample_size \in \{0.2, 0.6, 1\}$$

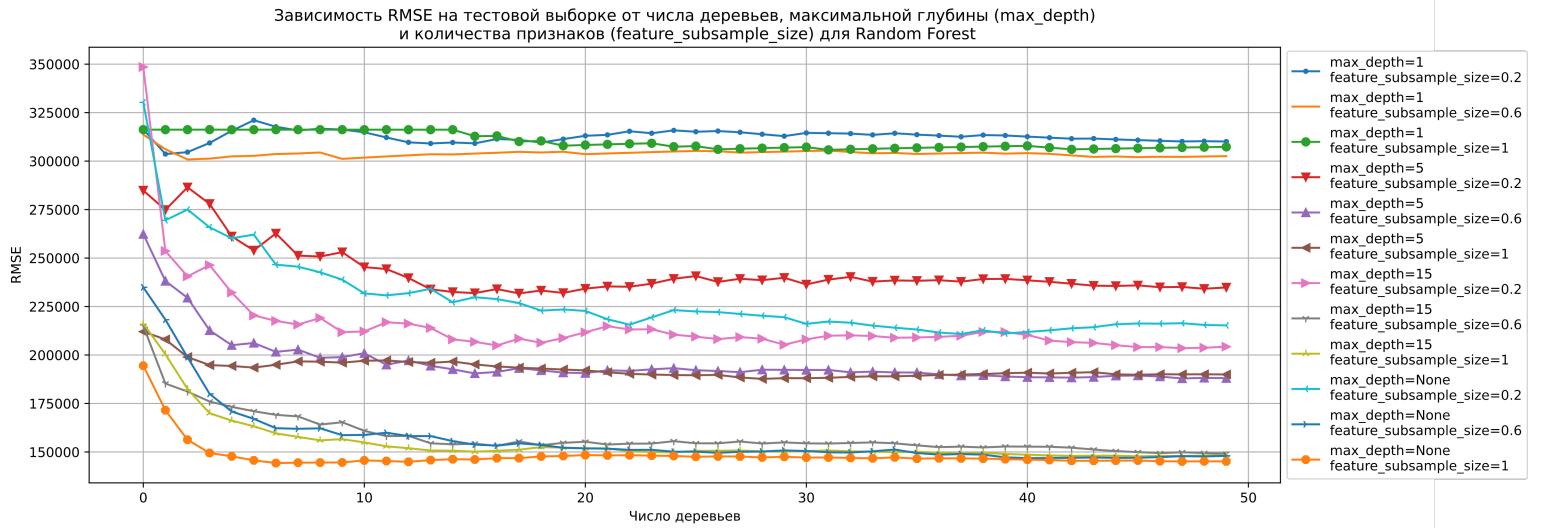


Рис. 2: Зависимость RMSE на тестовой выборке от параметров Random Forest

2.2 Результаты

Результат эксперимента можно увидеть на рисунках 2-3. Заметим, что лучший результат получается для глубоких деревьев и достаточно большом `feature_subsample_size`. При этом результат достаточно устойчивый. Худший результат показали "решающие пни" (`max_depth=1`). Скорость работы: ожидаемо, дольше всех обучаются деревья с большой глубиной и большим `feature_subsample_size`. Из 4 оптимальных решений самое быстрое - с параметрами `max_depth=15` `feature_subsample_size=0.6`.

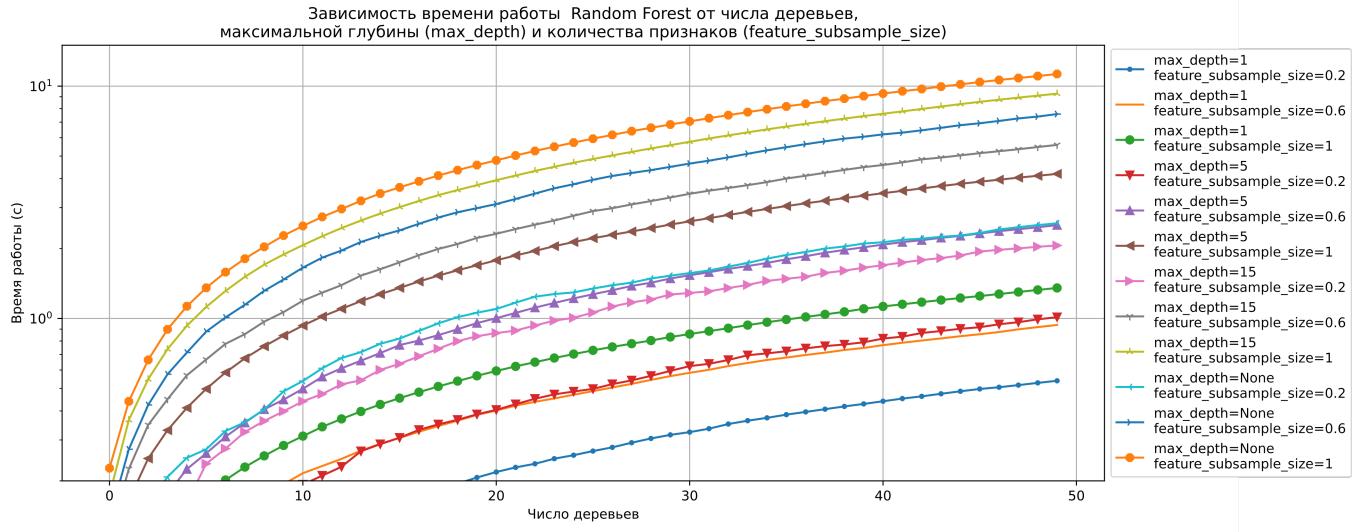


Рис. 3: Зависимость времени от параметров Random Forest

2.3 Выводы

Случайный лес достигает лучших результатов с глубокими деревьями.

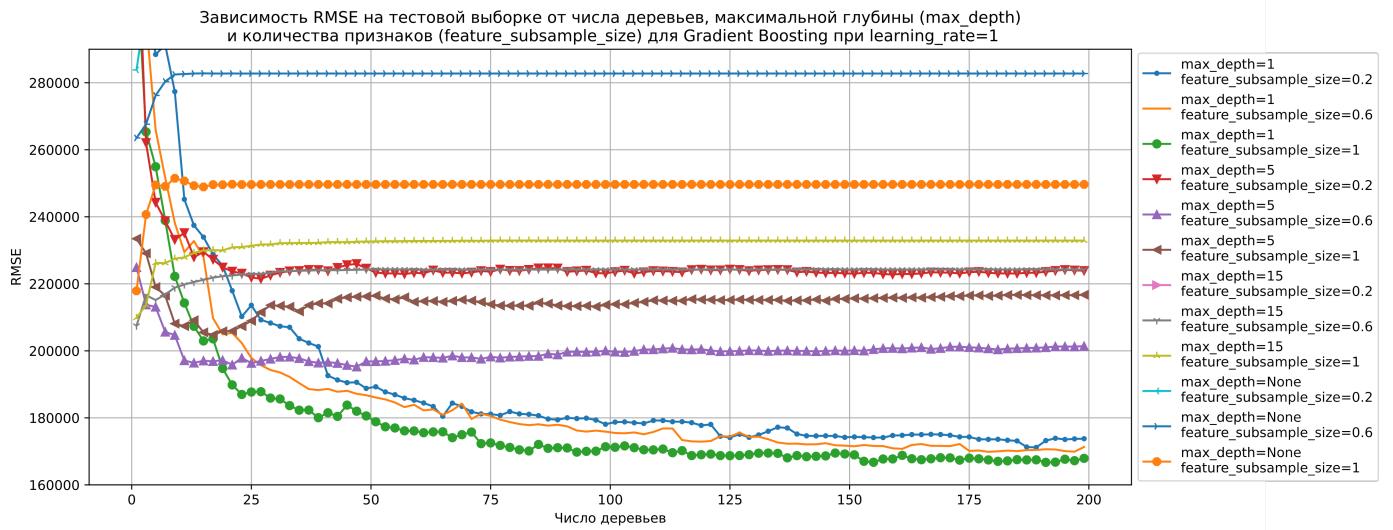


Рис. 4: Зависимость RMSE на тестовой выборке от параметров Gradient Boosting, `learing_rate=1`

3 Градиентный бустинг

3.1 Описание

Эксперимент проводится на изначальных данных. Перебираются параметры:

$$n_estimators \in \{1, \dots, 200\}, max_depth \in \{1, 5, 15, \infty\}, feature_subsample_size \in \{0.2, 0.6, 1\}, learning_rate \in \{0.01, 0.1, 1\}$$

3.2 Результаты

Результаты представлены на рисунках 4-6. Заметим, что теперь лучшее качество показывают неглубокие деревья. "Решающие пни" ($max_depth=1$) лучше справляются, чем деревья с неограниченной глубиной ($max_depth=None$). Особенно это заметно для $learning_rate=0.1$ (рис. 5). Чем меньше $learning_rate$, тем больше вероятность, что решение сходится, но при этом скорость сходимости уменьшается. Оптимальные параметры: $max_depth=5$, $learning_rate=0.1$. Скорость работы продемонстрирована на рисунках 7-9, ситуация схожа со случайным лесом, поэтому изображения графиков перенесены в приложение (вместо них можно смотреть на рисунок 3).

Самое быстрое оптимальное решение для $feature_subsample_size=0.2$

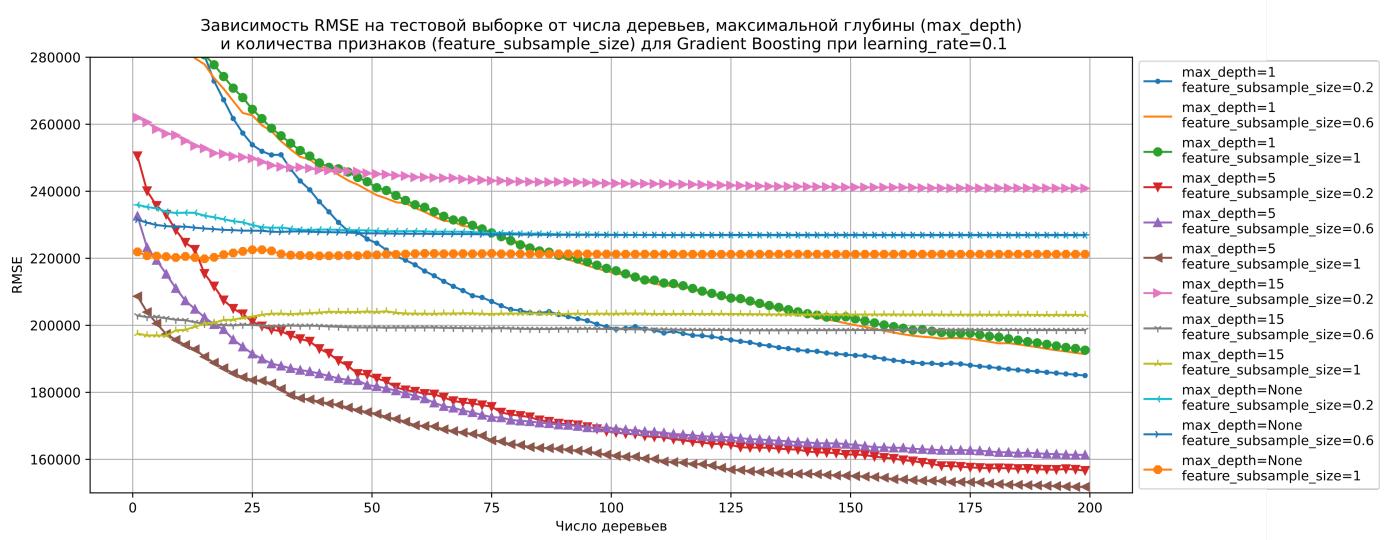


Рис. 5: Зависимость RMSE на тестовой выборке от параметров Gradient Boosting, learing_rate=0.1

3.3 Выводы

Градиентный бустинг достигает лучших результатов с неглубокими деревьями (для данной задачи: $max_depth=5$)

Вывод работы

Случайный лес устойчив к выбору гиперпараметров, достаточно лишь выбрать достаточно большую глубину и $feature_subsample_size$.

Градиентный бустинг тоже достаточно устойчив, но здесь уже требуется подбор глубины и скорости обучения по грубой сетке. Для градиентного бустинга нужны достаточно простые модели, в случае с деревьями: небольшая глубина, небольшой $feature_subsample_size$, не слишком маленький $learning_rate$

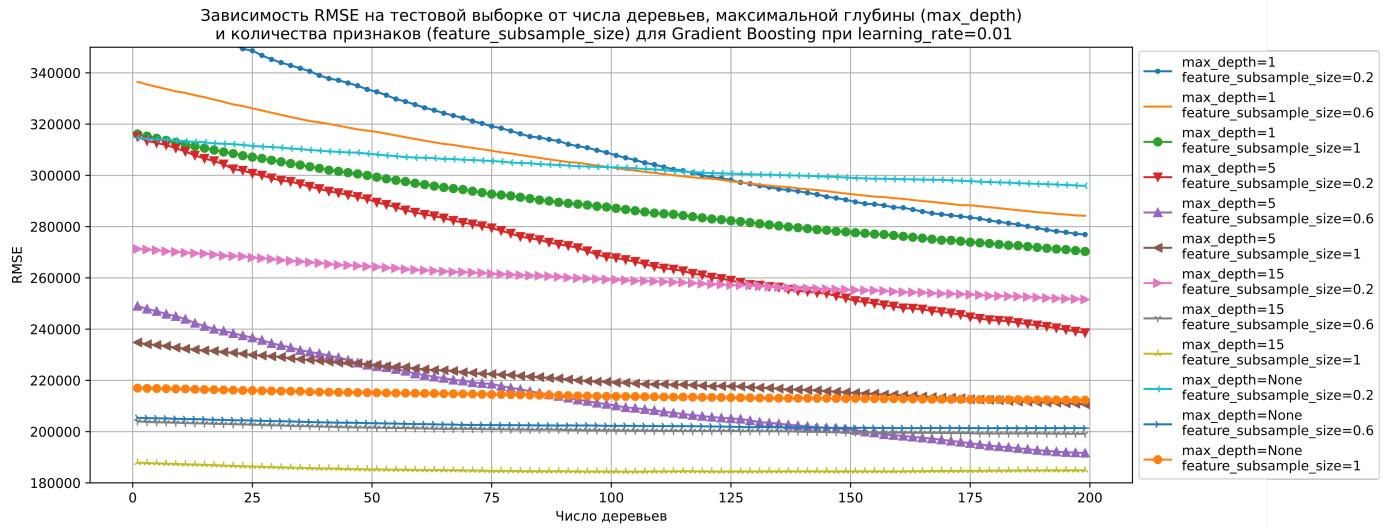


Рис. 6: Зависимость RMSE на тестовой выборке от параметров Gradient Boosting, learing_rate=0.01

и побольше итераций.

Также можно реализовать раннюю остановку на основе валидационных данных или out of bag prediction, чтобы не тратить много итераций.

Приложение

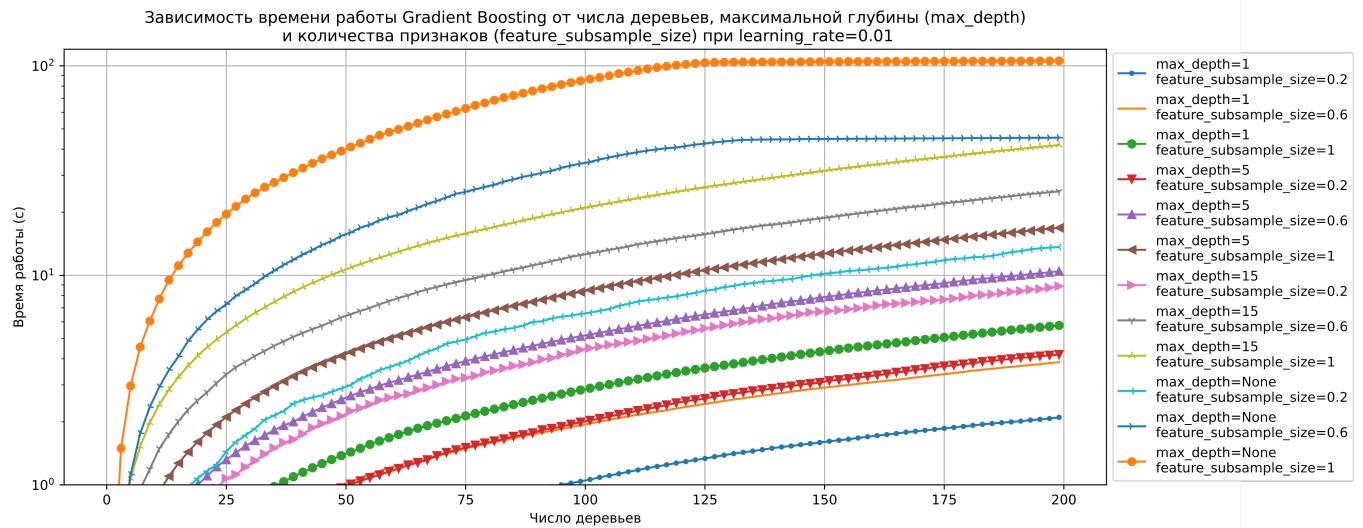


Рис. 7: Зависимость времени работы от параметров Gradient Boosting, learing_rate=1

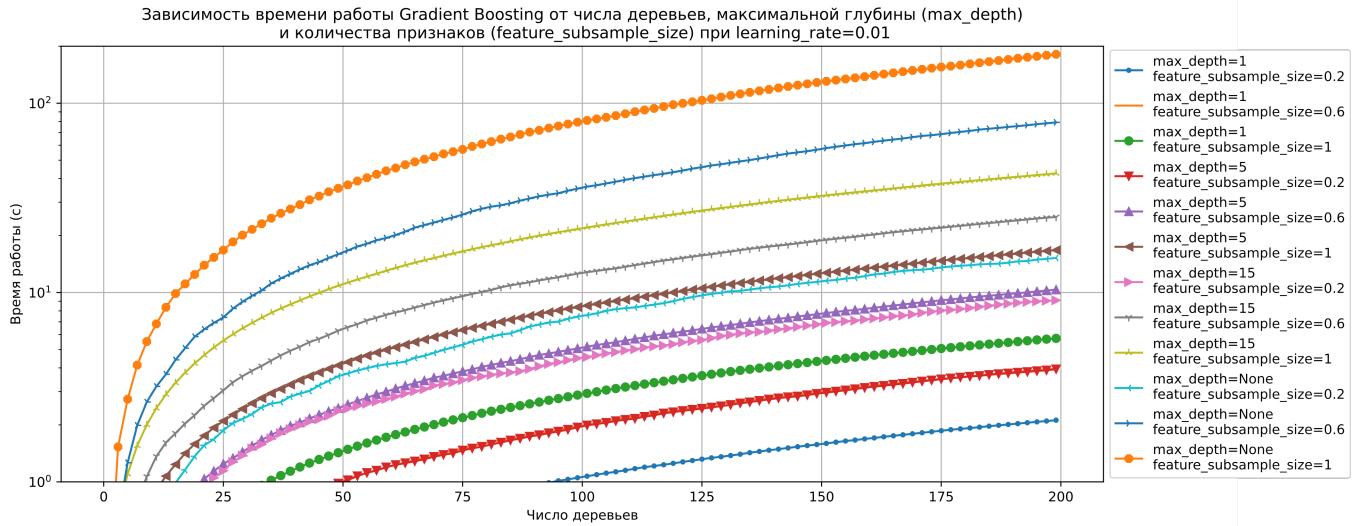


Рис. 8: Зависимость времени работы от параметров Gradient Boosting, learing_rate=0.1

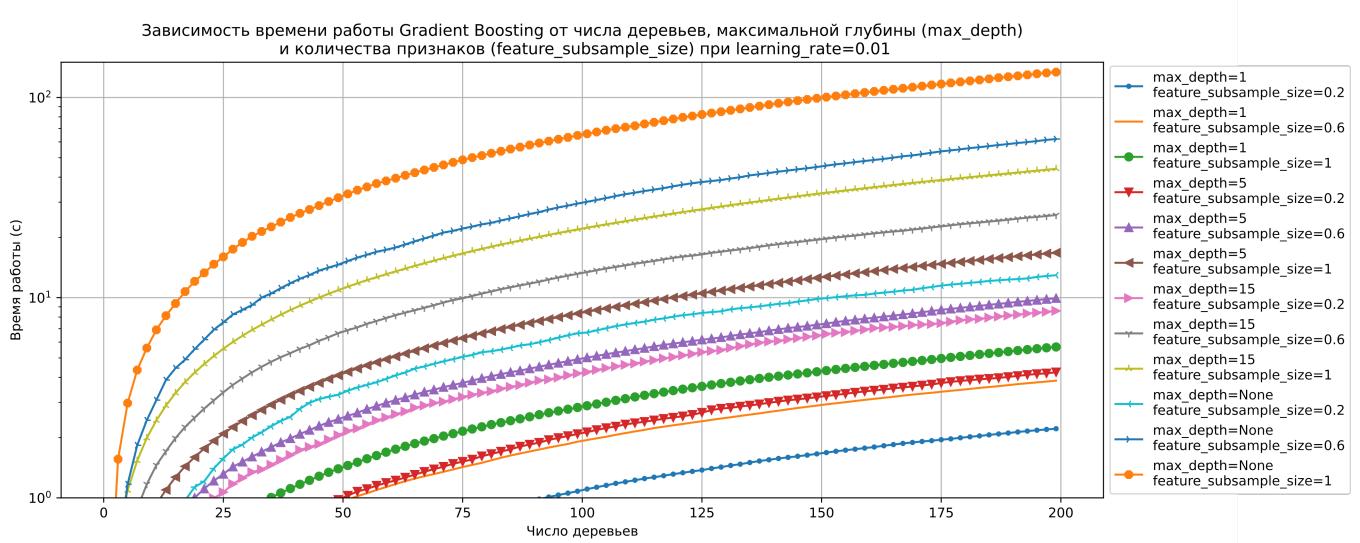


Рис. 9: Зависимость времени работы от параметров Gradient Boosting, learing_rate=0.01