МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

# Отчет по заданию №3

# «Ансамбли алгоритмов. Веб-сервер. Композиция алгоритмов для решения задач регрессии.»

Выполнила:
студентка 317 группы
Анисимова Д. В.

Москва
2021

# Содержание

# 1 Введение

В задании необходимо было реализовать композиции алгоритмов. Нужно было провести эксперименты, используя собственные реализации случайного леса и градиентного бустинга, исследовать, как различные параметры этих методов влияют на качество и время работы.

# 2 Эксперименты

## 2.1 Предобработка

Из данных был выделен стобец price — целевая переменная. Далее столбец date был преобразован в 3 столбца: day, month и year. Это было сделано для того, чтобы все данные таблицы были в числовом формате, так с ними работать гораздо удобнее, чем в изначальном виде. Наконец, был удален столбец id,который, как очевидно из его названия, ни на что не влияет.
Эти данные были преобразованы в numpy-массивы, а потом разделены на 2 выборки: обучающую и валидационную в соотношении 7:3.

## 2.2 Исследование поведения случайного леса

Рассмотрим сначала, как меняется ошибка при изменении числа деревьев.


Рис. 1: Случайный лес: зависимость значения функции потерь от количества деревьев

При очень маленьком количестве деревьев значение RMSE очень велико. Когда количество деревьев немного увеличивается (но все еще не превышает 100) функция потерь начинает осциллировать. Но потом, при увеличении параметра до 200 деревьев, RMSE практически не меняется. Таким образом, чтобы получить хорошую точность, нужно взять хотя бы 200 деревьев.
Удостоверимся, что модель с таким значением работает не очень долго, посмотрев на следующий график:

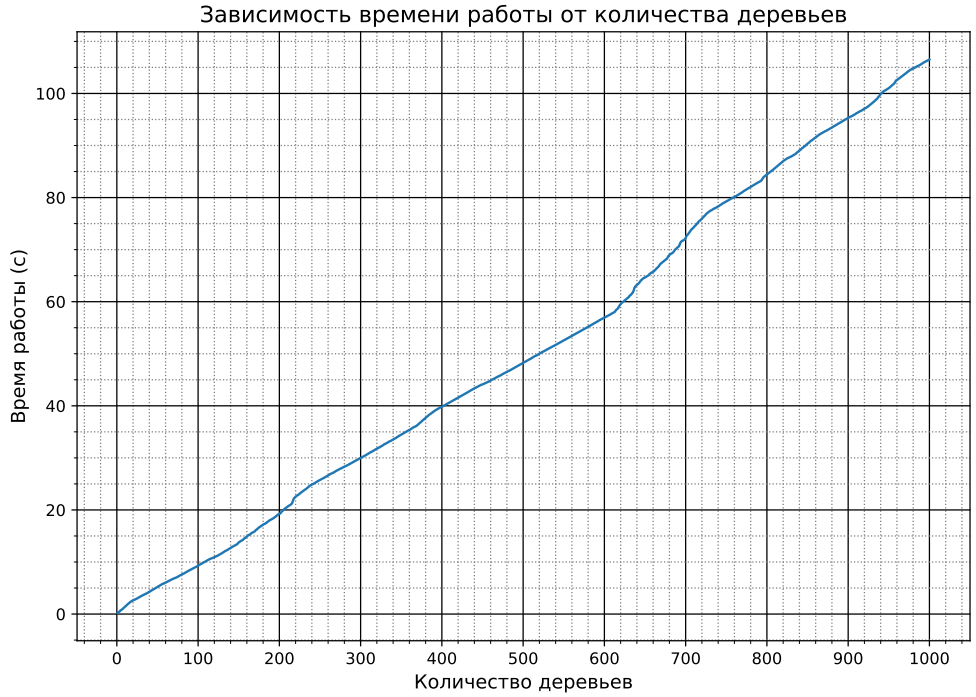


Рис. 2: Случайный лес: зависимость времени работы алгоритма от количества деревьев

Мы видим, что время увеличивается почти линейно с ростом числа деревьев, что вполне логично. Убедившись в не очень большом (примерно 20 секунд) времени работы алгоритма при параметре n_ estimators = 200, будем использовать его в дальнейших экспериментах.
Теперь посмотрим на то, как сильно влияет размерность подвыборки признаков на значения функции потерь.

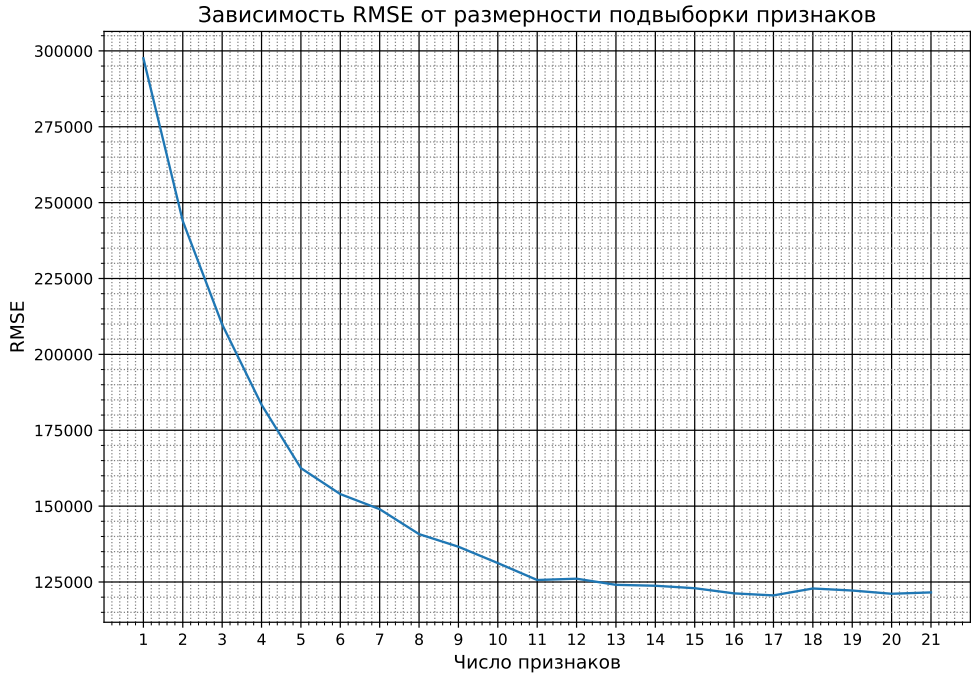


Рис. 3: Случайный лес: зависимость значения функции потерь от числа выбранных признаков

При маленькой размерности, как видно на графике, RMSE очень большое, но с увеличением размерности оно начинает очень сильно уменьшаться (функция почти экспоненциально убывает). В какой-то момент убывание замедляется, и примерно со значения размерности, равного 11, функция убывает не очень сильно. Возможно, если продолжить график далее, то на этом участке он стал бы похож на прямую. Значение размерности будем выбирать не меньше 11, но сначала посмотрим, как долго производятся вычисления:

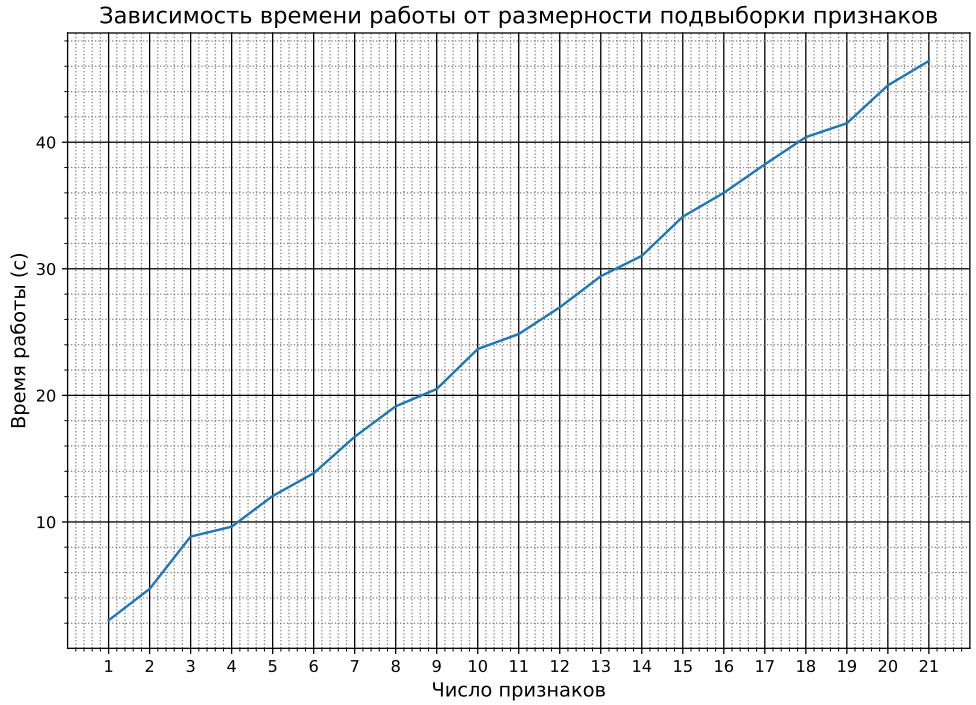


Рис. 4: Случайный лес: зависимость времени работы алгоритма от числа выбранных признаков

Функция возрастает практически линейно, поэтому брать слишком большое значение смысла нет. Для дальнейших экспериментов выберем значение, равное 13.
Теперь рассмотрим максимальную глубину деревьев.

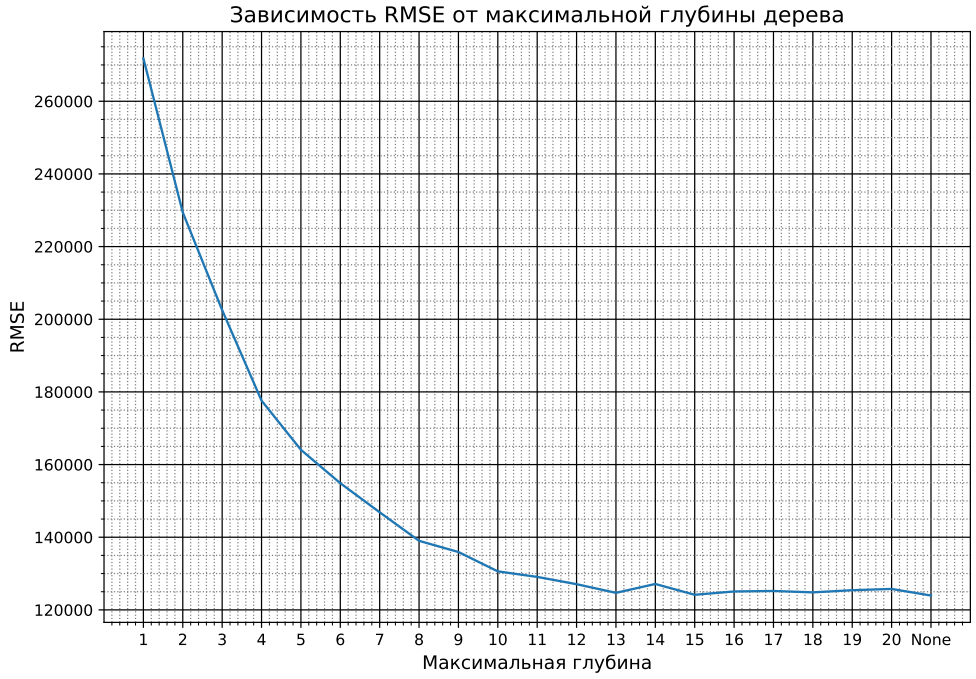


Рис. 5: Случайный лес: зависимость значения функции потерь от глубины дерева

При маленькой глубине значение функции потерь ожидаемо велико. В целом график очень похож на такой же график для размерности подвыборки признаков: сначала сильное убывание, которое потом замедляется, и график начинает походить на прямую. Поэтому разумно будет выбрать значение глубины, не меньше 13.
Посмотрим на время работы алгоритма при изменении глубины дерева:

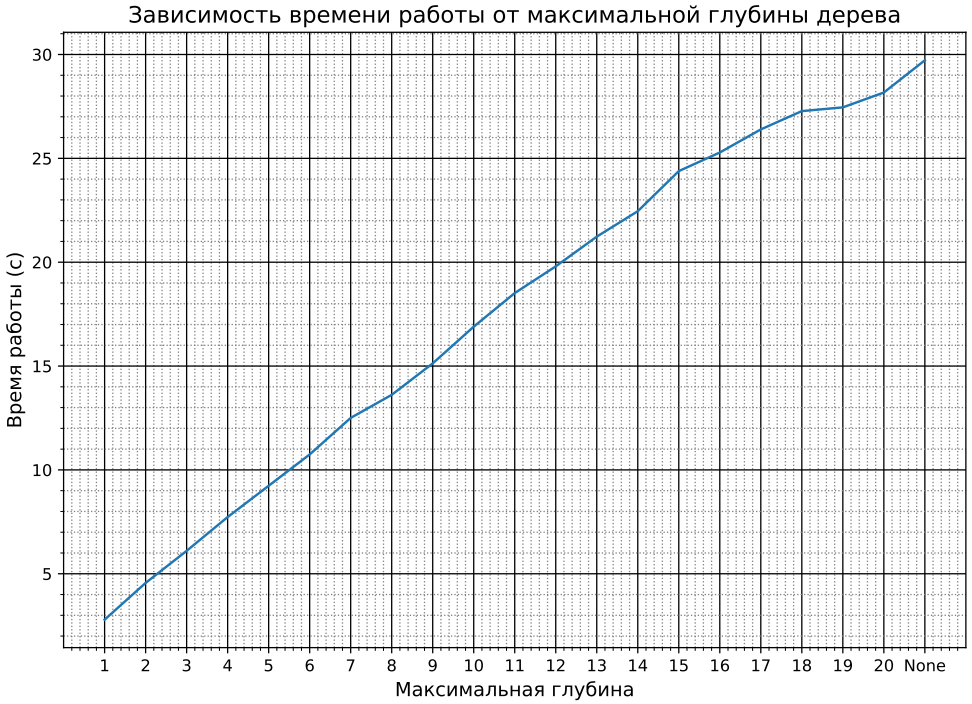


Рис. 6: Случайный лес: зависимость времени работы алгоритма от глубины дерева

Как и в предыдущих двух случаях, с увеличением параметра время работы возрастает, поэтому слишком большую глубину брать смысла нет. Одним из лучших значений, если посмотреть на эти 2 графика, является глубина, равная 15.

## 2.3 Исследование поведения градиентного бустинга

Рассмотрим сначала зависимость RMSE от числа деревьев.

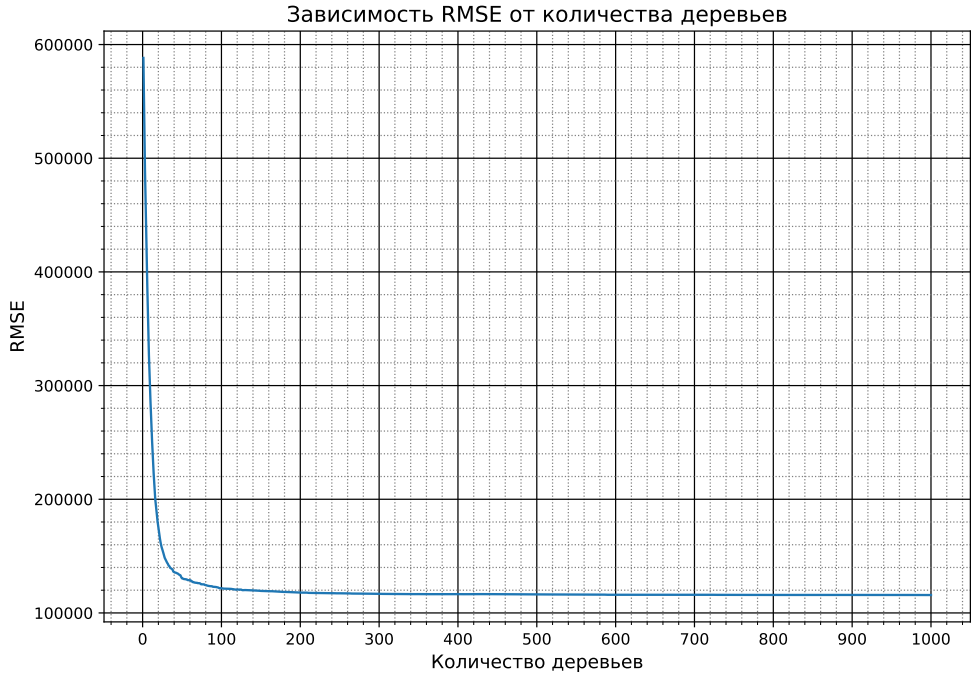


Рис. 7: Градиентный бустинг: зависимость значения функции потерь от количества деревьев

В отличие от случайного леса здесь практически не видны осцилляции при маленьких значениях параметра. И в целом поведение функции немного отличается: сначала идет монотонное убывание, но, в отличие от предыдущего алгоритма, здесь нет асимптоты, убывание, хоть и малозаметное, продолжается. Для выбора наилучшего значения опять воспользуемся графиком зависимости времени работы от числа деревьев:

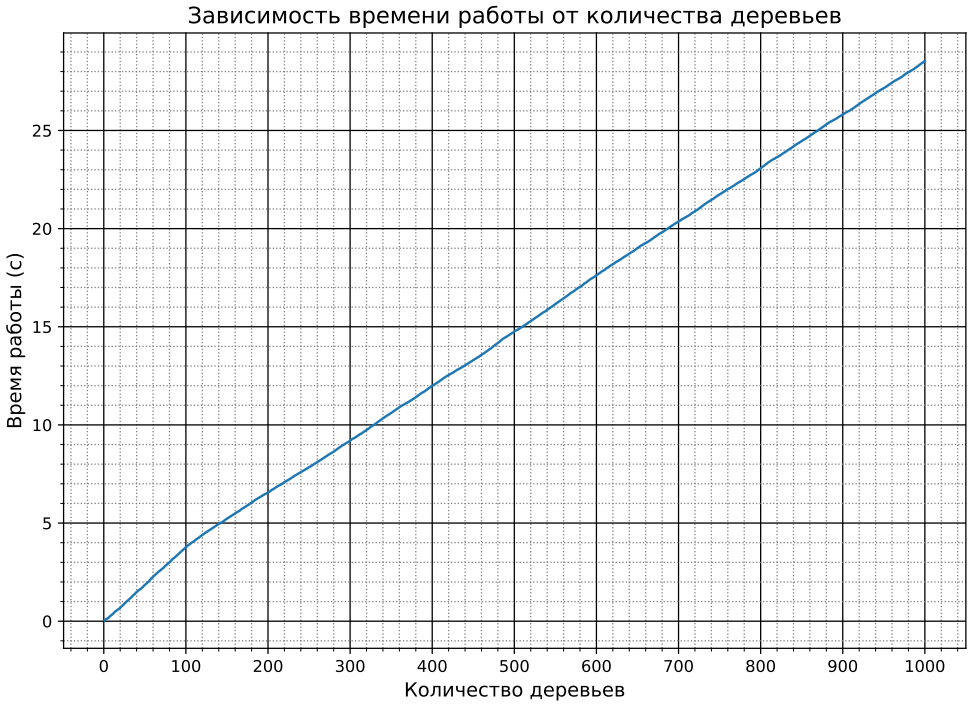


Рис. 8: Градиентный бустинг: зависимость времени работы алгоритма от количества деревьев

Как и ранее, этот время почти линейно растет с увеличением параметра. Поэтому нет смысла брать слишком большое число деревьев, для дальнейших экспериментов будем использовать значение 400.
Теперь посмотрим, как на нашу модель влияет размерность подвыборки признаков.


Рис. 9: Градиентный бустинг: зависимость значения функции потерь от числа выбранных признаков

График, хоть и напоминает отдаленно аналогичный график для случайного леса, все же имеет существенное отличие. Значение RMSE очень нестабильно, происходят постоянные скачки. Выберем значение 10, так как по графику видно, что при этом значении функция достигает локального минимума.


Рис. 10: Градиентный бустинг: зависимость времени работы алгоритма от числа выбранных признаков

Время, хоть и возрастает с ростом размерности, но тоже претерпевает скачки. Возможно, это связано с не очень хорошей предобработкой данных.
Теперь будем исследовать максимальную глубину дерева.

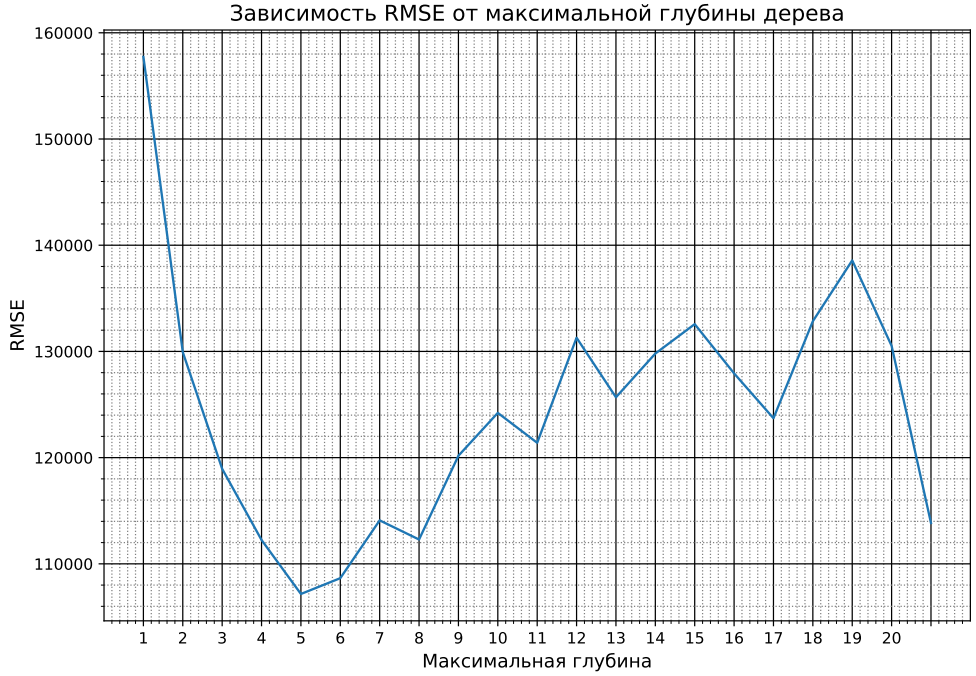


Рис. 11: Градиентный бустинг: зависимость значения функции потерь от максимальной глубины дерева

Здесь также происходят скачки. Но в целом поведение функции понятно: сначала ее значения уменьшаются, функция достигает локального минимума, а дальше RMSE увеличивается. Поэтому для следующих экспериментов стоит взять именно локальный минимум, то есть 5.

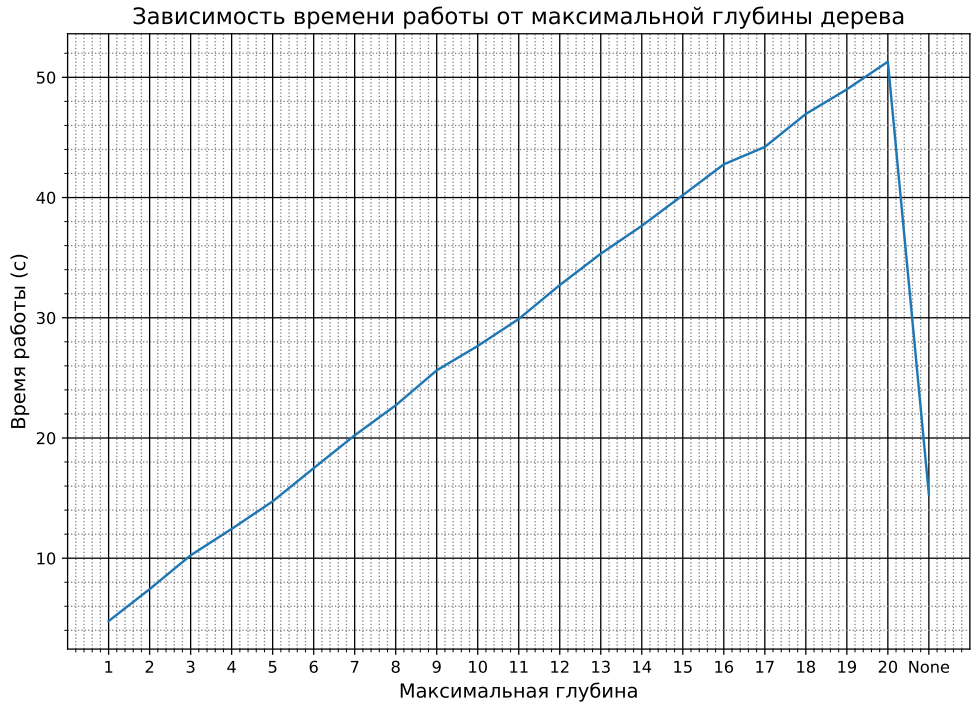


Рис. 12: Градиентный бустинг: зависимость времени работы алгоритма от максимальной глубины дерева

А вот время уже ведет себя нормально: линейно увеличивается с ростом глубины, что вполне логично.
Рассмотрим последний параметр — темп обучения:

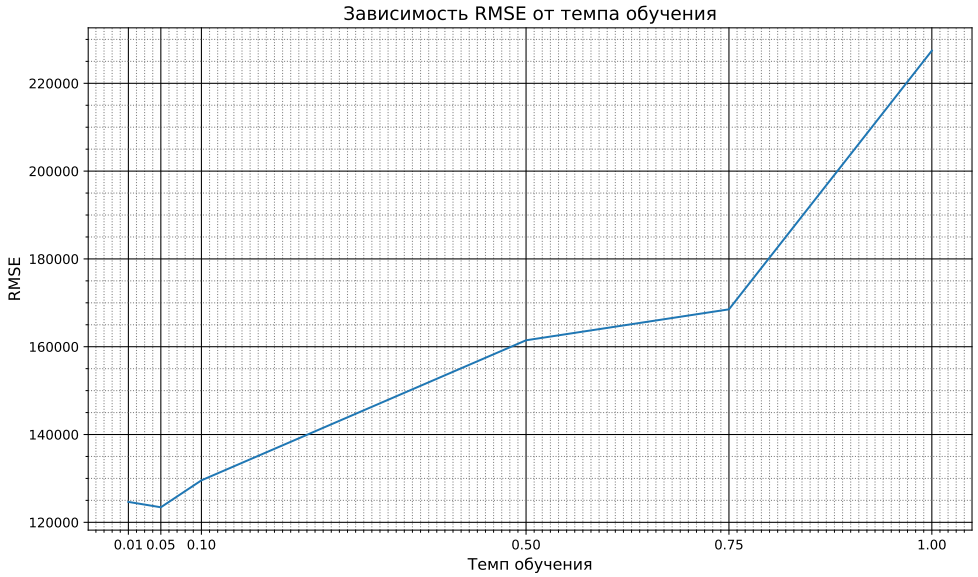


Рис. 13: Градиентный бустинг: зависимость значения функции потерь от темпа обучения

На графике видно, что при маленьких значениях темпа обучения RMSE тоже мало. А вот при увеличении оно монотонно растет. Казалось бы, поэтому нужно выбирать маленький темп обучения. Но у такого выбора есть существенный недостаток — большое время работы, что демонстрирует следующий график:

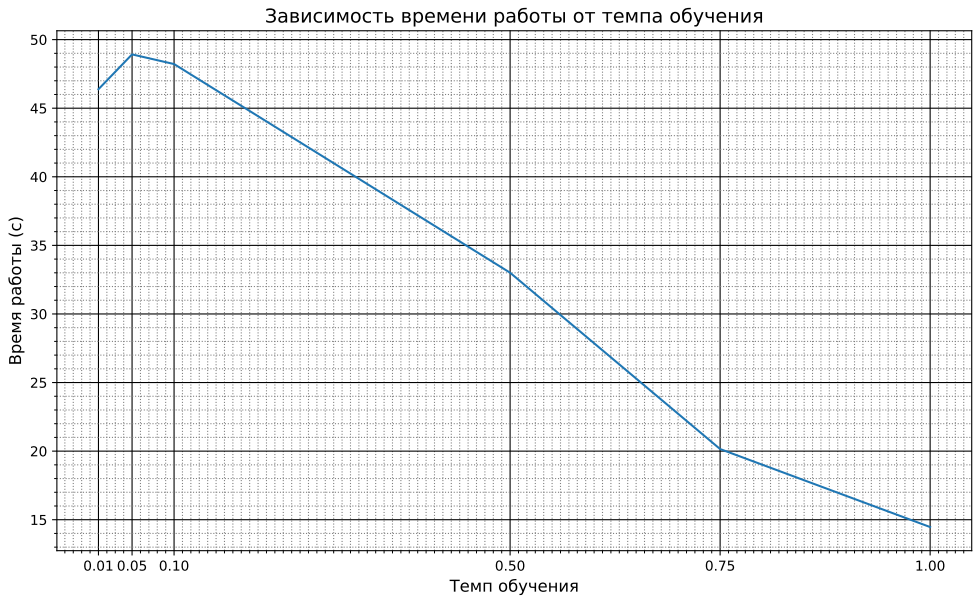


Рис. 14: Градиентный бустинг: зависимость времени работы алгоритма от темпа обучения

Но, поскольку это был последний эксперимент, в качестве финального значения выберем 0.05, хоть и время работы при нем велико.

# 3 Вывод

В результате проведение экспериментов было выяснено, что для случайного леса лучше всего подходят следующие параметры: число деревьев, равное 200; размерность признакового подпространства, равная 13; максимальная глубина, равная 15. Для градиентного бустинга оптимальными оказались такие параметры: число деревьев равно 400, размерность признакового пространства равно 10, максимальная глубина — 5, а темп обучения — 0.05. В итоге, запустив эти модели, мы получили следующий результат на валидационной выборке: случайный лес показал RMSE=124750 за время работы 28 секунд, а градиентный бустинг — RMSE=112131 за время работы 15 секунд. Таким образом, градиентный бустинг выигрывает по обоим критериям, а значит, лучше подходит для наших данных.