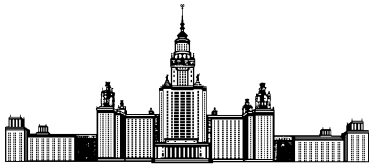
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

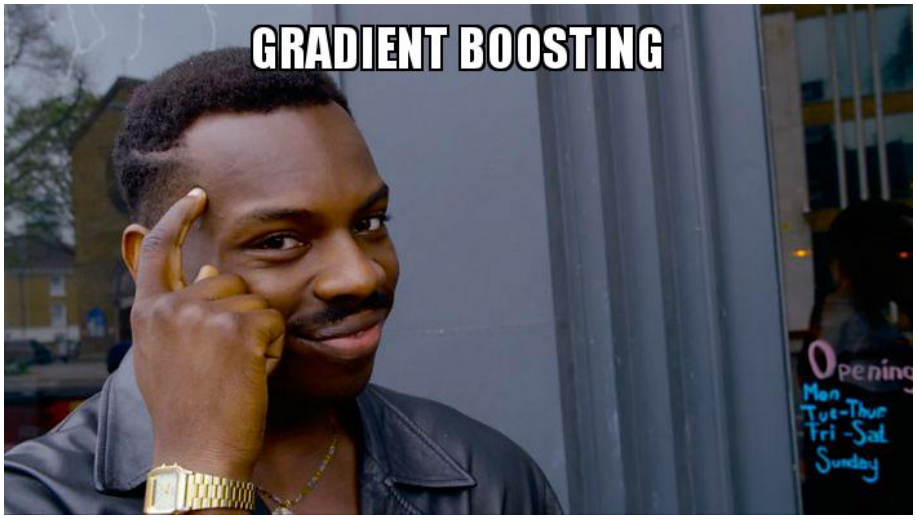
Кафедра Математических Методов Прогнозирования

Задание 3. Ансамбли алгоритмов. Веб-сервер. Композиция алгоритмов для решения задачи регрессии. Эксперименты.

Выполнил:

студент 3 курса 317 группы

Матвеев Артем Сергеевич



Москва, 2022

Содержание

1 Введение

Данное задание направлено на ознакомление с алгоритмами композиций. В рамках данного задание необходимо было написать на языке Python собственные реализации методов случайный лес и градиентный бустинг. Провести эксперименты с данными о продаже недвижимости **House Sales in King Country, USA** и написать реализацию веб-сервера. Отчет описывает часть с экспериментами.

2 Пояснение к задаче

В задаче оптимизируется метрика **RMSE**. Формула для нее имеет вид:

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

где y - истинные ответы, а $\hat{y}$ - предсказанные.

Для кодирования категориальных признаков в экспериментах будут использоваться счетчики со сглаживанием. Формула:

$$g_j(x, X) = \frac{\sum_{i=1}^{\ell}[f_j(x) = f_j(x_i)]y_i + C \times global\_mean}{\sum_{i=1}^{\ell}[f_j(x) = f_j(x_i)] + C}$$

где X - обучающая выборка, x - объект, j-ый признак которого мы кодируем, $f_j(x)$ - j-ый признак объекта x, $global\_mean$ - среднее значение целевой переменной по всей выборке, C - константа, отвечающая за то, как сильно мы сглаживаем счетчики. Также для уменьшения влияния проникновения целевой переменной в признаки, вычисление значения счетчика можно производить только по объектам, расположенным выше в данных, или делить всю выборку на фолды и считать значения счетчиков на текущем фолде по всем остальным.

3 Предобработка данных

Данный датасет изначально содержит 21613 записей о ценах домов. Каждая запись представляет из себя цену, которую мы хотим предсказать и набор признаков, описывающих выбранный дом: id, 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. Итого 20 признаков.

В датасете отсутствуют пропущенные значения, поэтому этап заполнения пропусков не рассматривается.

В данных присутствует признак **id**, который уникальный для всех записей и представляет из себя идентификатор дома, поэтому этот признак не будет информативным и сразу может быть убран из рассмотрения. Также в данных есть признак **date**, который представляет из себя строку с информацией о временной метке. В таком виде этот признак не будет корректно воспринят любой из рассматриваемых моделей, поэтому признак был заменен на признаки: год, месяц и день недели. Из интуитивных соображений эти признаки действительно могут быть полезны. Например, дома люди покупаю чаще всего ближе к началу зимы (поэтому месяц важен), а покупка может чаще совершаться на выходных из соображений того, что человек перед совершением покупки хотел бы еще раз придти в дом, все осмотреть, но в будний день это сделать затруднительно (день недели важен). Таким образом количество признаков увеличилось до 22.

Разделим исходную выборку на обучающую и валидационную в соотношении 7:3. Тогда размер обучающей выборки получится 15129 записей, а тестовой 6484.

По описанию данных можно сделать вывод о наличии двух категориальных признаков: **zipcode** и **waterfront**. Признак **zipcode** есть почтовый адрес дома, а **waterfront** принимает лишь значения 1 либо 0 и отвечает за расположение дома у водоема или нет соответственно. Для корректной работы моделей с категориальными признаками их необходимо закодировать. Рассматривалось два способа кодирования: one-hot-кодирование и mean-target кодирование. Т.к. признак **zipcode** имеет 70 уникальных значений, то one-hot-кодирование увеличило бы признаковое пространство в несколько раз, что негативно бы сказалось на времени обучения ансамблей, поэтому выбор был сделан в пользу mean-target кодирование, а конкретнее на счетчики со сглаживанием (см. Пояснение к задаче).

Для числовых признаков нормирование производить не нужно, т.к. семейство рассматриваемых алгоритмов умеет работать с признаками любых масштабов.

4 Линейная регрессия

Для того, чтобы увидеть, как качество, полученное с помощью Случайного леса и Градиентного бустинга соотносятся с качеством, полученным с помощью более простых моделей, на обучающей выборке была обучена обычная линейная регрессия. На валидации эта модель показала качество **RMSE=165896**.

5 Случайный лес

В данных экспериментах нужно изучить зависимость **RMSE** на отложенной выборке и время работы алгоритма в зависимости от различных параметров.

5.1 Количество деревьев в ансамбле

В эксперименте рассматривается зависимость **RMSE** на обучающей и валидационной выборках от числа деревьев в ансамбле.

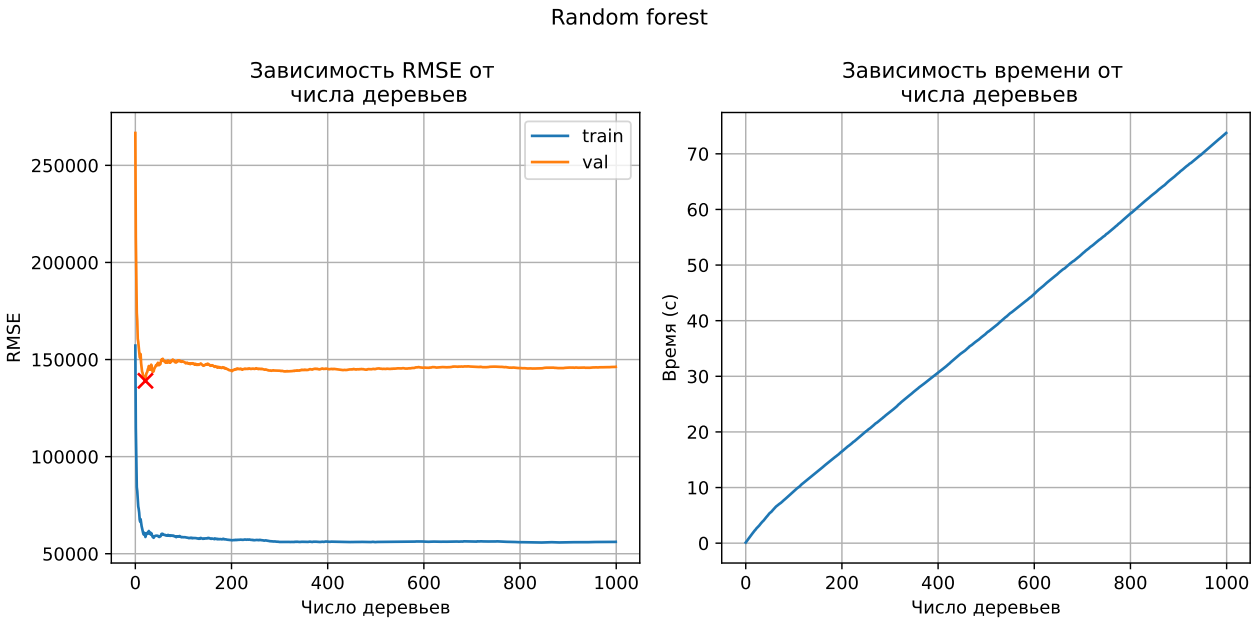


Рис. 1:

Из графиков видно, что время работы обучения ансамбля линейно зависит от числа деревьев в ансамбле, а минимальное значение функционала ошибки на валидационной выборке достигается там, где стоит красный крест, после чего начинается

переобучение (значение функционала ошибки уменьшается на обучающей выборке и увеличивается на валидационной). Красный крест соответствует **n_estimators=22**.

5.2 Размерность подвыборки признаков для одного дерева

В данном эксперименте рассматривается зависимость **RMSE** от размерности подвыборки признаков для одного дерева.

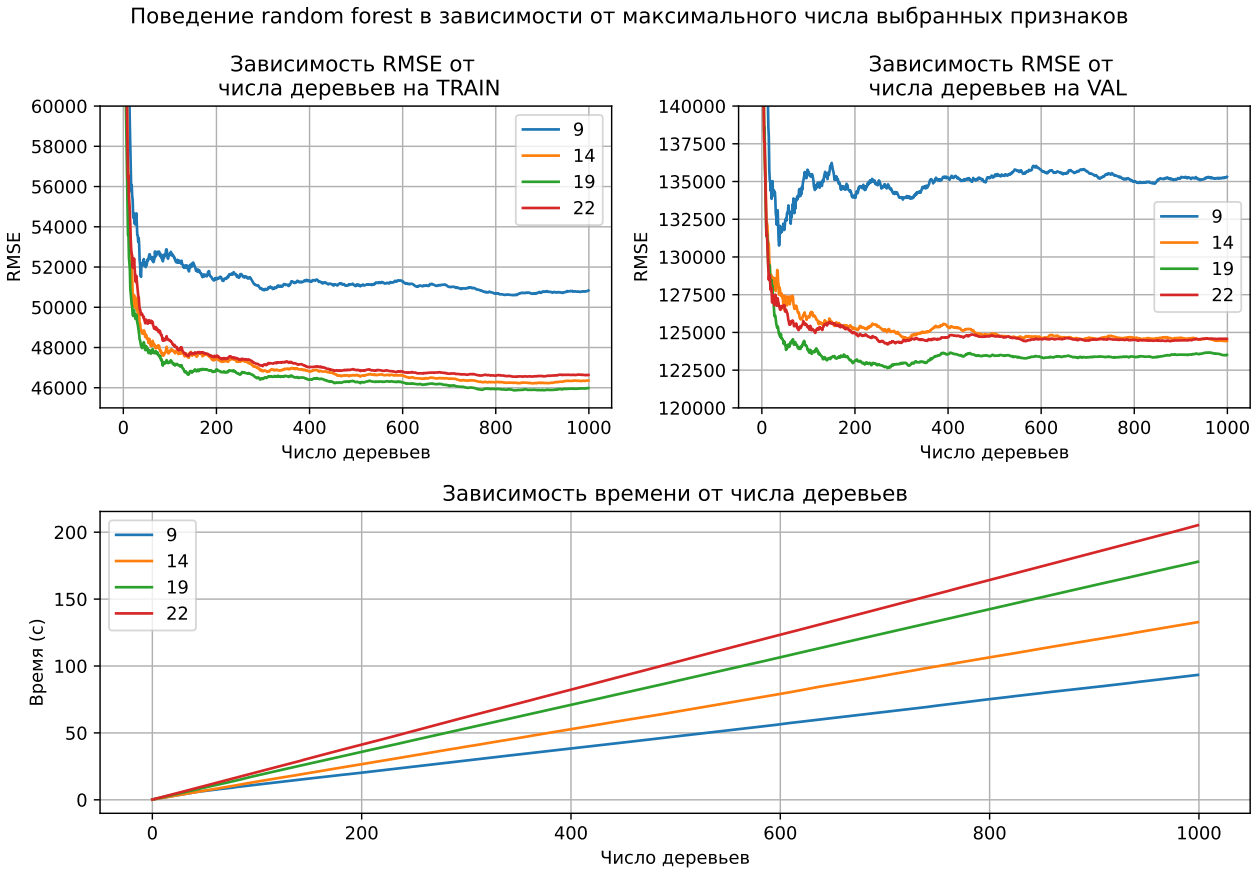


Рис. 2:

Из теории известно, что при подборе гипперпараметров для случайного леса самым важным является именно размерность признакого пространства. Из верхнего правого графика видим, что перебор данного параметра сильное меняет качество на валидации. Разница между лучшим и худшим результатами около 15000. Лучшее число признаков на валидации получилось **19**. Зафиксируем это значение для следующего эксперимента для дальнейшего улучшения качества.

Время обучения ожидаемо больше при большем числе рассматриваемых признаков.

5.3 Максимальная глубина дерева

В данном эксперименте рассматривается зависимость **RMSE** от максимальной глубины одного дерева в ансамбле.

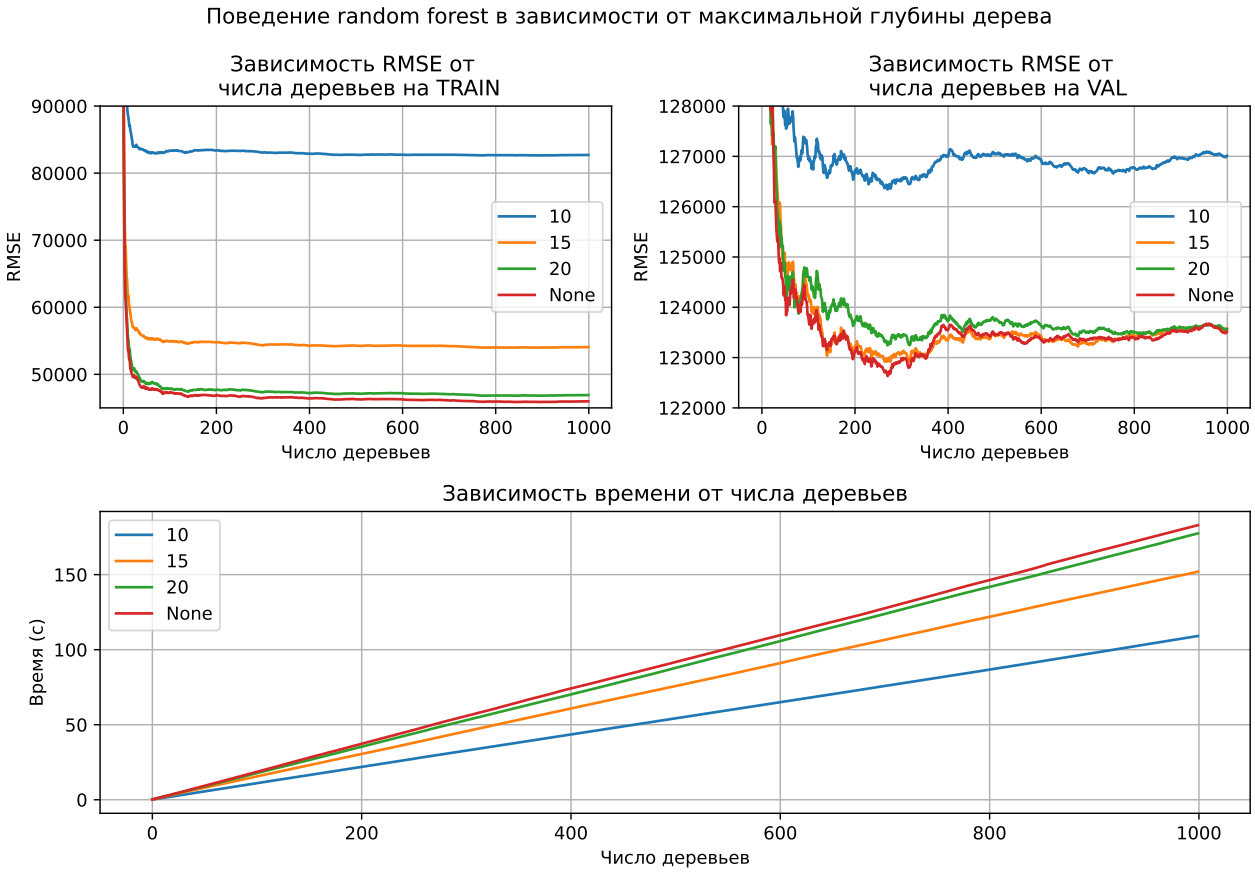


Рис. 3:

Из теории мы знаем, что суть случайного леса заключается в том, что каждый базовый алгоритм сильно переобучается, что позволяет уменьшить смещение, а разброс уменьшается с помощью усреднения базовых алгоритмов. У сильно переобученного дерева большая глубина, поэтому полученный здесь результат об оптимальном значении при неограниченной глубине (**None**) согласуется с теорией.

Время ожидаемо больше при росте глубины.

Наименьшее значение функционала ошибки для случайного леса получилось: **122630**

6 Градиентный бустинг

Все эксперименты совпадают со случаем для случайно леса, кроме того, что добавляется еще зависимость **RMSE** от learning_rate.

6.1 Количество деревьев в ансамбле

В эксперименте рассматривается зависимость **RMSE** на обучающей и валидационной выборках от числа деревьев в ансамбле.

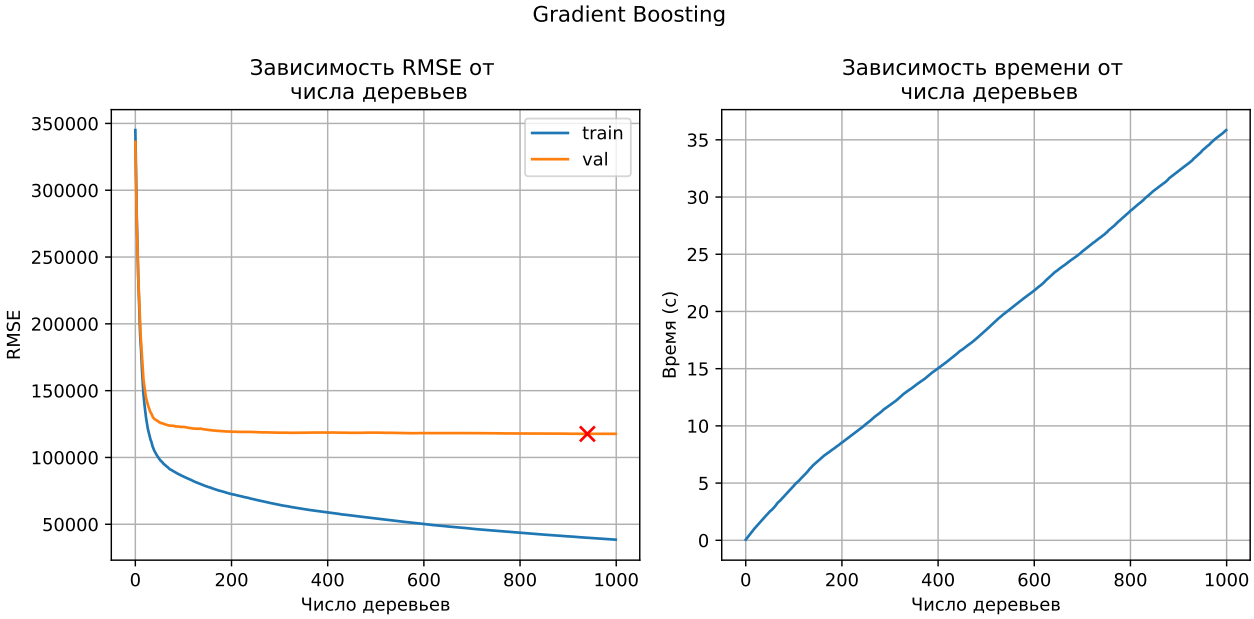


Рис. 4:

Заметим, что ошибка для тренировочной выборки монотонно убывает. Это связано с тем, что каждый следующий алгоритм оценивает направление антиградиента, а дальше происходит адаптивный подбор шага, что приводит каждый раз к шагу в локально оптимальную точку в направлении антиградиента, что означает уменьшение функционала ошибки на обучающей выборке.

В отличии от случайного леса здесь оптимальное число базовых моделей оказалось сильно больше. Это связано с тем, что в градиентном бустинге все деревья как

правило не глубокие, то есть это простые модели, а значит нам их нужно много. Оптимальное число **n_estimators=940**.

Время ожидаемо линейно зависит от числа базовых алгоритмов, причем градиентный бустинг с параметрами по умолчанию обучился быстрее, чем случайный лес для этого же случая.

6.2 Размерность подвыборки признаков для одного дерева

В данном эксперименте рассматривается зависимость **RMSE** от размерности подвыборки признаков для одного дерева.

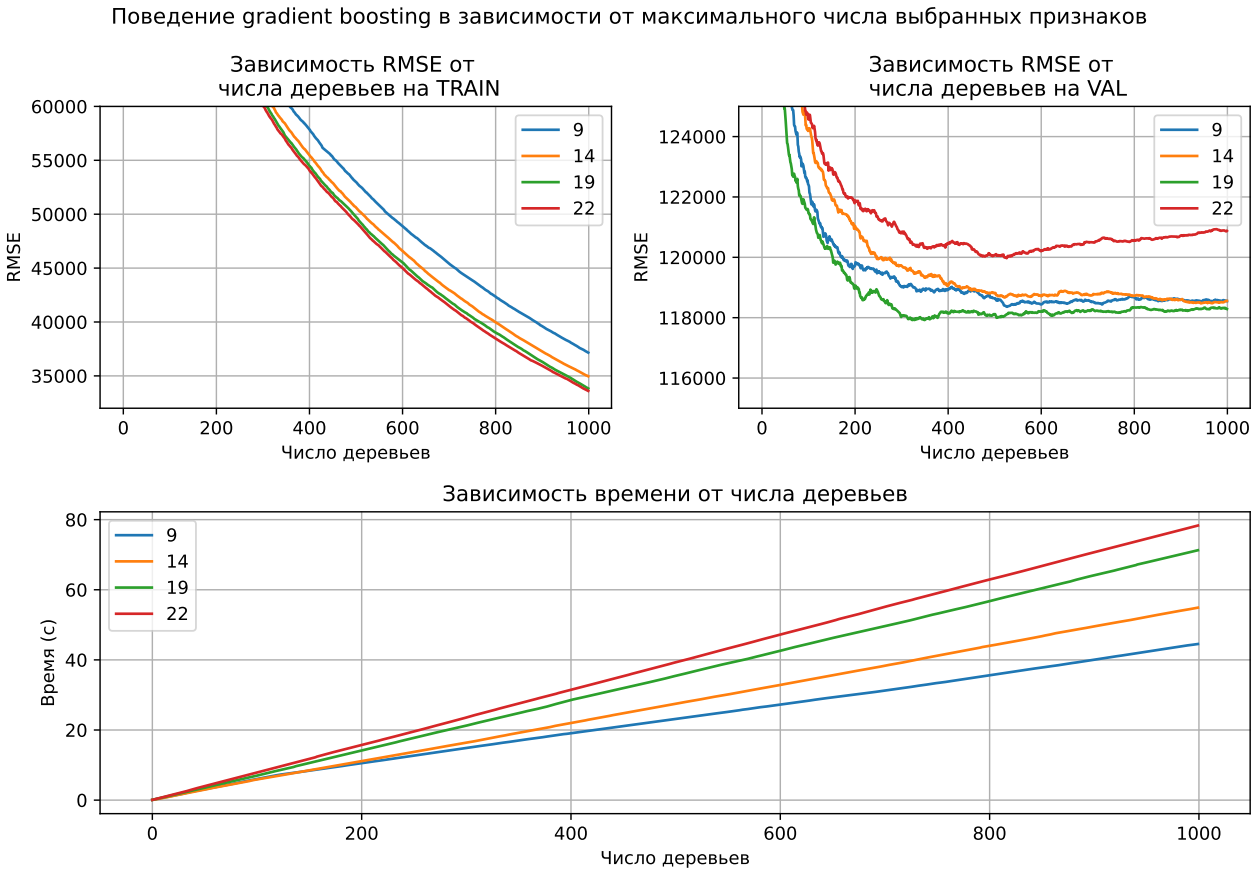


Рис. 5:

Заметим, что набор параметров для размеров подвыборок признаков совпадает с набором в эксперименте для случайного леса, но при этом разброс лучшего и худшего алгоритмов на валидации уже составляет около 2000, что гораздо меньше,

чем 15000 для случайного леса. Отсюда можно сделать вывод, что этот признак для градиентного бустинга менее важен, чем для случайного леса.

Оптимальным параметром является **19**. Зафиксируем его для дальнейших экспериментов.

С временим обучения ситуация аналогичная случайному лесу.

6.3 Максимальная глубина дерева

В данном эксперименте рассматривается зависимость **RMSE** от максимальной глубины одного дерева в ансамбле.

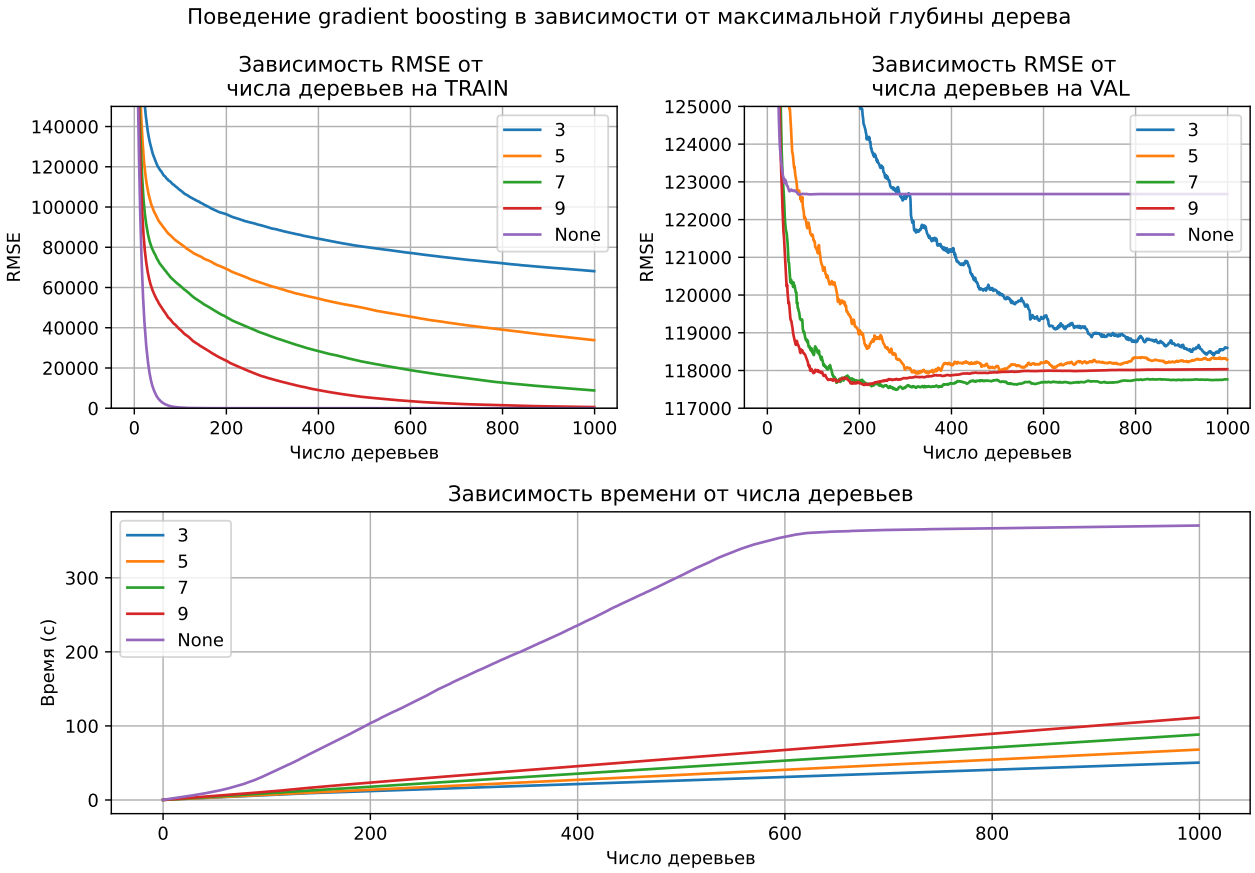


Рис. 6:

Т.к. градиентный бустинг хорошо работает с неглубокими базовыми алгоритмами, то масштаб сетки перебора здесь меньше, чем для случайного леса. При этом мы видим, что при глубине None (нет ограничения на глубину) модель настолько сильно

переобучается, что ошибка на обучающей выборке становится 0, а на валидационной становится большой соответственно.

Оптимальным параметром на валидации является **max_depth=7**. Зафиксируем его для следующего эксперимента.

Со временем обучения ситуациия аналогичная случайному лесу. Выход фиолетового графика на плато связан с тем, что ошибка на обучающей выборке стала равно 0.

6.4 Значение learning_rate

В данном эксперименте рассматривается зависимость **RMSE** от значения learning_rate.

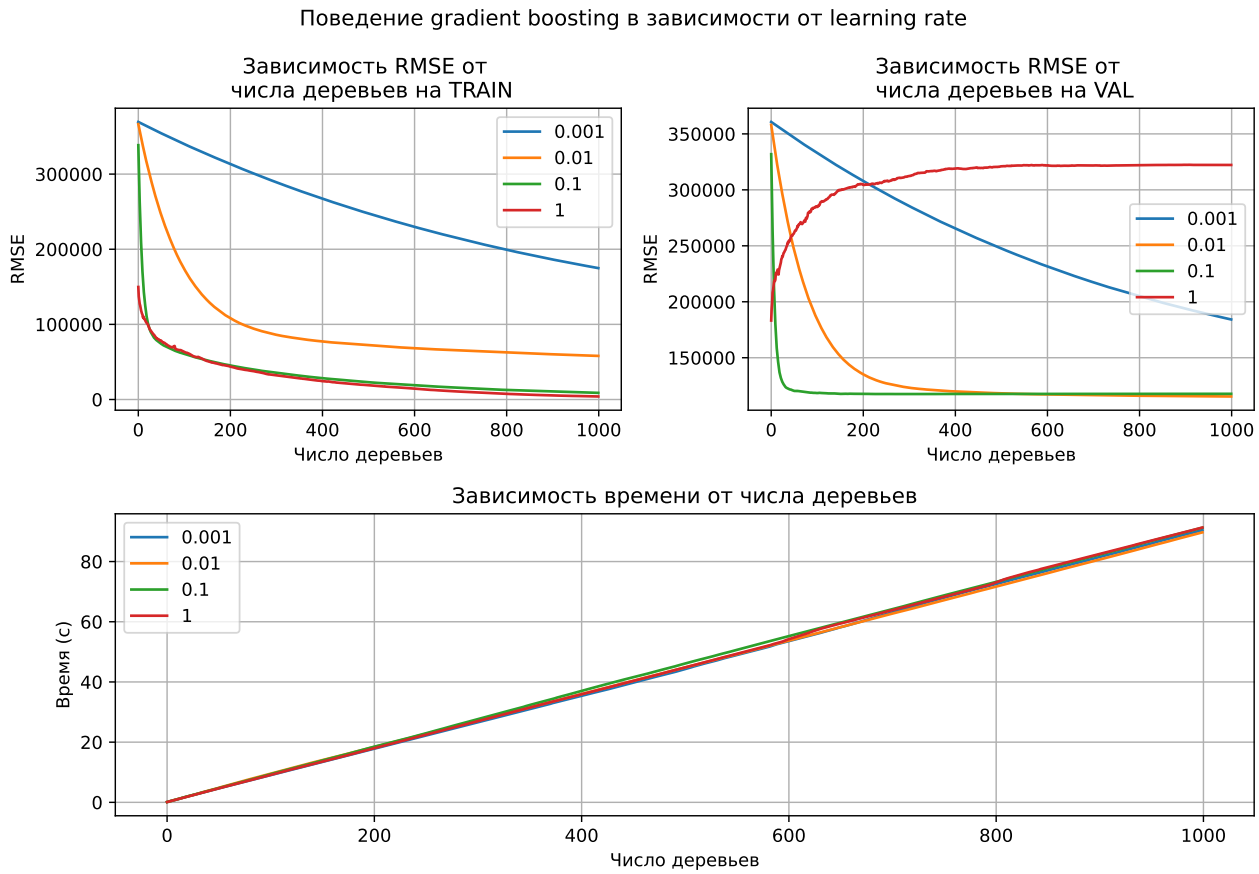


Рис. 7:

Видим, что время обучения практически не зависит от learning_rate. При learning_rate=1 наблюдается расходимость на валидации. Оптимальным значением по валидации будет **learning_rate=0.01**.

Наименьшее значение функционала ошибки для случайного леса получилось: **115386**.

7 Выводы

Случайный лес показал качество на валидации в **122630**, градиентный бустинг в **115386**, а обычная линейная регрессия **165896**. Видно, что две рассмотренные модели имеют существенный прирост по сравнению с линейной регрессией, что характеризует их как отличный вариант для задачи регрессии с табличными данными.

Для случайного леса на основе экспериментов самым важным параметром является максимальное число признаков для очередного базового дерева, глубину деревьев всегда можно делать неограниченной, а количество деревьев в ансамбле можно просто брать каким-то большим.

Для градиентного бустинга при подборе параметров важны как число признаков, так и глубина деревьев, при этом перебираемые глубины должны быть небольшими. Подбор оптимального learng_rate тоже важен, т.к. для слишком больших значений алгоритм может просто не сойтись, а для слишком маленьких сходиться долго.

В итоге лучшее качество продемонстрировал градиентный бустинг, что соотносится с реальность.