МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИМЕНИ М.В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики

ММП - 3 курс

Композиции алгоритмов для решения задачи регрессии.

Автор:
Грозный Сергей

Содержание

1 Введение

Целью данной лабораторной работы является собственная реализация таких методов машинного обучения, как случайный лес и градиентный бустинг, а также исследование зависимости этих методов от выбора гиперпараметров.

2 Список экспериментов

Для экспериментов возьмем данные о продажах недвижимости в США. В данной задаче необходимо предсказать стоимость недвижимости, опираясь на его характеристики : id, количество спален, площадь этажей, дата покупки и так далее. Заметим, что в данных нет пропусков.

Исследуем поведение RMSE и время работы алгоритма на отложенной выборке в зависимости от параметров. Разобьем нашу выборку на обучающую, отложенную и тестовую в соотношении 8:1:1. Отметим, что если не оговорено иное, то параметры по умолчанию: количество деревьев - 100, макс. глубина не ограничена, размерность признакового подпространства - количество признаков нацело делить на 3.

2.1 Предобработка данных

Для дальнейшей работы необходимо обработать сырой датасет. Выбросим столбец с id, а дату разобьем на три колонки: год, месяц, день. Опираясь на описания признаков, закодируем категориальные с помощью one hot encoding. Некоторые признаки можно считать как и некатегориальные, но так как в последующих экспериментах будем исследовать зависимость от размерности подпространства признаков, то будем отдавать предпочтение в сторону увеличения количества признаков. Также применим к некатегориальным признакам MinMaxScaler. Итого, после всех преобразований получаем общее количество признаков: 336.
Построим распределение цены недвижимости:

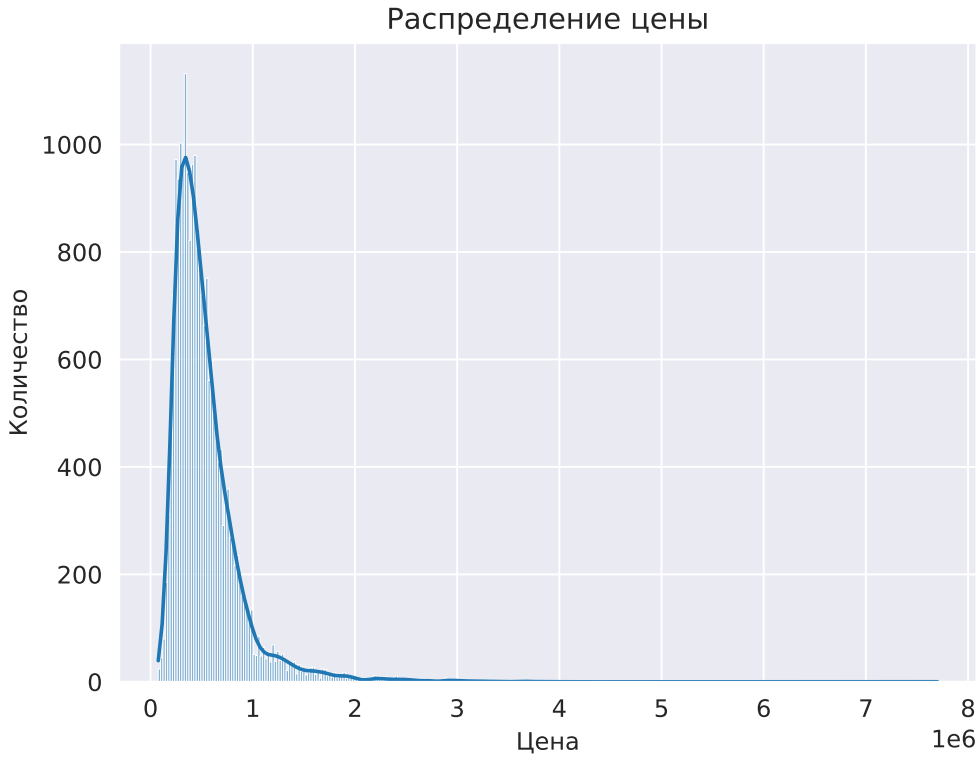

Рис. 1: Распределение цены. Выбросы достигают аж до $8 * 10^6$.

Как видно из графика 1, в данных присутствуют выбросы. Удалим их таким образом, чтобы итоговый датасет отличался от исходного на 1%.

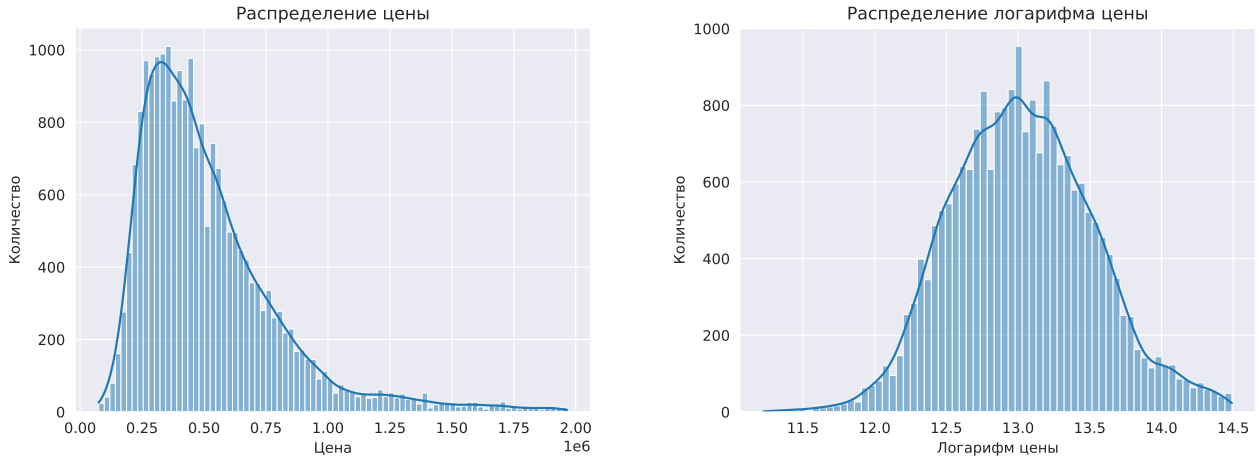

Рис. 2: **Слева** - распределение цены после удаления 1% данных. **Справа** - распределение логарифма цены.

Полученное распределение похоже на лог нормальное, поэтому в дальнейших исследованиях будем предсказывать логарифм цены.

2.2 Случайный лес

Случайный лес — модель, состоящая из множества деревьев решений. Для каждого дерева генерируется случайная выборка: индексы берутся случайно 'с возвращением', а признаки генерируются случайно, количество которых регулируется параметром. Так как наша задача относится к регрессии, то итоговый результат - это среднее ответов всех деревьев.

2.2.1 Количество деревьев

Зависимость RMSE и время работы алгоритма от количества деревьев изображены на Рис. 3:

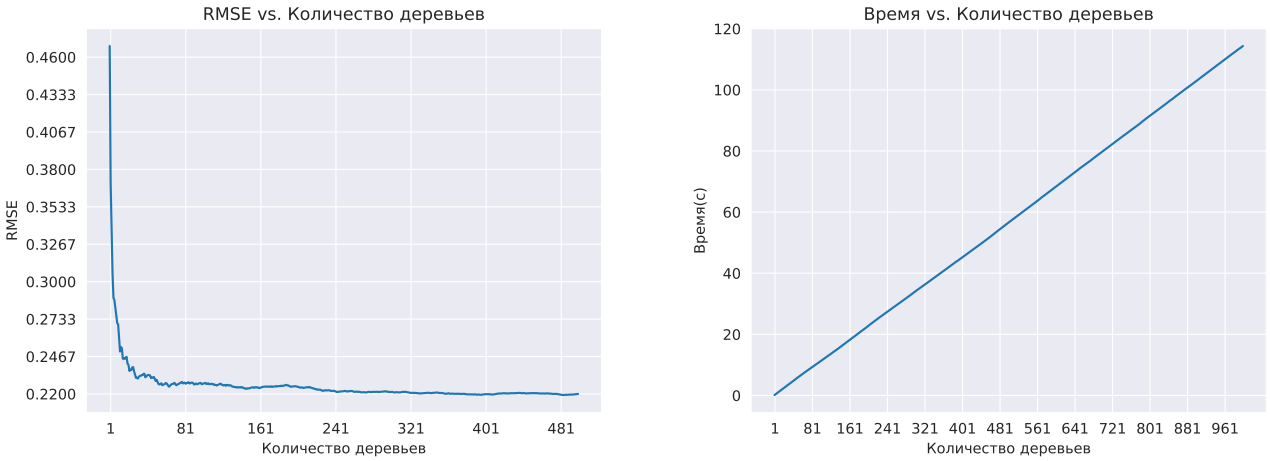

Рис. 3: Случайный лес. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Приблизительно до 240 деревьев видим колебания графика, а дальше идет стабилизация. При малом количестве деревьев ответ сильно зависит от случайности выборки. При большем количестве - меньше, по скольку случайность нивелируется усреднением ответов, и по итогу алгоритм начинает выдавать примерно один и тот же ответ. Время алгоритма с увеличением количества деревьев растет линейно. Это следует из реализации модели: деревья обучаются в цикле. Минимум достигается при 483.

2.2.2 Размерность признакового подпространства

Построим графики в зависимости от размерности признакового подпространства:

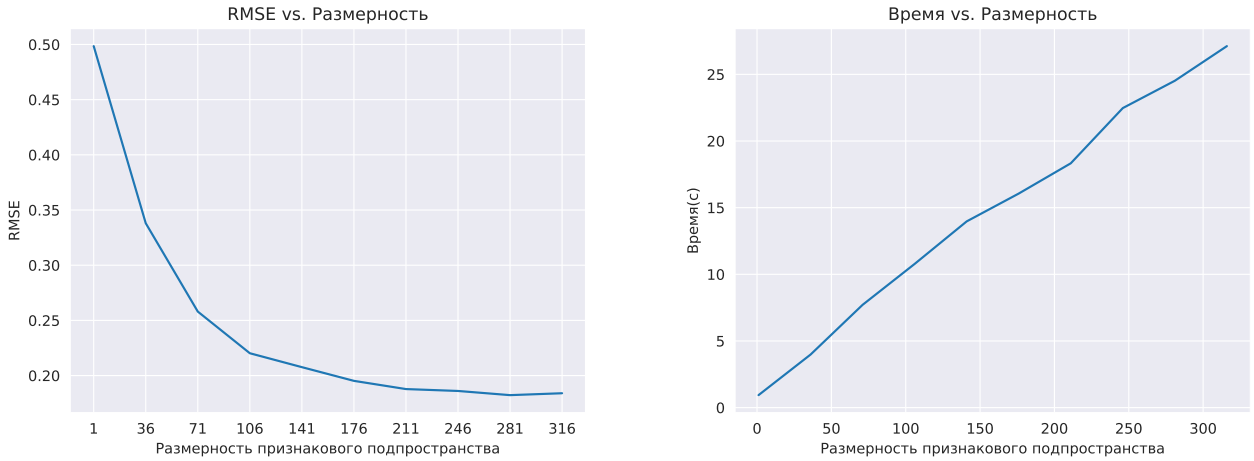

Рис. 4: Случайный лес. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Так как количество деревьев фиксировано, причем равное 100, низкой размерности признаков оказалось недостаточно, чтобы достичь низкой ошибки. Отметим, что время растет линейно. Минимум при 316.

2.2.3 Максимальная глубина

Проиллюстрируем зависимость RMSE и время работы алгоритма от максимальной глубины:

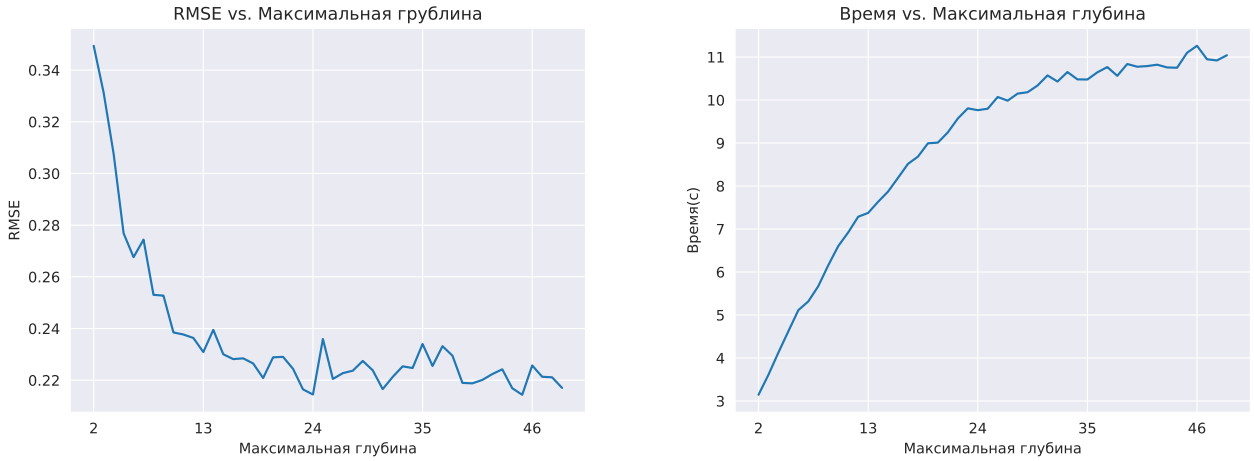

Рис. 5: Случайный лес. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Как видно из Рис. 5, сначала график убывает, так как алгоритм слишком рано "обрубает"деревья, а при большей глубине модель переобучается, поэтому происходят колебания на отложенной выборке. Заметим, что при увеличении максимальной глубины время растет нелинейно. Минимум при 24.

2.3 Градиентный бустинг

Градиентный бустинг – это продвинутый алгоритм машинного обучения для решения задач классификации и регрессии. Он строит предсказание в виде ансамбля слабых предсказывающих моделей, которыми в основном являются деревья решений. Из нескольких слабых моделей в итоге мы собираем одну, но уже эффективную.

Общая идея алгоритма – последовательное применение предсказателя таким образом, что каждая последующая модель сводит ошибку предыдущей к минимуму.
Аналогично случайному лесу исследуем поведение градиентного бустинг в зависимости от параметров.

2.3.1 Количество деревьев

Зависимость RMSE и время работы алгоритма от количества деревьев изображены на Рис. 6:

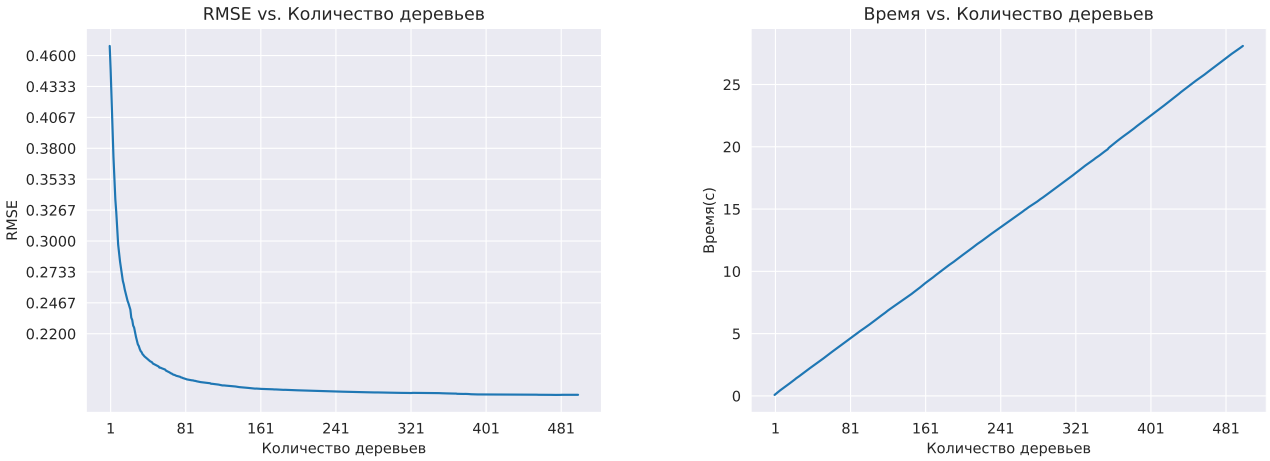

Рис. 6: Градиентный бустинг. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Так как у градиентого бустинга случайности меньше, чем у выше исследованного алгоритма, то шумов в графике RMSE меньше. С увеличением количества деревьев качество становится лучше. Время алгоритма растет линейно. Минимум при 477.

2.3.2 Размерность признакового подпространства

Построим графики в зависимости от размерности признакового подпространства:

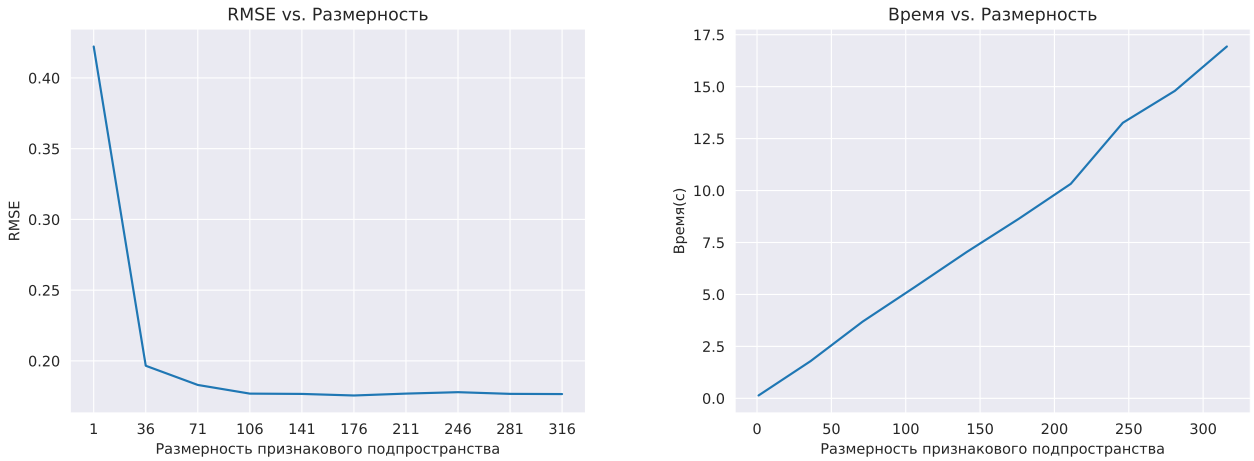

Рис. 7: Градиентный бустинг. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Отметим, что как и в случае случайного леса низкой размерности признаков оказывается недостаточно, чтобы достичь низкой ошибки. Время ожидаемо растет линейно. Минимум при 176.

2.3.3 Максимальная глубина

Проиллюстрируем зависимость RMSE и время работы алгоритма от максимальной глубины:

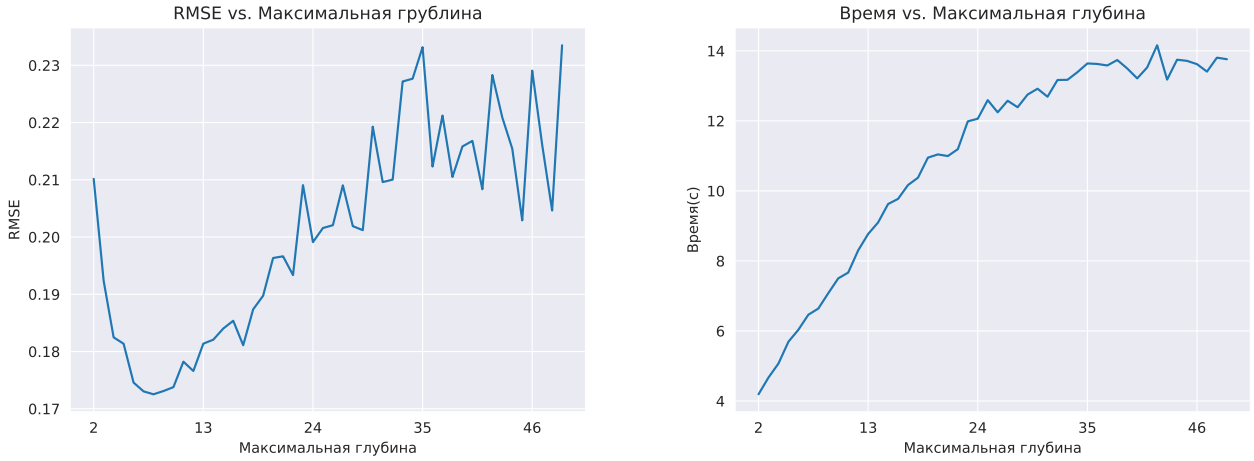

Рис. 8: Градиентный бустинг. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

На графике 8 наблюдается U-образная кривая. Это связано с тем, что максимальная глубина сильно влияет на переобучение алгоритма. В данном случае оптимальная глубина равна 8. Аналогично случайному лесу, время растет нелинейно.

2.3.4 Learning rate

Рассмотрим параметр, которого нет у случайного леса - learning rate. Этот параметр борется с переобучением модели. Зависимость RMSE от него следующая:

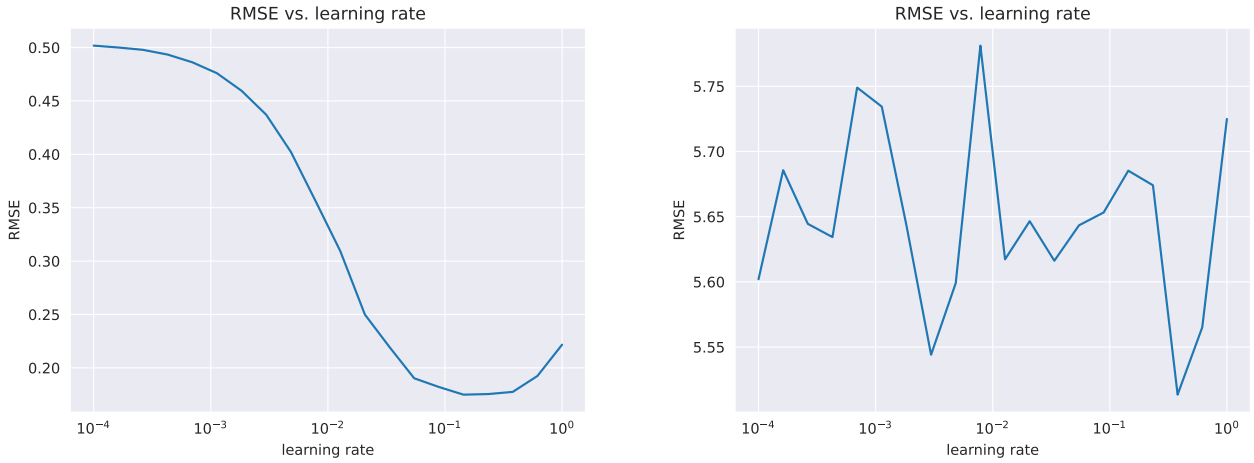

Рис. 9: Градиентный бустинг. **Слева** - RMSE на отложенной выборке. **Справа** - Время работы алгоритма.

Как видно из графика 9, RMSE довольно высокий при низком learnig rate($\leq 10^{-3}$). Оптимальный лежит на отрезке $[10^{-2}, 1]$ и равен 0.14. Отметим, что время практически не зависит от learning rate.

2.4 RMSE на тестовой выборке

Если обучить случайный лес и градиентный бустинг с оптимальными параметрами, полученными выше, то RMSE на тесте будет равен **0.177** и **0.167** соответственно. О чем можно сделать вывод, что в данной задаче градиентный бустинг показал себя лучше.

3 Вывод

Случайный лес и градиентный бустинг - два популярных алгоритма машинного обучения, которые используются для задач классификации и регрессии. Они оба строят несколько деревьев решений и объединяют их прогнозы для получения окончательного прогноза.
Однако между ними есть некоторые ключевые различия:

- Метод обучения: Случайный лес строит деревья независимо, в то время как градиентный бустинг строит деревья последовательно, где каждое дерево пытается исправить ошибки предыдущего дерева.
- Метод прогнозирования: Случайные леса делают прогнозы путем усреднения прогнозов каждого отдельного дерева, в то время как градиентное бустинг делает прогнозы путем сложения прогнозов каждого дерева.
- Скорость: В теории случайный лес можно ускорить, если распараллелить обучение деревьев, чего не сделаешь в градиентном бустинге.

Также отметим существенную разницу между графиками зависимостей от максимальной глубины. Градиентный бустинг имеет U-образную кривую, в то время как случайный лес имеет убывание при увеличении максимальной глубины. В целом, оба алгоритма могут быть эффективны в различных ситуациях, однако в нашей задаче побеждает градиентный бустинг.

4 Appendix

Таблица с описанием признаков:

Variable	Description
Id	Unique ID for each home sold
Date	Date of the home sale
Price	Price of each home sold
Bedrooms	Number of bedrooms
Bathrooms	Number of bathrooms, where .5 accounts for a room with a toilet but no shower
Sqft_living	Square footage of the apartments interior living space
Sqft_lot	Square footage of the land space
Floors	Number of floors
Waterfront	A dummy variable for whether the apartment was overlooking the waterfront or not
View	An index from 0 to 4 of how good the view of the property was
Condition	An index from 1 to 5 on the condition of the apartment,
Grade	An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design
Sqft_above	The square footage of the interior housing space that is above ground level
Sqft_basement	The square footage of the interior housing space that is below ground level
Yr_built	The year the house was initially built
Yr_renovated	The year of the house's last renovation
Zipcode	What zipcode area the house is in
Lat	Lattitude
Long	Longitude
Sqft_living15	The square footage of interior housing living space for the nearest 15 neighbors
Sqft_lot15	The square footage of the land lots of the nearest 15 neighbors

Рис. 10: Таблица признаков. Источник [1]

В качестве катигориальных признаков рассматривали: 'year', 'month', 'day', 'view', 'condition', 'grade', 'yr_built', 'yr_renovated', 'zipcode', 'waterfront', где 'year', 'month', 'day' - признаки, полученные из даты.

Список литературы

[1] https://www.slideshare.net/PawanShivhare1/predicting-king-county-house-prices