

# Задание 3. Композиции алгоритмов для решения задач регрессии

Бикметов Данил, 317 группа

Декабрь, 2020

## Введение

В практическом задании были реализованы собственные методы RandomForest и GradientBoosting. Экспериментальная часть задания проводилась на датасете данных о продажах недвижимости.

## Практическая часть

Для начала сделаем константное предсказание средним и посчитаем ошибку:  $Baseline RMSE = 361.469$ . Это поможет нам понять масштаб ошибки предсказания в дальнейшем. Теперь перейдём к экспериментам со случайным лесом и градиентным бустингом.

## Исследование поведения алгоритма случайный лес

В этом разделе мы изучим зависимость  $RMSE$  на отложенной выборке и время работы алгоритма в зависимости от следующих параметров:

- количество деревьев
- размерность подвыборки признаков для одного дерева
- максимальная глубина дерева (+ случай, когда глубина неограниченна)

## 1. Количество деревьев

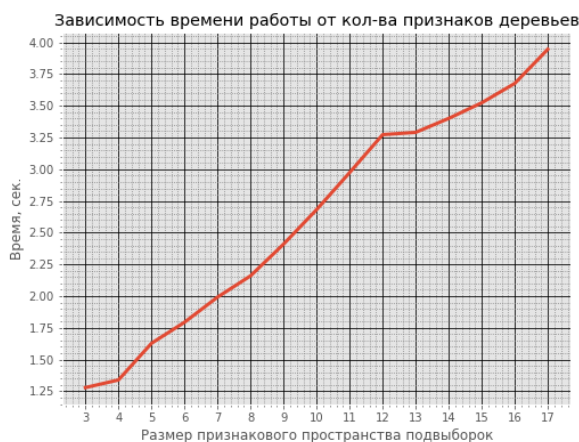
Изучим зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от количества деревьев. Изобразим зависимость на графике:



С ростом количества деревьев качество алгоритма на отложенной выборке растёт, но минимальная ошибка достигается при 500 деревьях. На графике присутствуют две "ямы": на 100 и на 500 деревьях. Логично предположить, что с дальнейшим ростом кол-ва деревьев алгоритм переобучается и качество на отложенной выборке будет меньше. Время работы алгоритма растёт линейно с ростом количества деревьев.

## 2. Размерность подвыборки признаков для одного дерева

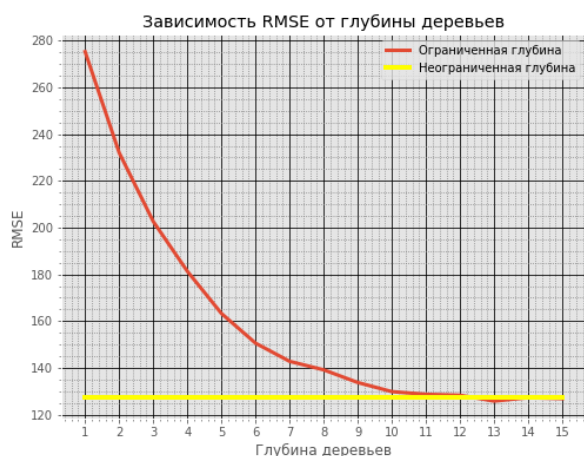
Теперь изучим зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от размера признакового пространства подвыборок.



Чем больше признаковое пространство деревьев, тем выше качество алгоритма. В данном случае это можно связать с тем, что общее количество признаков небольшое (всего 17). Однако оптимальный результат достигается при 13-15 признаках. Время работы алгоритма растёт линейно с ростом числа признаков в деревьях, хотя с 12 признаков время работы растёт заметно медленнее.

### 3. Максимальная глубина дерева

Изучим зависимость  $RMSE$  на отложенной выборке и время работы алгоритма в зависимости от глубины деревьев. Предварительно заметим, что глубину каждого дерева можно неограничивать вовсе. Отдельно посчитаем  $RMSE$  для неограниченной глубины деревьев и укажем её на графике константной желтой линией для сравнения.



Качество алгоритма улучшается с увеличением глубины деревьев. С небольшой погрешностью можно сказать, что ошибка алгоритма с ростом глубины деревьев асимптотически стремится к ошибке алгоритма с неограниченными деревьями. Время работы алгоритма растёт линейно. Стоит отметить, что время работы алгоритма в случае неограниченной глубины деревьев составило 9.83 секунды, что примерно соответствует глубине 17, если линейно продолжить зависимость на втором графике.

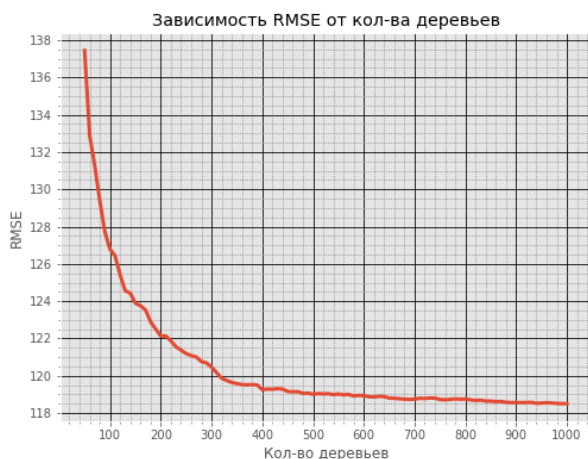
## Исследование поведения алгоритма градиентного бустинга

В этом разделе мы изучим зависимость  $RMSE$  на отложенной выборке и время работы алгоритма в зависимости от следующих параметров:

- количество деревьев
- размерность подвыборки признаков для одного дерева
- максимальная глубина дерева (+ случай, когда глубина неограниченна)
- выбранный  $learning\_rate$

### 1. Количество деревьев

Изучим зависимость  $RMSE$  на отложенной выборке и время работы алгоритма в зависимости от количества деревьев. Изобразим зависимость на графике:



В данном случае очевидно, что с ростом количества деревьев ошибка монотонно снижается. Это согласуется с интуицией, поскольку с каждой итерацией алгоритм находит всё лучшее приближение. Время работы алгоритма растёт линейно с ростом числа деревьев.

## 2. Размерность подвыборки признаков для одного дерева

Теперь изучим зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от размера признакового пространства подвыборки.

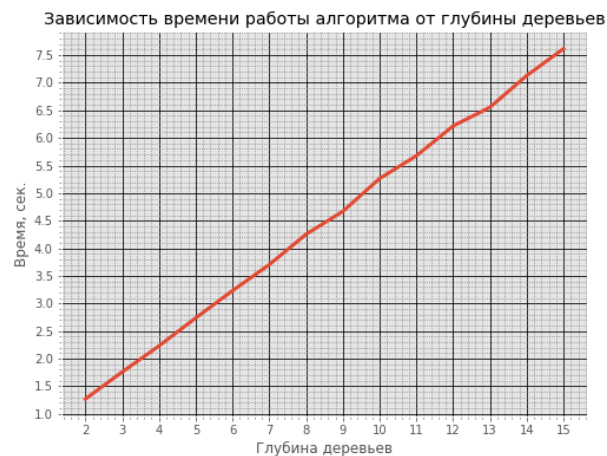
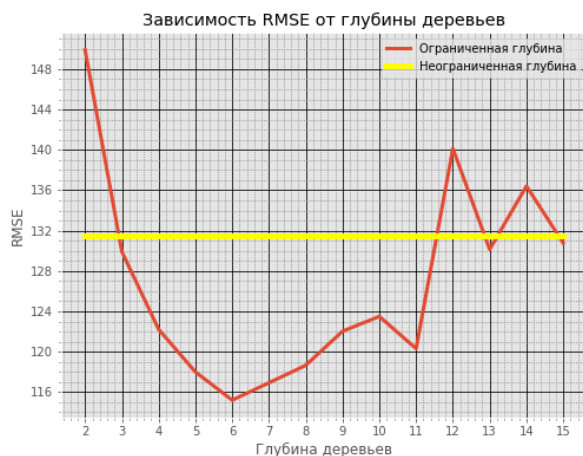


В этом случае результат не так очевиден: прослеживается своеобразная "яма" в середине графика, а оптимальное значение ошибки достигается при 10 признаках. В этом случае сложно делать какие либо выводы, поскольку общее количество признаков невелико. Время работы алгоритма растёт линейно с ростом размера признакового пространства деревьев.

## 3. Максимальная глубина дерева

Изучим зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от глубины деревьев. Как и в случае с RandomForest, глубину

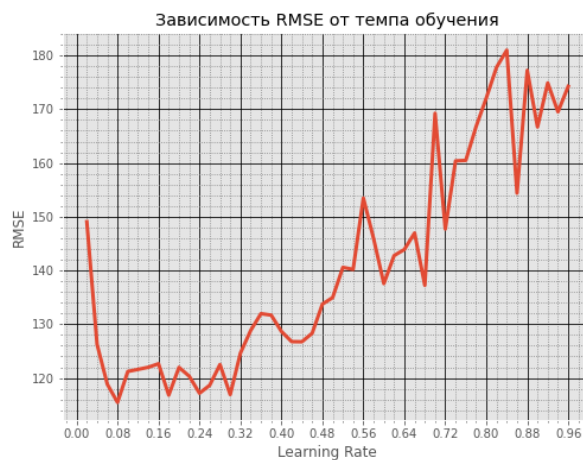
каждого дерева можно неограничивать вовсе. Поэтому отдельно посчитаем  $RMSE$  для неограниченной глубины деревьев и укажем её на графике константной желтой линией для сравнения.



Оптимальная глубина - 6. При дальнейшем увеличении глубины ошибка алгоритма растёт. Следовательно, при неограниченной глубине качество должно получиться относительно низким, что мы и видим на графике. Время работы алгоритма растёт линейно с ростом максимальной глубины. Время работы алгоритма в случае неограниченной глубины деревьев равно 10 секундам, что соответствует глубине 20, если продолжить линейную функцию на втором графике.

#### 4. Темп обучения

Наконец, изучим зависимость  $RMSE$  на отложенной выборке и время работы алгоритма в зависимости от выбора  $learning\_rate$ .



Ошибка минимальная при небольших  $learning\_rate$ . Оптимальным значением можно считать  $learning\_rate = 0.08$ , поскольку на нём достигается хорошее качество и алгоритм работает быстро. При дальнейшем увеличении параметра качество падает. Нет гладкой зависимости времени работы алгоритма от  $learning\_rate$ , но

при  $learning\_rate > 0.12$  время работы повышается. Однако отличие составляет десятые доли секунды, поэтому говорить о прямой зависимости нельзя.

## Пару слов о реализации сервера

К заданию был реализован сервер на Flask, с помощью которого пользователь, не знающий Machine Learning, имеет возможность обучить модель и получить предсказание. На сервере имеется возможность выбирать алгоритм, менять гиперпараметры, следить за обучением. Удалось реализовать следующие возможности:

- Файлы на сервер загружаются с помощью формы *input* типа *file*. Для этого я сохраняю выбранный пользователем файл в отдельную папку, относительный путь до которой известен. Далее работа ведётся непосредственно с этим файлом.
- Принимаются только файлы формата .csv. Пользователю нужно отдельно загрузить обучающий файл без целевой переменной и файл со значениями целевой переменной.
- Настраивать можно следующие параметры:
  - количество деревьев
  - размер признакового пространства деревьев (можно установить по умолчанию: берётся  $d/3$ , где  $d$  - кол-во всех признаков)
  - глубина деревьев (можно не ограничивать)
  - $learning\_rate$  для градиентного бустинга. Для удобства пользователя его можно выбирать только кратным 0.01 из отрезка  $[0.01; 1]$
- После обучения выводится информация о модели: гиперпараметры и значение функции потерь на каждой итерации
- Есть возможность загрузить файл и скачать предсказание по нему. На компьютер пользователя скачивается .csv файл с названием prediction, в котором два столбца: с индексом и со значением целевой переменной.

Важное замечание: сервер автоматически убирает категориальные признаки, имеющие формат, отличный от `int`, `float`. Это сделано по той причине, что пользователь не имеет возможности изменить способ кодирования категориальных признаков. Стало быть, программа не должна решать за него.