Задание 3. Композиции алгоритмов для решения задачи регрессии

Деев Александр Сергеевич
3 курс «Практикум на ЭВМ» ММП ВМК МГУ

13 декабря 2019 г.

1. Задача

При помощи моделей RandomForest и GradientBoosting определить цену на недвижимость. Исследовать поведение алгоритмов при различных значениях параметров моделей.

2. Эксперименты

Перед исследованием проведем минимальную обработку данных:

Дату можно удалить, так как скорее всего это дата выхода на рынок (дата постройки дома есть, и это, по моему мнению, должно влиять на ответ). Также можно удалить колонки id и index, так как по смыслу задачи они не должны влиять на цену недвижимости.

Разделим датасет с размером X_test = 0.25 * Data на обучение и контроль.

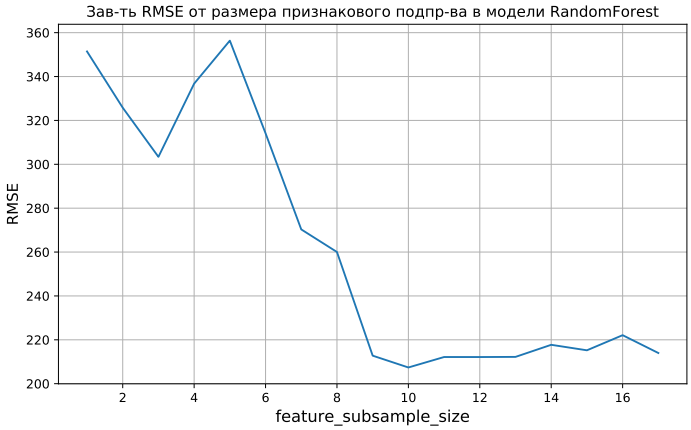
2.1 Random Forest

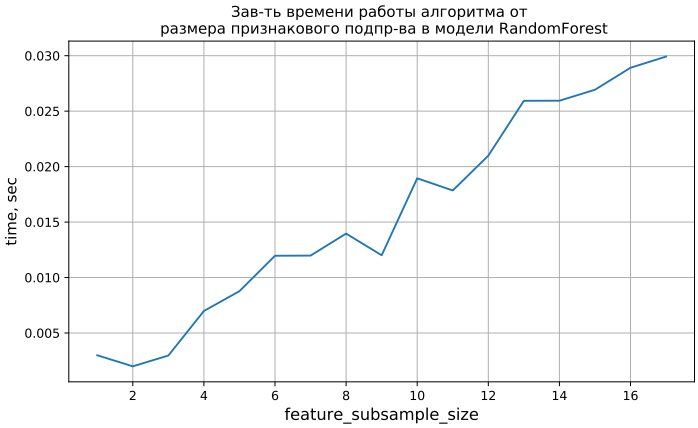
2.1.1 Задача

Исследовать алгоритм RandomForest на качество прогноза RMSE и время его работы в зависимости от количества деревьев (n_estimators), размерности признакового подпространства (feature_subsample_size) и максимальной глубины дерева (max_depth).

2.1.2 Результаты

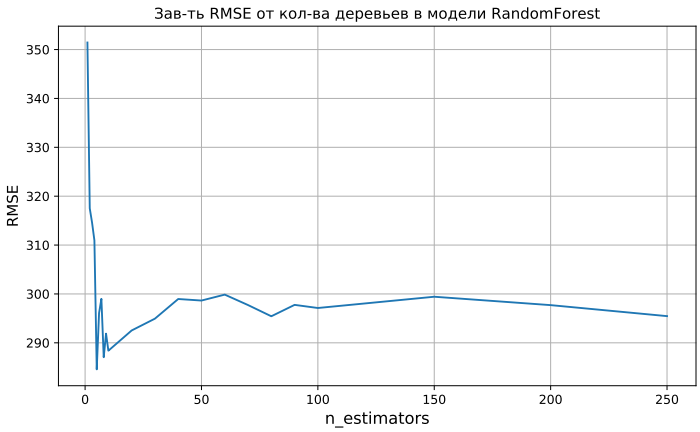
Начнем исследование с подбора параметра feature_subsample_size. Ниже приведены графики качества и времени в зависимости от размера признакового подпространства при количестве деревьев равном 1. Значения перебираются по естественной шкале от 1 до максимального количества признаков (17).

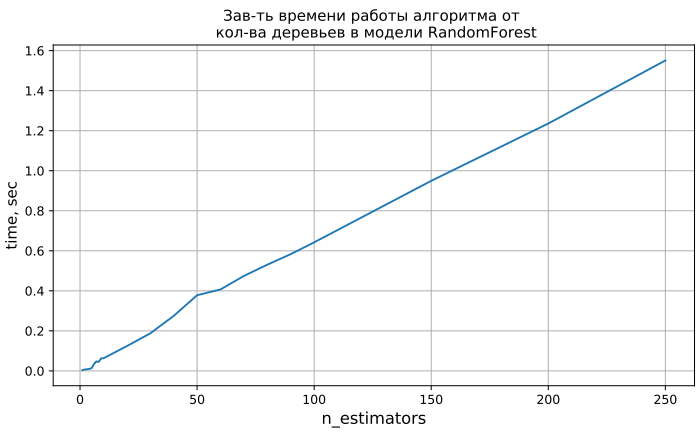




Время настройки алгоритма представляет из себя ломанную, стремящуюся к линейной зависимости от параметра. RMSE же падает до значения 9, после чего начинает колебаться вокруг одного значения качества.

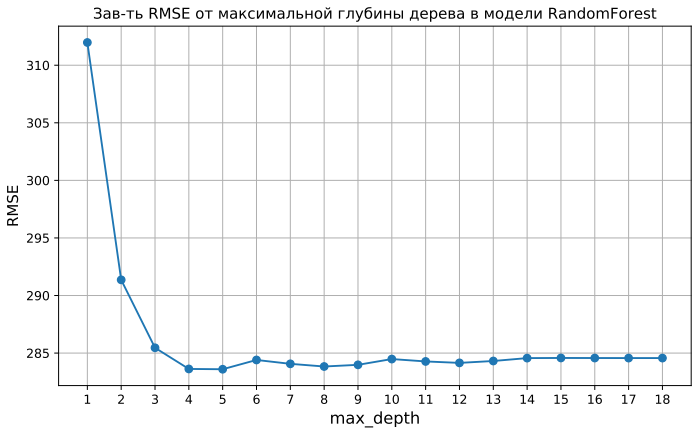
Далее подберем оптимальное количество деревьев из диапазона от 1 до 300 при feature_subsample_size = 1.

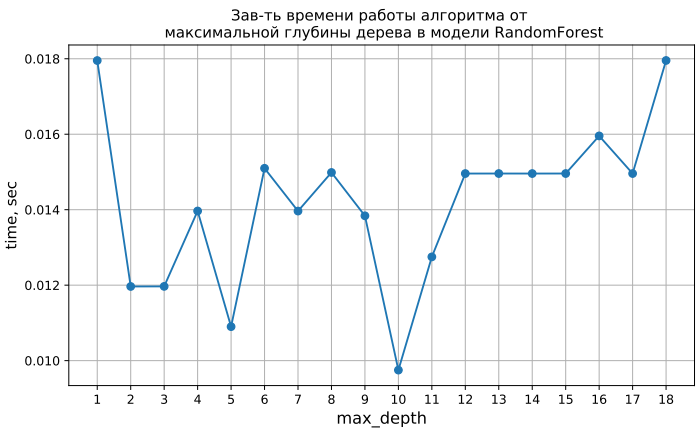




По графикам видно, что время выполнения зависит строго линейно от количества деревьев в модели. Качество становится лучше практически сразу, уже на 5 деревьях достигается минимум, после чего выходит на ассимптоту.

Ниже приведены графики зависимости качества и времени работы алгоритма от максимальной глубины дерева. Значения перебираются от 1 до количества признаков в датасете при 5 деревьях и размерности признакового подпространства 1. Последняя точка графика соответствует неограниченной глубине (max_depth = None).





Ошибка резко падает с увеличением глубины, дойдя до минимума при глубине 4, после чего качество практически не изменяется с ростом глубины. График зависимости времени трудно описать.

2.1.3 Краткие выводы

Первые два эксперимента, касающиеся расчета времени, согласуются с теорией о том, что чем больше количество деревьев или размерность признакового подпространства, тем больше времени требуется для настройки алгоритма, причем время возрастает линейно с ростом значения исследуемого параметра. Что касается глубины дерева, то в среднем можно тоже утверждать, что последние значения параметра ведут к более долгой настройке алгоритма.

Говоря о качестве прогноза: увеличение размера признакового подпространства ведет к улучшению качества до определенного момента; увеличение глубины дерева ведет также к улучшению качества до определенного момента. Так как оптимальная глубина не 1, то на предсказание влияют комбинации признаков, что довольно-таки логично для нашей задачи, причем для выбора этих комбинаций нужно оптимальное кол-во признаков, а значит и размер признакового пространства не должен быть равен 1 или 2.

Оптимальными значениями являются: n_estimators=5, feature_subsample_size=10, max_depth=5.

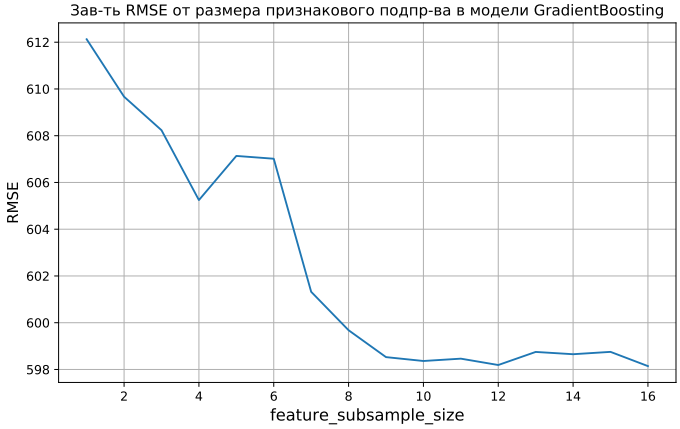
2.2 Gradient Boosting

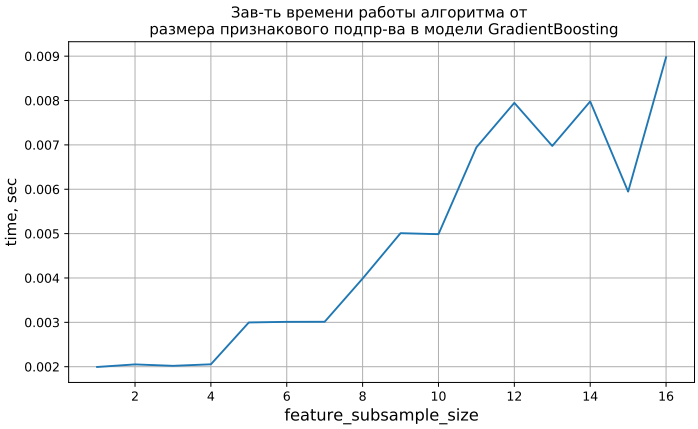
2.2.1 Задача

Исследовать алгоритм GradientBoosting на качество прогноза RMSE и время его работы в зависимости от количества деревьев (n_estimators), размерности признакового подпространства (feature_subsample_size), максимальной глубины дерева (max_depth) и коэффициента learning_rate.

2.2.2 Результаты

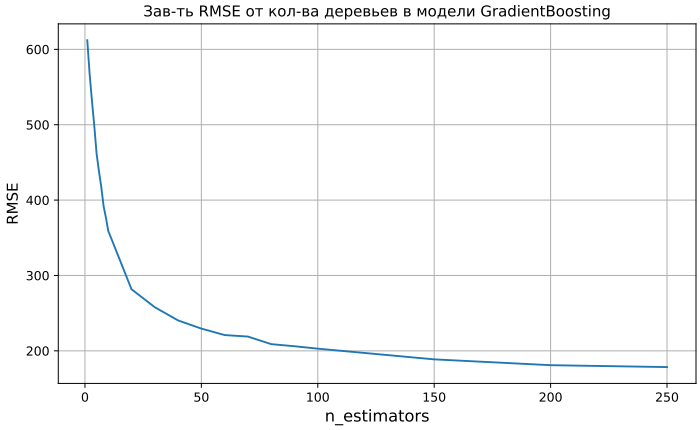
Также начнем исследование с подбора параметра feature_subsample_size. Ниже приведены графики качества и времени в зависимости от размера признакового подпространства при количестве деревьев равном 1. Значения перебираются по естественной шкале от 1 до максимального количества признаков (17).

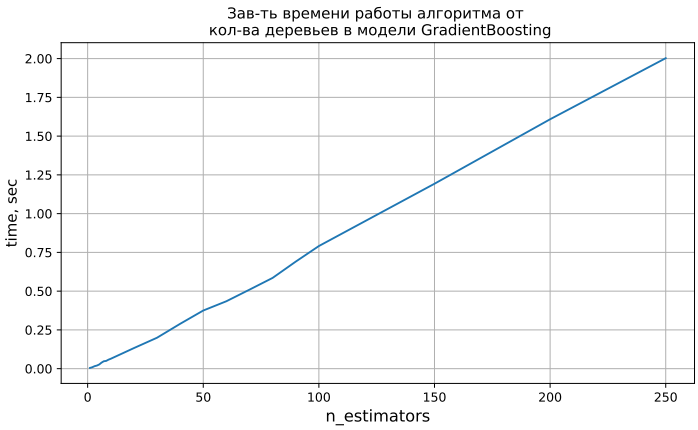




Время увеливается с ростом размера признакового подпространства. Качество тоже становится лучше, выходя на ассимптоту после значения парметра 9.

Найдем минимальное количество деревьев для модели GradientBoosting при минимальном размере признакового подпространства.

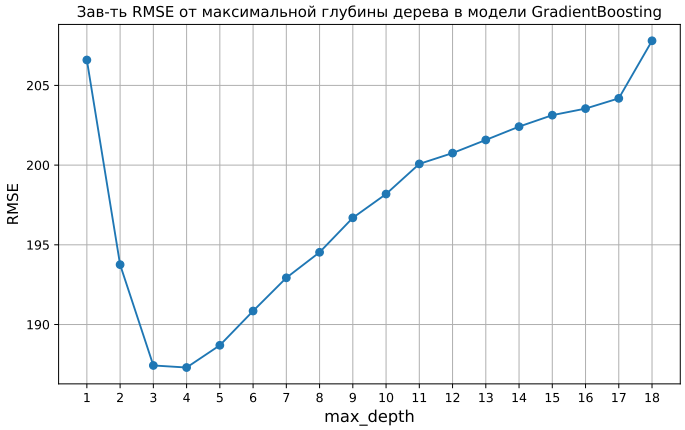


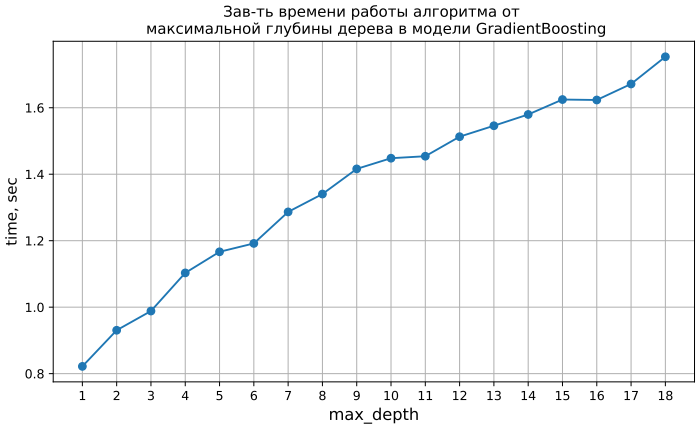


Как и в RF время растет линейно с ростом числа дереьев. Качество же постоянно падает,

выходя на некоторую ассимптоту примерно на 100 деревьях.

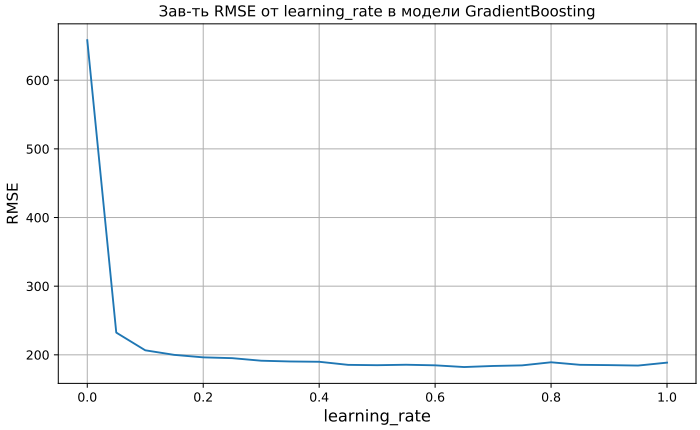
Далее подберем максимальную глубину дерева, перебирая параметр max_depth от 1 до количества признаков при кол-ве деревьев 150 и feature_subsample_size=1. Последняя точка графика соответствует неограниченной глубине (max_depth = None).





Время линейно возрастают с увеличением глубиныю Качество сначала падает до минимума, после чего снова возрастает.

Найдем оптимальное значение параметра learning_rate, перебирая значения от 0 до 1 при кол-ве деревьев 150, feature_subsample_size=1 и max_depth=1.



Качество выходит на ассимптоту с значением параметра около 0.2, а время выполнения растет до значения параметра 0.2, после чего начинает флуктуировать около 0.95 сек.

2.2.3 Краткие выводы

Время, затраченное на обучение, везде возрастает, причем нелинейно только при росте параметра learning_rate и feature_subsample_size.

Для выйгрыша в качестве и времени работы оптимальными параметрами метода GradientBoosting для данной задачи являются: n_estimators=150, feature_subsample_size=10, max_depth=3, learning_rate=0.1.

Параметр максимальный глубины дерева практически одинаковый (оба метода имеют луч-

шие показатели качества при значении параметра 4, но в силу своих особенностей каждый имеет второе лучшее значение качества GB-3, RF-5), что является логичным, так как оба метода используют решающие деревья на одних данных.

Размер признакового подпространства совпадает у методов весьма вероятно по той же причине.

Learning_rate является таким же, как у той же модели из Sklearn по умолчанию.

Методы только сильно различаются в количестве деревьев, скорее всего это связано с использованием встроенной функции поиска минимума в методе GB, которое приводит к замедлению вычислений при малом общем количестве данных. Весьма вероятно на больших выборках метод GB обгонит RF.

3. Выводы

В проделанной работе исследовались модели GradientBoosting и RandomForest, которые являются одними из лучших в машинном обучении. Они показали хорошие результаты с точки зрения времени, но сошлись к разным ассимптотическим значениям качества.

Экспериментально не доказано, что метод GradientBoosting должен сходиться быстрее за счет того, что использует идею градиентного спуска в пространстве функций (базовых моделей). Скорее всего это связано с тем, что выборка мала, и более простой Random Forest сходится на меньших данных быстрее.