Отчет о практическом задании «Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии».

Практикум 317 группы, ММП ВМК МГУ.

Павленко Иван Александрович.

Декабрь 2024.

Содержание

1 Введение

В данном задании проводился анализ алгоритмов, основанных на ансамблях решающих деревьев: случайного леса и градиентного бустинга. На наборе данных, содержащих информацию о квартирах ([1]), изучалось поведение данных алгоритмов на задаче регрессии: время обучения, ошибки, в зависимости от различных гиперпараметров для деревьев и алгоритмов.

1.1 Способ работы методов

Случайный лес - алгоритм, основанный на агрегации предсказаний многих решающих деревьев на случайных выборках, полученных из начальной обучающей выборки с помощью бэггинга. Этот метод позволяет уменьшить разброс предсказаний решающего дерева для различных тестовых элементах, что положительно сказывается на результате ([2]).

Градиентный бустинг - алгоритм, также основанный на обучении многих решающих деревьев, однако, в отличие от случайного леса, он обучает их последовательно на одной и той же выборке, но с разными целевыми переменными: каждое следующее дерево пытается предсказать ошибку предыдущего на элементах обучающей выборки. Далее предсказания всех деревьев суммируются, однако, для более плавного движения предсказаний в сторону целевой переменной, при этом умножаются на некоторый коэфициент, меньший единицы (темп обучения). Такой метод позволяет уменьшить сдвиг предсказания от верного на тестовых элементах ([2]).

1.2 Детали реализации

В экспериментах были использованы классические подходы к реализации описанных методов. Все решающие деревья были реализованы как sklearn.DecisionTreeRegressor.

В экспериментах для обоих методов максимальное количество решающих деревьев равно 100. Этот выбор обоснован, так как эксперименты показали, что ошибка методов перестаёт уменьшаться уже при меньших количествах деревьев. Завершение обучения происходило при отсутствии уменьшения ошибки на трёх итерациях (новых деревьях) подряд.

В число исследуемых гиперпараметров для обоих методов вошли: количество деревьев, максимальная глубина деревьев, максимальная доля признаков, рассматривающихся для разделения выборки на каждой вершине. Для градиентного бустинга также был исследован параметр темпа обучения. Значения остальных гиперпараметров использовались предусмотренные по умолчанию в sklearn.

В качестве ошибки для обучения использовалась RMSLE (поскольку логарифм целевой переменной распределён нормально). Для оценочной метрики использовалась RMSE.

2 Эксперименты

2.1 Предобработка данных

Целевой переменной в данных ([1]) является цена квартиры. Несодержательные признаки, не относящиеся к самому объекту такие как id, date, zipcode, не использовались при обучении. Остальные признаки были поделены на категориальные и численные по такому принципу: признаки, принимающие малое число различных значений на наборе данных (до 100), причислялись к категориальным. Спорным в данном смысле являлся только признак, отвечающий за год реновации квартиры, принимающий около 70-ти различных значений. Он был отнесён к численным по причине значительной корреляции с таргетом (он, очевидно, имеет определённый порядок). Остальные признаки по смыслу явно причислялись к численным либо к категориальным.

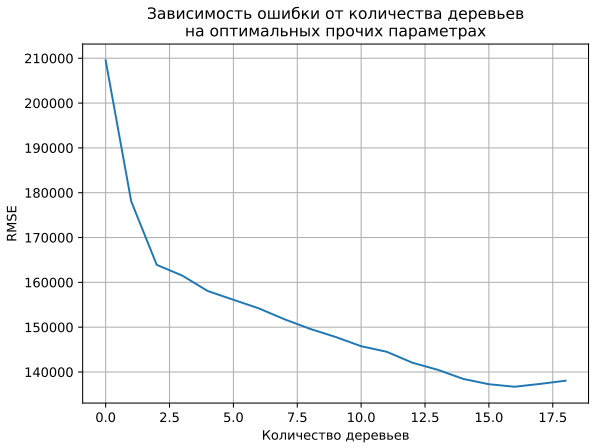
Категориальные признаки были закодированы с помощью OneHotEncoder. К численным признакам не применялось масштабирование, так как оно не является принципиальным для решающих деревьев (они одинаково легко отделяют значения признаков, распределённые с разными дисперсиями и средними.

Выборка была случайно разделена на обучающую и тестовую в отношении 5 : 1. Во всех экспериментах обучение проводится на этой обучающей выборке, а демонстрируемые результаты получены на тестовой.

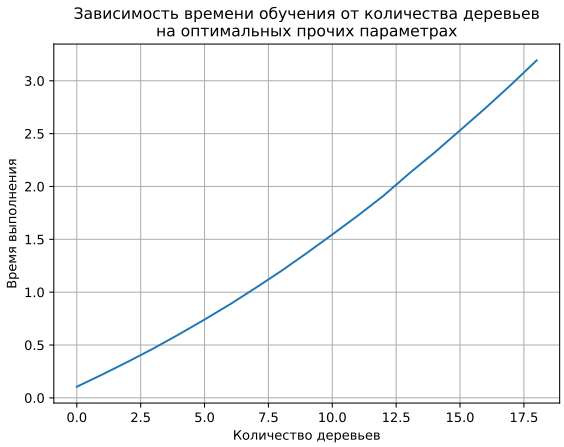
2.2 Изучение случайного леса

Для данного алгоритма перебирались три гиперпараметра, указанных в деталях реализации. Значения времени выполнения и ошибки в зависимости от количества деревьев в ансамбле получались автоматически при обучении алгоритма для каждого набора прочих параметров, поскольку деревья добавляются в алгоритм поочерёдно. На рисунке 1 представлены зависимости метрики и времени обучения от различных значений гиперпараметров (при этом для каждого графика значение всех остальных параметров выбирается оптимальное, то есть то, на котором получен лучших из всех результат).

Для максимальной глубины деревьев и максимального количества деревьев видна явная зависимость: при увеличении их значений до какого-то момента ошибка убывает, а после начинает слабо расти или стабилизируется. Это связано с тем, что при больших значениях максимальной глубины деревьям легче переобучиться, а при больших количествах деревьев новые деревья дают всё меньшее усреднение для ошибки, пока их вклад не становится вовсе незначительным или вредным. Время обучения растёт при увеличении этих параметров, хотя для максимальной глубины деревьев оно имеет сильные колебания, что связано со стохастическим получением выборок, из-за которого обучение останавливается на разных этапах.



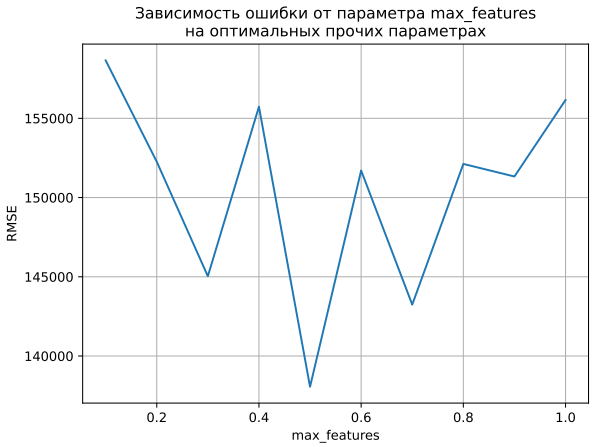
(a) Количество деревьев, RMSE



(b) Количество деревьев, время

(c) Максимальная глубина деревьев, RMSE

(d) Максимальная глубина деревьев, время



(e) Максимальное количество признаков, RMSE

(f) Максимальное количество признаков, время

Рис. 1: Зависимость ошибки и времени обучения случайного леса от значений гиперпараметров

Параметр максимального количества признаков, проверяемых перед делением вершины дерева, не имеет видимой зависимости от ошибки и времени. Это может быть связано с тем, что при увеличении данного параметра деревья обучаются дольше на каждой вершине, однако общее число вершин уменьшается из-за более обоснованных сплитов.

Лучшим набором гиперпараметров стал: $max\_depth = 17$, $max\_features = 0.5$, $n\_estimators = 16$. На них был достигнут результат RMSE = 138058. Учитывая среднее значение целевой переменной, равное 540088, и стандартное отклонение, равное 367127 такой результат можно считать приемлемым. Таким образом, алгоритм выбрал использовать небольшое количество деревьев большой глубины.

Также отдельно был проведён подбор оптимальных параметров, при неограниченной глубине деревьев. Оптимальными значениями других параметров стали: $max\_features = 0.6$, $n\_estimators = 25$, значение метрики: RMSE = 137973, что почти совпадает со значением на предыдущем оптимальном наборе. При этом данных алгоритм обучился медленнее предыдущего (5 и 3 секунды).

2.3 Градиентный бустинг

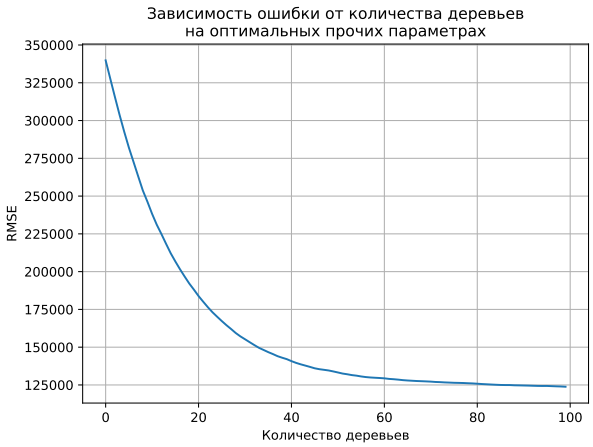
Проведённые с градиентным бустингом эксперименты аналогичны предыдущим, однако был добавлен новый гиперпараметр - темп обучения. Результаты экспериментов представлены на рисунках 2 и 3. В целом повторяются тренды зависимостей из случайного леса, однако они более сглажены из-за отсутствия стохастики: градиентный бустинг обучает все деревья на одной выборке. Сильнее видно, что алгоритм переобучается при больших значениях максимальной глубины деревьев (2c). Теперь для всех параметров, кроме темпа обучения, время обучения явно возрастает. Это объяснимо тем, что все эти параметры усложняют структуру деревьев.

Лучшим темпом обучения стал минимальный из рассматриваемых, 0.05, на нём же достигается самое долгое время обучения. Оно убывает при увеличении данного гиперпараметра, поскольку шаги градиентного бустинга становятся больше, а значит, что требуется меньшее число деревьев.

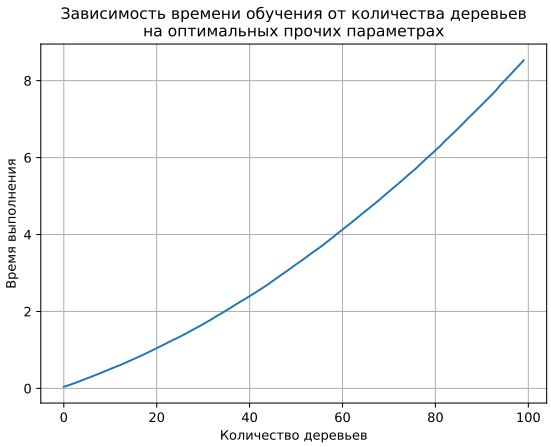
Лучший результат алгоритма на тестовой выборке был достигнут при следующих значениях параметров: $max\_depth = 9$, $max\_features = 0.3$, $n\_estimators = 100$, $learning\_rate = 0.05$. На таких параметрах был достигнут результат RMSE = 123816, что значительно лучше результата случайного леса. Видно, что для градиентного бустинга требуется для лучших результатов намного большее число деревьев, однако меньшей глубины.

Изучения поведения бустинга при неограниченной глубине деревьев проводилось, однако лучшим результатом стал RMSE = 137048, что сильно хуже полученного выше, и сравнимо с результатом случайного леса. Таким образом, для оптимальной работы градиентного бустинга необходимо ограничение глубины деревьев (в том числе для избежания переобучения).

Время обучения градиентного бустинга значительно (в 2-3 раза) превышало время обучения случайного леса.



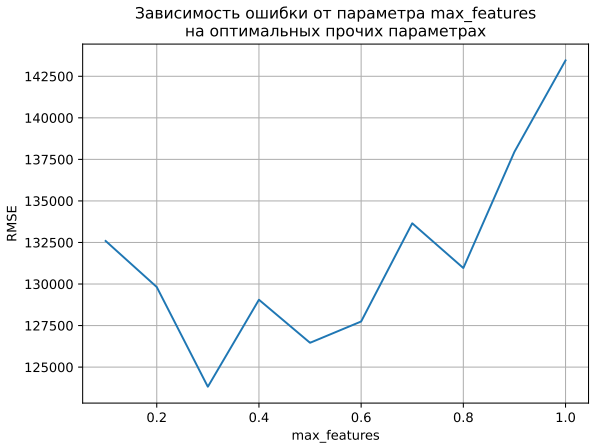
(a) Количество деревьев, RMSE



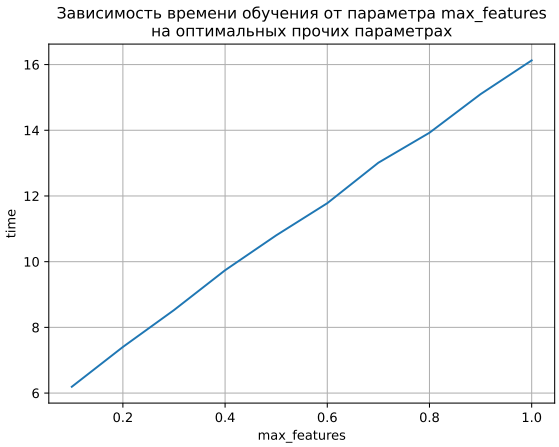
(b) Количество деревьев, время

(c) Максимальная глубина деревьев, RMSE

(d) Максимальная глубина деревьев, время

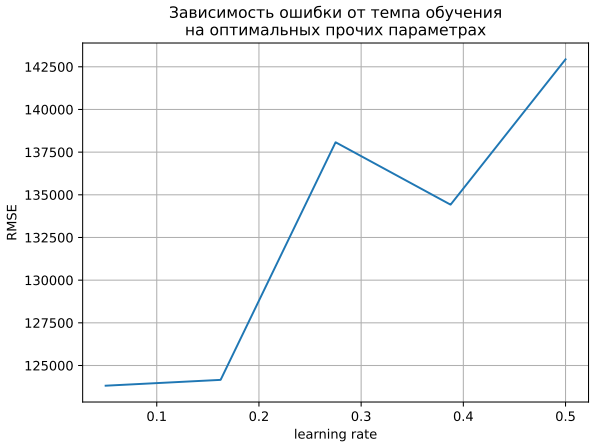


(e) Максимальное количество признаков, RMSE

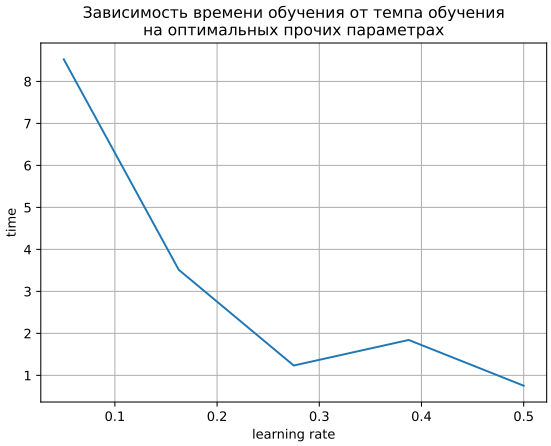


(f) Максимальное количество признаков, время

Рис. 2: Зависимость ошибки и времени обучения градиентного бустинга от значений гиперпараметров



(a) Темп обучения, RMSE



(b) Темп обучения, время

Рис. 3: Зависимость ошибки и времени обучения градиентного бустинга от значений гиперпараметров

3 Выводы

Сравнение и анализ алгоритмов градиентного бустинга и случайного леса показали, что ансамбли алгоритмов и правда помогают справляться с уменьшением ошибки (сдвига или смещения). При этом градиентный бустинг показал результаты лучше случайного леса. Было продемонстрировано, как изменение структуры и способов построение деревьев в данных алгоритмах влияет на время их обучения и результаты. Было показано, что градиентный бустинг и случайный лес выбирают разные стратегии построения деревьев (для случайного леса оптимально меньшее число глубоких деревьев, в то время как для градиентного бустинга - большое число деревьев небольшой глубины).

Список литературы

[1] *House Sales in King County, USA dataset.* https://www.kaggle.com/datasets/harlfoxem/housesalesprediction. 2015.

[2] Лекция ММРО. *Ансамбли. Градиентный бустинг.* https://github.com/mmp-mmro-team/mmp$_m mro_f all_2$024/*blob*/*main*/*seminars*/$Seminar_1 1_i ntro_t o_e nsembles/le$ 2024.