Отчет по заданию №3: Композиции алгоритмов для решения задачи регрессии

Васильев Руслан ВМК МГУ, 317 группа

25 декабря 2020 г.

Содержание

1 Введение

В заключительном практическом задании предлагается реализовать композиции алгоритмов машинного обучения и провести эксперименты, а также спроектировать веб-сервис для взаимодействия с моделью. Весь проект доступен в репозитории[1]. Данный отчет иллюстрирует результаты экспериментов с моделями на датасете данных о продажи недвижимости.

2 Постановка задачи

Итак, рассматривается задача регрессии с метрикой качества RMSE:

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

где N — размер выборки, y_i — истинное значение целевой переменной на i-м объекте, $\hat{y}_i$ — предсказанное.

Для решения реализованы две модели, представляющие собой ансамбли решающих деревьев: случайный лес и градиентный бустинг. Исследование алгоритмов включает в себя измерение функции ошибки и времени работы при варьировании гиперпараметров (порядок экспериментов соответствует стандартной настройке данных моделей).

3 Эксперименты

3.1 Предобработка данных

Исходные данные о недвижимости были разделены на обучение (80%) и контроль (20%, она же валидационная выборка). И здесь сразу учитывается особенность задачи. Хотя в задании отсутствует описание признаков и целевой переменной, можно с уверенностью предположить, что столбец `date` связан со временем поступления данных (даты имеют небольшой диапазон 2014–2015, монотонно возрастают, дублируются, следуют сразу за `ID`, а столбцы `build_year` и `renovation_year` с ними не связаны). По этой причине было бы некорректно перемешать выборку перед разделением на обучение и контроль — из-за утечки такая стратегия может дать ложную оценку качества моделей и привести к неправильным выводам. В качестве валидационной выборки берутся последние 20% данных, соответствующие хронологическому порядку по столбцу `date`.

3.2 Случайный лес

3.2.1 Количество деревьев

Количество деревьев в случайном лесе регулирует число алгоритмов, по которому проводится ансамблирование (усреднение). На рис. 1 можно видеть, что с ростом

[1] https://github.com/artnitolog/mmf_prac_2020_task_3

числа деревьев ошибка практически монотонно убывает на обучении выборке. Тем не менее на контроле по достижении оптимального числа базовых алгоритмов функционал затем немного увеличивается (переобучение), а затем выходит на асимптоту. Для нашей задачи нам оказалось достаточно взять 250 деревьев.

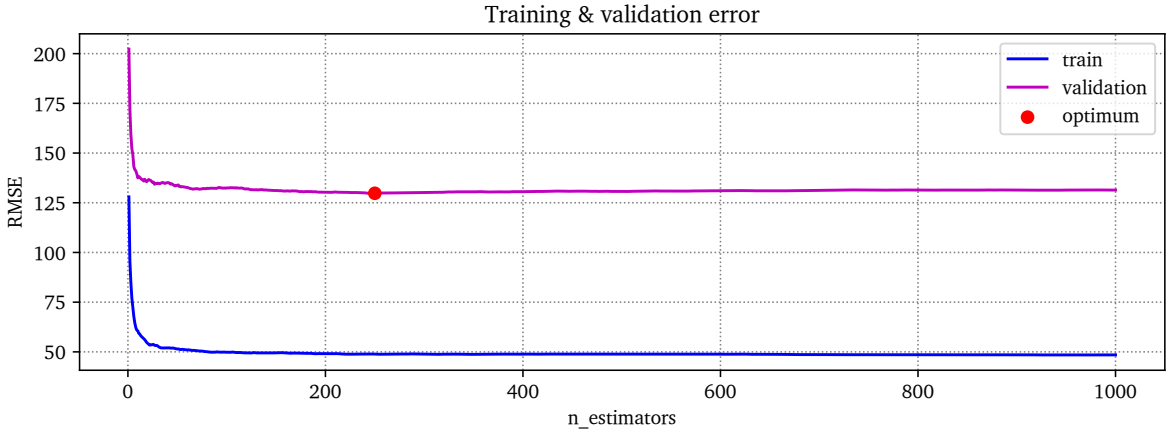

Рис. 1: Зависимость RMSE от количества деревьев в случайном лесе

Что касается времени работы, то понятно, что оно должно линейно зависеть от числа деревьев в лесе. Для более честной оценки обучим с нуля несколько моделей с разным количеством деревьев, результаты приведены на рис. 2. И действительно, время обучения растет линейно.

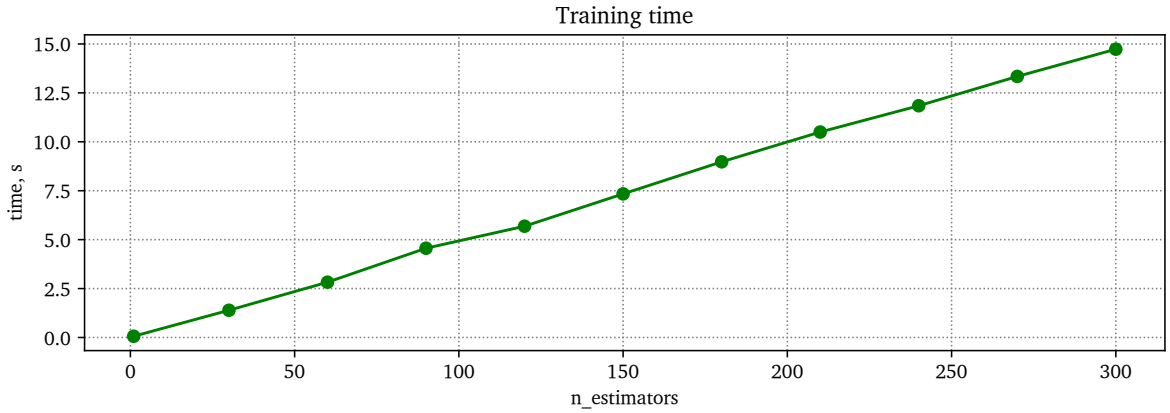

Рис. 2: Зависимость времени обучения случайного леса от числа деревьев

3.2.2 Размерность подвыборки признаков для дерева

3.2.3 Глубина дерева

3.3 Градиентный бустинг

3.3.1 Количество деревьев и темп обучения

В отличие от случайного леса, в градиентном бустинге базовые алгоритмы не являются независимыми — каждый следующий исправляет ошибки предыдущих.

Поэтому при настройке гиперпараметров количество деревьев не подбирается отдельно, а рассматривается в паре с темпом обучения. Рассмотрим зависимость RMSE на обучающей и контрольной выборках (рис. 3).

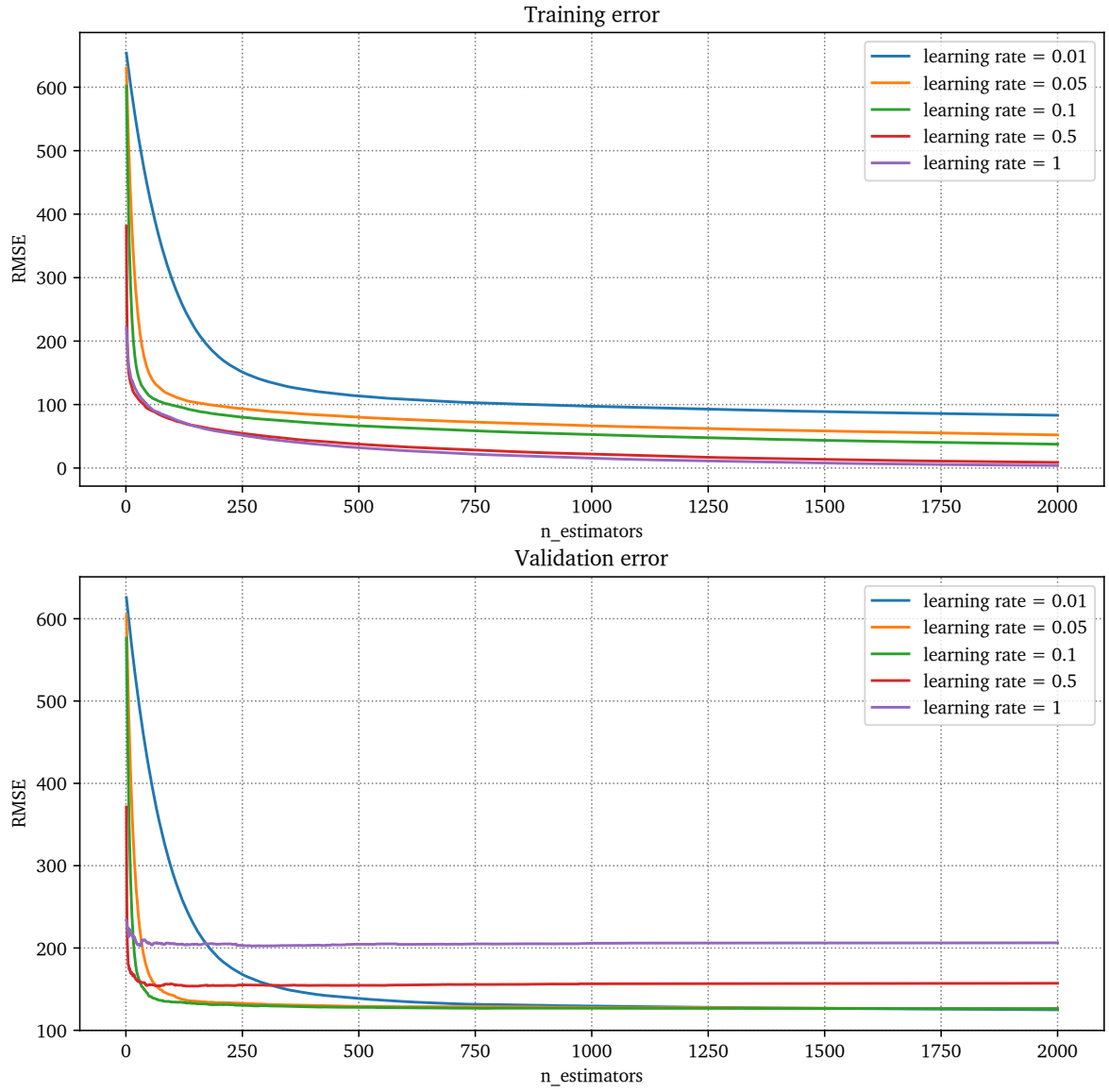

Рис. 3: Зависимость RMSE от числа деревьев и темпа обучения в градиентном бустинге

На обучении (рис. 3) ошибка с ростом числа деревьев монотонно стремится к нулю. Но на валидации и тесте в градиентном бустинге обычно монотонной зависимости нет. При высоком темпе обучения, как видим, качество действительно начинает ухудшаться с некоторого момента. Но при «умеренном» значении `learning_rate` ошибка, пусть крайне медленно, но продолжает уменьшаться даже после тысячи деревьев. Такое поведение связано с использованием классической реализации бустинга и особенностями задачи.

Если на обучающей выборке ошибка уменьшается с ростом темпа обучения, то на валидации зависимость обратная (на большом числе деревьев). Но для дальней-

шего проведения экспериментов особого смысла в 2000 деревьев и темпе 0.01 нет — разница в качестве незначительная. Поэтому дальше рассмотрим 500 деревьев с `learning_rate` = 0.1. Время работы в нашей реализации по-прежнему линейно растет с числом базовых алгоритмов, не не зависит от темпа обучения[2]. Подтвердим это экспериментом с обучением разных моделей с нуля, но на уменьшенном числе деревьев.

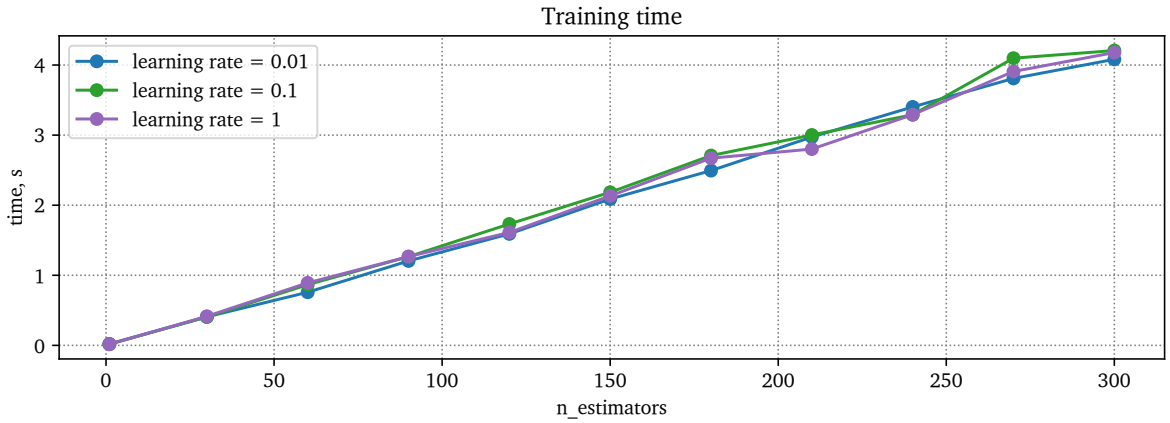

Рис. 4: Зависимость времени обучения градиентного бустинга от числа деревьев и темпа

Итак, рис. 4 показывает, что в нашей реализации бустинга время обучения растет линейно с числом деревьев и не зависит от темпа обучения. Но если сравнить со случайным лесом (рис. 2), то при бустинге обучение происходит быстрее. Но почему? Причина заключается в разных подходах к настройке глубины для бустинга и леса.

3.3.2 Глубина дерева

3.3.3 Размерность подвыборки признаков для дерева

4 Заключение

[2]Хотя время могло бы уменьшаться с ростом темпа обучения, если добавить критерий останова при отсутствии улучшения качества на валидации.