Отчет по заданию №3: «Композиции алгоритмов для решения задач регрессии.»

Практикум на ЭВМ 2020/2021

Елистратов Семен, 317 гр.
ММП ВМК МГУ

24 декабря 2020

1. Постановка задачи.

В рамках практического задания необходимо самостоятельно реализовать модели случайных лесов и градиентного бустинга и проанализировать зависимость функции потерь от гиперпараметров. В качестве датасета для экспериментов, будем использовать данные о продажах недвижимости. Метрику для анализа качества возьмем RMSE (корень из среднеквадратичного отклонения).

2. Эксперименты.

Прежде чем проводить эксперименты, преобразуем единственный не количественный признак, дату, в удобное представление: разобьем на год, месяц и день. Так как мы используем алгоритмы композиции над решающими деревьями, нет нужды масштабировать признаки.

2.1 Случайный лес

Сначала исследуем поведение функции потерь RMSE в зависимости от количества деревьев в модели.

Рис. 1: График зависимости RMSE от числа деревьев для Random Forest.

Таким образом, видно, что с увеличением количества деревьев, график выходит на горизонтальную ассимптоту, хотя и имеет видимые флуктуации. Оптимальное значение 250 деревьев. Теперь рассмотрим зависимость от максимальной глубины:

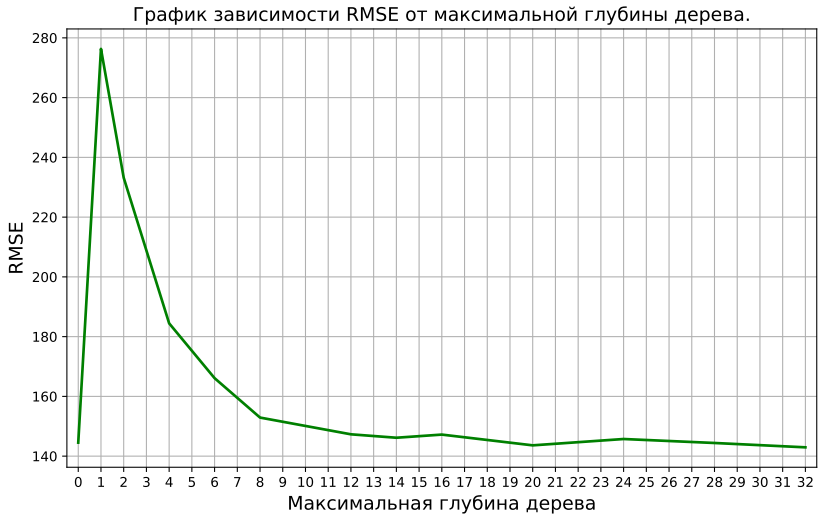

Рис. 2: График зависимости RMSE от глубины для Random Forest. (Значение глубины равное 0 соответствует неограниченности деревьев.)

По графику видно, что лучшее качество при глубине 20 и при неограниченном случае. Из-за того что оптимальная глубина получилась достаточно большой, можно сделать вывод, что для данного датасета лучше использовать композицию сложных деревьев.

Проанализируем влияние коэффициента размерности подвыборки признаков для одного дерева:

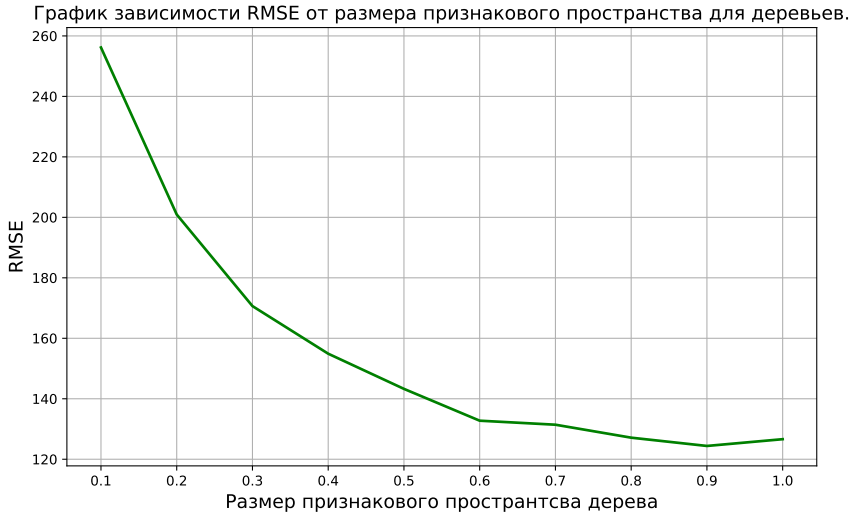

Рис. 3: График зависимости RMSE от коэффициента размерности подвыборки признаков для одного дерева.

Лучший результат при коэффициента равном 0.9, то есть в данных есть важные признаки, удаление которых увеличивает функцию потерь.

2.1 Градиентный бустинг

Исследуем поведение функции потерь RMSE в зависимости от числа деревьев.

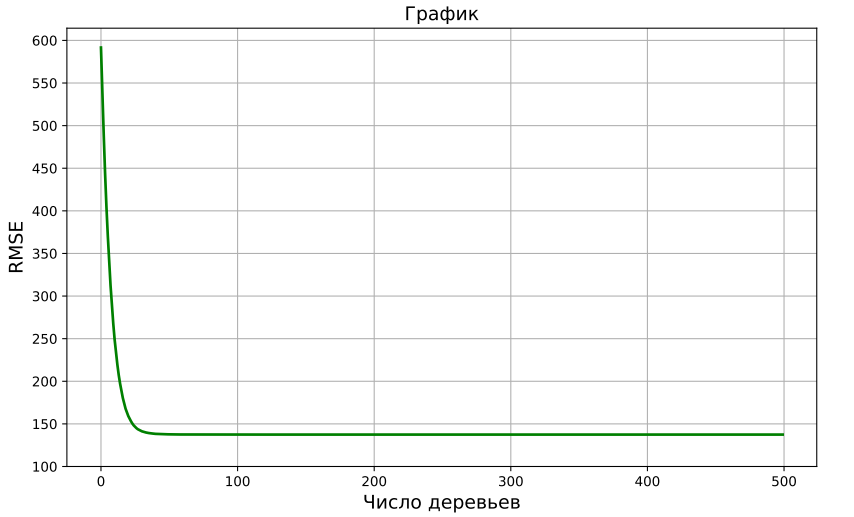

Рис. 4: График зависимости RMSE от числа деревьев для градиентного бустинга.

По сравнению с предыдущим алгоритмом, здесь график более гладкий и не имеет выраженных флуктуаций, так как каждый последующий алгоритм исправля-

ет ошибки предыдущего. График достаточно быстро выходит на ассимптоту. Оптимальное значение: 100.

Теперь рассмотрим зависимость от максимальной глубины дерева:

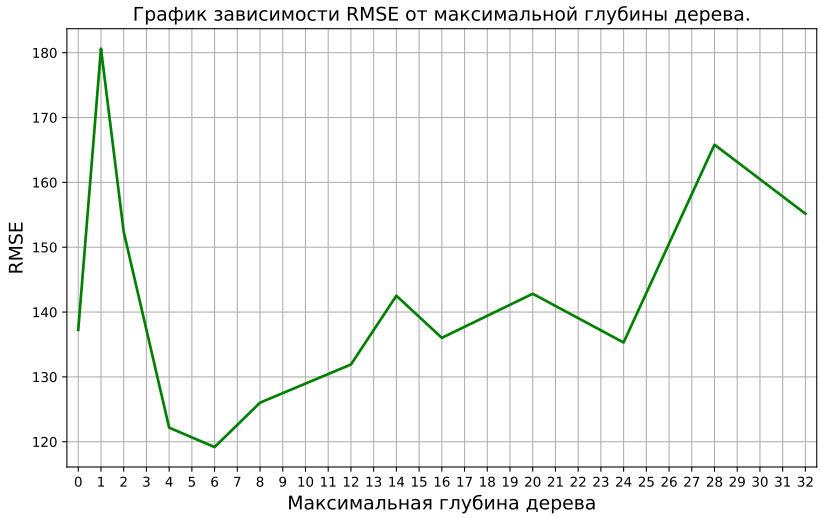

Рис. 5: График зависимости RMSE от глубины для градиентного бустинга. (Значение глубины равное 0 соответствует неограниченности деревьев.)

В отличии от случайного леса, в данном эксперименте лучшее значение глубины: 6. Так как каждый последующий алгоритм уменьшает ошибки предыдущего, то невыгодно, чтобы первоначальный алгоритм сразу переобучился, поэтому базовая модель выбирается более простой.

Аналогично, для следующего эксперимента:

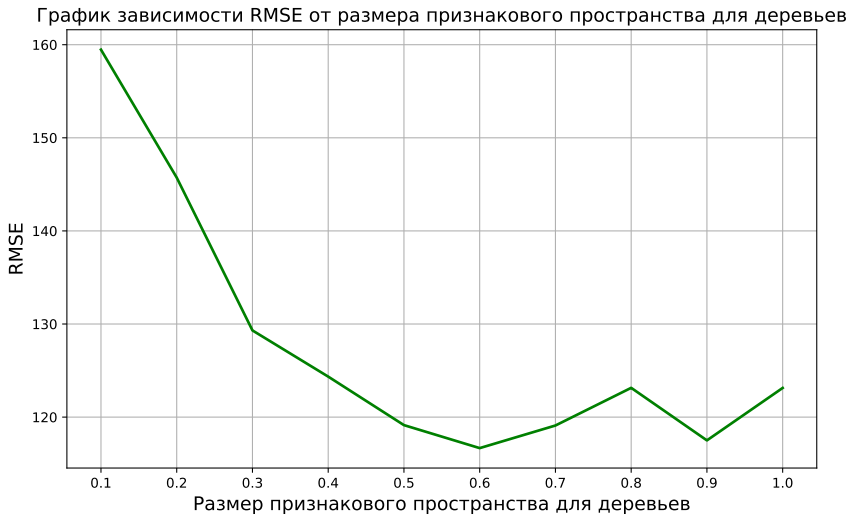

Рис. 6: График зависимости RMSE от коэффициента размерности подвыборки признаков для одного дерева (алгоритм градиентного бустинга).

Оптимальное значение: 0.6, что меньше, чем для случайного леса (0.9), по причине того, что требуется более простая модель.

И последний эксперимент: зависимость от $learning\_rate$.

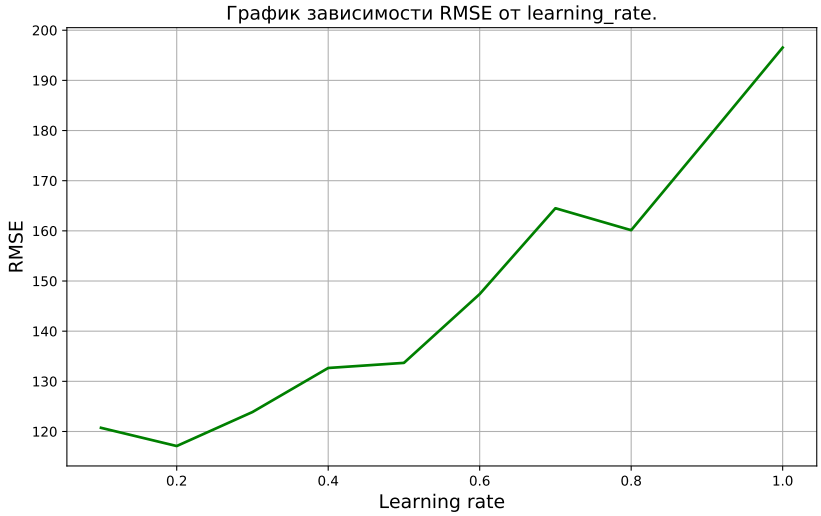

Рис. 7: График зависимости RMSE от $learning\_rate$ для градиентного бустинга.

Лучшее значение: 0.5. Подбор правильного параметра $learning\_rate$ - ключевой процесс настройки модели, он контролирует с какими весами новые деревья будут добавляться в модель.

3. Веб-сервер

Кроме выполнения экспериментов, также был реализован веб-сервер с помощью фреймворка Flask, позволяющий обучить модель на пользовательском датасете и получить предсказания. Сервер поддерживает настройку гиперпараметров, вывод значений по умолчанию, вывод информации о модели и датасете, на котором она обучалась, запуск на валидационных данных.

Для унификации были сформулированы следующие требования к датасетам: обучающий и валидационный датасет загружается одним файлом формата $.csv$, один из стобцов которого (под заголовком $target$) является зависимой переменной. Модель учитывает только количественные признаки, остальные игнорирует. Столбец с заголовком $target$ в датасете для предсказания игнорируется. Для обученной модели можно делать неограниченное число предсказаний и валидации.

3. Выводы

Мы рассмотрели два важных алгоритма композиции над решающими деревьями, представляющими два возможных подхода ансамблирования: параллельный

(случайный лес) и последовательный (градиентный бустинг). Каждый из них обладает своими достоинствами и недостатками. Например, градиентный бустинг имеет хорошую обобщающую способность, но на зашумленных данных его бывает гораздо труднее настроить, в сравнении с случайным лесом.