Лабораторная работа по методам ансамблирования (Random-Forest и Gradient-tree-boosting)

Федоров Артем Максимович

December 2023

1 Введение

Решение прикладных задач любой области человеческих знаний всегда начинается с грамотной и подробной постановки. Отсутствие явного и четкого соглашения о желаемом результате, исходных данных и, возможно, методов решения с конечными условиями применимости, приводит к неоформленности задания и отсутствию возможности и смысла его решать. Задачи машинного обучения не являются исключением из правил, напротив, именно в прикладных областях данной дисциплины крайне четко прослеживается зависимость всего подхода решения задачи от ее природы, от особенных потребностей конечного потребителя, специфики выбора критерия качества, от формы исходных данных.

Последний аспект играет наиважнейшую роль в успехе решения. Если пожелания заказчика, выбор метрик качества алгоритма, класс ответов, им даваемых, — влияют на наш вектор поиска наилучшей стратегии (будь–то наши эмпирические, эвристические представления о целесообразных подходах к конкретной задаче, или же сухой подход с точки теории), но именно исходные данные, предоставленные нам в распоряжение, определяют жизнеспособность и применимость каждой из стратегий. К сожалению, на практике не всегда попадаются такие данные, на которых даже самые простые модели способны показать хорошие результаты; их форма, наполнение сильно варьируются от проблемы к проблеме, от предметной области к предметной области.

Один из подходов в нахождении "сильных" моделей, способных качественно решать даже сложные задачи, является построение ансамблей — объединения обученных алгоритмов, работающих вместе лучше, чем по отдельности. Данная работа ставит своей целью рассмотреть основные приемы построения ансамблей: Random-Forest и градиентный бустинг на деревьях, — оценить их особенности, сравнить между собой.

2 Ошибки алгоритмов

Пространство признаков объектов X и множество скрытых параметров объектов считаются известными (при чем в общем случае ничего не утверждается о счетности K; необходимо лишь условие его ограниченности). В таком случае объекты, рассматриваемые в контексте последующей работы, будут отождествляться с объектом $(x, k) \in X \times K$. Обозначим $q(x) : X \to K$, как общий вид алгоритма, восстанавливающий по наблюдениям $x \in X$ скрытый параметр соответствующего объекта $k \in K$. Тогда, доопределив функцию потерь, ошибка алгоритма на объекте отождествляется с функционалом $\mathcal{L}$, а функция эмпирического риска с $Q(X^l, q, k^l) = \sum_{i=1}^{l} \mathcal{L}(q(x_i), k_i)$

В таких обозначениях конкретная ошибка err модели q на объекте (x, k) есть ничто иное, как $q(x) = k^\star, k^\star \neq k \Rightarrow err = \mathcal{L}(x, k, q)$. Однако одна ошибка алгоритма не способна в полной мере описать его классификационную или регрессирующую способность. Нужен способ оценить его ошибку в среднем. Сузим класс проблем, рассматриваемых в работе, ограничившись задачами регрессии со всюду плотным множеством K и квадратичной функцией потерь, равной $\mathcal{L}(x, k, q) = \frac{1}{2}(q(x) - k)^2$. Эмпирически, ошибка моделей в данном случае разделима на три составляющих: ошибки, исходящей из неидеальности считывании реального параметра k, ошибки от константного смещения модели в сторону и случайного разброса модели, получаемого из зависимости ответа модели от обучающей выборки. Введение вероятностной постановки задачи, а именно вероятностного поля в пространство $X \times K$, позволяет строго подтвердить оценками данное заключение.

Пусть $\exists p_{X,K}(x, k)$ — совместное распределение вероятности над объектами из X и K. Будем говорить, что $\hat{X}$ есть некоторая обучающая подвыборка фиксированного размера, а $q(\hat{X})$ есть алгоритм, обученный на ней. Тогда математическое ожидание квадратичной функции потерь модели $q(x)$ можно представить как:

$$\mathbb{E}_{x,k}\big[\mathbb{E}_{\hat{X}}\big[(k-q(\hat{X})(x))^2\big]\big] = \underbrace{\mathbb{E}_{x,k}\big[(k-\mathbb{E}[k|x])^2\big]}_{\text{шум}} + \underbrace{\mathbb{E}_{x}\big[(\mathbb{E}_{\hat{X}}[q(\hat{X})(x)]-\mathbb{E}[y|x])^2\big]}_{\text{смещение}} + \underbrace{\mathbb{E}_{x}\big[\mathbb{E}_{\hat{X}}\big[(q(\hat{X})(x)-\mathbb{E}_{\hat{X}}[q(\hat{X})(x)])^2\big]\big]}_{\text{разброс}}$$

Средняя ошибка алгоритма распадается на три параметра, описывающих регрессионную способность модели:

- Параметр шума (**noise**) описывает параметры самих данных, их зашумленность. Является характеристикой исходных данных и избавиться от нее не представляется возможным на уровне модели.
- Параметр смещения (**bias**) наоборот является прямой характеристикой модели, показывая, как в среднем будет смещен ее ответ относительно действительных скрытых параметров объектов при обучении ее на случайно подобранной выборке. Смещение показывает способность семейства моделей подстраиваться под целевую переменную и именно он отвечает за то, подходит ли данная модель для конкретной задачи или же нет.
- Параметр разброса (**variance**) так же описывает характеристику модели, в свою очередь определяя степень зависимости итогового ответа от обучающей выборки. Разброс показывает, насколько семейство моделей устойчиво к обучающим данным.

В реальных задачах, в особенности на больших данных, сложно восстановить распределения напрямую, к тому же это бы означало решение всей задачи в целом. Потому для полученных оценок можно рассматривать лишь приближенные оценки, позволяющие судить о применимости архитектуры решения к конкретной задаче.

> Становится очевидным, почему некоторые модели имеют стабильно плохие результаты на некоторых наборах данных. Первая и самая простая причина этому закладывается в показателе шума выборки, что мешает адекватно строить и оценивать конечную модель. В то же время, если модель слаба для конкретных данных, ее обучаемая способность мала (о чем свидетельствует большой bias). Чувствительность модели к малейшим изменениям обучающей выборки (на что указывает большой показатель variance), показывает неустойчивость конечной модели к внесениям изменениям в обучающую выборку. Оба данных факта крайне плохо сказываются на конечной применимости модели.

3 Улучшение базовых моделей

Статистическая оценка ошибки позволяет определить неявную задачу оптимизации для моделей: требуется свести к минимуму суммарную ошибку **bias** и **variance** на валидационной выборке. Задача представляется естественной, однако пути достижения ее решения могут быть неочевидны: применение сложных моделей может усложнить процесс создания и калибровки решения. На практике широко применим обратный подход, использующий не одну сложную модель, но набор (ансамбль) элементарных, базовых. Будем считать, что $\mathcal{B}$ есть класс базовых алгоритмов, оптимизирующих поставленную ранее регрессионную задачу. В таком случае рассматривается стратегия $q(x)$ в виде: $q(x) = \sum_{t=1}^{N} \alpha_t b_t$, $b_t \in \mathcal{B}$, $\alpha_t \in \mathbb{R}$, что есть взвешенная сумма простых моделей.

3.1 Random-Forest

Рассмотрим частный случай взвешенных сумм, где каждый из базовых алгоритмов имеет единый вес в принятии итогового решения: $\frac{1}{N}\sum_{t=1}^{N} b_t$. В таком случае смещение всей стратегии выражаемо в виде:

$$q(x) = \mathbb{E}_x\big[(\mathbb{E}_{\hat{X}}[q(\hat{X})(x)]-\mathbb{E}[y|x])^2\big] = \mathbb{E}_x\big[(\frac{1}{N}\sum_{t=1}^{N}\mathbb{E}_{\hat{X}}[b_t(\hat{X})(x)]-\mathbb{E}[y|x])^2\big] = \mathbb{E}_x\big[(\mathbb{E}_{\hat{X}}[b_t(\hat{X})(x)]-\mathbb{E}[y|x])^2\big]$$

легко заметить, что это есть формула **bias** (смещение) для одного базового алгоритма. Таким образом порождается первое требование к классу $\mathcal{B}$: модели из класса должны обладать малой смещенностью. Однако подобные преобразования, как для **bias**, можно применить и к **variance** (разбросу):

$$\mathbb{E}_x\big[\mathbb{E}_{\hat{X}}\big[(q(\hat{X})(x)-\mathbb{E}_{\hat{X}}[q(\hat{X})(x)])^2\big]\big] = \mathbb{E}_x\big[\mathbb{E}_{\hat{X}}\big[\frac{1}{N}\sum_{t=1}^{N}(b_t(\hat{X})(x)-\frac{1}{N}\sum_{t=1}^{N}\mathbb{E}_{\hat{X}}[b_t(\hat{X})(x)])^2\big]\big] =$$

$$= \frac{1}{N^2}(\sum_{t=1}^{N} variance(b_t) + \sum_{i\neq j} cov(b_i,b_j)) = \frac{1}{N}variance(b_t) + \frac{1}{N^2}\sum_{i\neq j} cov(b_i,b_j)$$

получаем формулу, крайне походящую на таковую для дисперсии среднего арифметического набора случайных одинаково распределенных величин. **Variance** конечного алгоритма так же выражается через разброс единичной базовой модели, при чем данный член входит с коэффициентом $\frac{1}{N}$, что говорит о его монотонном уменьшении при росте количества моделей. Однако в данную формулу так же входит и второй член, а именно $\frac{1}{N^2}\sum_{i\neq j} cov(b_i, b_j)$, при чем заметим, что заменить сумму ковариаций на ковариацию по $i - j$ алгоритмам в общем случае нельзя, так как обучающие выборки алгоритмов могут попарно различаться. Данная сумма в общем случае может быть не ограничена, а потому для получения наилучшего прироста качества в среднем при применении алгоритма, коррелированность ответов базовых моделей требуется свести к нулю.

3.1.1 Выбор моделей

Полученные выводы суммируются в два главных аспекта выбора базовых моделей:

- Модели должны обладать малой смещенностью — для задачи регрессии с квадратичной функцией это выливается в способность базовых моделей приближать функцию скрытого параметра достаточно точно и почти всюду на X
- Ответы моделей должны быть некоррелированны — может быть достигнуто специальными стратегиями обучения моделей, например бутстрапом или же обучением моделей на подвыборках признаков объектов X

> Стоит отметить, что во втором пункте стоит отдать предпочтение именно обучению моделей на подвыборках из **признаков** объектов, так как при применении бутстрапа и дроблении генеральной совокупности (обучающей выборки), в процессе обучения некоторых базовых моделей могут быть упущены **важные** объекты распределения p_{XK}, из-за чего $\mathbf{bias(q)} \neq \mathbf{bias(b}_t\mathbf{)}$

Данным условиям отвечает стратегии построения решающих деревьев. Второй пункт выполняется из реализации, остается лишь уточнить, что оптимальным для задач регрессии считается именно подвыборка размера $\frac{1}{3}$ от числа всех признаков объекта. Для доказательства выполнения первого условия определим вид итогового решения, строящегося деревьями. Решающее дерево, обучаясь на данных, разбивает пространство на области с помощью гиперплоскостей. Стоит отметить, что получаемые области пространства являются измеримыми, и на каждом таком множестве модель принимает константное значение. Таким образом решающее дерево в нашей задаче строит простую и измеримую функцию. Так же будем считать, что функция скрытого параметра, что мы приближаем алгоритмом, принадлежит пространству функций L^2 на некоторой ограниченной области из X, что гарантированно покрывает все значения признаков объектов x (прочие объекты считаем выбросами или редкими примерами, не входящими в генеральную совокупность, это будет оговорено далее). Тогда существует такое разбиение пространства X на измеримые множества, что существует такая простая функция, определенная на данном разбиении, что приближает функцию скрытого параметра по норме на заранее известное значение. Ограничением в таком приближении является лишь не бесконечность обучающих данных, из-за чего приходится лишь говорить о стремлении.

3.1.2 Принцип построения Random-Forest

Настройки модели **Random-Forest** будут определяться набором параметров: числом деревьев N, максимальной глубиной каждого из деревьев $depth$, что в общем случае может не быть заданной, числом признаков $fcount$, выбираемых случайным образом каждым из деревьев для обучения, стратегией ветвления (случайным образом или наилучшим для функции потерь), а так же коэффициентом разбиения обучающей выборки для каждой из моделей (коэффициентом бутстрапа $bscoef$). Алгоритм реализации приведен на схеме 1. Операция **subset** позволяет взять случайное подмножество мощности bscoef от мощности первоначальных.

3.2 Градиентный бустинг

Рассмотрим задачу иначе. Будем строить более сложную модель ансамбля, в котором коэффициенты вхождения так же являются обучаемыми параметрами, а базовые алгоритмы приближают не искомый скрытый параметр, а будут являться решением следующего выражения:

$$\mathcal{Q}(\alpha, b, X^l) = \sum_{i=1}^{l} \mathcal{L}(\sum_{t=1}^{T-1} \alpha_t b_t(x_i) + \alpha b(x_i), y_i) \longrightarrow \min_{\alpha\in\mathbb{R}, b\in\mathcal{B}}$$

Идея подхода заключается в том, что модели b_t приближают вектор антиградиента $\mathcal{L}'_q(q_{t-1}, y)$, из чего вытекает решение задачи оптимизации $b_t = \arg\min_{b\in\mathcal{B}} \sum_{i=1}^{l}(b(x_i) - \mathcal{L}'_q(q_{t-1}, y))$. Однако такой ход в общем

Algorithm 1 Алгоритм построения Random-Forest

Require: $N > 0$, fcount $\in (0; 1]$, depth $\in \mathbb{Z}_+$, bscoef $\in [0; 1]$, strategy $\in \{best, random\}$
Require: $X^l \in \mathbb{R}^{l \times m}$, $k^l \in K$
  $q_0(x) \leftarrow 0$ ▷ начальное состояние ансамбля
  $\mathcal{B} =$ **RegressorTree(**depth, fcount, strategy**)** ▷ определение класса алгоритмов
  for each $t \in \{1, 2, ..., N\}$ **do**
    if bscoef $\neq 0$ **then**
      $X_{work}, k_{work} \leftarrow$ **subset(**X^l, k^l, bscoef**)**
    else
      $X_{work}, k_{work} \leftarrow X^l, k^l$
    end if
    $n_t \leftarrow |X_{work}|, x_i \in X_{work}, k_i \in k_{work}$
    $b_t \leftarrow \arg\min\limits_{b \in \mathcal{B}} \sum_{i=1}^{n_t} (b(x_i) - k_i)^2$
    $q_t(x) := q_{t-1}(x) + \frac{1}{N} b_t(x)$ ▷ Добавление новой модели в ансамбль
  end for

случае может привести к неустойчивости модели, потому обучаемые веса α_t получаются как произведение $coef_t = \text{learning\_rate} \times \alpha_t$, $\alpha_t = \arg\min_{\alpha > 0} \sum_{i=1}^{l} \mathcal{L}(q_{t-1}(x_i) + \alpha b_t(x_i), y_i)$. Для **градиентного бустинга** все так же используются решающие деревья по причинам, описанным ранее для **Random-Forest**.

Алгоритм построения **градиентного бустинга** будет схож с таковым для **Random-Forest** 2. Главным отличием является появление дополнительного параметра *learning-rate* (или как указан lr), а так же оптимизационной задачей для градиента функции эмпирического риска вместо изначальной задачи.

Algorithm 2 Алгоритм построения Градиентного бустинга

Require: $N > 0$, fcount $\in (0; 1]$, depth $\in \mathbb{Z}_+$, bscoef $\in [0; 1]$, strategy $\in \{best, random\}$, lr $\in \mathbb{R}$
Require: $X^l \in \mathbb{R}^{l \times m}$, $k^l \in K$
  $q_0(x) \leftarrow 0$ ▷ начальное состояние ансамбля
  $\mathcal{B} =$ **RegressorTree(**depth, fcount, strategy**)** ▷ определение класса алгоритмов
  for each $t \in \{1, 2, ..., N\}$ **do**
    if bscoef $\neq 0$ **then**
      $X_{work}, k_{work} \leftarrow$ **subset(**X^l, k^l, bscoef**)**
    else
      $X_{work}, k_{work} \leftarrow X^l, k^l$
    end if
    $n_t \leftarrow |X_{work}|, x_i \in X_{work}, k_i \in k_{work}$
    $b_t \leftarrow \arg\min\limits_{b \in \mathcal{B}} \sum_{i=1}^{n_t} (b(x_i) - \mathcal{L}'(q_{t-1}(x_i), k_i))^2$ ▷ $\mathcal{L}'$ есть градиент функции потерь
    $\alpha_t \leftarrow \arg\min\limits_{\alpha > 0} \sum_{i=1}^{n_t} \mathcal{L}(q_{t-1}(x_i) + \alpha b_t, k_i)$ ▷ Задача одномерной оптимизации
    $q_t(x) := q_{t-1}(x) + \text{lr} \cdot \alpha_t \cdot b_t(x)$ ▷ Добавление новой модели в ансамбль
  end for

4 Постановка задачи

Рассматривается задача определения ценности жилья, исходя из его параметров, таких как число квадратных метров, количество этажей, число комнат и так далее. Объекты описываются 19 признаками из X^l — пространства признаков, где l размерность выборки. $K \in \mathbb{R}$ есть множество возможных значений скрытой переменной. Задачей является восстановление зависимости $q(x) = k(x)$, минимизирующий суммарный квадратичный функционал ошибки $\mathcal{L}(x, q, y) = (q(x) - y)^2$

Особенностью данной обучающей выборки является то, что почти все признаки в ней являются числовыми параметрами, на объектах каждого из которых определено бинарное соотношение сравнения. Из особенностей разбиения пространства признаков решающими деревьями, легко видно, что они устойчивы к масштабу и смещению распределения. Таким образом, остается лишь 3 поля, что требуют обработки: *id*, что для задачи без сохранения имен объектов не несет никакой полезной информации, *date* - поле, хранящее дату закрытия сделки, а так же *zipcode* - аналог кода почтамта. В рамках предобработки удалим поле *id*, заменим запись дат в поле *date* на представление целыми числами (в $\overline{year - month - day}$), а так же закодируем *zipcode* с помощью целевой переменной.

Сохранение возможности прямого сравнения дат (*date*) является целесообразным решением, как продолжение бинарного отношения с самих дат на их представление числами. Кодирование любым другим способом меняло бы суть данного поля признаков. В свою очередь *zipcode* имеет большое количество уникальных значений (70), что могло бы слишком сильно увеличить размерность признакового пространства, сделав его менее плотным.

5 Анализ параметров моделей

Сравнение моделей и оценка их качества производится по двум метрикам: основной $RMSE = \sum_{i=1}^{l} (k_i - q(x_i))^2$, или же квадратичный функционал, используемый в качестве функции потерь при обучении алгоритмов, а так же дополнительная метрика $R^2score = 1 - \frac{\sum_{i=1}^{l}(k_i - q(x_i))}{\sum_{j=1}^{l}(k_i - \bar{y})}$. Последняя метрика выбрана из-за своих свойств учитывать статистическую дисперсию самого скрытого параметра, тем самым нивелировав влияние "шума" (**noise**) на результаты алгоритма.

5.1 Random-Forest

Начнем с оценки **Random-Forest**. Оценка включает в себя анализ поведение алгоритма на полученном наборе данных в зависимости от выбранных параметров, для этого выборка была разделена в соотношении 7 к 3 случайным образом, и далее использовалась во всех экспериментах. Влияющие на работу ансамбля параметры подразумевают количество деревьев, их глубину, размер подвыборки признаков из X, используемых при обучении, стратегию ветвления и коэффициента бутстрапа. Так же важно понимать сколько затрачивается в среднем на обучение моделей, потому вместе с метриками качества далее рассматривается так же и дельта по времени.

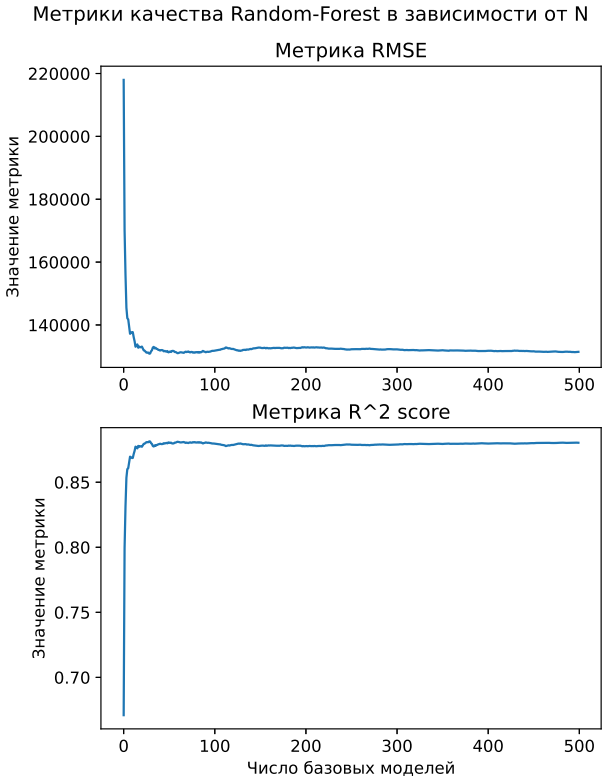

Figure 1: Метрики качества Random-Forest

5.1.1 Число деревьев N

Для формул **bias** и **variance** справедлива скорость убывания пропорциональная $\frac{1}{N}$ при условии независимости ответов моделей в совокупности. Однако последнего аспекта обучения на практике добиться крайне трудно. Потому ожидается монотонное улучшение качества работы ансамбля с ростом N с постепенным уменьшение прироста качества при добавлении каждой последующей модели в ансамбль. Обучать деревья будем на подвыборках признаков, размерности $\frac{1}{3}$ от общего числа, со стратегией ветвления по лучшему предикату, не используя бутстрапинг. Для минимизации ошибки **bias** будем использовать лишь деревья с неограниченной глубиной, так как именно они доставляют наименьшее смещение при условии, что обучающая выборка репрезентативна, а функция скрытого параметра (как говорилось ранее) "достаточно хорошая".

На графике 1 отображена зависимость метрик качества ансамбля в зависимости от N, пробегающего значения от 1 до 500. Видно, что уже при малых количествах базовых моделей, качество алгоритма выходит на плато, где остается на всем протяжении обучения. Для метрики RMSE это значение составляет примерно 131444, для R^2 — 0.88. Подтверждается изначальное предположение о динамике обучения **Random-Forest**, что исходит из коррелированности ответов моделей. Произойти такое могло из-за малого количества независимых признаков в обучающей выборке.

Применение бутстрапа на первый взгляд может исправить положение. Обучаясь на различных подвыборках коррелированность моделей должна в среднем снижаться, однако на практике это не дает никаких положительных результатов. Наоборот, при обучении деревьев на подвыборках размера 0.4 от всей обучающей, RMSE увеличилось на ≈ 5000, а R^2 уменьшился на одну сотую.

Такой же неудачей закончится и попытка использовать стратегию ветвления по случайному правилу.

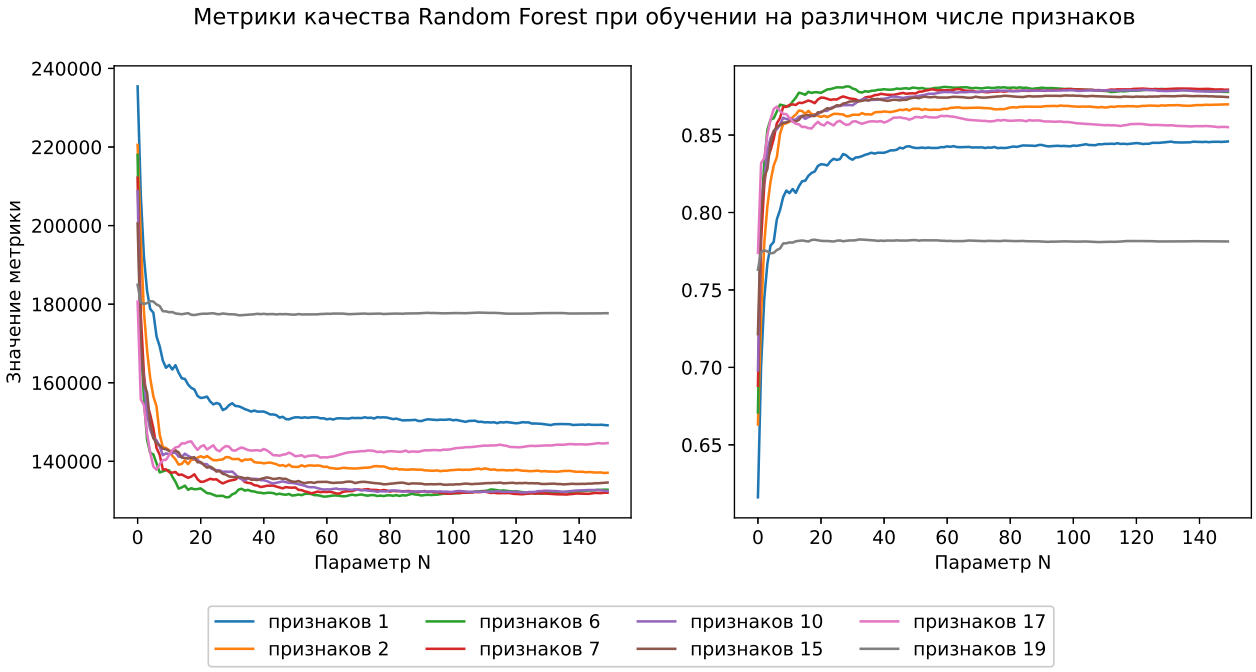

Figure 2: Метрики качества Random Forest при различной мощности подмножества признаков

> Такой способ может улучшить ситуацию с показателем **variance**, однако на практике лишь серьезно ухудшил показатели ансамбля (5 тысяч по RMSE), что могло произойти из-за серьезно ухудшившегося **bias** каждого отдельно взятого дерева.

Время: для **Random Forest** время обучения можно представить в виде конечной суммы времен обучения каждого из деревьев. Для рассмотренного случая на 6500 объектов время обучения базовых алгоритмов без ограничения по высоте на $\frac{1}{3}$ признаков в среднем составило 0.059 сек. со стандартным отклонением 0.015 сек. При чем с ростом числа деревьев тренда на изменение этого времени нет, так как каждое дерево решает изначальную задачу обособленно, что является отличительной чертой данного метода ассемблирования. При использовании бутстрапа ситуация меняется: среднее время, затрачиваемое на обучение одной модели становится уже 0.038 сек. со стандартным отклонением в 0.001 сек. Такое происходит из-за уменьшения числа объектов, на котором каждое из деревьев должно обучиться. Но сильнее всего уменьшает вычислительную сложность параметр ветвления по случайному предикату: 0.017 сек. и 0.001 сек. для среднего и отклонения соответственно.

5.1.2 Величина подвыборки признаков

Классическая рекомендация, брать $\frac{1}{3}$ от всего множества признаков для каждой модели, может привести к тому, что с ростом N слишком быстро наступает момент, когда ответы моделей становятся зависимы между собой. Все так же деревья имеют максимальную глубину; обучение производилось на меньшем числе деревьев, исходя из крайне быстрой сходимости ансамбля (150 моделей).

Графики 2 приводят зависимость метрик качества **Random Forest** от числа признаков, на которых строилось каждое из деревьев. Большинство моделей ведет себя схоже и стремятся примерно к одному и тому же плато. Однако видно, что модели с числом признаков в районе 6 (что равняется $\frac{1}{3}$ от всех признаков) достигают наилучших показателей качества. Модели, обучаемые на малом количестве параметров или же на слишком большом имеют ощутимо большую ошибку при валидации, что так же объяснимо. Малый коэффициент приводит к тому, что деревья, входящие в ансамбль, не могут хорошо приблизить функцию скрытого параметра, из-за чего имеют большой **bias**, в то время как модели с большим количеством признаков имеют большое **variance** из-за коррелированности моделей. Таким образом **Random Forest** обладает экстремумом

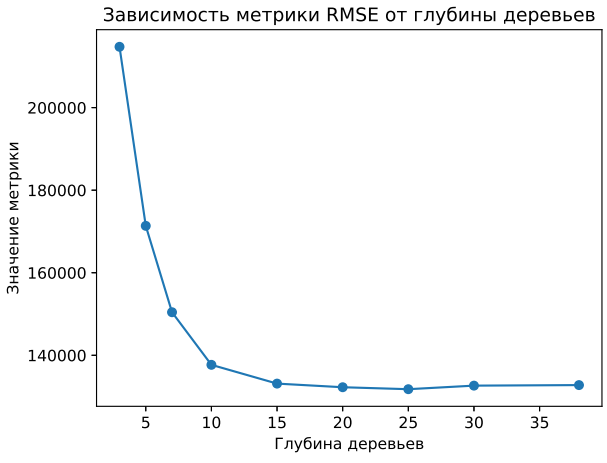

Figure 3: RMSE **Random Forest** от глубины моделей

по качеству относительно рассматриваемого параметра в районе значения по молчанию.

Время: Для стратегии ветвления по лучшему предикату с уменьшением числа признаков, из которых нужно выбирать лучший на каждом этапе, скорость обучения растет. Обратное так же верно, при чем среднее время с увеличением числа признаков растет линейно. В свою очередь стандартное отклонение не подчиняется никакому тренду и остается в райне 0.01. При стратегии ветвления случайным образом время уменьшается, как и скорость увеличения затрачиваемого времени с ростом признаков. Объяснением такому поведению является уменьшенные относительные затраты на построение деревьев.

5.1.3 Выбор глубины деревьев

Пусть $k(x) : X \to K$ есть восстанавливаемая зависимость скрытого параметра от признаков объекта. Тогда для решающих деревьев справедливо монотонное убывание числа ветвлений при уменьшении максимальной глубины, в свою очередь приводящее к уменьшению разбиения пространства признаков на множества. Из представления регрессионных деревьев в виде простых функций можно получить, что при таком процессе ухудшается и аппроксимация скрытой функции: $\sqrt{\int\cdots\int_{\tilde{X}\subset X} \left(k(x) - tree(x)\right)^2}$ в среднем увеличивается, где $\tilde{X}$ есть ограниченное выпуклое множество из X, что гарантированно содержит все объекты генеральной совокупности. Это приводит к увеличению **bias** как каждой из моделей, так и в случае **Random Forest** к увеличению **bias** всего ансамбля.

> Приходится ограничиваться ограниченным множеством $\tilde{X}$ чтобы расширить класс функций $k(x)$, иначе с теоретической точки зрения метод был бы неприменим. Выйти из ситуации, когда $k(x)$ определена на всем пространстве можно, на это указывает сама форма простой функции $tree(x)$. Будем считать, что всюду вне $\tilde{X}$ дерево принимает значение равное пределу функции на границе множества (так оно и есть в случае дерева). И, таким образом, задача из интерполяции $k(x)$ переходит в разряд экстраполяции. Чтобы такой факт не отразился сильно на результате и требуется, чтобы обучающая выборка была генеральной совокупностью.

Отображена данная зависимость на графике 3. Действительно, при малых значениях глубины базовых моделей, общая ошибка стремительно увеличивается и достигает максимума при значении глубины 1. Обучение производилось по 250 деревьям со стандартной величиной подвыборок признаков. Деревья, не имеющие ограничения по высоте в среднем имели глубину в 38-39 слоев, что отмечено крайне правой точкой на графике. Видно, что уже при малых глубинах, начиная с 15, модели ансамблей стабильно приносят примерно одинаковый результат.

Время: так же как и для величины признакового подмножества, с ростом глубины деревьев увеличивается и время по линейному закону (от примерно 0.01 сек. при глубине 3 до 0.08 при неограниченном числе деревьев).

5.2 Градиентный бустинг на деревьях

В отличие от **Random Forest**, **градиентный бустинг** является итеративным методом, где построение каждого алгоритма опирается на результат построения предыдущего. При этом каждый из таких алгоритмов решает не исходную задачу, а лишь приближает градиент функции потерь от ансамбля на предыдущем шаге. Для такого метода появляется понятие $learning - rate$, а обучение больше походит на градиентный спуск. В последующих экспериментах использовалась обучающая и тестовая выборки из секции **Random Forest**.

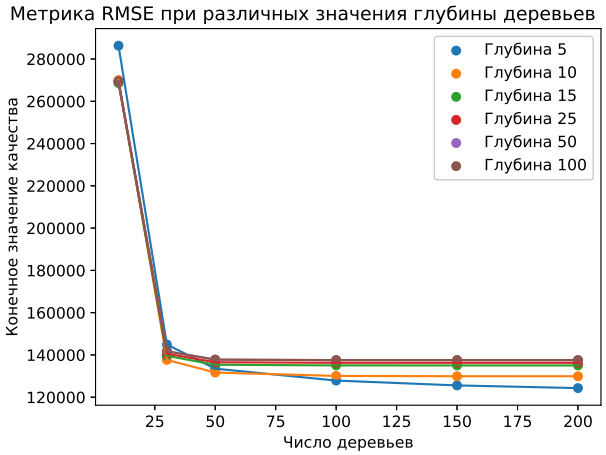

Figure 4: RMSE **Градиентного бустинга** от глубины моделей

5.2.1 Параметр глубины деревьев

Градиентный бустинг каждым своим базовым алгоритмом **не** решает исходную задачу оптимизации. Идея состоит в том, чтобы каждая последующая модель исправляла накопленную суммарную ошибку предыдущих $(\sum_{i=1}^{l}(k_i - \sum_{t=1}^{T-1} b_t(x_i)))$. Чтобы подобная модель была лишена проблемы переобучения, каждый отдельный алгоритм должен плохо решать поставленную задачу приближения

скрытого параметра, иначе все остальные модели решали бы задачу приближения нулевого вектора (остатка градиента). Тем самым работа ансамбля сводилась бы к работе одного дерева, имеющего относительно большой **variance**. Ожидаемым результатом является ухудшение конечного качества при увеличении глубины деревьев.

График 4, с приведенной динамикой обучения алгоритмов подтверждает такие выводы. Проводилось усреднение конечных метрик по параметру величины подмножества признаков, используемых при обучении базовых моделей; $learning - rate$ принят за стандартное значение в 0.1. Прослеживается монотонное улучшение качества ансамбля от максимальной глубины, при чем наименьшая глубина способна доставлять наилучшее качество. Однако при уменьшении глубины увеличивается необходимое число деревьев в ансамбле для выхода на устойчивое плато, хотя данный фактор слишком мал, чтобы брать его в расчет при таком выигрыше по качеству.

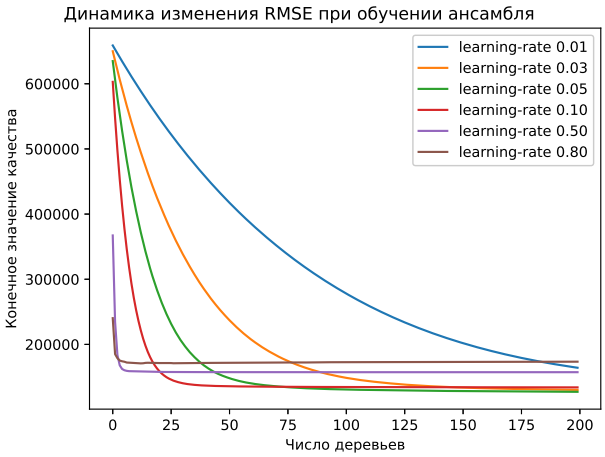

Figure 5: RMSE **Градиентного бустинга** от числа моделей при различных $learning - rate$

Время: Для **Градиентного бустинга** время обучения так же как и для **Random Forest** представимо в виде суммы времен обучения деревьев. Однако теперь деревья зависимы между собой. Тем не менее точно так же видна практически линейная зависимость среднего времени обучения ансамбля от времени, с его увеличением при росте глубины деревьев. При чем лучшим по качеству оказывается алгоритм практически наилучший и по времени, что сильно отличается от **Random Forest**. Тем не менее в рамках одного обучения трендов по увеличению или уменьшению времени обучения очередного базового алгоритма так же нет. Это вызвано самим устройством решающих деревьев, способом построения гиперплоскостей в пространстве, не зависящих от масштаба приближаемой функции.

5.2.2 Число деревьев N и параметр learning-rate

В прошлом пункте уже прослеживалась динамика изменения качества **Градиентного бустинга** при росте числа моделей (качество достаточно быстро росло). Однако из особенностей данного подхода построения ансамбля вытекает и тот факт, что с ростом числа деревьев, качество может не улучшаться монотонно, а даже падать. Например, при большом параметре $learning - rate$ модель ожидаемо будет подвержена осцилляции около глобального минимума (из условия выпуклости функции потерь). Оценки сходимости по метрике RMSE брались для ансамблей от 10 до 200 базовых моделей, обучаемых на подмножестве признаков размера $\frac{1}{3}$ от всего X; $learning - rate$ брался по сетке $[0.01, 0.03, 0.05, 0.1, 0.5, 0.8]$; результаты усреднялись относительно мощности множества признаков.

График 5 позволяет увидеть явную зависимость качества обучения как от параметра $learning - rate$, так и целевой N. При любом малом значении $learning - rate$ алгоритм стремится выйти на плато. При чем при всех коэффициентах плато находятся близко друг другу, что говорит об общей сходимости алгоритма. Однако даже так график способен проиллюстрировать сразу два нежелательных состояния: переобучение и недообучения. К первому состоянию относятся кривые $learning - rate \in \{0.5, 0.8\}$, при котором ансамбль, резко подстроился под данные обучающей выборки, вследствие чего показывает результаты хуже других. Некоторые деревья слишком быстро решили исходную задачу, из-за чего последующие уже не могли вносить свой вклад в уменьшение статистической ошибки алгоритма, нивелируя преимущества ансамбля. Данный факт хорошо иллюстрирует идею построения ансамблей: каждый алгоритм должен вносить свой вклад в конечное решение, в лучшем случае, если этот вклад в среднем одинаков. Противоположная ситуация с моделью $learning - rate = 0.01$. Она сходится слишком медленно, чтобы за 200 итераций хотя бы приблизиться к своему оптимальному значению. Такое решение неудобно из-за больших трат вычислительных ресурсов. Из всех оставшихся самым лучшим является параметр по умолчанию, равный 0.1

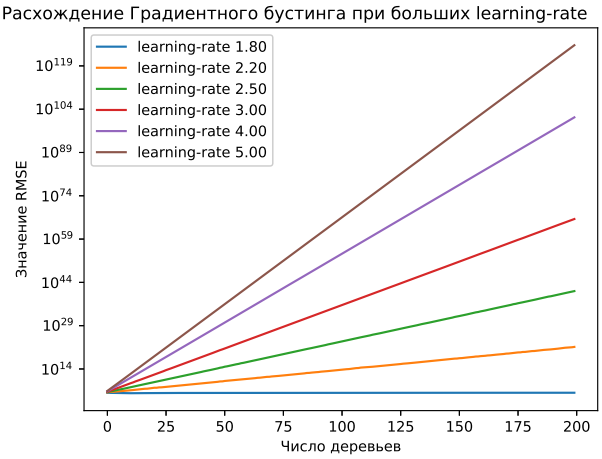

Figure 6: Метрика RMSE для неправильно выбранных $learning - rate$

При более экстремальных значениях параметра $learning{-}rate$ наблюдается оговоренная выше расходимость

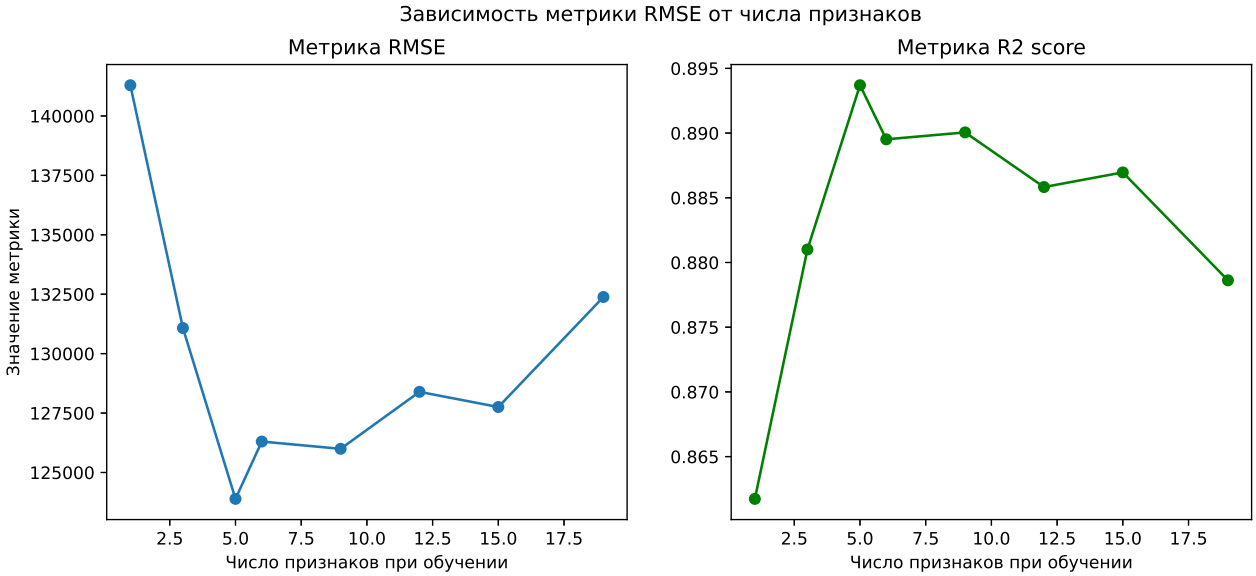

Figure 7: Метрики качества **Градиентного бустинга** при различной мощности подмножества признаков

алгоритма. Иллюстрация такому эффекту приведена на графике 6, где уже при относительно небольших значениях ансамбль не сходится, ошибка стремительно растет.

Время: градиентный бустинг все так же повторяет особенности **Random Forest** относительно времени. С ростом числа базовых моделей время, затрачиваемое на обучение, растет линейно. При чем все так же нет трендов на изменение скорости обучения деревьев в процессе для одного ансамбля. $learning - rate$ так же не влияет на суммарное и удельное время. Из особенностей решающих деревьев одинаково хорошо справляться с данными различного масштаба, любой ход алгоритма в процессе градиентного спуска не меняет время обучения конкретной базовой модели.

5.2.3 Величина подвыборки признаков

По данному аспекту **градиентный бустинг** повторяет требования **Random Forest**. Для улучшения качества работы ансамбля требуется уменьшить коррелированность моделей, в таком случае должно существовать некоторое положение оптимума, в окрестностях которого доставляется наилучшее качество (меньше $\Rightarrow$ большое **bias** (смещение), больше $\Rightarrow$ меньше возможных комбинаций признаков для обучения алгоритмов, и как итог увеличение **variance**). Рассматриваются ансамбли, состоящие из 100 базовых моделей имеющих ограничение в 5 слоев в глубину, с критерием ветвления по наилучшему разбиению и обученные без применения бутстрапа.

Изначальное суждение соответствует действительности. На графиках 7 обеих метрик RMSE и R^2 видна ярко выраженная точка глобального максимума в районе на 5, в которой алгоритм лучше всего работает на валидационной выборке. При этом качество резко падает при подходе к 1, намного быстрее чем при подходе к максимальным 19. Это указывает на большую роль ошибки смещенности над ошибкой случайного шума.

Время: Линейная зависимость обучения каждого дерева ансамбля присутствует и для параметра мощности подвыборки признаков. Здесь так же отсутствуют тренды на изменение времени обучения каждого из базовых алгоритмов внутри ансамблей.

5.3 Использование бутстрапинга

При построении ансамблей важным параметром является независимость базовых моделей. Главным используемым способом в рамках поставленной задачи является оговоренный уже не раз случайный выбор подмножества признаков из X. Его главная идея заключается в обучении базовых моделей на всей обучающей выборке, при этом обеспечивая их независимость друг от друга. Однако как можно убедиться из экспериментов, даже он не лишен проблем: он работает не при всех размерах подвыборки, и работает не на всем процессе обучения (быстро он упирается в предел своих возможностей, например из-за малой размерности исходных данных).

Более радикальные приемы, такие как бутстрап, должны были бы изменить ситуацию, однако на практике это ни к чему не привело. Как для **Random Forest**, так и для **Градиентного бустинга**,

применение бутстрапа лишь ухудшает конечное качество на данной задаче, однако для каждого ансамбля по своему. Графики 8 изображают данную дельту значения метрик алгоритмов с применением бутстрапа и без, где стандартным алгоритмом (baseline) является соответсвующий ансамбль без. Видно, что с ростом коэффициента (что говорит о росте числа объектов, на которых обучается каждый из базовых алгоритмов) растет в среднем и качество, приближаясь к baseline, указывая на то, что в выборке присутствуют редкие, но важные примеры объектов. Разбивая выборку на обучении, есть возможность, что такие объекты в очередную подвыборку не попадут, тем самым испортив предсказательную способность базовой модели. Вместе с тем, для обеих стратегий построения ансамблей при применении одинаковых коэффициентов бутстрапа асимптотика качества практически одинакова, что может указывать на причину такого поведения в данных, нежели самих моделях. Тем не менее на данной задаче применение таких методов уменьшения корреляции бесполезно.

6 Сравнение моделей

Обе стратегии построения ансамблей смогли себя показать в разы лучше более простых моделей на представленной выборке. Однако каждая из них отличается своими достоинствами и недостатками. **Random Forest** показал наиболее устойчивые результаты, он не склонен к переобучению, крайне быстро выходит на плато качества и остается на нем, имеет широкий диапазон применимости для всех своих входных параметров, при которых модель остается сравнительно эффективной. При этом он и лишен возможности преодолеть качественный барьер, закладываемый данными. С другой стороны **Градиентный бустинг** способен решить данную задачу. Он способен получать лучшие качественные оценки, однако при этом затрачивая большее время для приближения оптимального ответа, он много больше зависит от аккуратного выбора параметров, имеет возможность переобучиться, разойтись в процессе обучения, и при этом более чувствителен к обучающим данным (что подтверждает эксперимент с использованием бутстрапа).

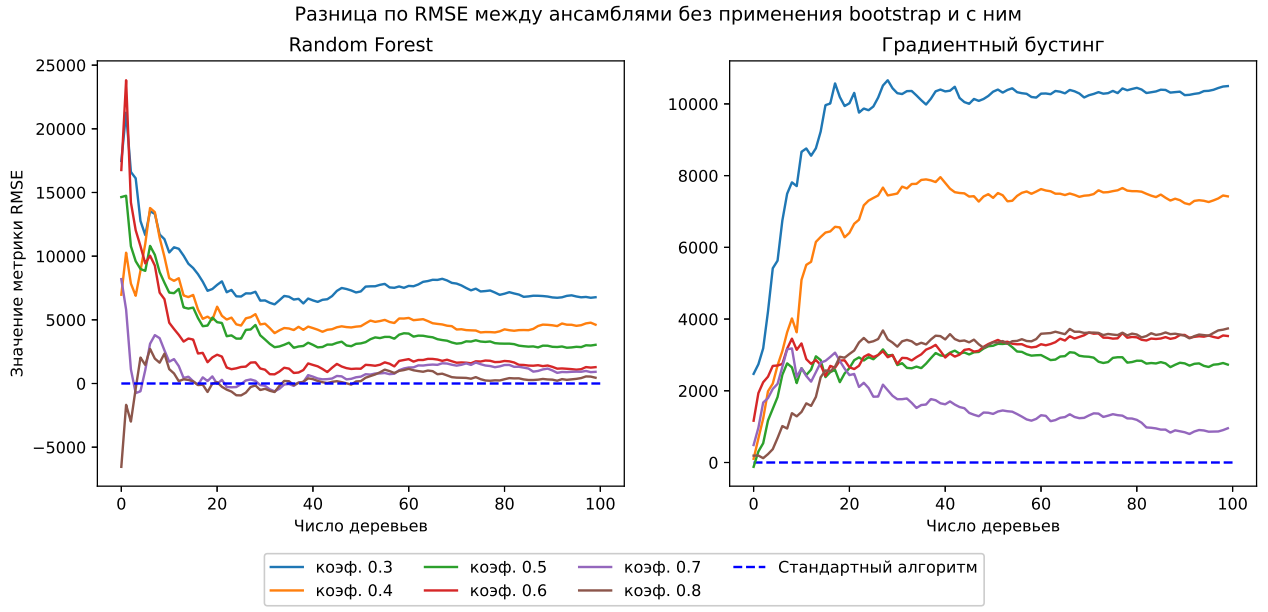

Figure 8: Разница в качестве между алгоритмами с применением bootstrap и без