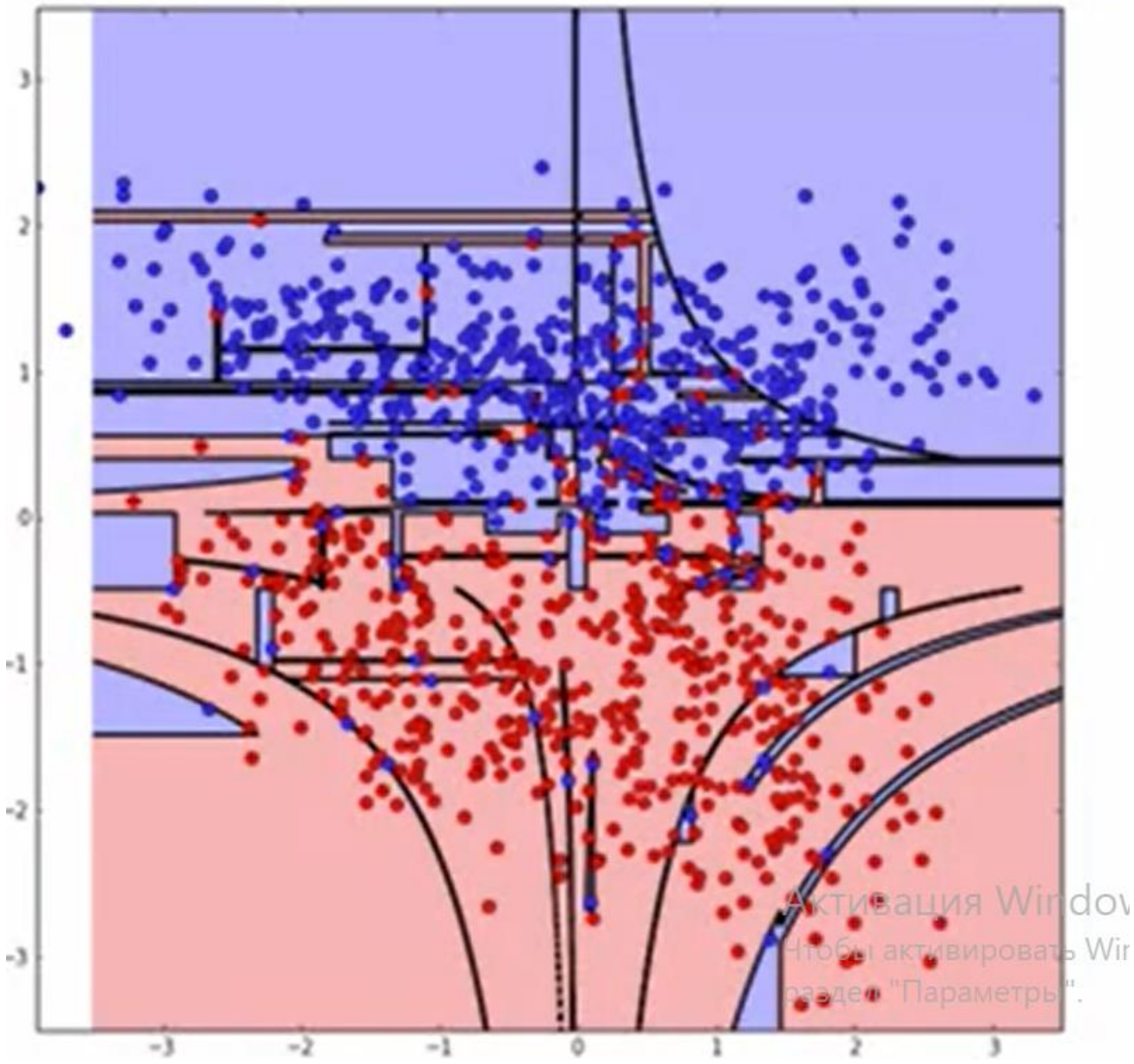
Композиционные методы машинного обучения. Random forest.

Достоинства и недостатки деревьев решений

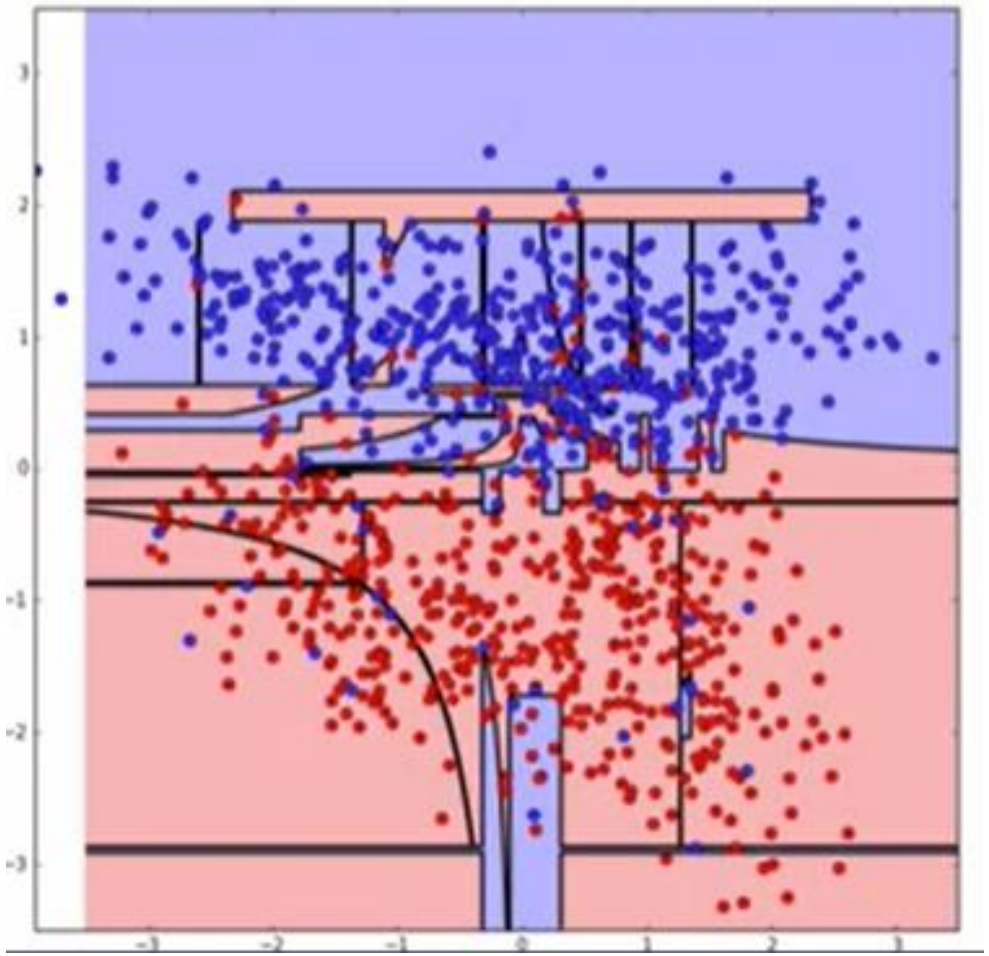
Достоинства:

- Поддерживают работу с входными переменными разных (смешанных) типов
- Возможность обрабатывать данные с пропущенными значениями
- Устойчивы к выбросам
- Нечувствительность к монотонным преобразованиям входных переменных
- Поддерживают работу с большими выборками
- Возможность интерпретации построенного решающего правила

Переобучение



Неустойчивость к изменениям



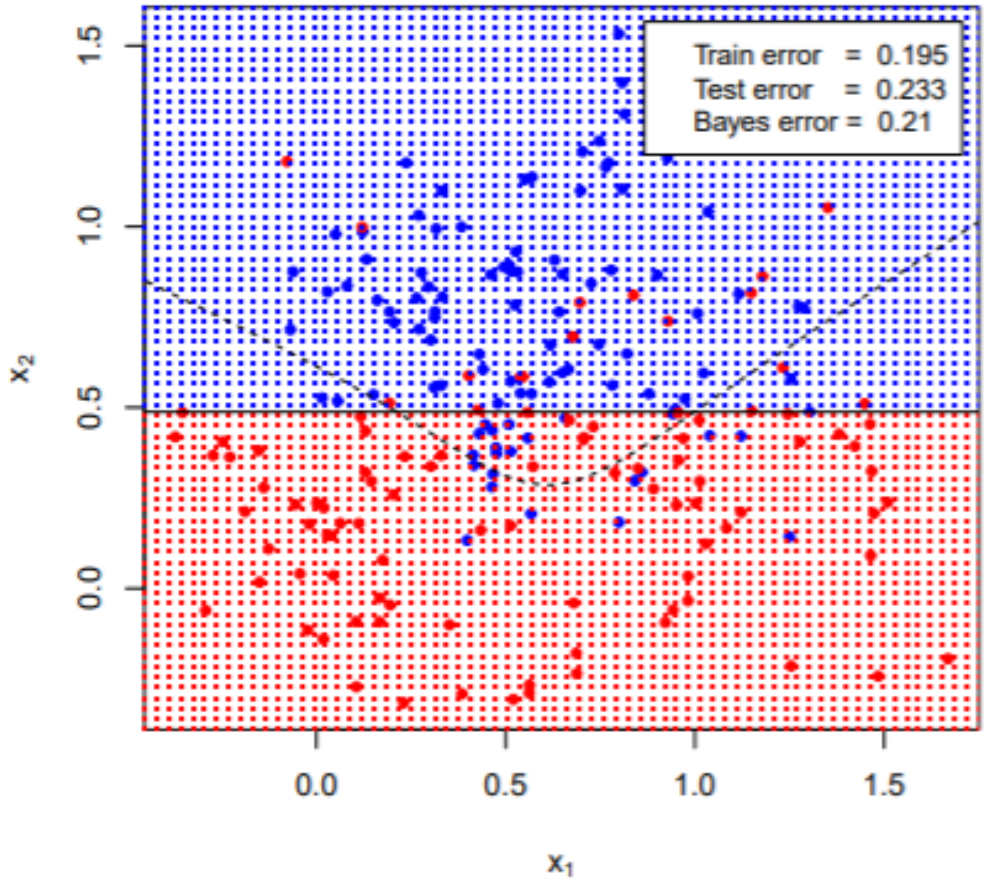
Метод ансамбля

- Мудрость толпы
- Можно ли научиться комбинировать слабые классификаторы, чтобы получить сильный [Kearns, Valiant, 1988]?

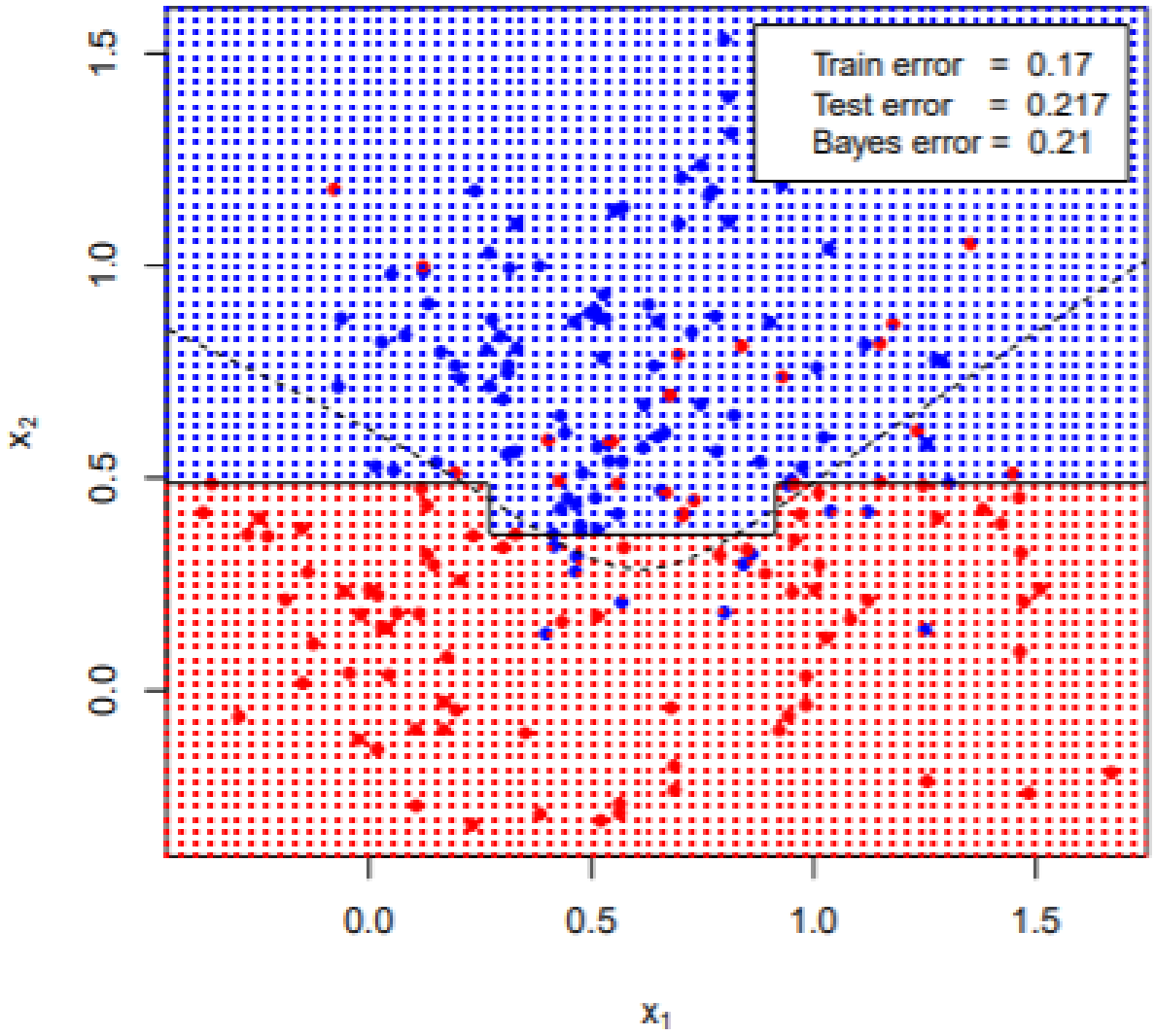
- Единство –это сила

Пример Слабые классификаторы — деревья решений высоты 1 (stumps)

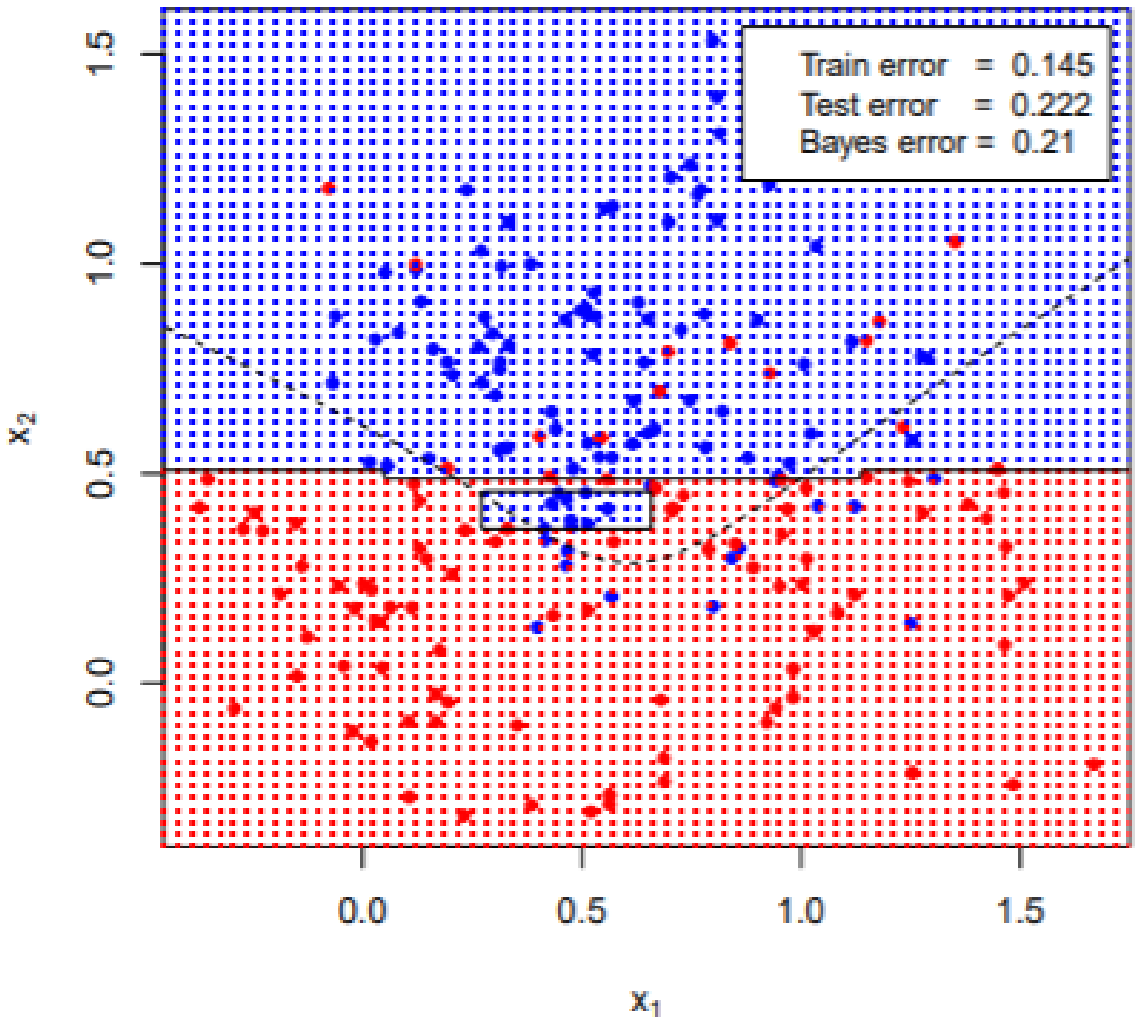
$M = 1$



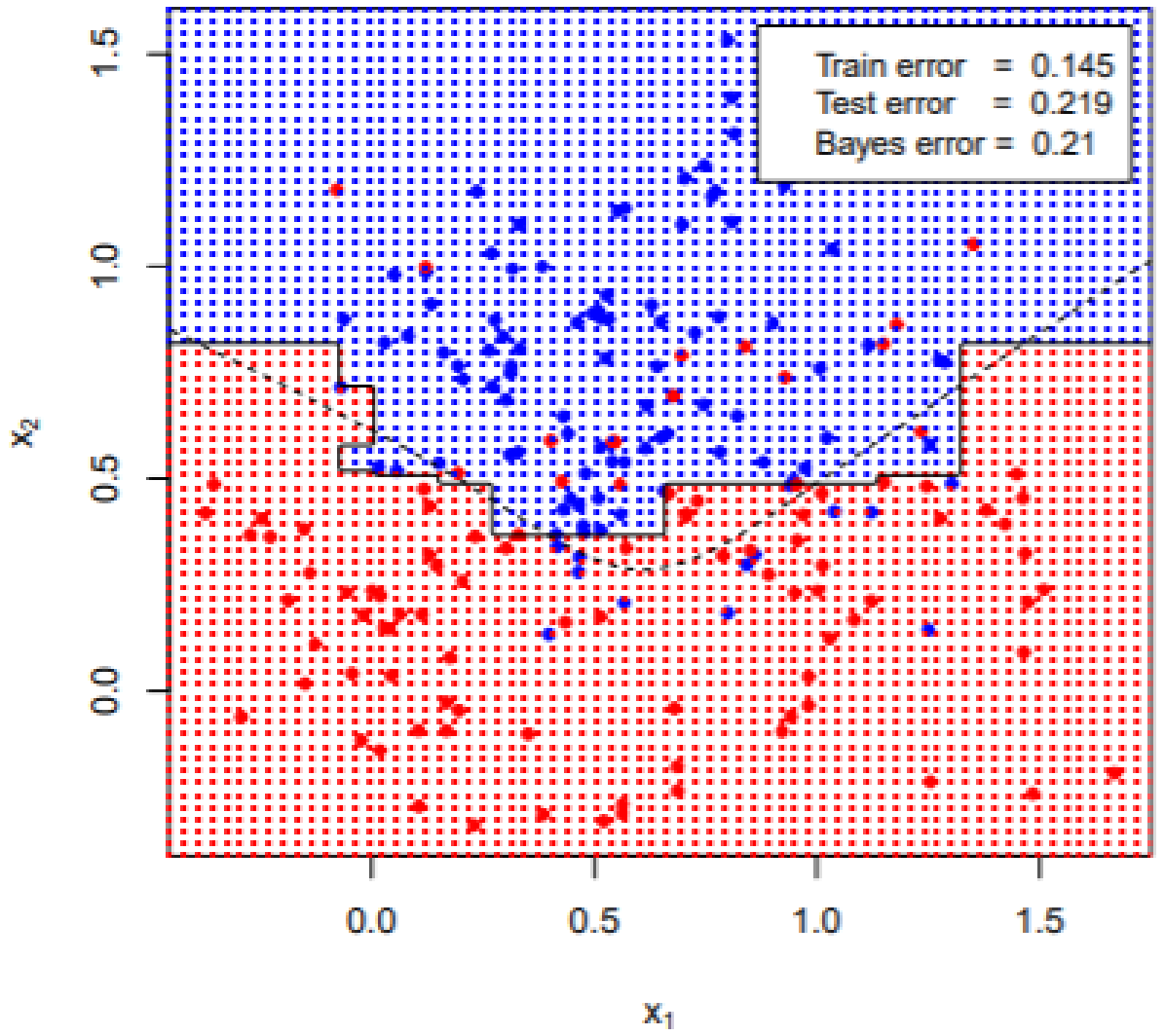
$M = 25$


Train error = 0.17
Test error = 0.217
Bayes error = 0.21
0.0
0.5
1.0
1.5
x_2
x_1

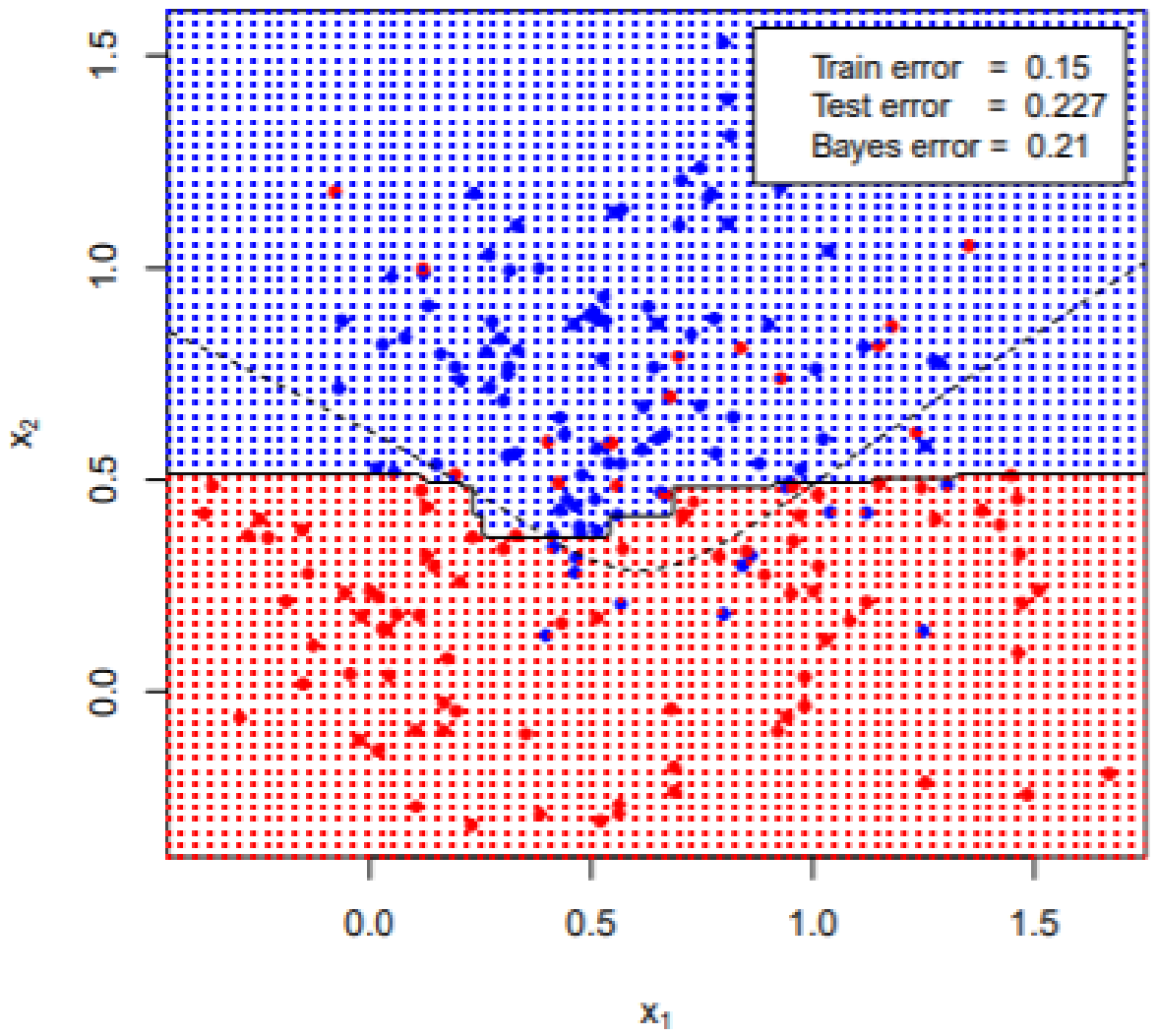
$M = 50$


Train error = 0.145
Test error = 0.222
Bayes error = 0.21
0.0
0.5
1.0
1.5
x_1
x_2

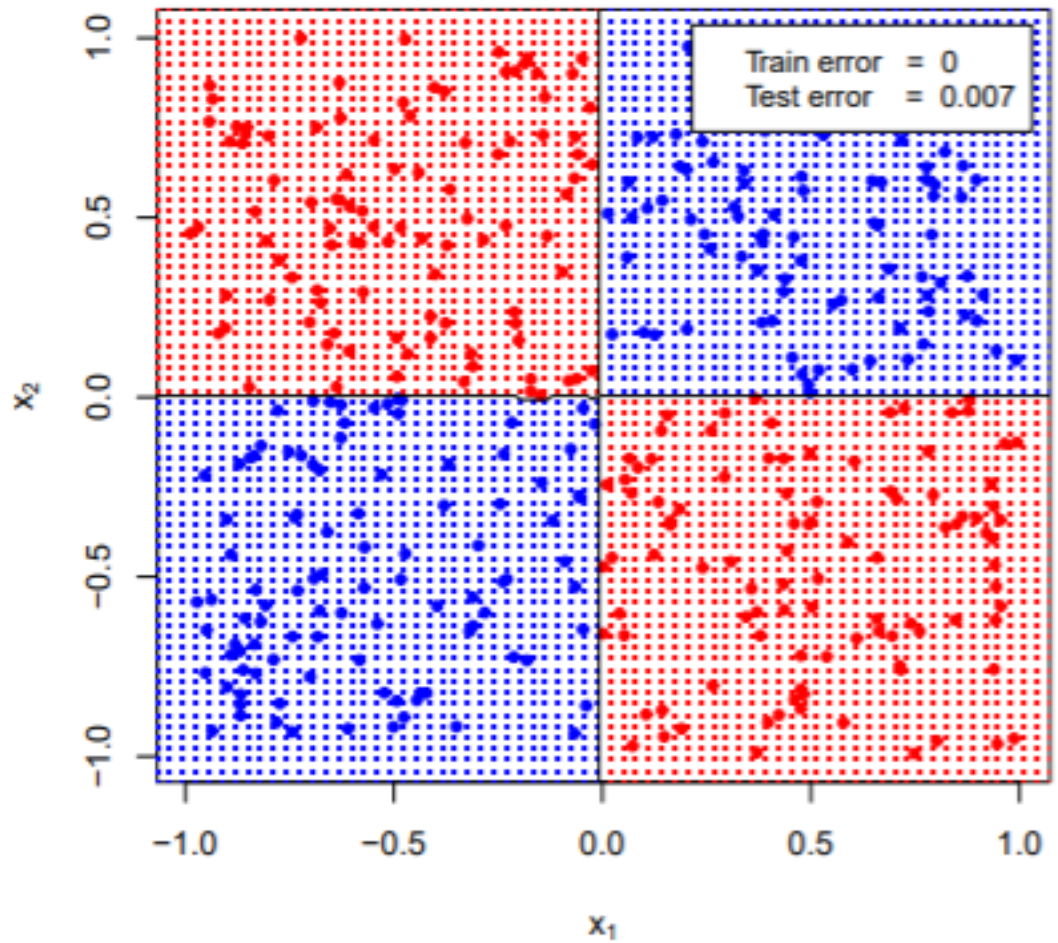
$M = 100$


Train error = 0.145
Test error = 0.219
Bayes error = 0.21
0.0
0.5
1.0
1.5
x_1
x_2

$M = 1000$


Train error = 0.15
Test error = 0.227
Bayes error = 0.21
0.0
0.5
1.0
1.5
x_1
x_2

50 деревья решений высоты 2



(Если брать деревья решений высоты 1 — не хватает даже 1000)

- ***Ансамблевые методы*** — это парадигма машинного обучения, где несколько моделей (часто называемых «слабыми учениками») обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов. Основная гипотеза состоит в том, что при правильном сочетании слабых моделей мы можем получить более точные и/или надежные модели.

Ансамблирование

- Цель **методов ансамбля** состоит в том, чтобы объединить предсказания нескольких базовых оценок, построенных с данным алгоритмом обучения, чтобы улучшить обобщаемость / устойчивость по одной оценке.

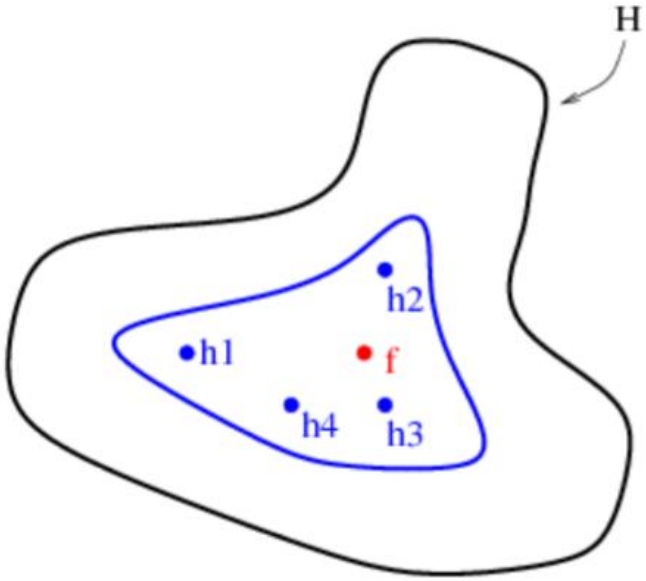
Выделяют два семейства ансамблевых методов:

- В **методах усреднения** основной принцип состоит в том, чтобы построить несколько классификаторов независимо, а затем усреднить их прогнозы. В среднем комбинированный классификатор обычно лучше, чем любой из базовых классификаторов, потому что его дисперсия уменьшается.

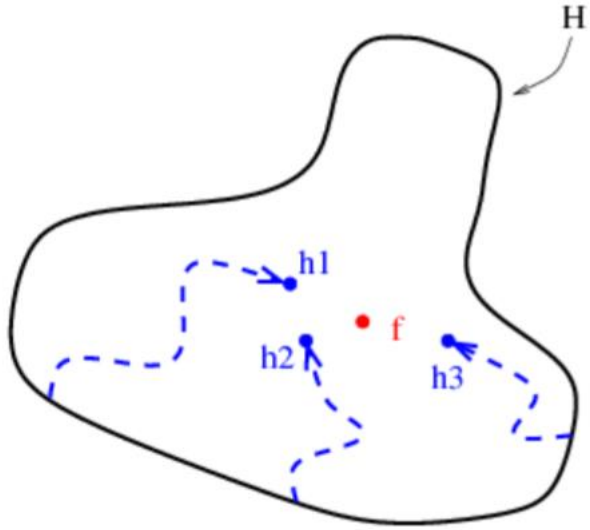
- Напротив, в **методах повышения** базовые оценки строятся последовательно, и каждый пытается уменьшить смещение объединенной оценки. Мотивация состоит в том, чтобы объединить несколько слабых моделей для создания мощного ансамбля.

Три причины, по которым объединение классификаторов может быть успешным:

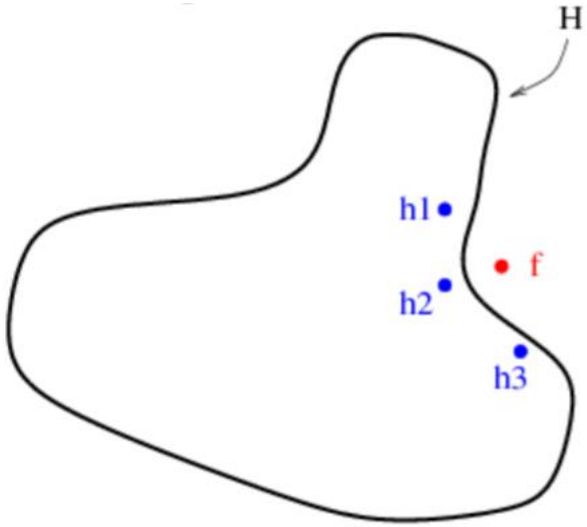
- Статистическая причина. Классификационный алгоритм можно рассматривать как процедуру поиска в пространстве гипотез H о распределении данных с целью поиска наилучшей гипотезы f. Обучаясь на конечном наборе данных, алгоритм может найти множество различных гипотез одинаково хорошо описывающих обучающую выборку. Строя ансамбль моделей, мы «усредняем» ошибку каждой индивидуальной гипотезы и уменьшаем влияние нестабильностей и случайностей при формировании гипотез.



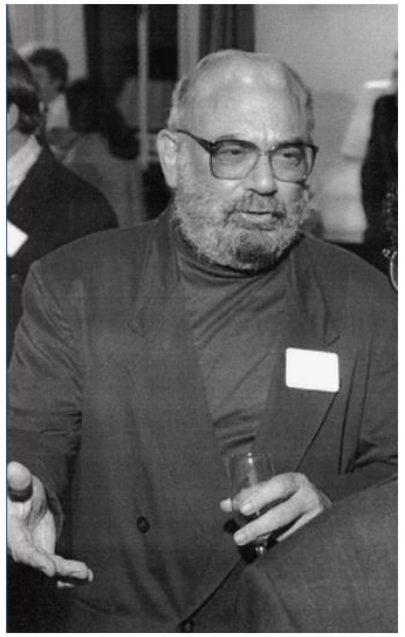
- Вычислительная причина. Большинство обучающих алгоритмов используют методы нахождения экстремума некой целевой функции. Например, деревья решений – жадные алгоритмы роста дерева, минимизирующие энтропию данных, нейронные сети используют 3 методы градиентного спуска для минимизации ошибки прогноза, и т.д. Эти алгоритмы оптимизации могут «застрянуть» в точке локального экстремума. Ансамбли моделей, комбинирующие результаты прогноза базовых классификаторов, обученных на различных подмножествах исходных данных, имеют больший шанс найти глобальный оптимум, так как ищут его из разных точек исходного множества гипотез



- Репрезентативная причина. Комбинированная гипотеза может не находиться в множестве возможных гипотез для базовых классификаторов, т.е. строя комбинированную гипотезу, мы расширяем множество возможных гипотез.



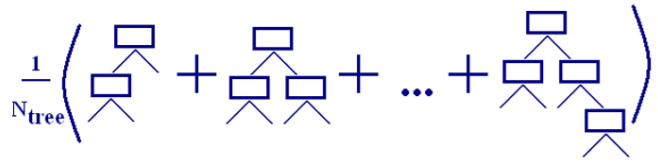
Лео Брейман, 1928 – 2005

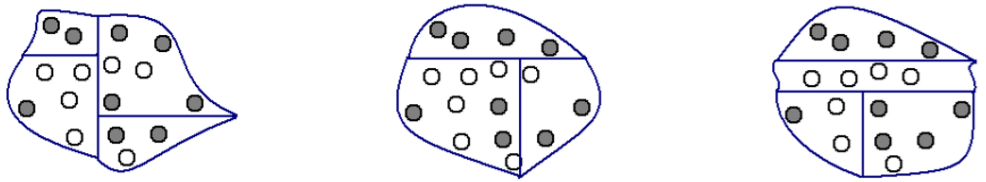


Случайные леса
Random forest (Breiman, 2001

- леса решений – несколько деревьев, результат классификации определяется путем голосования (ответом выбирается тот класс, который предсказало наибольшее число деревьев).

Что такое случайный лес?





- Идея:
  - Обучим много деревьев $b_1(x), \ldots, b_N(x)$
  - Усредним ответы:

$$a(x) = \text{sign}\, \frac{1}{N} \sum_{n=1}^{N} b_n(x)$$

Рассмотрим задачу классификации на K классов.

$$\mathscr{Y} = \{1, 2, \ldots, K\}\,.$$

Пусть имеется M классификаторов («экспертов») $f_1, f_2, \ldots, f_M$

$$f_m : \ \mathscr{X} \to \mathscr{Y}, \qquad f_m \in \mathscr{F}, \qquad (m = 1, 2, \ldots, M)$$

Построим новый классификатор:

простое голосование:

$$f(x) = \underset{k=1,\ldots,K}{\operatorname{argmax}} \sum_{m=1}^{M} I\big(f_m(x) = k\big),$$

- Чтобы дать прогноз для случайного леса, алгоритм сначала дает прогноз для каждого дерева в лесе.
- Решая задачу классификации, каждое дерево сначала вычисляет для наблюдения листовые вероятности классов.
- Листовая вероятность класса — это доля объектов класса в листе, в который попало классифицируемое наблюдение.
- Каждое дерево голосует за класс с наибольшей листовой вероятностью. Таким образом, объект относится к тому классу, за который проголосовало большее число деревьев.

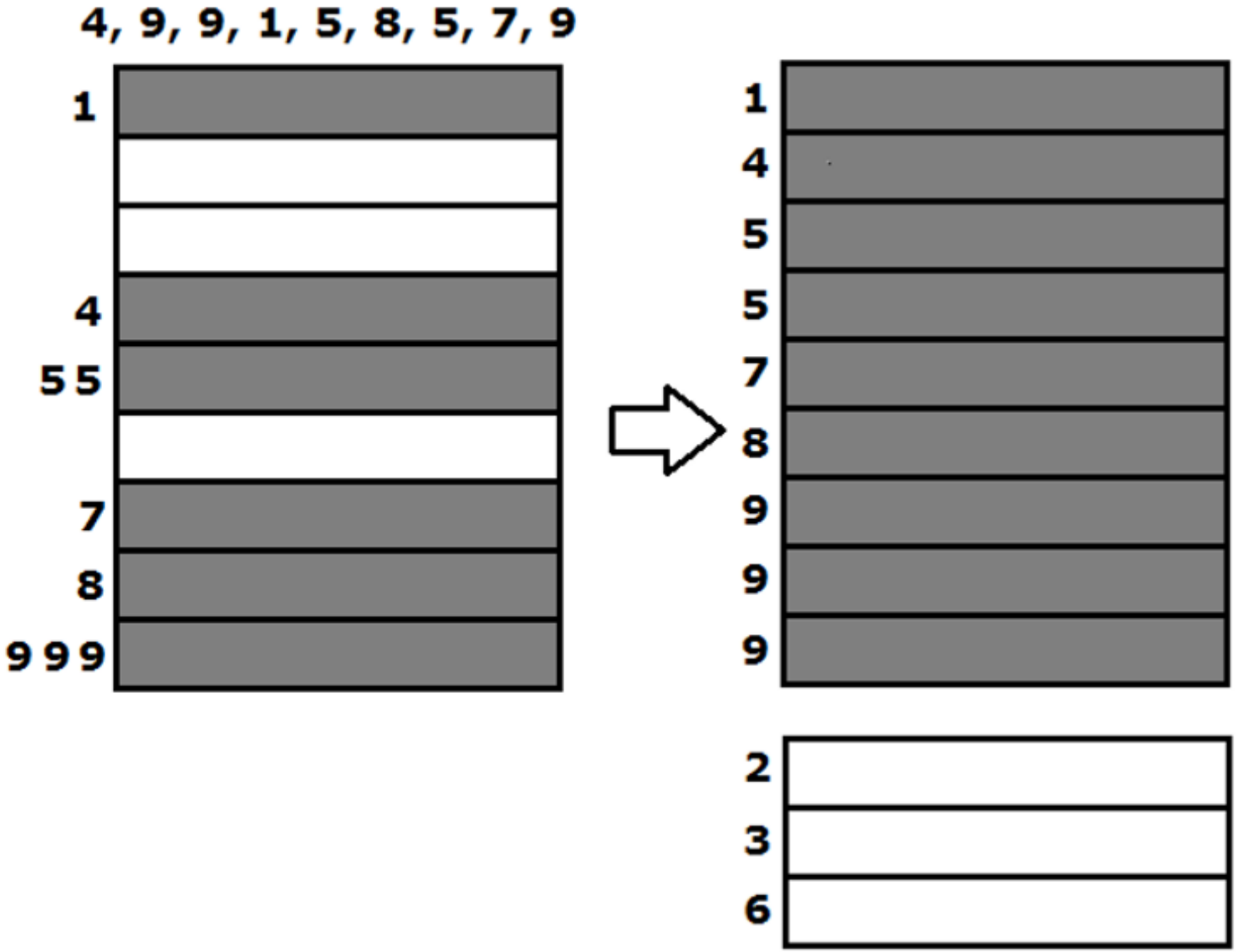
ПРИМЕР

- Прогнозы деревьев: −1, −1, 1, −1, 1, −1

$$a(x) = \text{sign} -\frac{2}{6} = -1$$

- Как сделать деревья разными?
- Обучать по подвыборкам – кросс валидация?
- Рандомизация подвыборок

Бутстреп



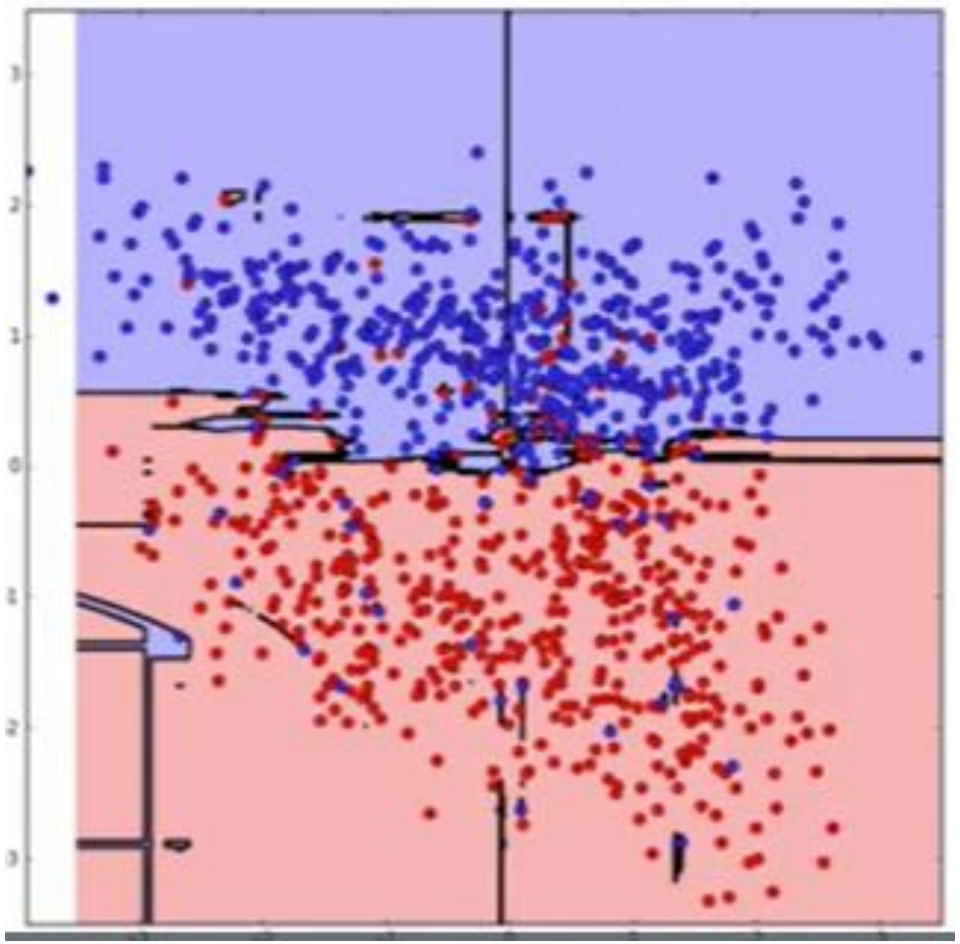
обучение

контроль

- Выбираем из обучающей выборки ℓ объектов с возвращением
- Пример: $\{x_1, x_2, x_3, x_4\} \to \{x_1, x_2, x_3, x_4\}$
- Примерно $0.632 * \ell$ различных объектов

Бутстреп + композиция

- 100 деревьев



Этапы Случайного леса

- Независимое построение определенного количества M (например, 500) деревьев.
- генерация случайной bootstrap-выборки из обучающей выборки (50–70% от размера всей обучающей выборки);

Механизм работы бутстрепа

Исходная выборка	1	2	3	4	5	6	7	8	9	10
Бутстреп-выборка I	10	9	7	8	1	3	9	10	10	7
Бутстреп-выборка II	4	8	5	8	3	9	2	6	1	6
Бутстреп-выборка III	6	2	6	10	2	10	3	6	5	1
Бутстреп-выборка IV	6	7	8	10	6	10	9	10	8	2
Бутстреп-выборка V	5	8	1	8	5	7	10	1	10	9

Бутстреп-выборка должна иметь тот же самый размер, что и исходная выборка

Поскольку отбор с возвращением, одно и то же наблюдение может попасть в бутстреп-выборку несколько раз

- На основе каждой сформированной бутстреп-выборки строится полное бинарное дерево решений
- построение дерева решений по данной подвыборке: в каждом новом узле дерева переменная для разбиения выбирается не из всех признаков, **а из случайно выбранного их подмножества небольшой мощности.**

Как выбрать количество признаков?

- Рекомендуется в задачах классификации брать $m = \lfloor \sqrt{d} \rfloor$, а в задачах регрессии — $m = \lfloor d/3 \rfloor$, где d — число признаков.
- Также рекомендуется в задачах классификации строить каждое дерево до тех пор, пока в каждом листе не окажется по одному объекту.

Алгоритм Random Forest

1: **для** $n = 1, \ldots, N$

2: Сгенерировать выборку $\tilde{X}_n^\ell$ с помощью бутстрэпа

3: Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}_n^\ell$:

- дерево строится, пока в каждом листе не окажется не более $n_{\min}$ объектов
- при каждом разбиении сначала выбирается m случайных признаков из p, и оптимальное разделение ищется только среди них

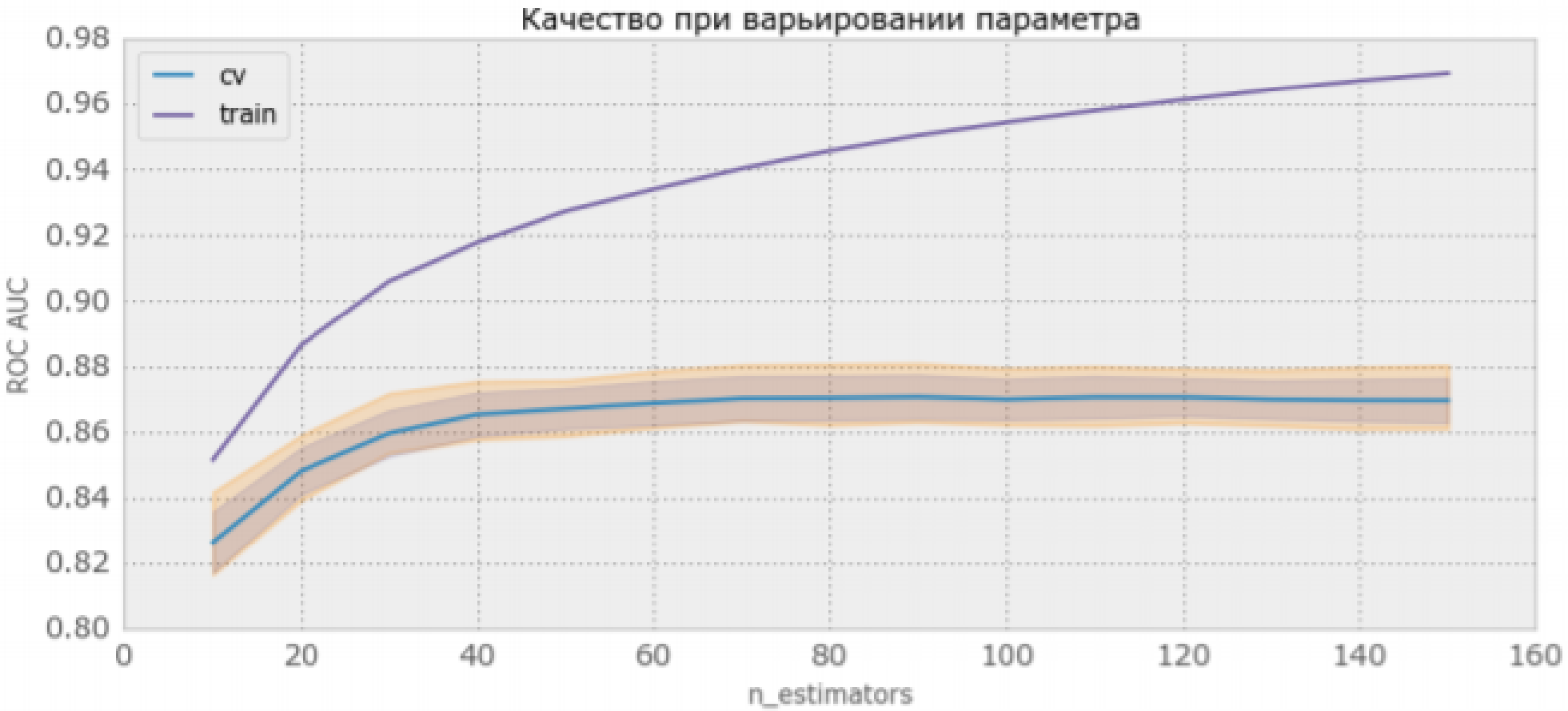
4: Вернуть композицию $a_N(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$

Параметры случайного леса

```
class
sklearn.ensemble.RandomForestClassifier
(n_estimators=10,
criterion='gini',
max_depth=None,
min_samples_split=2,
min_samples_leaf=1,
min_weight_fraction_leaf=0.0,
max_features='auto',
max_leaf_nodes=None,
bootstrap=True,
oob_score=False,
n_jobs=1,
random_state=None,
verbose=0,
warm_start=False,
class_weight=None)
```

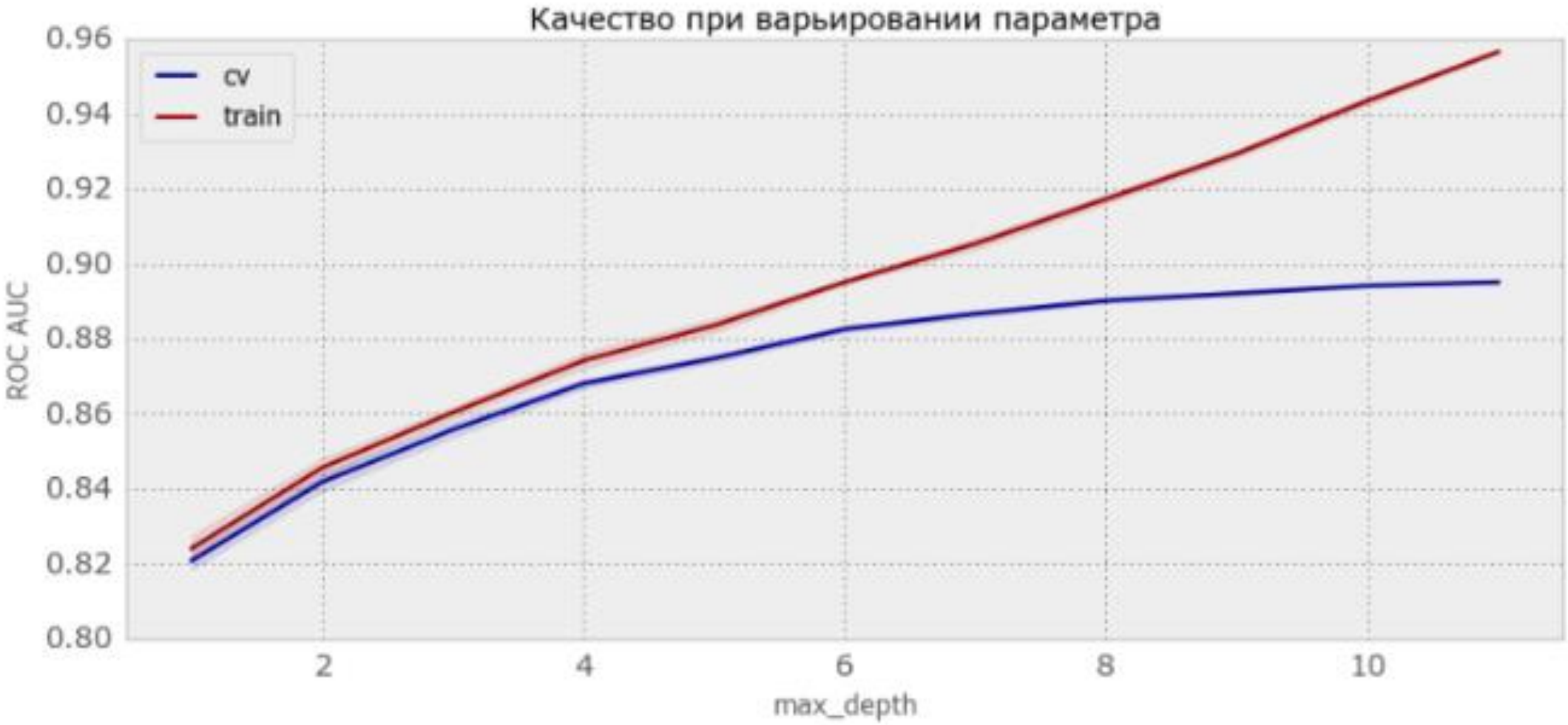
```
{randomForest} randomForest(
x, y, xtest, ytest,
ntree=500,
mtry=if (!is.null(y) &&
!is.factor(y))
max(floor(ncol(x)/3), 1) else
floor(sqrt(ncol(x))),
replace=TRUE,
classwt=NULL,
cutoff,
strata,
sampsize = if (replace) nrow(x)
else ceiling(.632*nrow(x)),
nodesize = if (!is.null(y) &&
!is.factor(y)) 5 else 1,
maxnodes = NULL,
importance=FALSE,
localImp=FALSE,
nPerm=1,
proximity, oob.prox=proximity)
```

```
class
sklearn.ensemble.RandomForestClassifier(n_estimators=1
0,
criterion='gini', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features='auto', max_leaf_nodes=None,
min_impurity_split=1e-07,
bootstrap=True, oob_score=False, n_jobs=1,
random_state=None, verbose=0, warm_start=False,
class_weight=None)
```



Чем больше деревьев – тем лучше!

- **Максимальная глубина деревьев — max_depth**



- «Если не удалось повысить качество модели за счет увеличения количества деревьев и количества отбираемых признаков, попробуй стартовое значение».

- Случайный лес использует рандомизацию, установка различных стартовых значений генератора случайных чисел (или вообще отказ от использования стартового значения) может кардинально изменить построение модели.
- random_state — начальное значение для генерации случайных чисел

Задание- исследовать влияние это параметра на качество

- Пусть в выборке ℓ объектов.
- На каждом шаге все объекты попадают в подвыборку с возвращением равновероятно, т.е отдельный объект — с вероятностью 1ℓ.
- Вероятность того, что объект НЕ попадет в подвыборку (т.е. его не взяли ℓ раз):

$$(1 - \frac{1}{\ell})^{\ell}.$$

При $\ell \to +\infty$ получаем один из "замечательных" пределов $1e$.

Тогда вероятность попадания конкретного объекта в подвыборку $\approx 1 - 1e \approx 63\%$.

Каждый базовый алгоритм обучается на ~63% исходных объектов. Значит, на оставшихся ~37% его можно сразу проверять.

Out-of-Bag оценка — это усредненная оценка базовых алгоритмов на тех ~37% данных, на которых они не обучались.

Качество модели

- Оценка out of bag для части множества, не попавшей в обучающую выборку.
- Для леса деревьев решений не обязательно проводить кросс-валидацию или тестирование на отдельной выборке. Достаточно ограничиться оценкой out of bag.
- Каждое дерево строится с использованием разных образцов бутстрэпа из исходных данных. Примерно 37% примеров остаются вне выборки бутстрэпа и не используются при построении k-го дерева.

Смещение и разброс

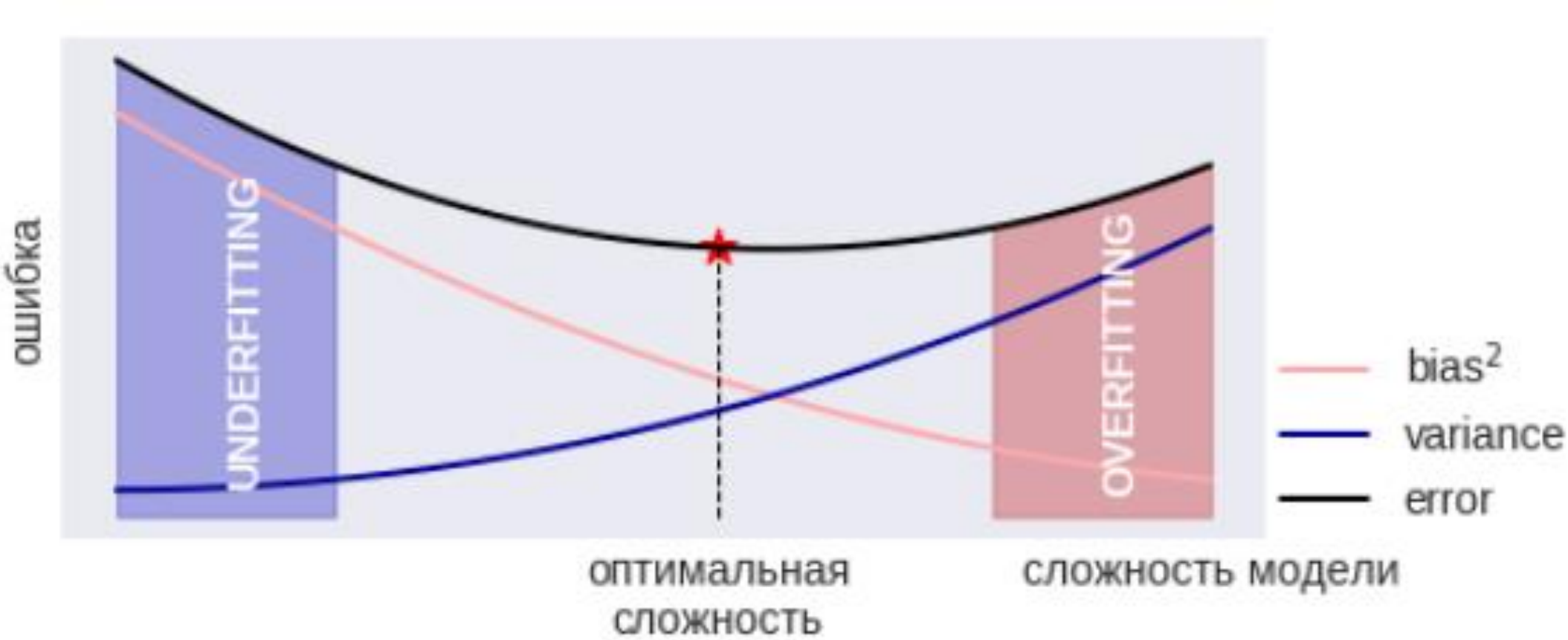
- Усреднение ответов деревья повышает качество. Почему?

Разложение ошибки классификации:

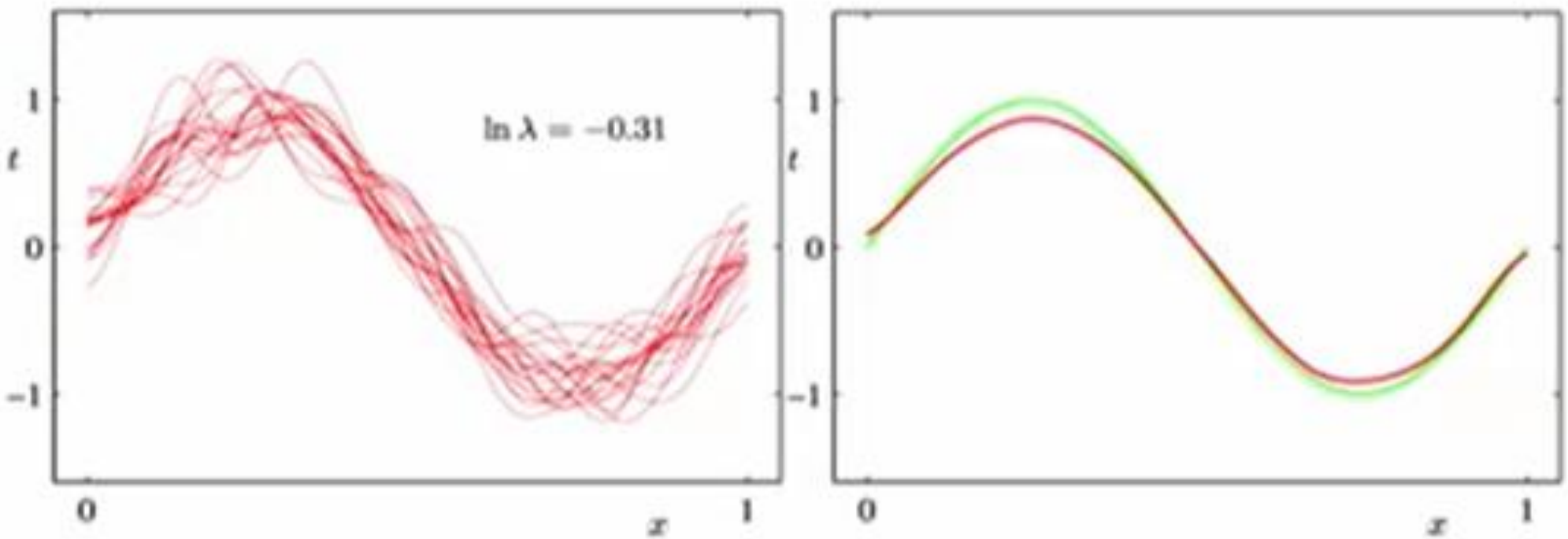
Ошибка на новых данных-=Шум+Смещение+разброс

- **Шум** - ошибка лучшей модели
- **Смещение**- отклонение средних ответов нашей модели ответов от ответов лучшей модели
- **Разброс-** дисперсия ответов наших моделей


Малое смещение
Большое смещение
Малый разброс
Большой разброс


ошибка
UNDERFITTING
OVERFITTING
bias^2
variance
error
оптимальная сложность
сложность модели

ПРИМЕР



РЕШАЮЩИЕ ДЕРЕВЬЯ

- Низкое смещение
- Большой разброс

УСРЕДНЕНИЕ АЛГОРИТМОВ

- Не меняет смещение
- Разброс = 1/N (разброс базового алгоритма) + (корреляция между базовыми алгоритмами)

- **Плюсы**:
  — имеет высокую точность предсказания, на большинстве задач будет лучше линейных алгоритмов; точность сравнима с точностью бустинга
  — практически не чувствителен к выбросам в данных из-за случайного сэмлирования
  — не чувствителен к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков, связано с выбором случайных подпространств
  — не требует тщательной настройки параметров,
- — способен эффективно обрабатывать данные с большим числом признаков и классов
  — одинаково хорошо обрабатывет как непрерывные, так и дискретные признаки
  — редко переобучается, на практике добавление деревьев почти всегда только улучшает композицию, но на валидации, после достижения определенного количества деревьев, кривая обучения выходит на асимптоту
  — для случайного леса существуют методы оценивания значимости отдельных признаков в модели
  — хорошо работает с пропущенными данными; сохраняет хорошую точность, если большая часть данных пропущена
  — предполагает возможность сбалансировать вес каждого класса на всей выборке, либо на подвыборке каждого дерева
  — вычисляет близость между парами объектов, которые могут использоваться при кластеризации, обнаружении выбросов или (путем масштабирования) дают интересные представления данных
  — возможности, описанные выше, могут быть расширены до неразмеченных данных, что приводит к возможности делать кластеризацию и визуализацию данных, обнаруживать выбросы
  — высокая параллелизуемость и масштабируемость.

- **Минусы**:
  — в отличие от одного дерева, результаты случайного леса сложнее интерпретировать
  — нет формальных выводов (p-values), доступных для оценки важности переменных
  — алгоритм склонен к переобучению на некоторых задачах, особенно на зашумленных данных
  — для данных, включающих категориальные переменные с различным количеством уровней, случайные леса предвзяты в пользу признаков с большим количеством уровней: когда у признака много уровней, дерево будет сильнее подстраиваться именно под эти признаки, так как на них можно получить более высокое значение оптимизируемого функционала (типа прироста информации)
  — если данные содержат группы коррелированных признаков, имеющих схожую значимость для меток, то предпочтение отдается небольшим группам перед большими
  — больший размер получающихся моделей.

- Преимуществом применения этих методов является то, что качество окончательного решения, полученного с помощью классификатора, обычно становится выше.
- Но эти результаты получены исключительно эмпирическим путем.
- Нет никакой гарантии того, что комбинированные классификаторы проявят себя как самые лучшие во время решения всей задачи в целом, хотя для закрытых множеств предложено доказательство, что полученные с их помощью результаты должны быть не хуже всех прочих.
- Затраты на создание комбинированных классификаторов возрастают линейно с увеличением количества используемых деревьев решений. Для хранения структур данных требуются дополнительные объемы памяти, а выработка каждого решения связана с дополнительными затратами на прохождение по каждому из деревьев

Случайные леса

+ наиболее универсальный
(~75% задач машинного обучения)

+ все типы задач
(классификация, регрессия, кластеризация)

+ настраивается сразу под все функционалы
(или можно преобразовать – не всегда надо)

+ нечувствителен к монотонным преобразованиям признаков
не совсем так...

+ легко реализуется
(лучшие реализации: R и Python)

Домашнее задание

1) Исследуйте зависимость тестовой ошибки от количества деревьев в ансамбле для алгоритма forest Random на наборе данных Organics. Постройте график зависимости тестовой ошибки при числе деревьев, равном 1, 11, 21, . . . , 301, объясните полученные результаты. Получить графики зависимостей от других параметров моделей.

- Пример дерева решений и критериев информативности
- https://logic.pdmi.ras.ru/~sergey/oldsite/teaching/ml/01-dectrees.pdf

- Оценка качества алгоритмов
- https://ru.coursera.org/lecture/supervised-learning/otsienivaniie-kachiestva-alghoritmov-sjbVd

- Спасибо за внимание!