Ансамбли.
Бэггинг

Разложение ошибки (Bias-Variance Decomposition)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели* $a(x)$ *можно представить в виде*

$$\boxed{\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}}$$

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели* $a(x)$ *можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2}.$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2}.$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.

Смещение показывает, насколько в среднем модель хорошо предсказывает целевую переменную:

✓ ***маленькое смещение - хорошее предсказание***

✓ ***большое смещение – плохое предсказание***

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели* $a(x)$ *можно представить в виде*

$$\boxed{\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}}$$

- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.

Разложение ошибки (Bias-Variance decomposition)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели* $a(x)$ *можно представить в виде*

$$\boxed{\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2}.}$$

- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.

Большой разброс означает, что ошибка очень чувствительна к изменению обучающей выборки, т.е.:

✓ ***большой разброс – сильно переобученная модель***

Разложение ошибки (bias-variance decomposition)

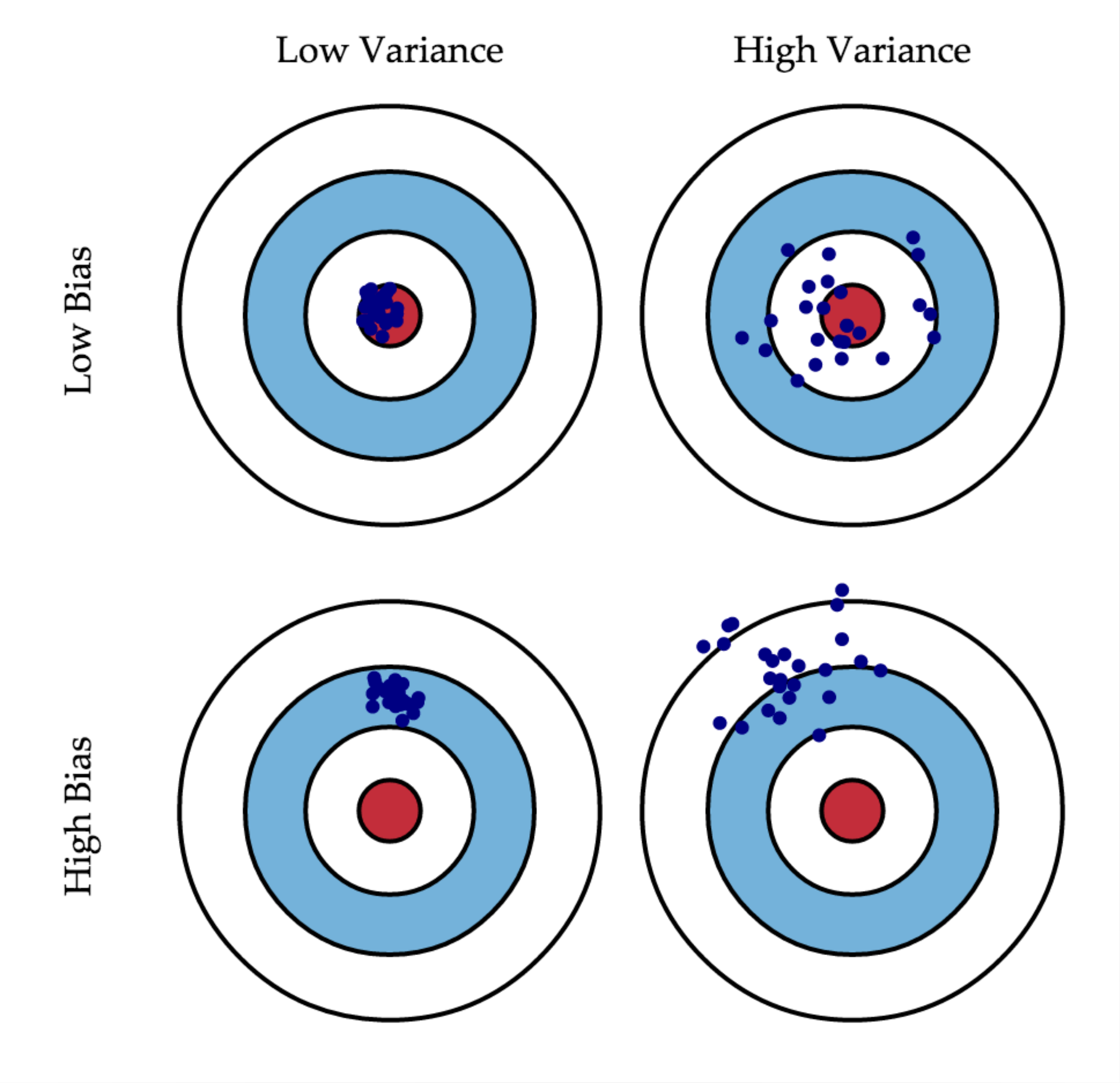
Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

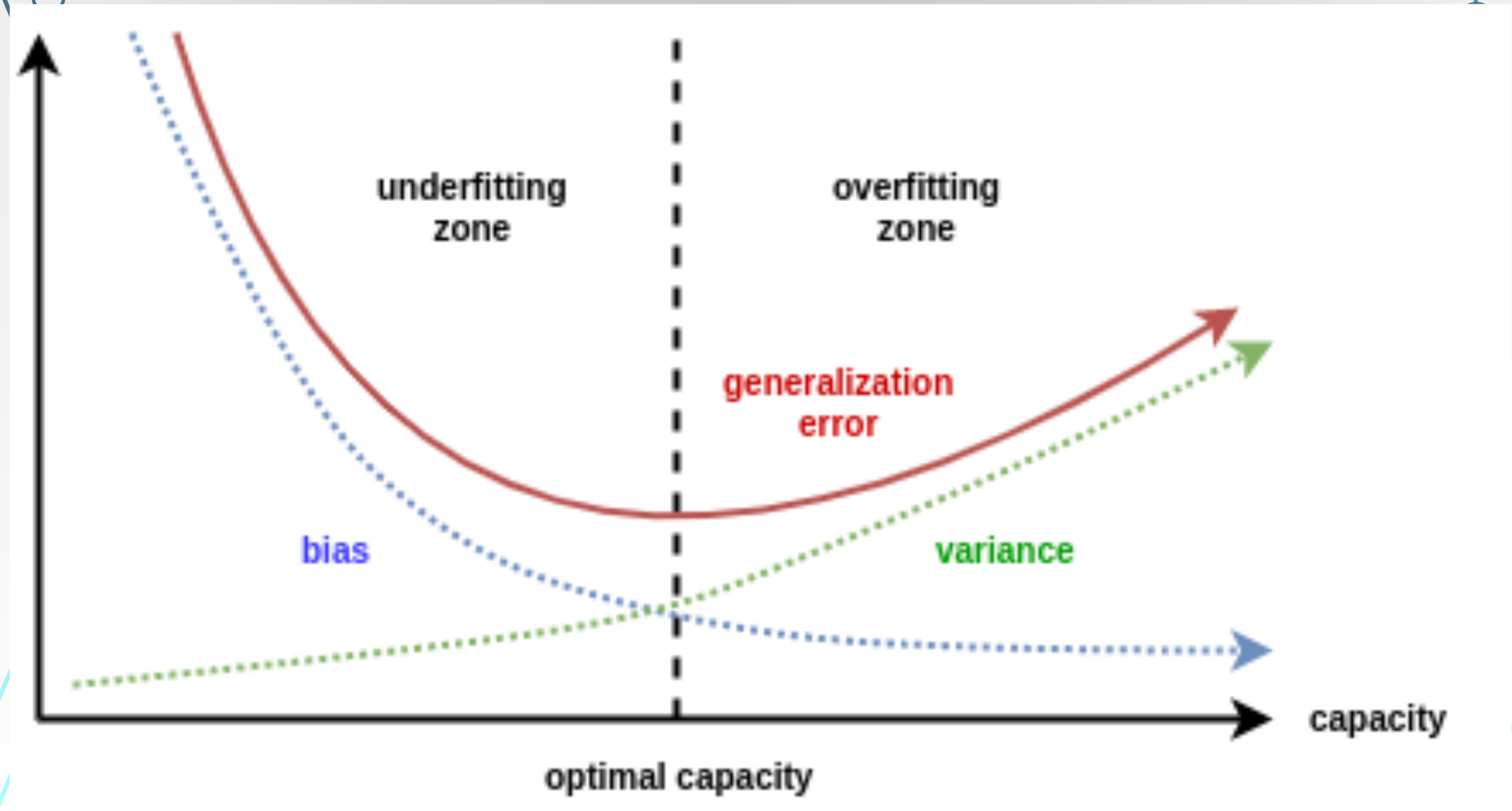
$$\boxed{\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2}.}$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.
- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.
- $\boldsymbol{\sigma}^\mathbf{2}$ - неустранимая ошибка – **шум**.

СМЕЩЕНИЕ И РАЗБРОС



BIAS-VARIANCE TRADEOFF

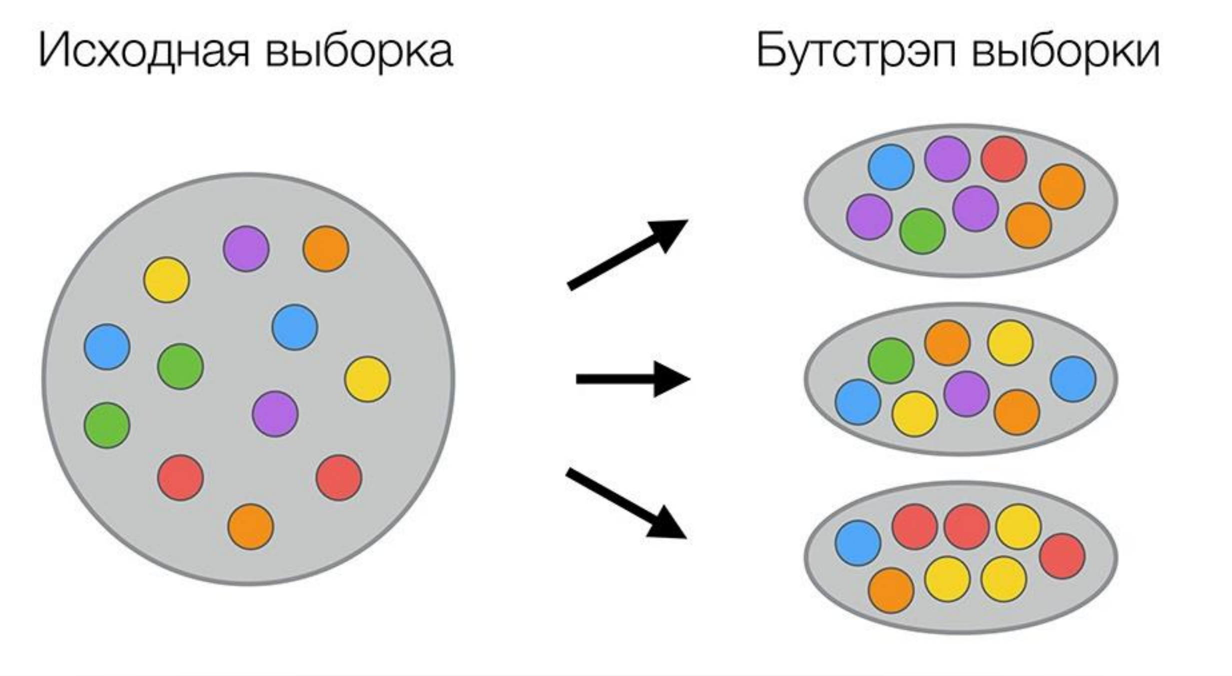


БУТСТРЭП

Дана выборка X.

Бутстрэп: равномерно возьмем из выборки X l объектов с возвращением (т.е. в новой выборке будут повторяющиеся объекты). Получим выборку X_1.

- Повторяем процедуру N раз, получаем выборки $X_1, \dots, X_N$.



БЭГГИНГ (BOOTSTRAP AGGREGATION)

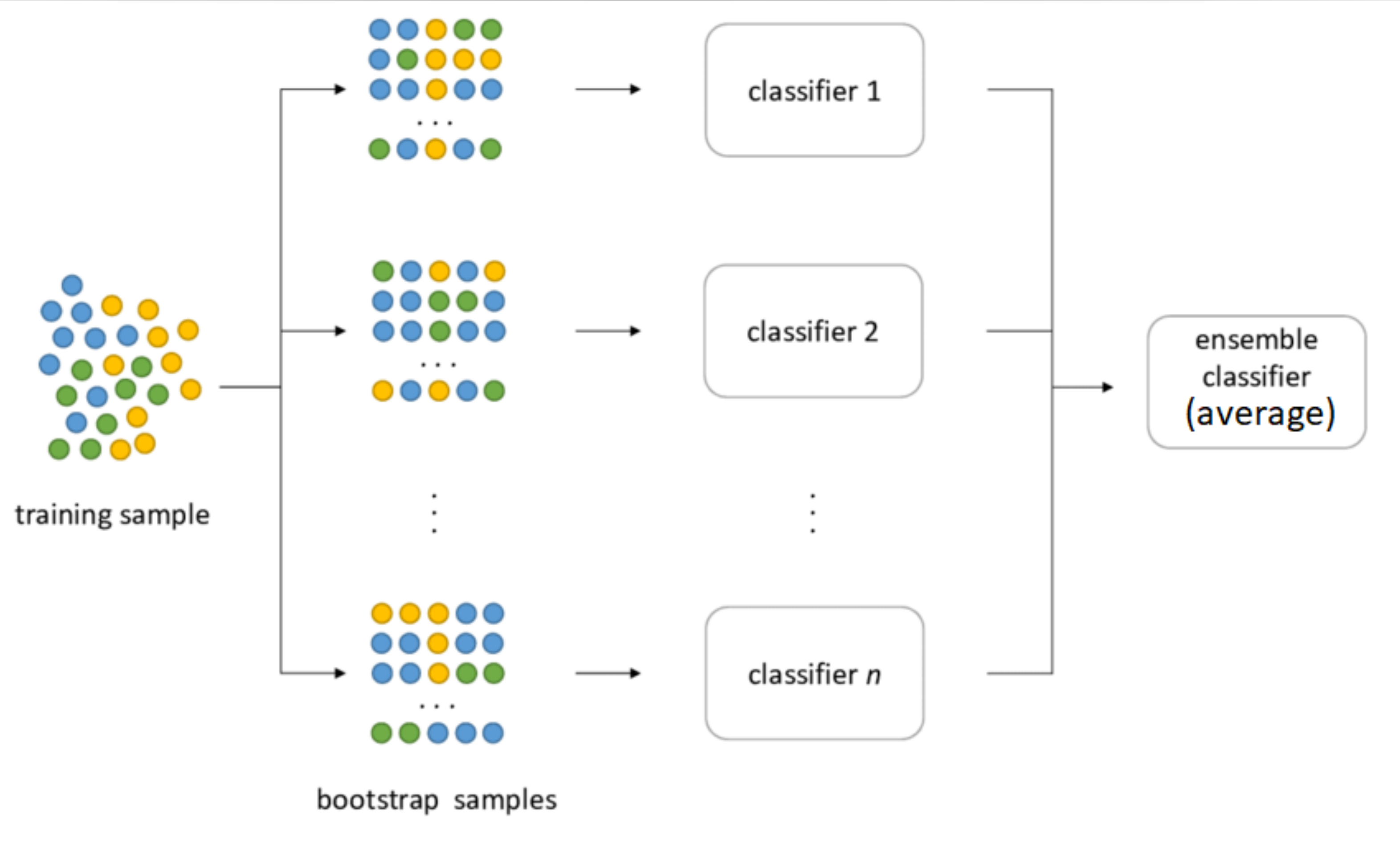
С помощью бутстрэпа мы получили выборки $X_1, \dots, X_N$.

- Обучим по каждой из них модель – получим базовые алгоритмы $b_1(x), \dots, b_N(x)$.
- Построим новую функцию регрессии:

$$\boldsymbol{a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)}$$

БЭГГИНГ (BOOTSTRAP AGGREGATION)

$$a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)$$



РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

- *Модель переобучена?*
- *Модель плохо предсказывает целевую переменную?*
- *В самих данных много неточностей (шумов)*

СМЕЩЕНИЕ И РАЗБРОС У БЭГГИНГА

Бэггинг: $a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x) = \frac{1}{N}\sum_{n=1}^{N} \tilde{\mu}(X)\,(x)$

(здесь $\tilde{\mu}(X) = \mu(\tilde{X})$ – алгоритм, обученный на подвыборке $\tilde{X}$)

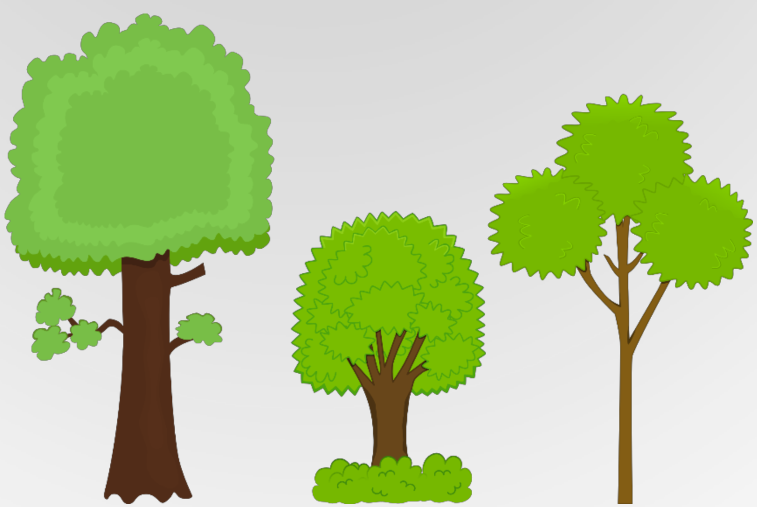
Утверждение (с док-вом):

1) ***Бэггинг не ухудшает смещенность модели***, *т.е. смещение $a_N(x)$ равно смещению одного базового алгоритма.*

2) *Если базовые алгоритмы некоррелированы, то* ***дисперсия бэггинга $a_N(x)$ в N раз меньше дисперсии отдельных базовых алгоритмов.***

СЛУЧАЙНЫЙ ЛЕС (RANDOM FOREST)

- Возьмем в качестве базовых алгоритмов для бэггинга ***решающие деревья***, т.е. каждое случайное дерево $b_i(x)$ построено по своей подвыборке X_i.
- В каждой вершине дерева будем искать ***разбиение не по всем признакам, а по подмножеству признаков***.
- Дерево строится до тех пор, пока в листе не окажется n_{min} объектов.



RANDOM FOREST

Алгоритм 3.1. Random Forest

1: **для** $n = 1, \dots, N$

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

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

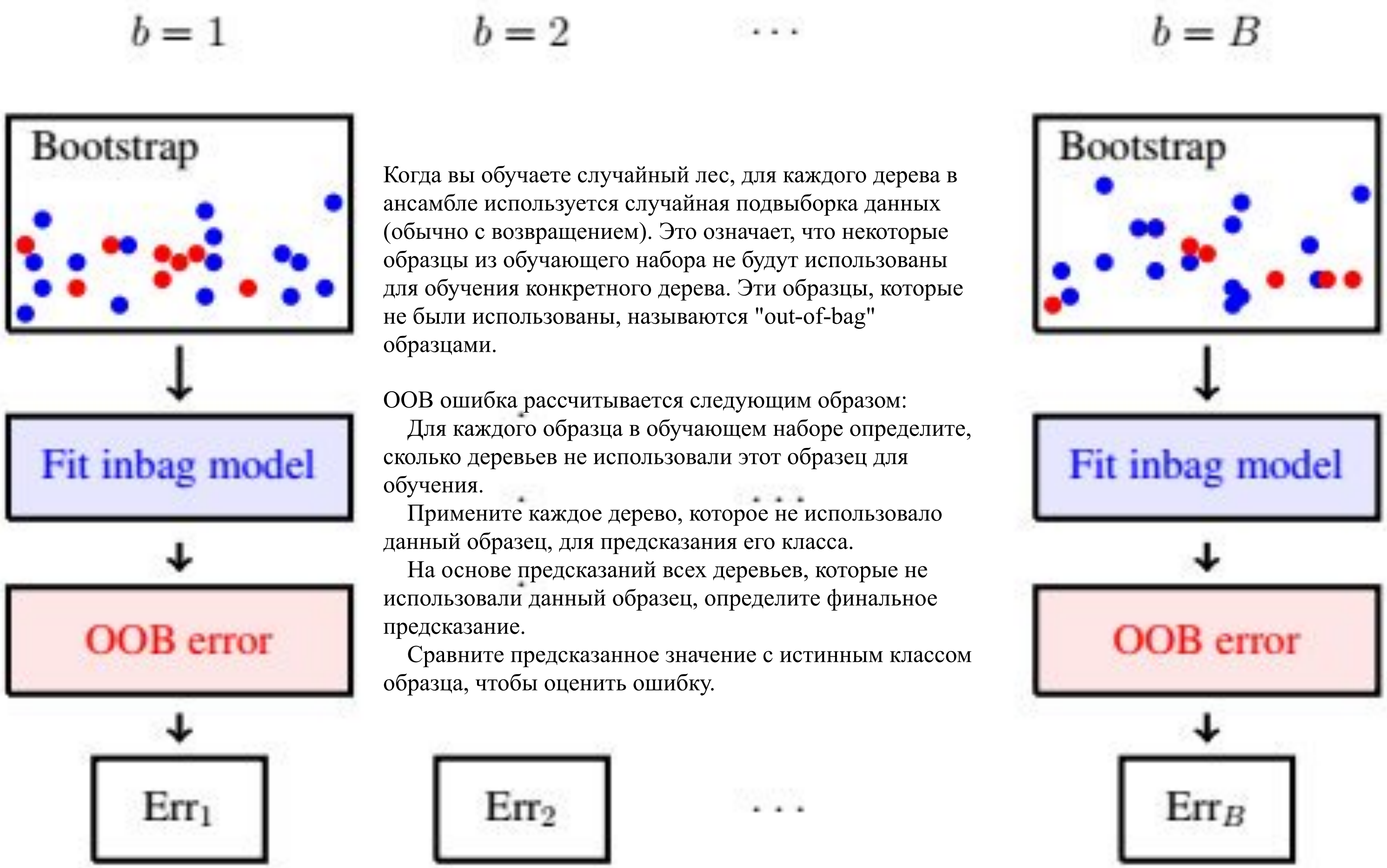
- дерево строится, пока в каждом листе не окажется не более $n_{\min}$ объектов
- при каждом разбиении сначала выбирается m случайных признаков из p, и оптимальное разделение ищется только среди них

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

RANDOM FOREST – ПРАКТИЧЕСКИЕ РЕКОМЕНДАЦИИ

- Если p – количество признаков, то при классификации обычно берут $m = [\sqrt{p}]$, а при регрессии - $m = [\frac{p}{3}]$ признаков
- При классификации обычно дерево строится, пока в листе не окажется $n_{min} = 1$ объект, а при регрессии $n_{min} = 5$

OUT-OF-BAG ОШИБКА



Когда вы обучаете случайный лес, для каждого дерева в ансамбле используется случайная подвыборка данных (обычно с возвращением). Это означает, что некоторые образцы из обучающего набора не будут использованы для обучения конкретного дерева. Эти образцы, которые не были использованы, называются "out-of-bag" образцами.

ООВ ошибка рассчитывается следующим образом:

Для каждого образца в обучающем наборе определите, сколько деревьев не использовали этот образец для обучения.

Примените каждое дерево, которое не использовало данный образец, для предсказания его класса.

На основе предсказаний всех деревьев, которые не использовали данный образец, определите финальное предсказание.

Сравните предсказанное значение с истинным классом образца, чтобы оценить ошибку.

$$\mathrm{Err}_{\mathrm{oob}} = \frac{\mathrm{Err}_1 + \cdots + \mathrm{Err}_B}{B} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{Err}_b$$

OUT-OF-BAG ОШИБКА

- Каждое дерево в случайном лесе обучается по некоторому подмножеству объектов
- Значит, для каждого объекта есть деревья, которые на этом объекте не обучались.

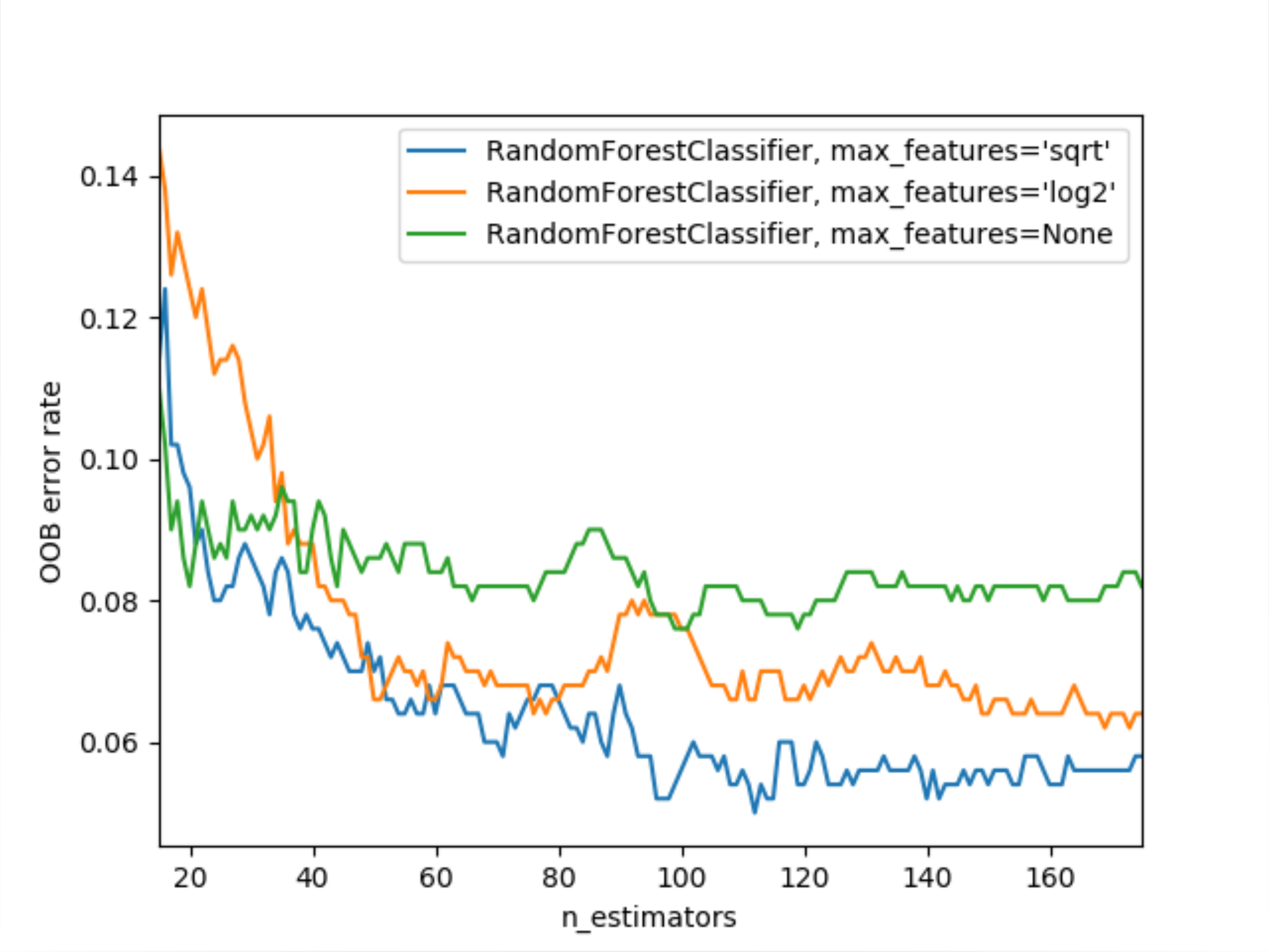
Out-of-bag ошибка:

$$OOB = \sum_{i=1}^{l} L\left(y_i, \frac{\sum_{n=1}^{N} [x_i \notin X_n]\, b_n(x_i)}{\sum_{n=1}^{N} [x_i \notin X_n]}\right)$$

Утверждение. *При $N \to \infty$ OOB оценка стремится к leave-one-out оценке.*

OOB-SCORE

- По графику out-of-bag ошибки можно, например, подбирать количество деревьев в случайном лесе



КАЛИБРОВКА ВЕРОЯТНОСТЕЙ

Калибровка вероятностей - приведение ответов алгоритма к значениям, близким к вероятностям объектов принадлежать конкретному классу.

Зачем это нужно?

- Вероятности гораздо проще интерпретировать
- Вероятности могут дать дополнительную информацию о результатах работы алгоритма

КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

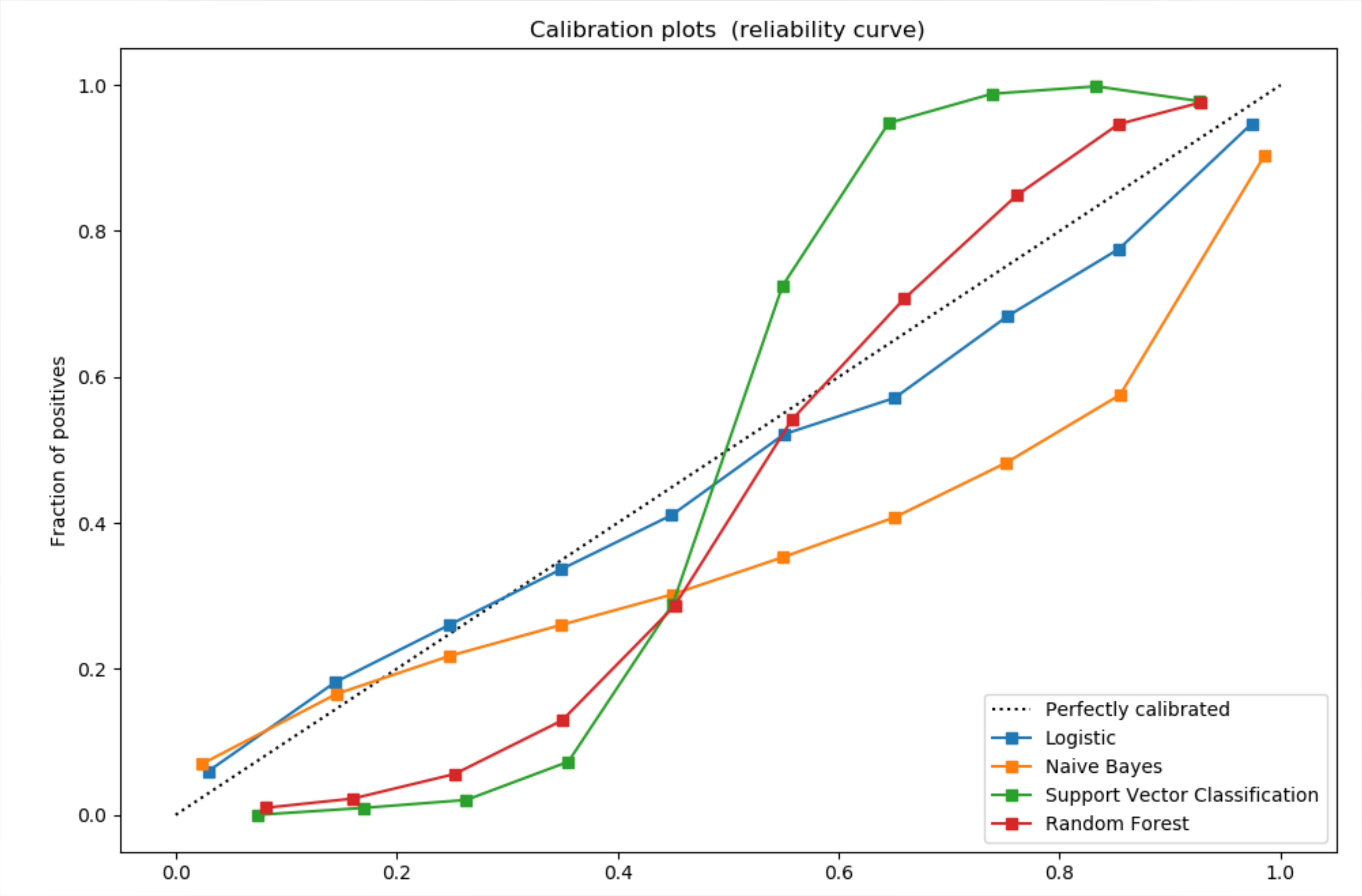
КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

Идея: обучаем логистическую регрессию на ответах классификатора $a(x)$.

ПРИМЕР ИЗ SKLEARN



КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

Идея: ***обучаем логистическую регрессию на ответах классификатора $a(x)$***.

КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

Идея: ***обучаем логистическую регрессию на ответах классификатора $a(x)$***.

- $\pi(x; \alpha; \beta) = \sigma(\alpha \cdot \boldsymbol{a(x)} + \beta) = \frac{1}{1+e^{-(\alpha \cdot a(x)+\beta)}}$

КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

Идея: ***обучаем логистическую регрессию на ответах классификатора $a(x)$***.

- $\pi(x; \alpha; \beta) = \sigma(\alpha \cdot a(x) + \beta) = \frac{1}{1+e^{-(\alpha \cdot a(x)+\beta)}}$

- Находим α и β, минимизируя логистическую функцию потерь (*то есть обучаем логистическую регрессию*):

$$-\sum_{y_i=-1} \log\big(1 - \pi(x; \alpha; \beta)\big) - \sum_{y_i=+1} \log\big(\pi(x; \alpha; \beta)\big) \to \min_{\alpha,\beta}$$