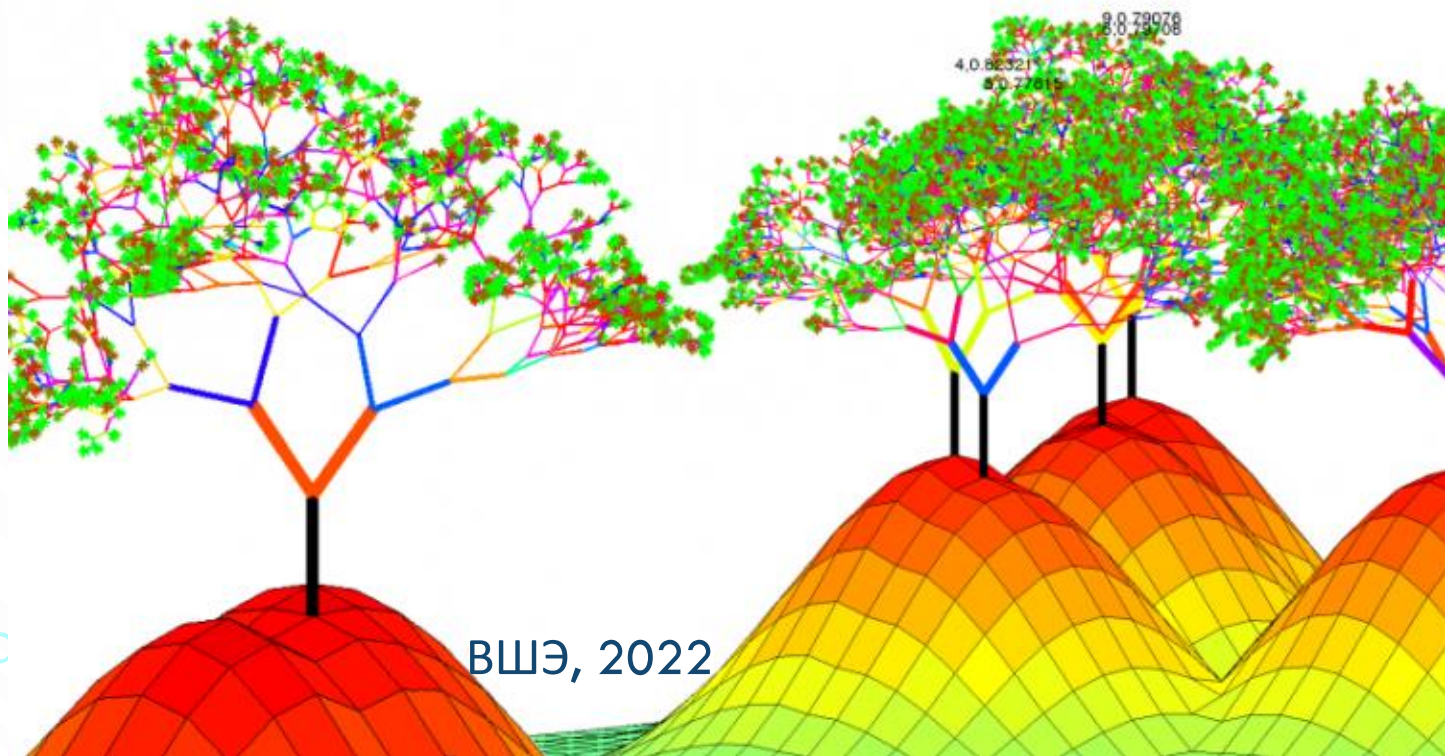


Лекция 12

Композиции алгоритмов. Часть 1.

Кантонистова Е.О.



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

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

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

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

- **$\text{Bias}(a(x))$** - 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

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

- **$\text{Bias}(a(x))$** - 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

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

- **$\text{Var}(a(x))$** - 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

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

- $\text{Var}(a(x))$ - 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)

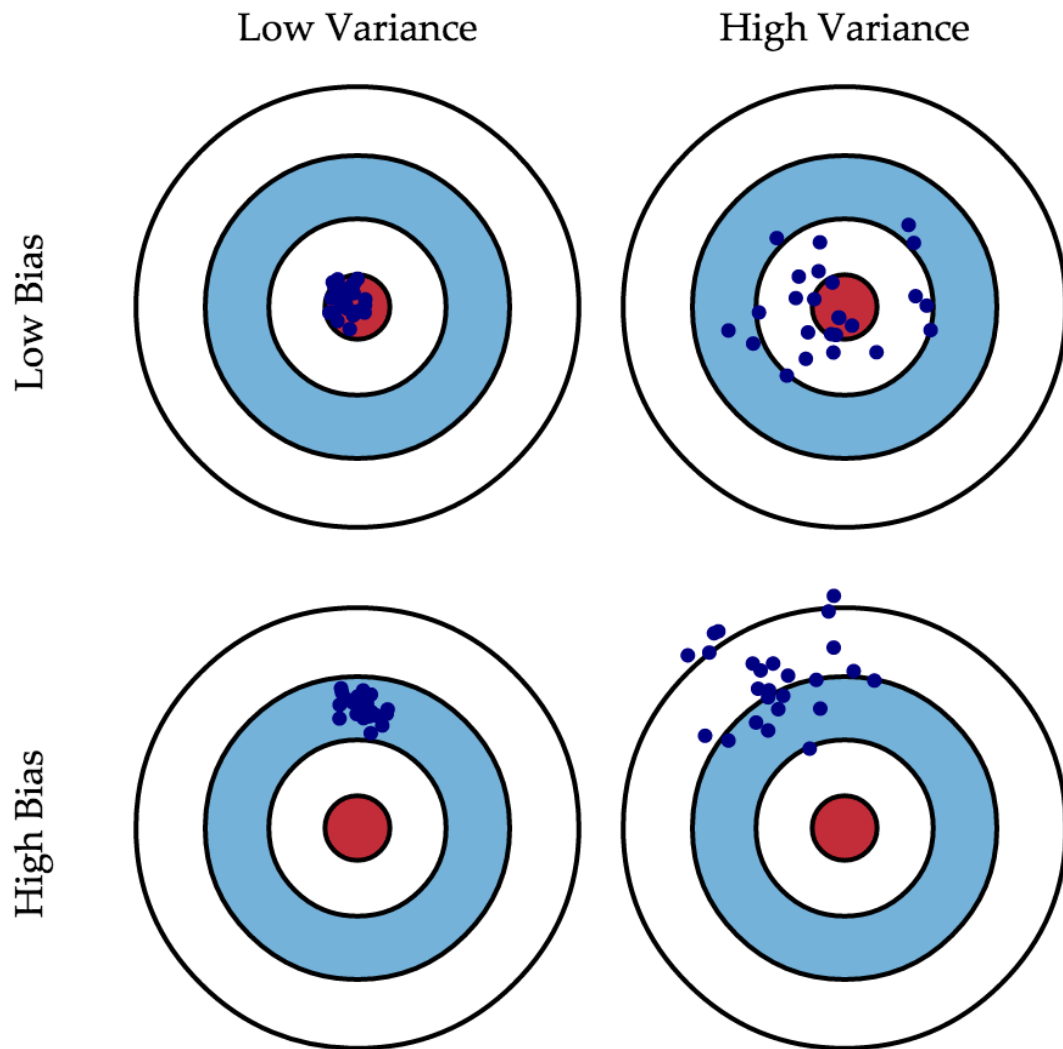
Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

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

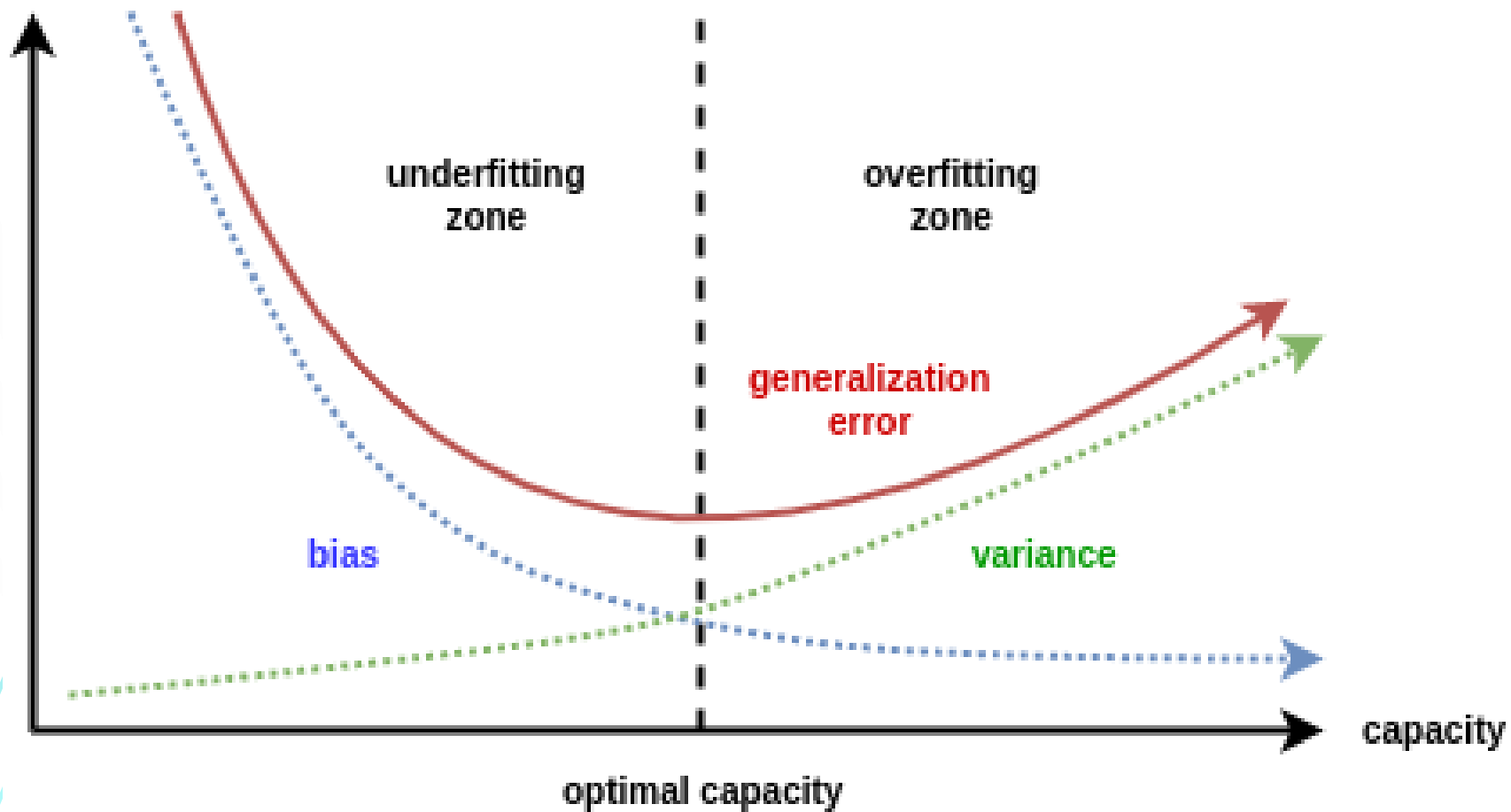
$$\text{Err}(x) = \text{Bias}^2(a(x)) + \text{Var}(a(x)) + \sigma^2.$$

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

СМЕЩЕНИЕ И РАЗБРОС



BIAS-VARIANCE TRADEOFF



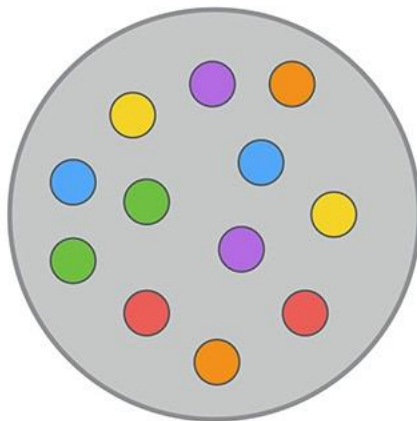
БУТСТРЭП

Дана выборка X .

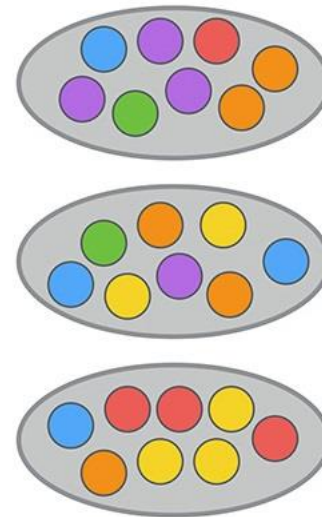
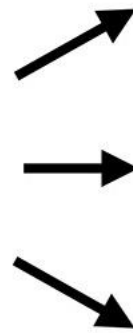
Бутстрэп: равномерно возьмем из выборки X l объектов с возвращением (т.е. в новой выборке будут повторяющиеся объекты). Получим выборку X_1 .

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

Исходная выборка



Бутстрэп выборки



БЭГГИНГ (BOOTSTRAP AGGREGATION)

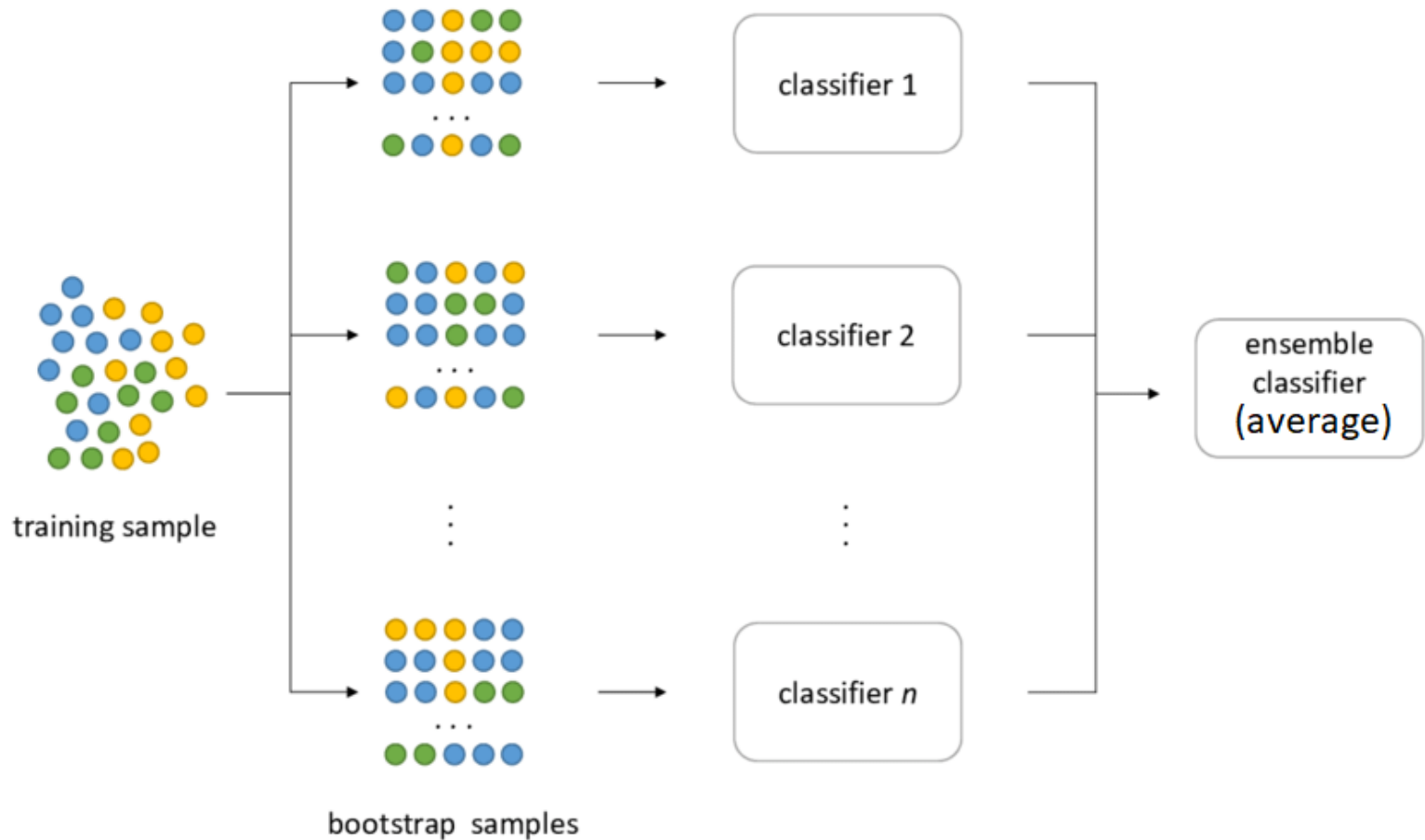
С помощью бутстрэпа мы получили выборки X_1, \dots, X_N .

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

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

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})

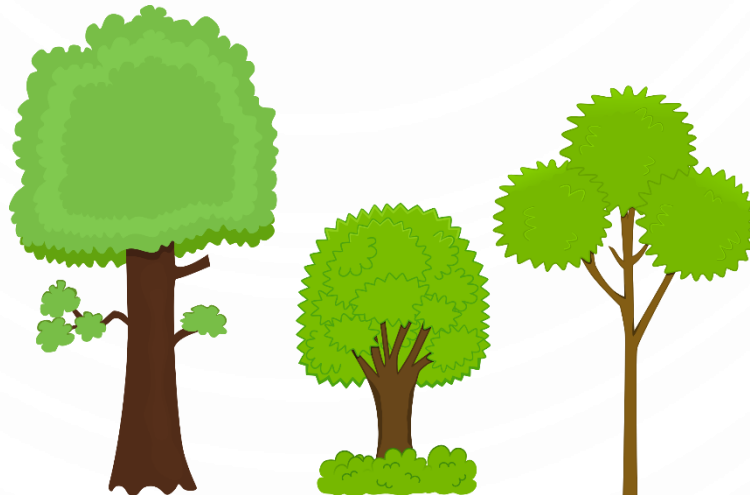
Утверждение (с док-вом):

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

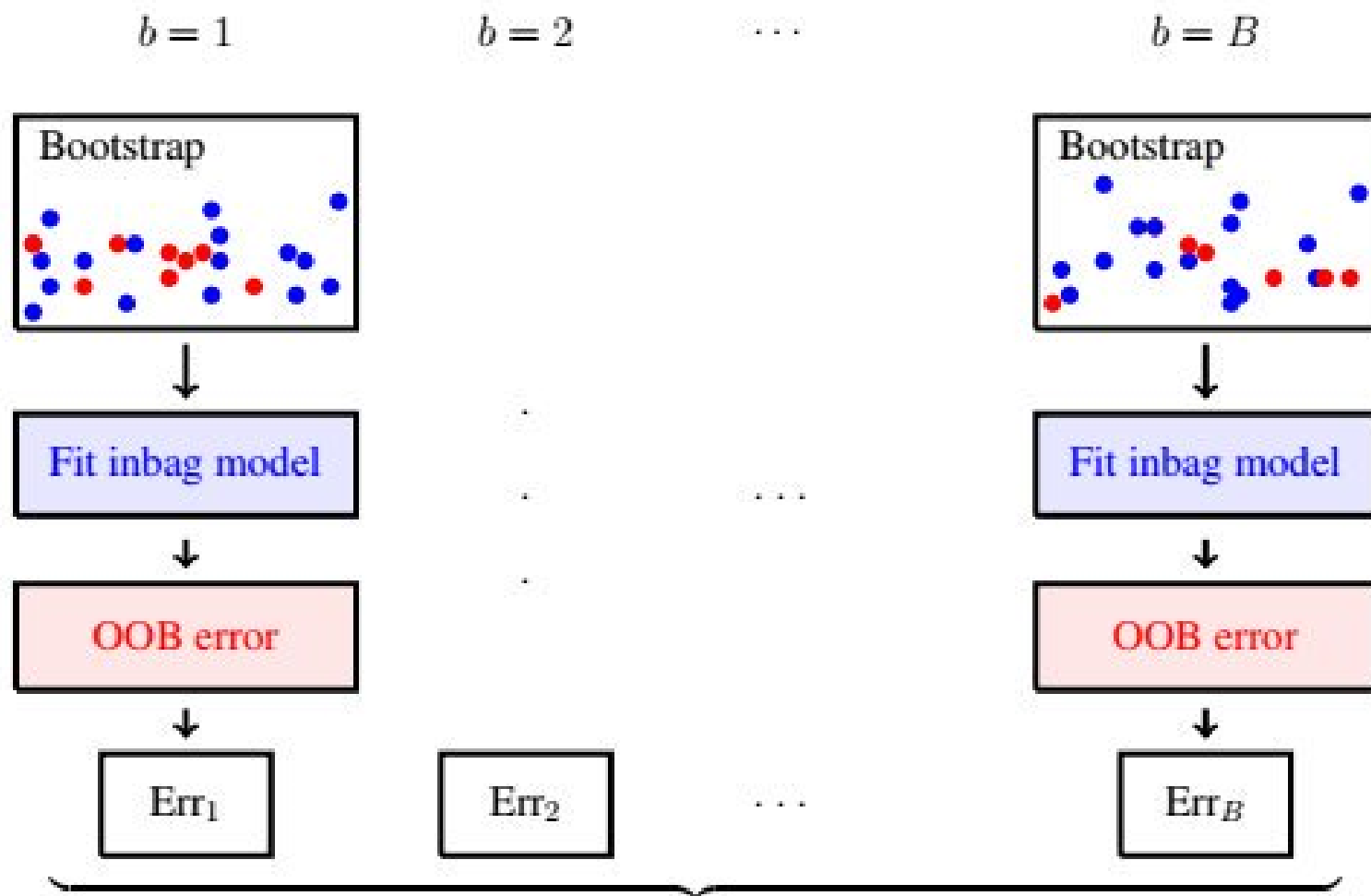
Алгоритм 3.1. Random Forest

- 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 $m = \lfloor \sqrt{p} \rfloor$, а при регрессии - $m = \lfloor \frac{p}{3} \rfloor$ 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



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.

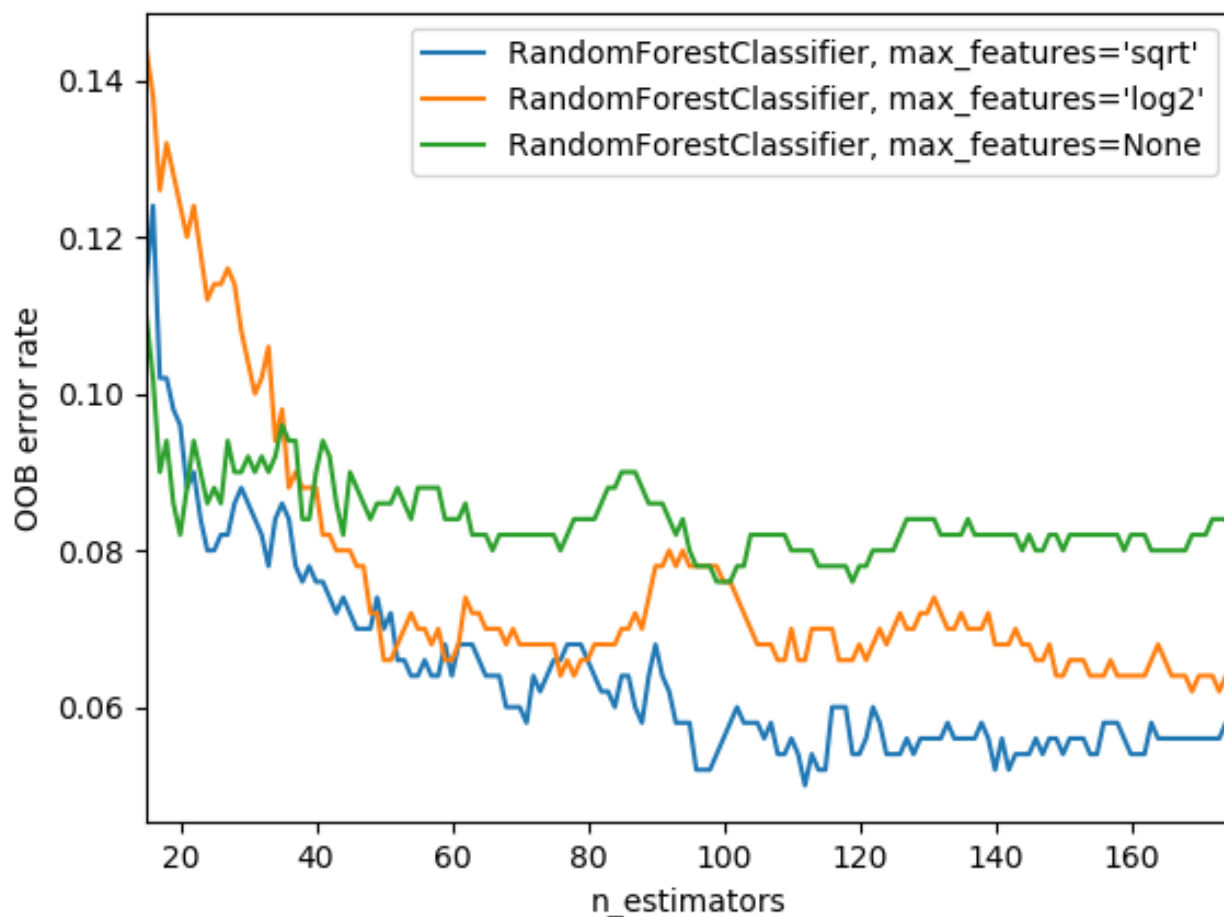
Out-of-bag ошибка:

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

Утверждение. При $N \rightarrow \infty$ 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 $p(y = +1|x)$.

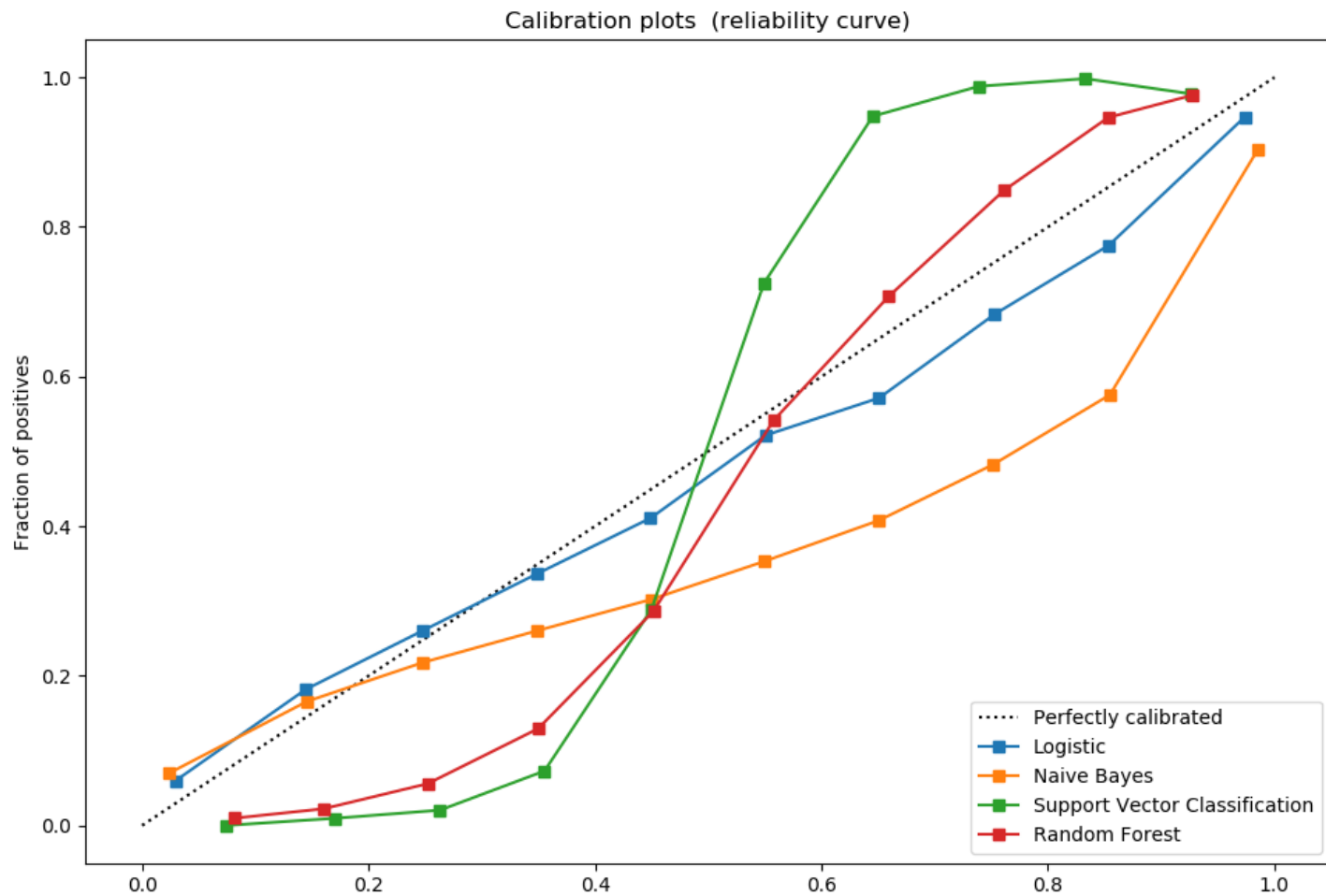
КАЛИБРОВКА ПЛАТТА

- 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 $a(x)$.*

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

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*):

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