

Машинное обучение

Лекция 7

Композиции алгоритмов. Бэггинг.

Михаил Гуцин

mhushchyn@hse.ru

НИУ ВШЭ, 2025



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Эволюция решающих деревьев



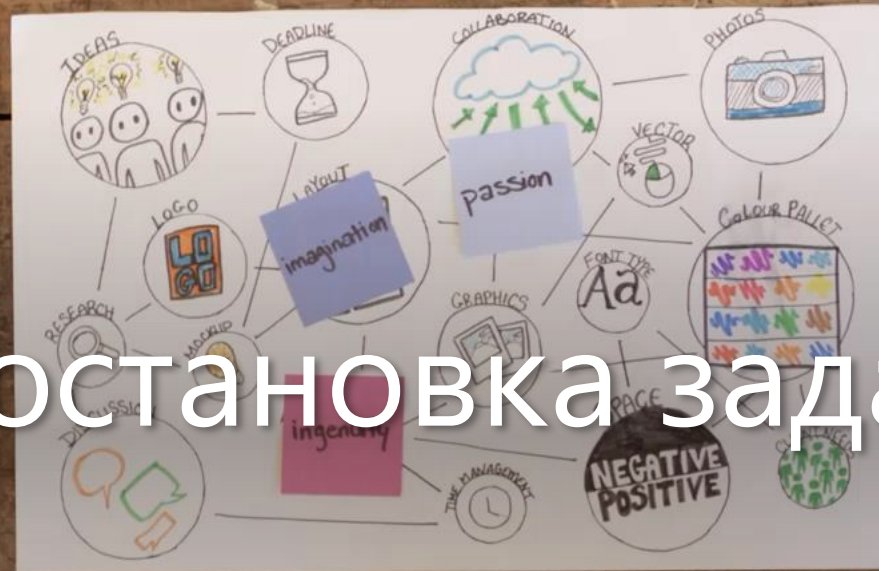
Решающие
деревья



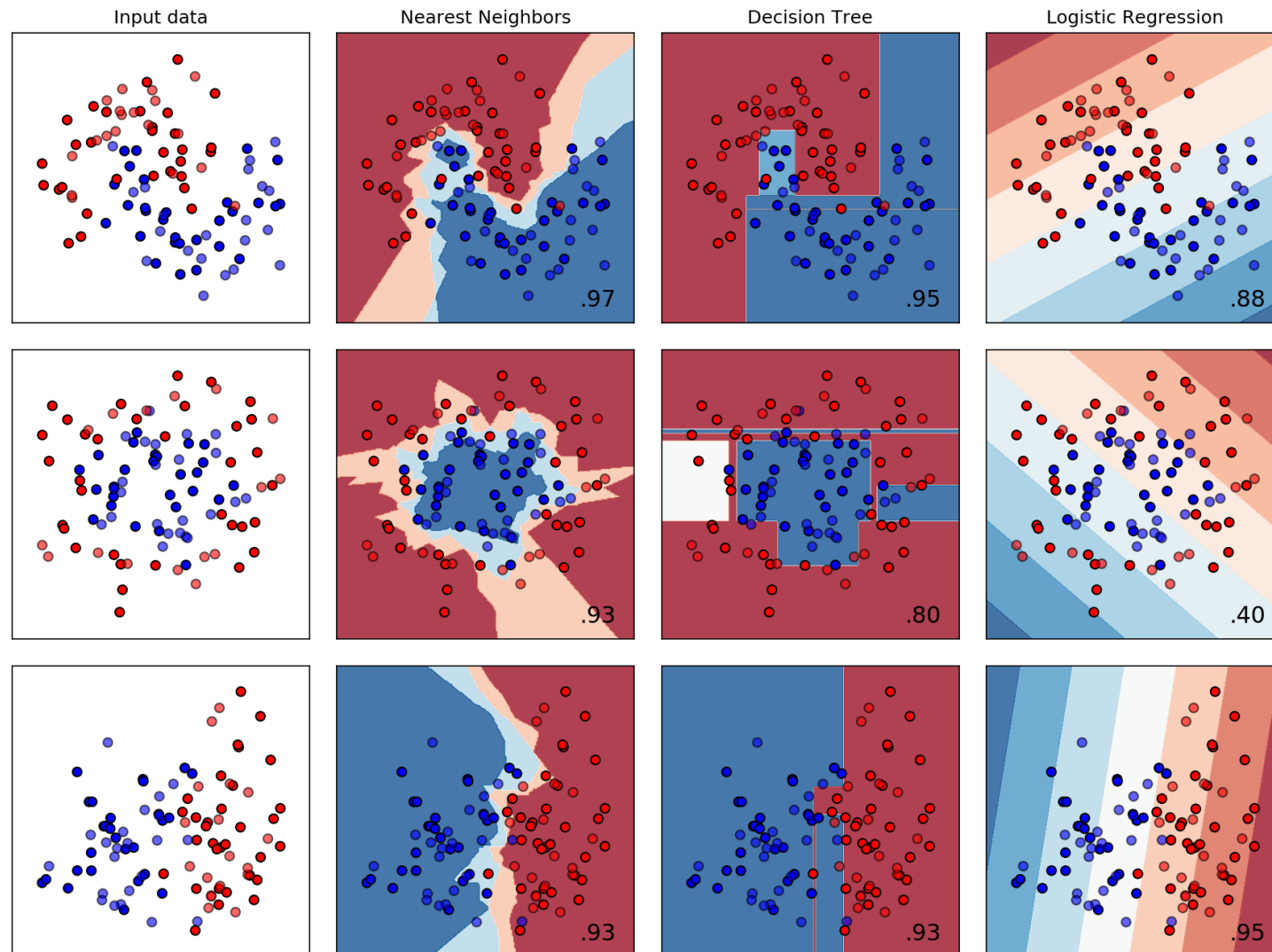
Случайный лес
решающих
деревьев



Постановка задачи



Классификаторы



Задача

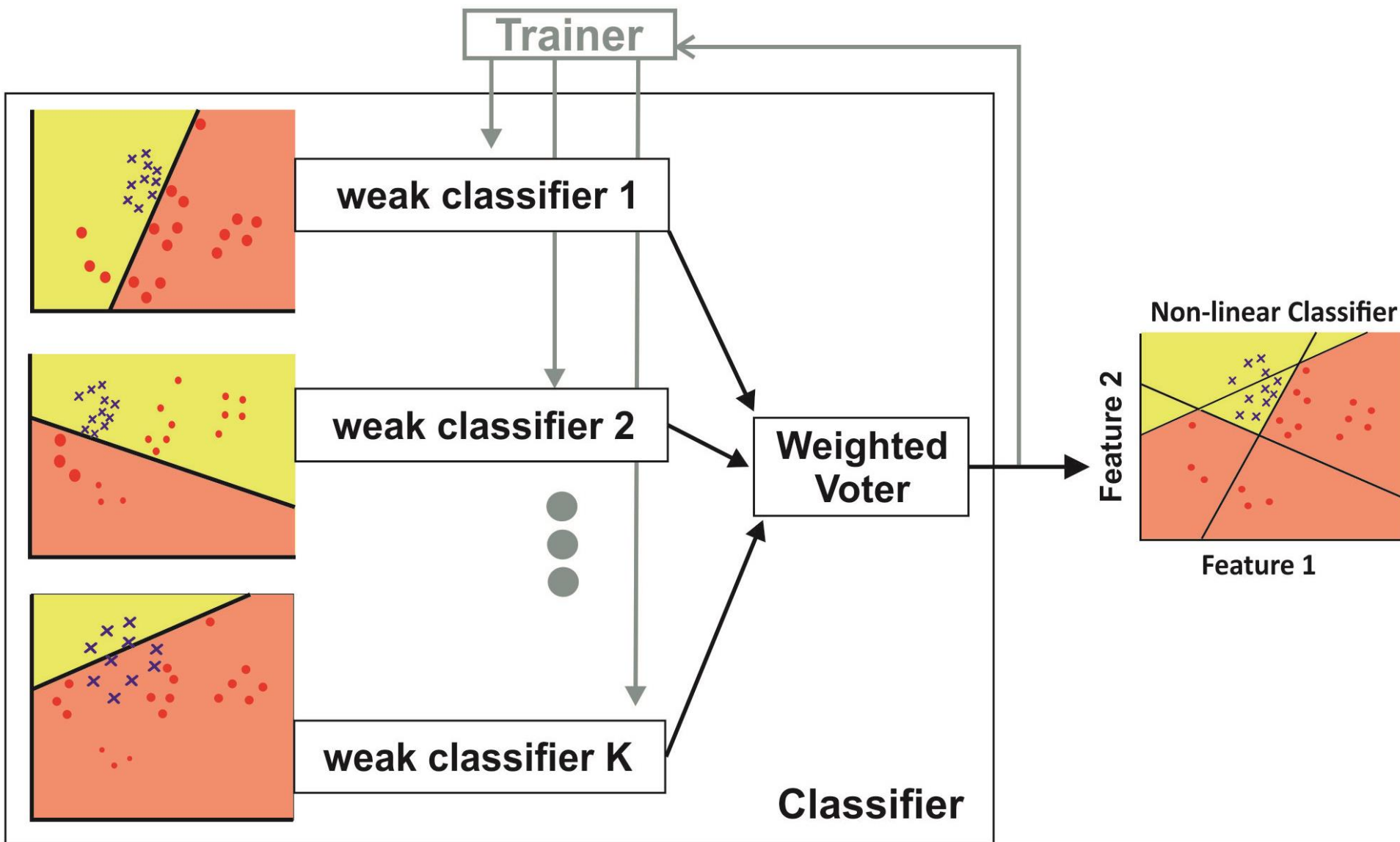
▶ Нам дано:

- Набор данных X, y
- Несколько классификаторов

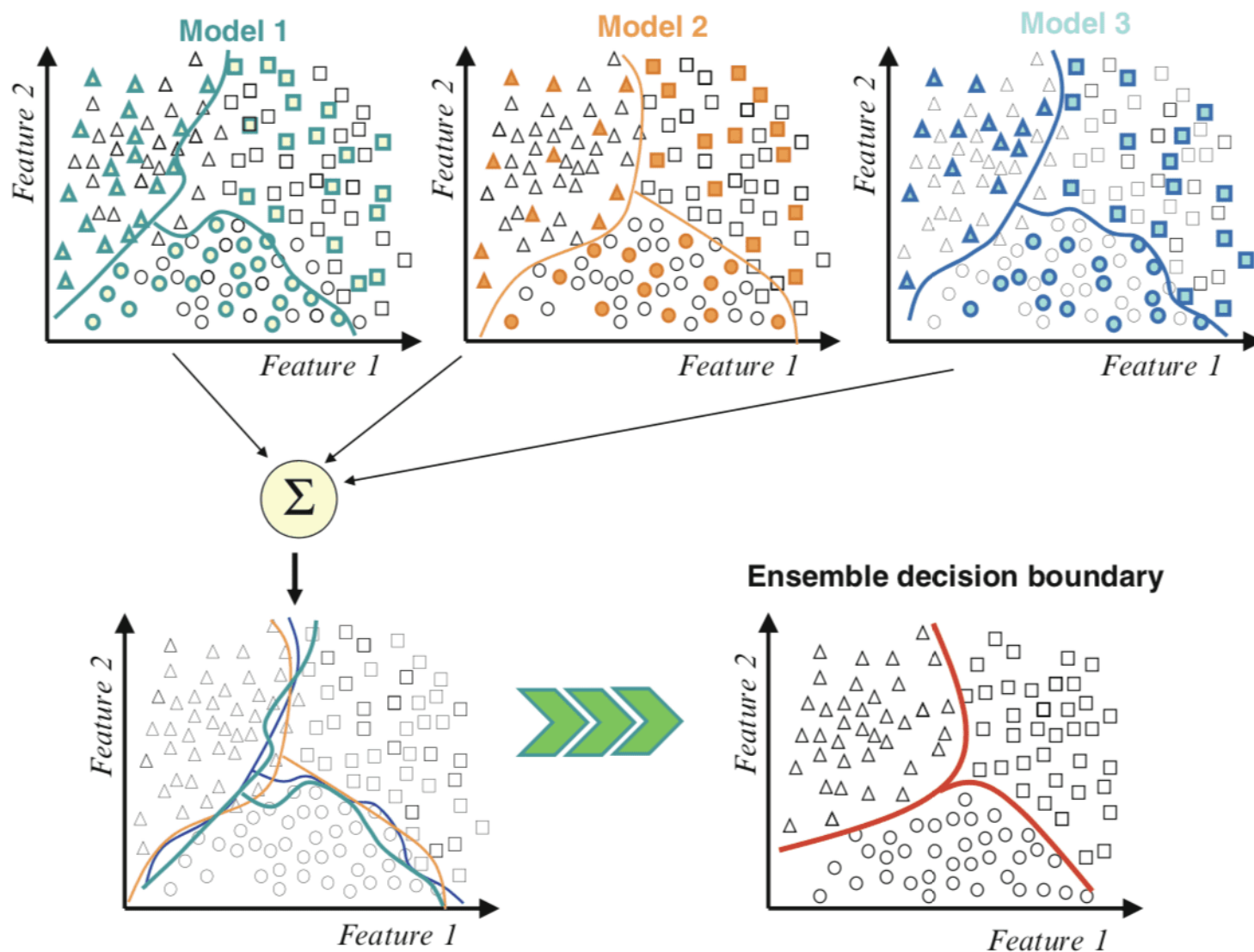
▶ Наши задачи:

- Как улучшить прогнозы классификаторов?
- Можно ли объединить разные классификаторы в одну модель?

Пример



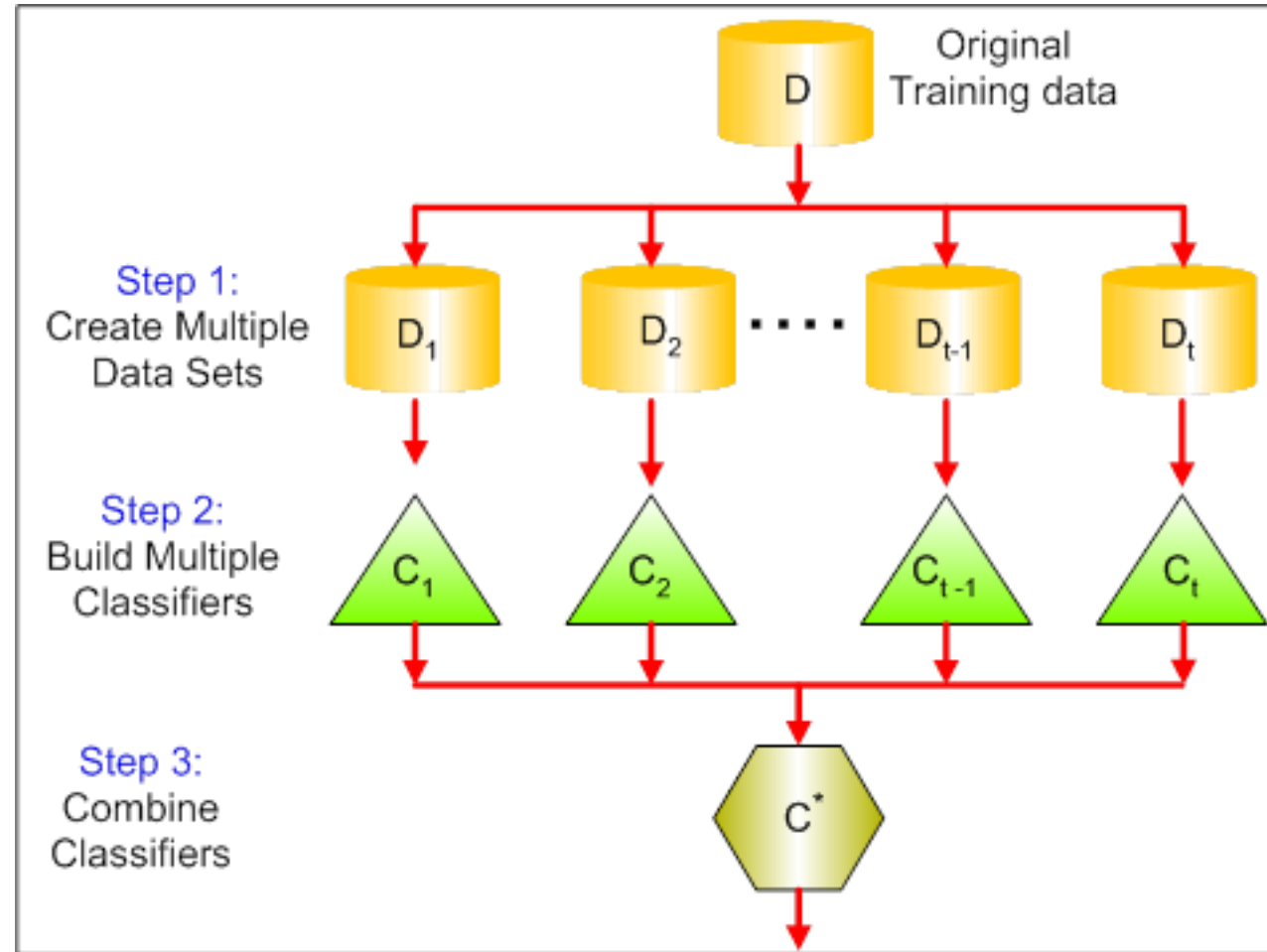
Пример





Бэггинг (bagging)

Бэггинг (bagging)



Подвыборки

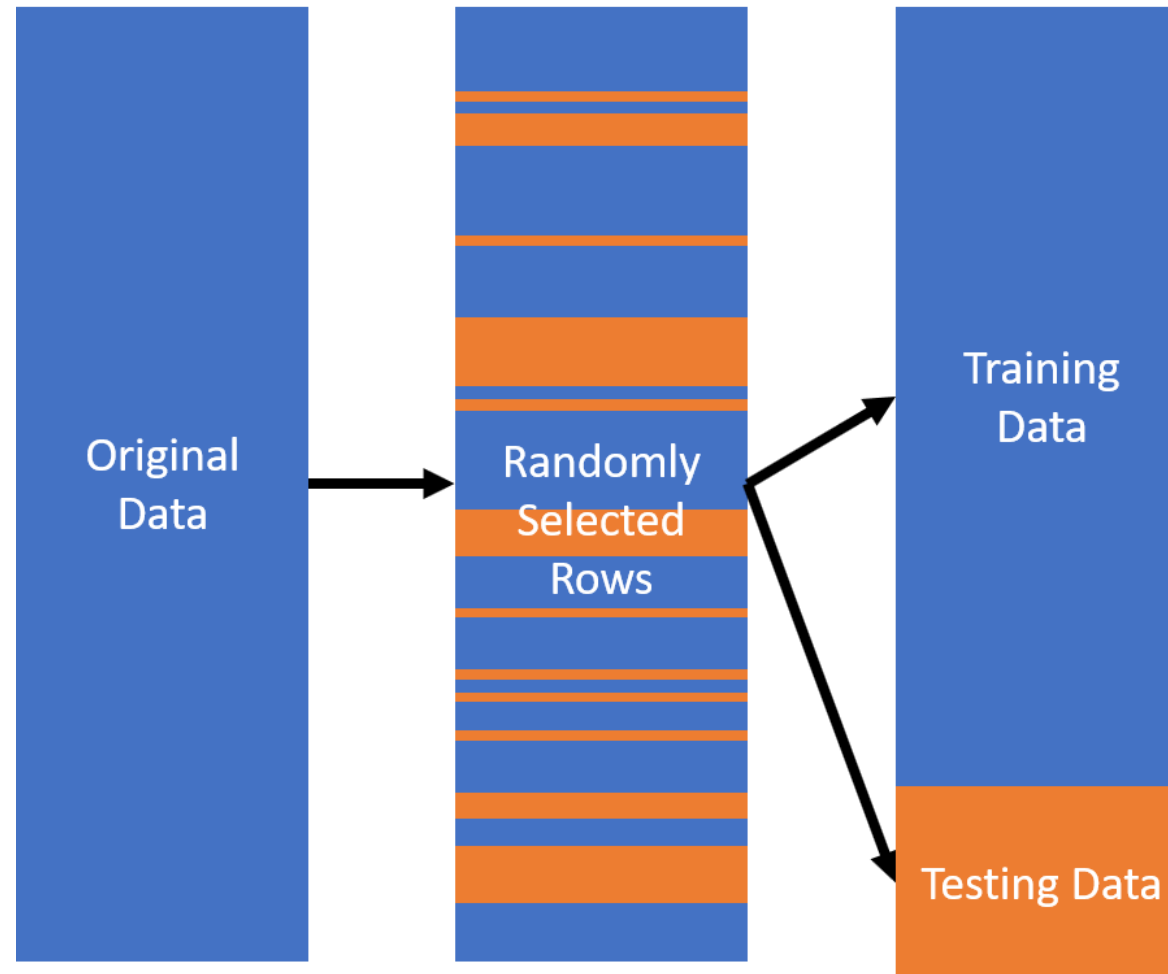
- ▶ Как создать разные подвыборки имеющихся данных?

Два наиболее популярных способа:

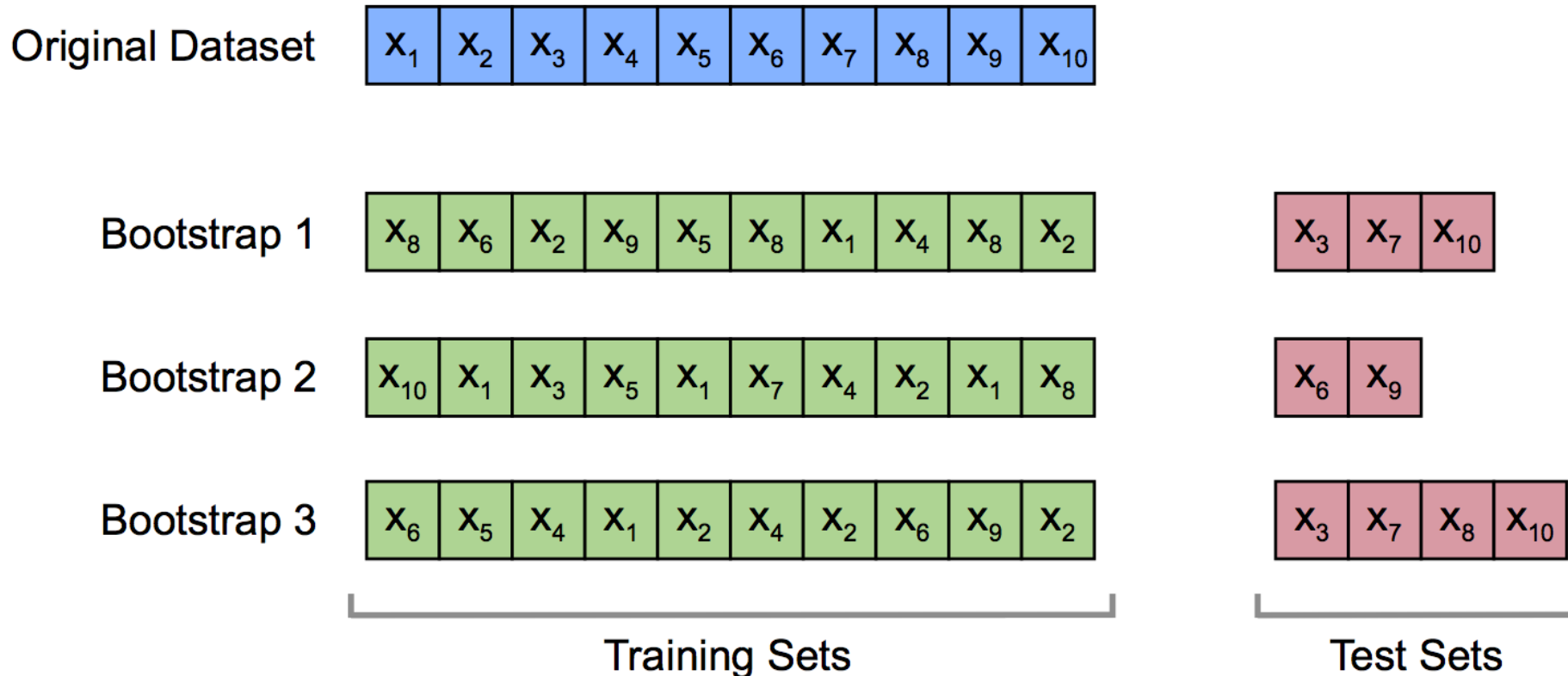
- ▶ Случайная подвыборка без повторений (train-test split)
 - Случайным образом выбираем объекты из всей выборки
 - Размер подвыборки меньше самой выборки
- ▶ Бутстрап
 - Случайная подвыборка с повторениями
 - Размер подвыборки совпадает с размером всей выборки

Случайная подвыборка

Splitting Data for Machine Learning



Бутстрап (bootstrap)



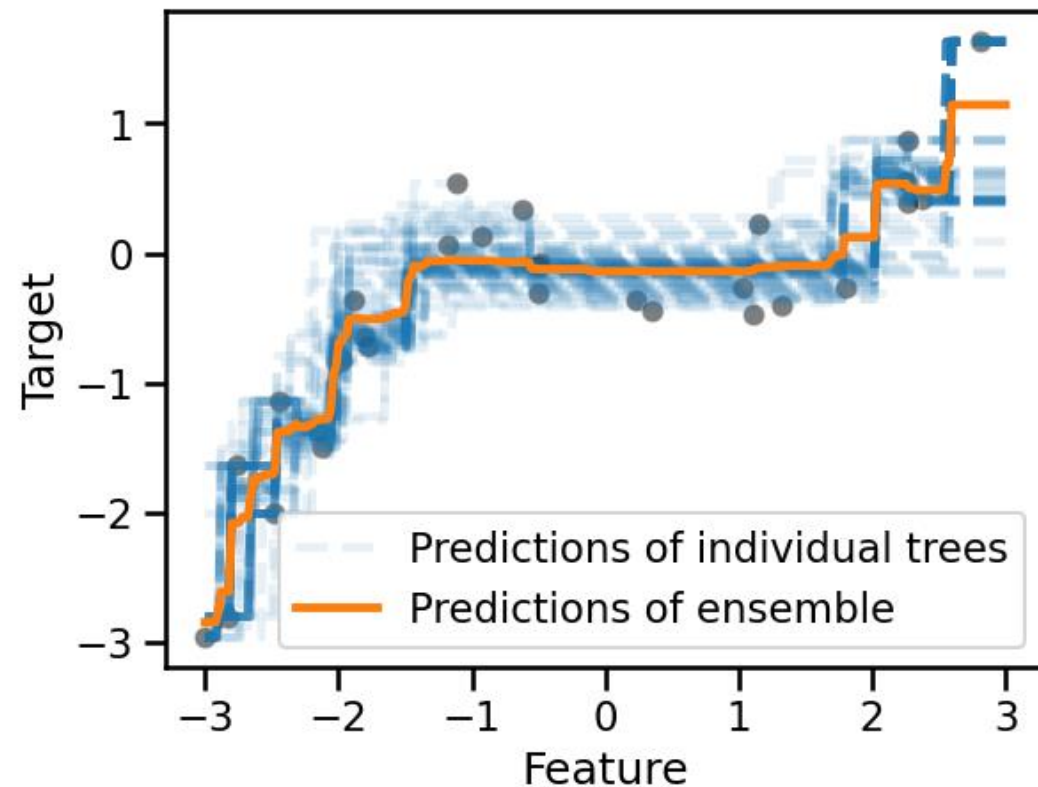
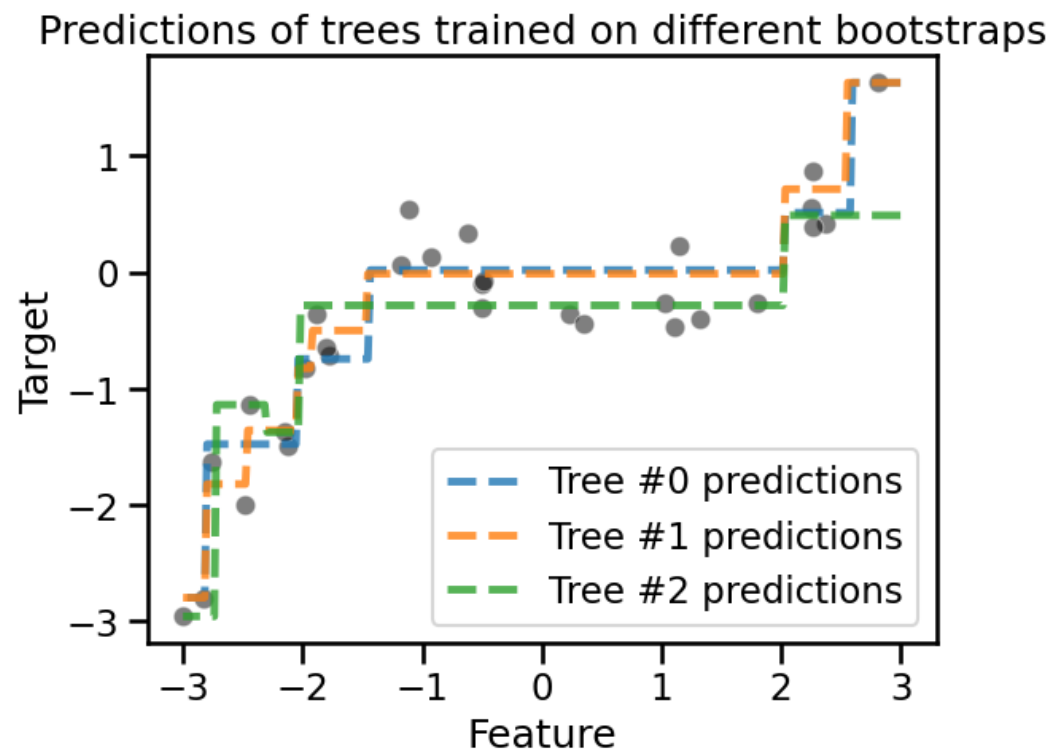
This work by Sebastian Raschka is licensed under a Creative Commons Attribution 4.0 International License.

Алгоритм бэггинга

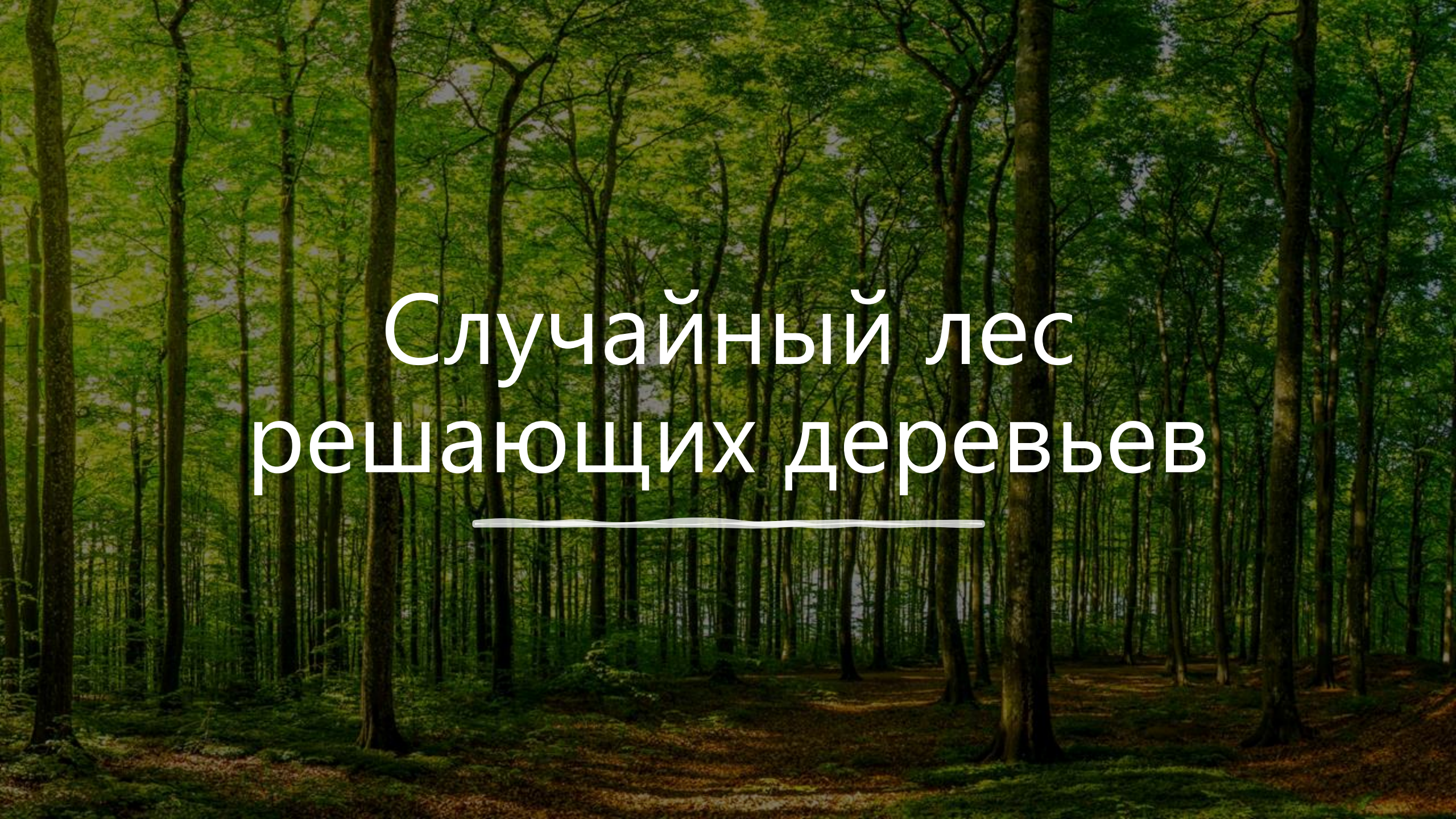
- ▶ Дана выборка данных X, y
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрапа** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем модель классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Пример



Источник: https://inria.github.io/scikit-learn-mooc/python_scripts/ensemble_bagging.html

A photograph of a dense forest with tall, thin trees and a lush green canopy. The text is overlaid in the center.

Случайный лес решающих деревьев

Случайный лес (random forest)

Training dataset

Obs	X_1	X_2	X_3	X_4	X_5	Y
1	X_{11}	X_{21}	X_{31}	X_{41}	X_{51}	0
2	X_{12}	X_{22}	X_{32}	X_{42}	X_{52}	1
3	X_{13}	X_{23}	X_{33}	X_{43}	X_{53}	0
4	X_{14}	X_{24}	X_{34}	X_{44}	X_{54}	0
5	X_{15}	X_{25}	X_{35}	X_{45}	X_{55}	1

Bootstrap

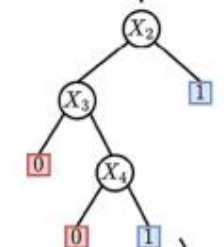
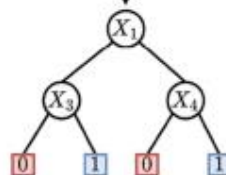
Obs	X_1	X_3	X_4	Y
1	X_{11}	X_{31}	X_{41}	0
2	X_{12}	X_{32}	X_{42}	1
5	X_{15}	X_{35}	X_{45}	1
1	X_{11}	X_{31}	X_{41}	0
5	X_{15}	X_{35}	X_{45}	1

Obs	X_2	X_3	X_4	Y
1	X_{21}	X_{31}	X_{41}	0
3	X_{23}	X_{33}	X_{43}	0
4	X_{24}	X_{34}	X_{44}	0
3	X_{23}	X_{33}	X_{43}	0
2	X_{22}	X_{32}	X_{42}	0

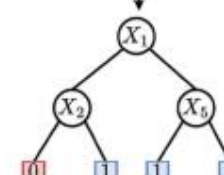
...

Obs	X_1	X_2	X_5	Y
2	X_{12}	X_{22}	X_{52}	1
3	X_{13}	X_{23}	X_{53}	0
5	X_{15}	X_{25}	X_{55}	1
5	X_{15}	X_{25}	X_{55}	1
3	X_{13}	X_{23}	X_{53}	0

Ensemble of trees



...



Aggregation

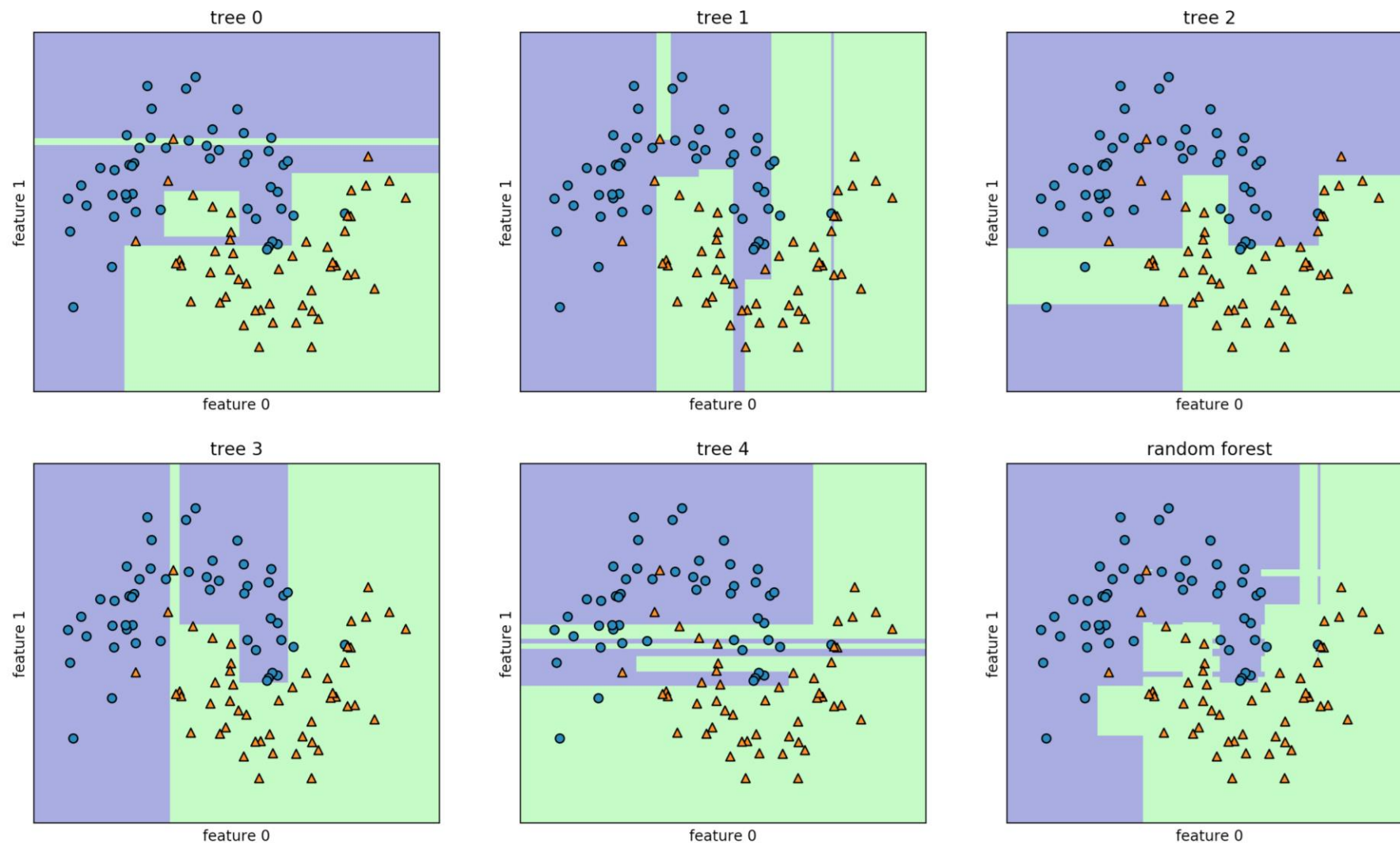
Majority decision

Алгоритм случайного леса

- ▶ Дана выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрапа** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем решающее дерево классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
 - При каждом разбиении дерева выбирается **t случайных признаков из d** .
Оптимальное разбиение ищется только среди них.
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Пример



Бутстрап (bootstrap)



This work by Sebastian Raschka is licensed under a Creative Commons Attribution 4.0 International License.

Out-of-Bag (OOB) ошибка

- ▶ Строим деревья на бутстрап подвыборках
- ▶ Объекты, которые не попали в подвыборку, можно использовать для тестирования дерева
- ▶ Для каждого объекта x_i можно найти деревья, которые были обучены без него, и посчитать по их прогнозам Out-of-Bag ошибку:

$$OOB = \sum L(y_i, \frac{1}{\sum_{k=1}^K [x_i \notin X^{(k)}]} \sum_{k=1}^K [x_i \notin X^{(k)}] b_k(x_i))$$

- где $L(y, z)$ – функция потерь или метрика качества

Важные замечания

- ▶ Бэггинг можно применять для любых алгоритмов
- ▶ Лучше всего он работает на слабых моделях
- ▶ Как правило, в композиции объединяют переобученные модели



Bias-Variance decomposition

Вопросы

- ▶ Почему композиции алгоритмов работают?
- ▶ Почему прогноз композиции лучше, чем прогнозы отдельных моделей?
- ▶ Почему используют переобученные модели?

Задача

- ▶ Рассмотрим задачу регрессии с функцией потерь MSE
- ▶ Пусть ответ $y(x)$ для заданного x – некоторая случайная величина:

$$y(x) = f(x) + \epsilon$$

– где $\epsilon \sim N(0, \sigma^2)$

- ▶ Обозначим прогноз нашей модели $a(x)$
- ▶ Посчитаем мат. ожидание ошибки прогноза для заданного x :

$$Error = E[(a(x) - y(x))^2]$$

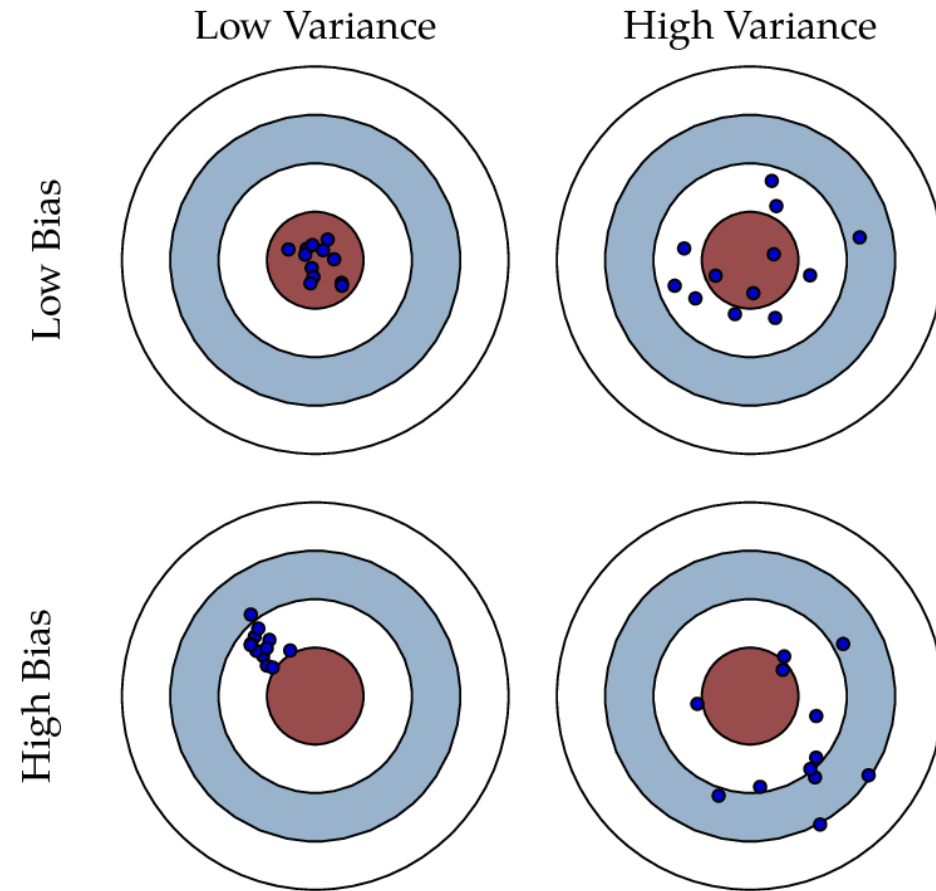
Расчет ошибки

$$\begin{aligned} \text{Error} &= E \left[(a(x) - y(x))^2 \right] = \\ &= E[(a(x) - f(x) - \epsilon + E[a(x)] - E[a(x)])^2] = \\ &= \textcolor{red}{E[(a(x) - E[a(x)])^2]} + (\textcolor{blue}{E[a(x)]} - \textcolor{blue}{f(x)})^2 + \textcolor{green}{\sigma^2} = \\ &= \textcolor{red}{Variance} + \textcolor{blue}{Bias^2} + \textcolor{green}{Noise} \end{aligned}$$

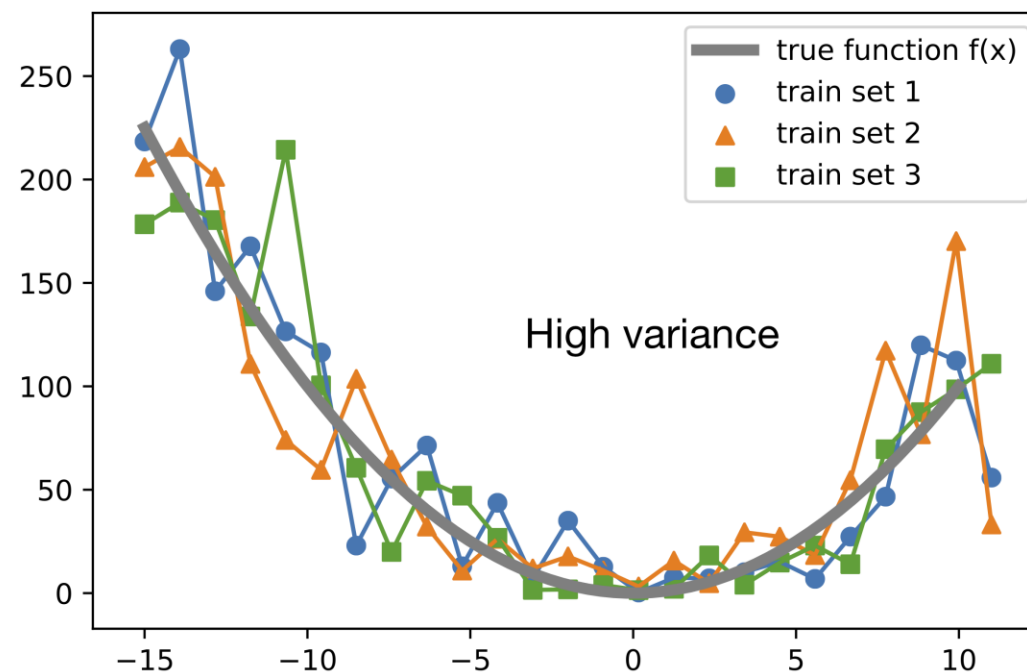
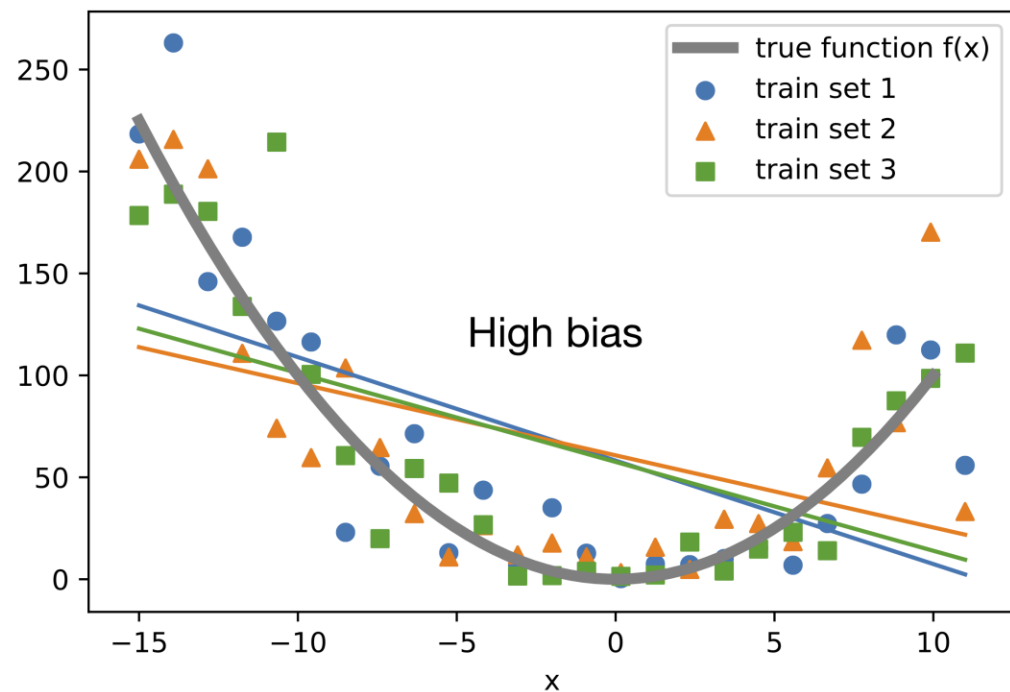
Bias-Variance decomposition

- ▶ **Variance (разброс)** – разброс ответов обученных алгоритмов относительно среднего ответа.
- ▶ **Bias (смещение)** – отклонение среднего ответа алгоритма от идеального ответа
- ▶ **Noise (шум)** – шум в данных

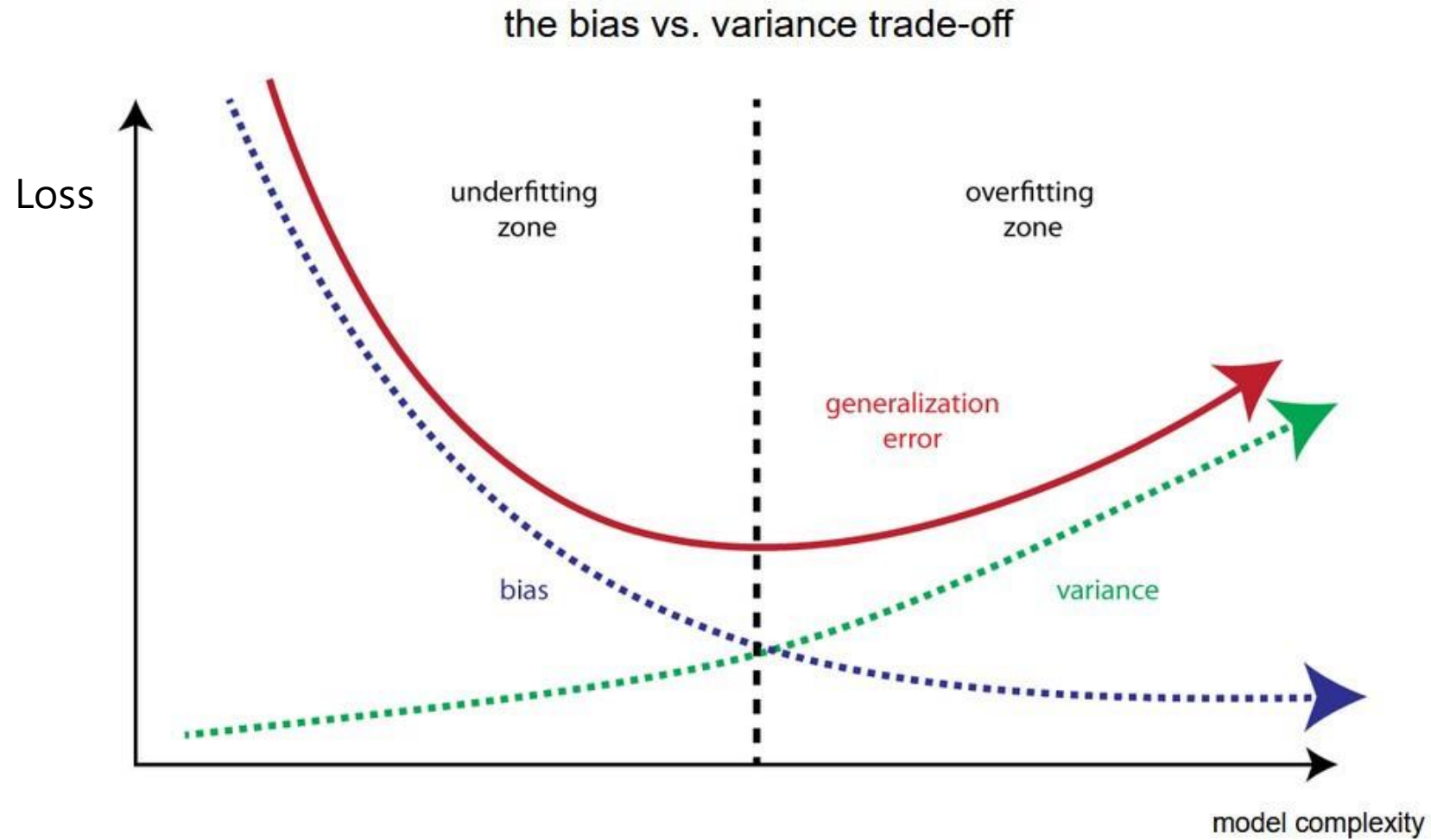
Пример



Пример

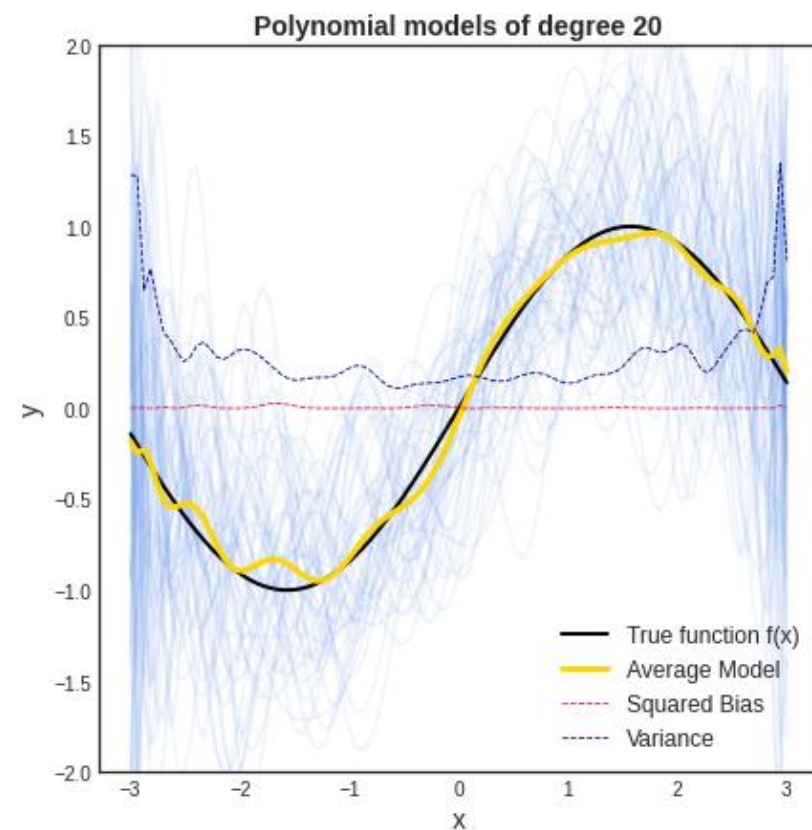
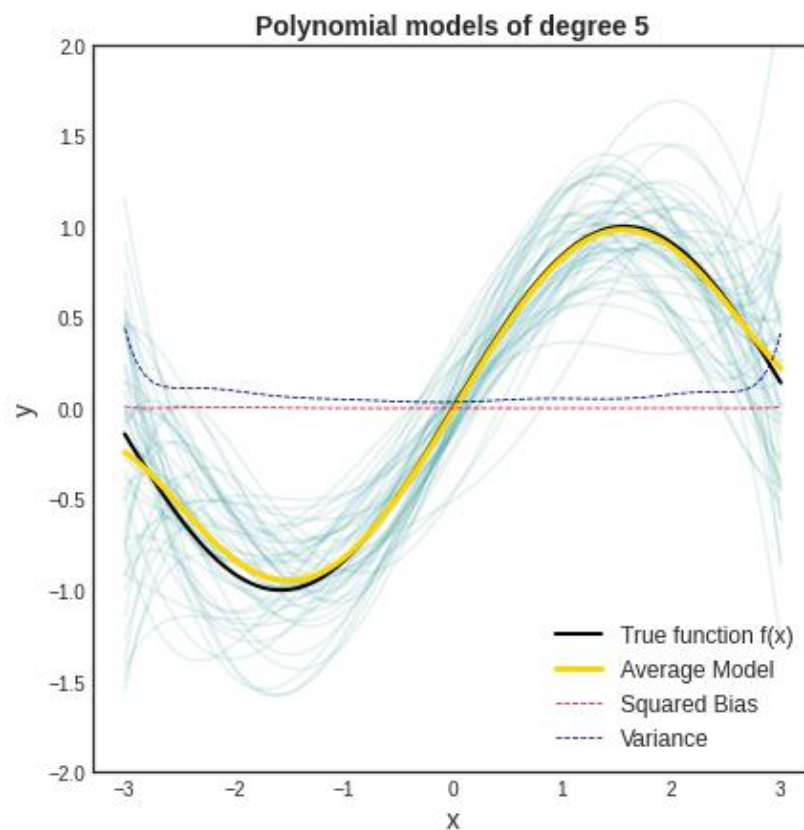
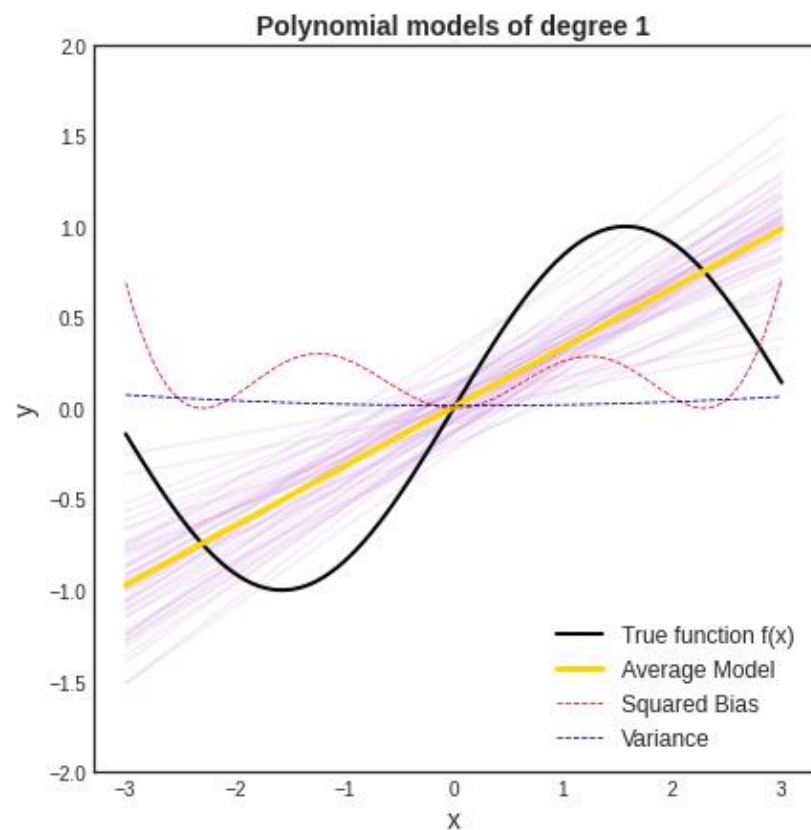


Переобучение



Пример

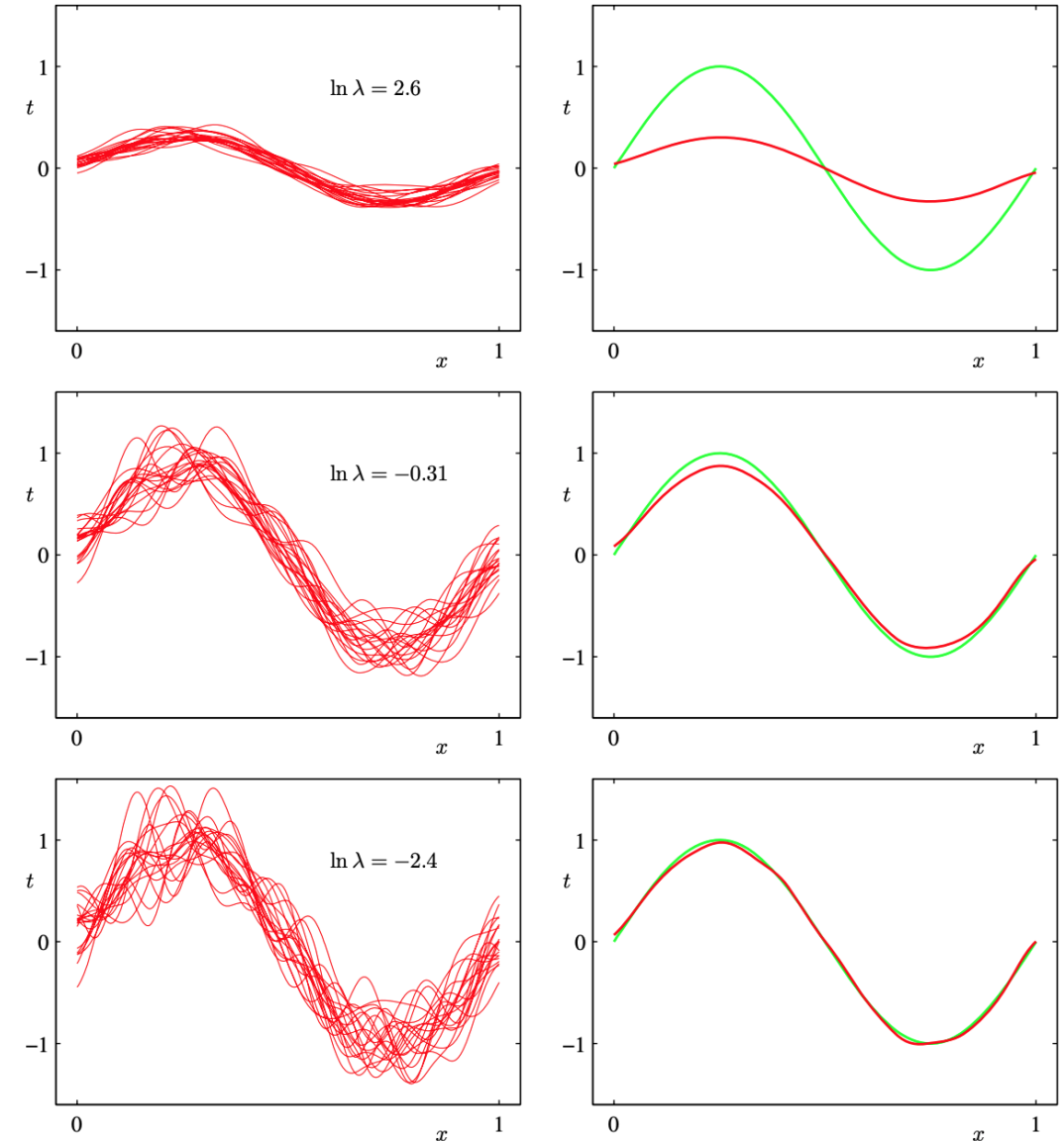
Polynomial models of different degrees fit on random data



Источник: <https://allenkunle.me/bias-variance-decomposition>

Эффект регуляризации

- ▶ Большой коэффициент регуляризации приводит к росту **bias** и снижению **variance**
- ▶ Маленький коэффициент регуляризации приводит к снижению **bias** и росту **variance**



Композиции алгоритмов

$$Error = E[(a(x) - E[a(x)])^2] + (E[a(x)] - f(x))^2 + \sigma^2$$

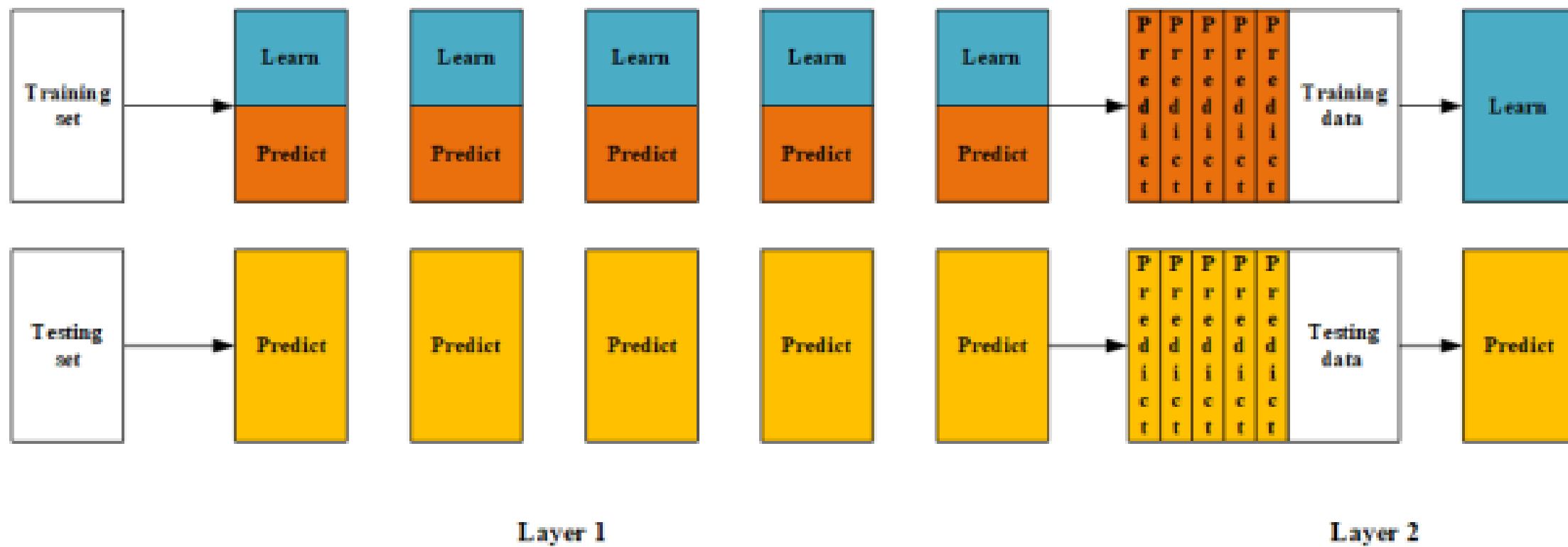
- ▶ Пусть $a(x)$ - композиция алгоритмов
- ▶ Тогда $a(x) \approx E[a(x)]$
- ▶ Тогда ошибка для композиции:

$$Error \approx 0 + (E[a(x)] - f(x))^2 + \sigma^2$$



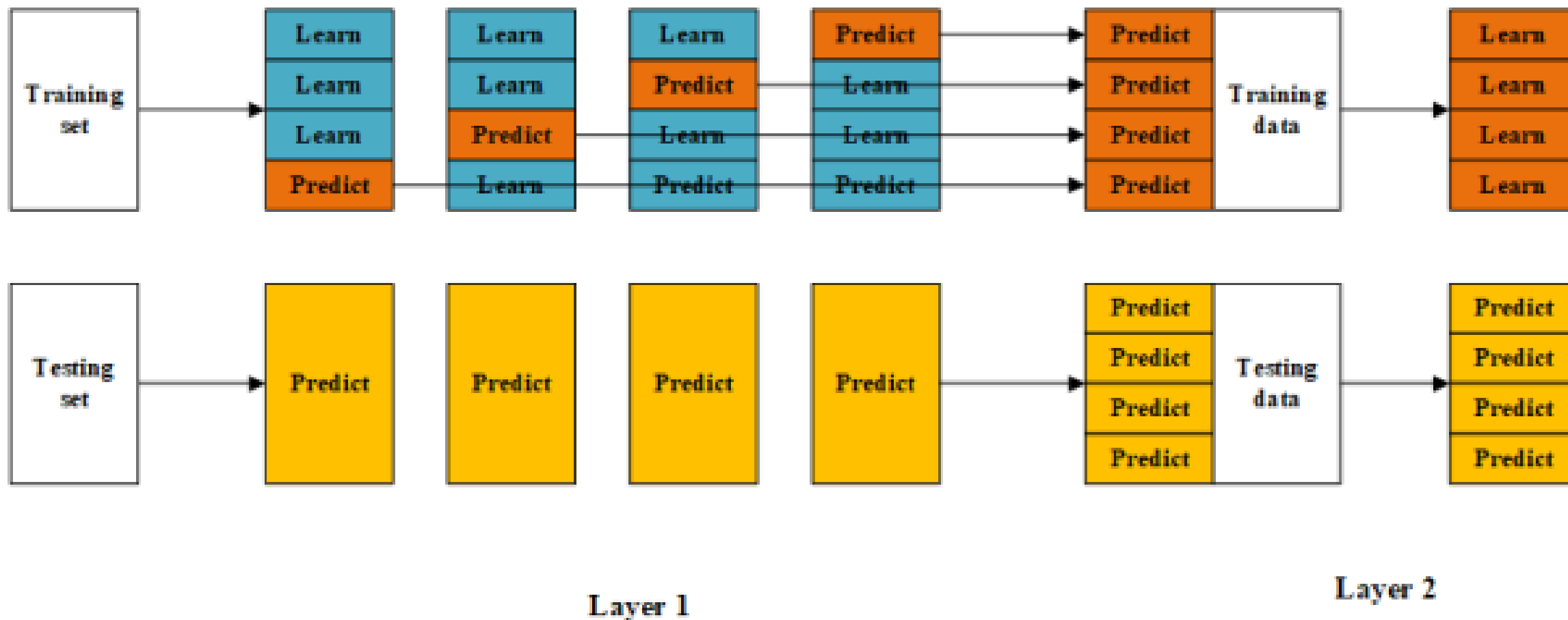
Blending и Stacking (не обязательно)

Blending



Источник: <https://www.codeproject.com/Tips/4354591/Step-by-Step-Guide-to-Implement-Machine-Learning-5>

Stacking



Источник: <https://www.codeproject.com/Tips/4354591/Step-by-Step-Guide-to-Implement-Machine-Learning-5>

Заключение



Вопросы

- ▶ Что такое композиция алгоритмов машинного обучения? Покажите, что в предположении некоррелированных ошибок базовых алгоритмов, ошибка композиции будет в N раз меньше, чем средняя ошибка базовых алгоритмов, где N - число базовых алгоритмов.
- ▶ Что такое бэггинг? Что такое случайный лес? Что такое out-of-bag ошибка, для чего она используется?