Boosting и Bagging

План

1. Композиции алгоритмов

2. Что нужно помнить

3. ML Puzzles

1. Композиции алгоритмов

Теорема Кондорсе о жюри присяжных

Каждый член жюри имеет независимое мнение

Вероятность правильного решения члена жюри выше случайного

Вероятность принятия правильного решения путем голосования стремится к единице с увеличением количества присяжных

Теорема Кондорсе о жюри присяжных

Каждый член жюри имеет **независимое** мнение

Вероятность правильного решения члена жюри **выше случайного**

Вероятность принятия правильно решения путем голосования стремится к единице **с увеличением количества присяжных**

Бэггинг

Базовые алгоритмы обучаются на случайной подвыборке исходной выборки с повторениями

Компенсация ошибок алгоритмов, выбросы не попадают в часть выборок

Композиция: усреднение или простое голосование

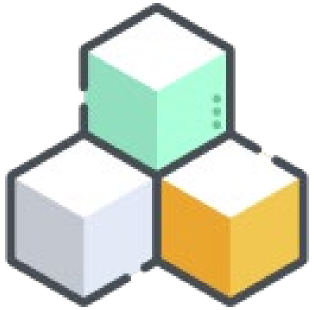
Бэггинг: математика

Пусть $\xi_1, \xi_2, \ldots, \xi_n$ — случайные одинаково распределённые случайные величины с дисперсией σ^2 и коэффициентом корреляции ρ. Тогда

$$\mathbf{Var}\frac{1}{n}\sum_{i=1}^{n}\xi_i = \frac{1}{n^2}\mathbf{Var}\sum_{i=1}^{n}\xi_i = \frac{1}{n^2}\mathbf{cov}(\sum_{i=1}^{n}\xi_i, \sum_{j=1}^{n}\xi_j) = \frac{1}{n^2}\left(\sum_{i=1,j=1}^{n}\mathbf{cov}(\xi_i,\xi_j)\right) =$$

$$= \frac{1}{n^2}\left(\sum_{i=1}^{n}\mathbf{cov}(\xi_i,\xi_i) + \sum_{i\neq j}\mathbf{cov}(\xi_i,\xi_j)\right) = \frac{1}{n^2}(n\sigma^2+n(n-1)\rho\sigma^2) = \frac{1+\rho(n-1)}{n}\sigma^2$$

Метод случайных подпространств



Базовые алгоритмы обучаются на случайных подмножествах признаков

При большом числе признаков и небольшом числе объектов

Композиция: усреднение или простое голосование

Random Forest

Случайное подмножество объектов для каждого дерева

Признак в каждой вершине выбирается из случайного подмножества

Композиция: усреднение или простое голосование

Extremely Randomized Trees *

Для каждого признака пороговые значения выбираются случайно

Для разделения узла выбирается наилучший из случайно сгенерированных порогов

Использовать, когда вы сильно переобучаетесь на RF

* sklearn.ensemble. ExtraTreesClassifier & sklearn.ensemble.ExtraTreesRegressor

Короеды случайного леса

Глубокие деревья требуют много вычислительных ресурсов, особенно для большой или широкой выборки

Процесс построения деревьев ненаправленный, нужно много деревьев

Неглубокие деревья не могут улавливать сложные закономерности, увеличивается сдвиг

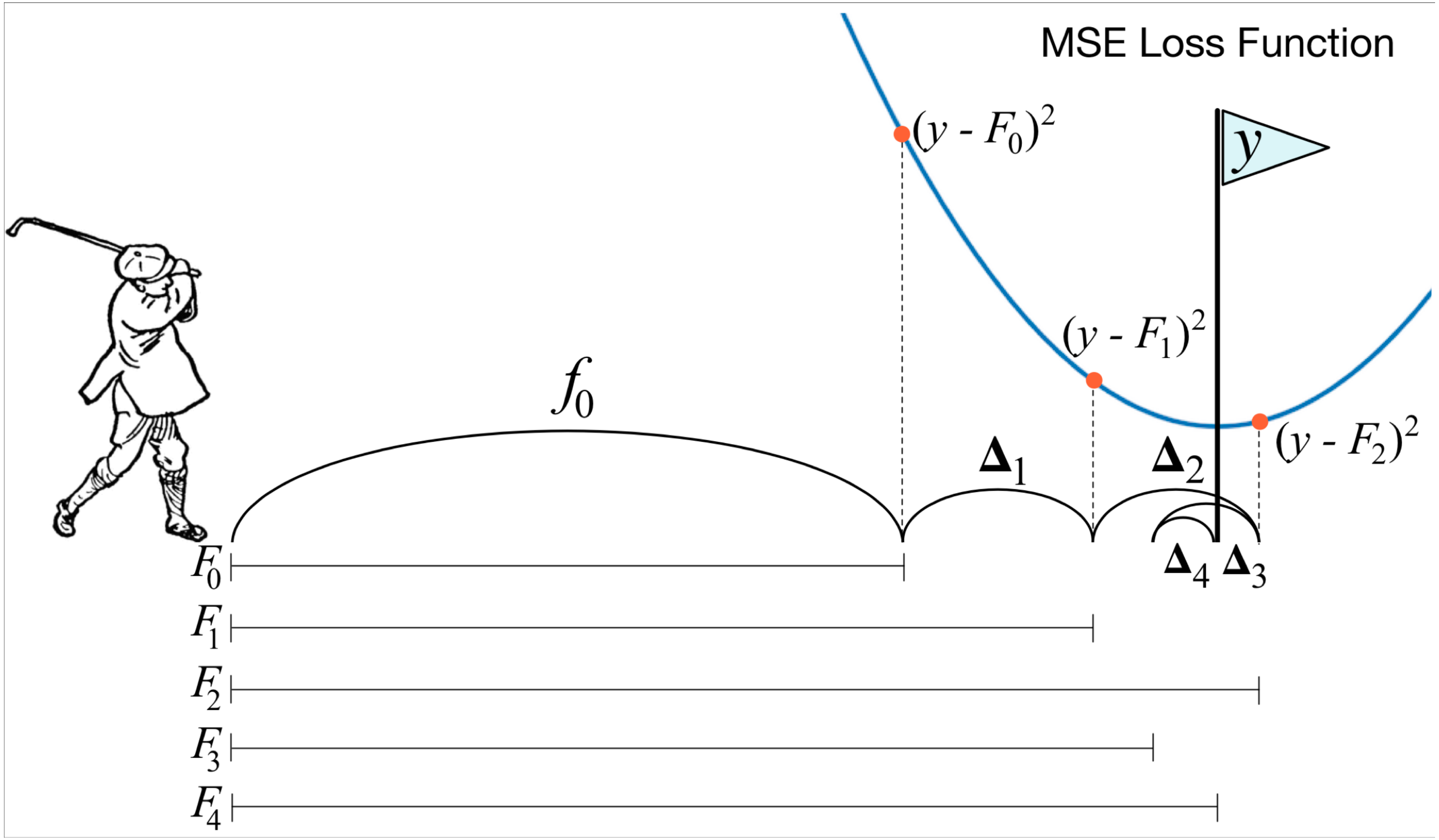
Бустинг

Композиция простых базовых алгоритмов

Алгоритмы строятся последовательно

Каждый следующий алгоритм настраивается на ошибку композиции предыдущих

Бустинг: аналогия



Бустинг: математика

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + \gamma_N b_N(x_i)) \to \min_{b_N, \gamma_N}$$

1. $s_i = -\left.\frac{\partial L}{\partial z}\right|_{z=a_{N-1}(x_i)}$
2. $b_N = \arg\min_b \sum_{i=1}^{l} (b(x_i) - s_i)^2$
3. $\gamma_N = \arg\min_\gamma \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + \gamma b_N(x_i))$

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \approx$$

$$\approx \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i)) + g_i b_N(x_i) + \frac{1}{2} h_i b_N(x_i)^2 + \Omega(b_N)$$

Regularization: $\Omega(b_N) = \gamma T + \lambda \|w\|^2$, T – leaf count, w – leaf values.

Полезные ссылки

- Greedy function approximation: A gradient boosting machine
- XGBoost: A Scalable Tree Boosting System
- LightGBM: A Highly Efficient Gradient Boosting Decision Tree
- Fighting biases with dynamic boosting
- А. Дьяконов «Градиентный бустинг»
- Видео про градиентный бустинг и фишки

2. Что стоит помнить

Bias-Variance tradeoff

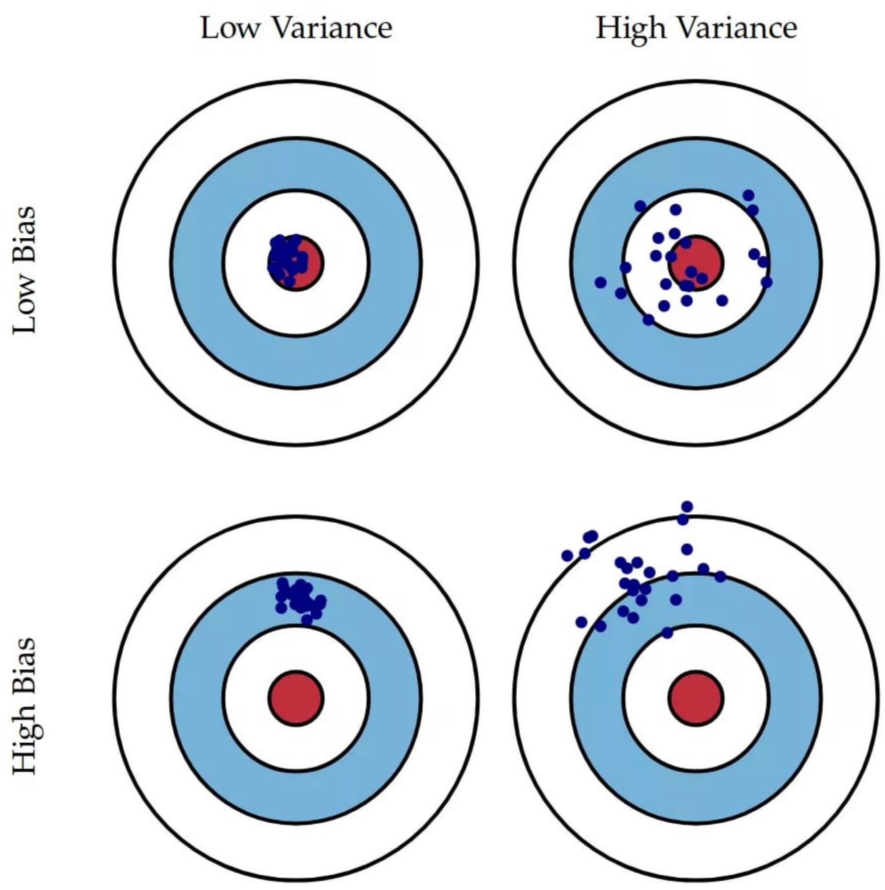


Fig. 1 Graphical illustration of bias and variance.

Bias-Variance tradeoff

Bagging уменьшает ?

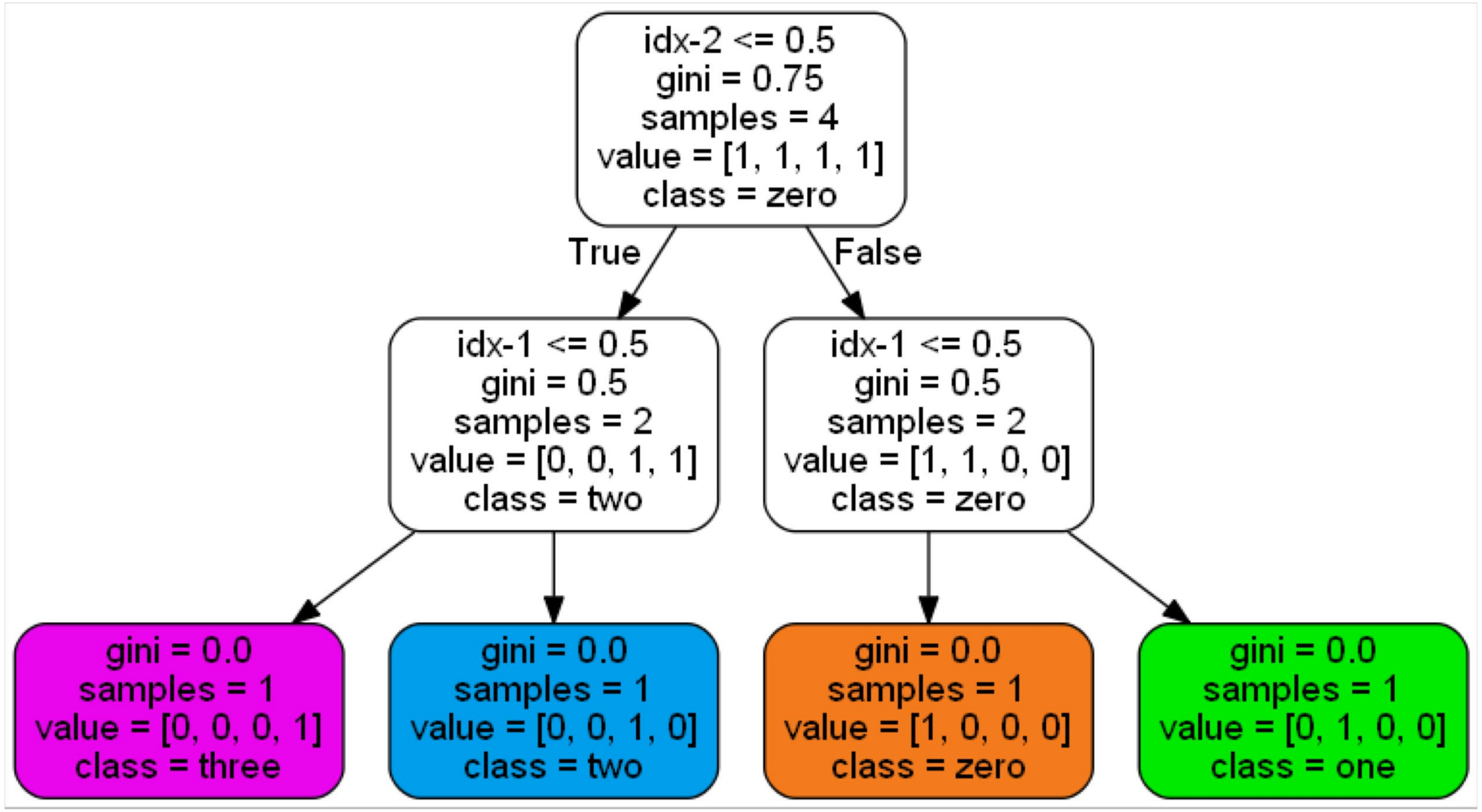
Boosting уменьшает ?

Bias-Variance tradeoff

Bagging уменьшает **variance**

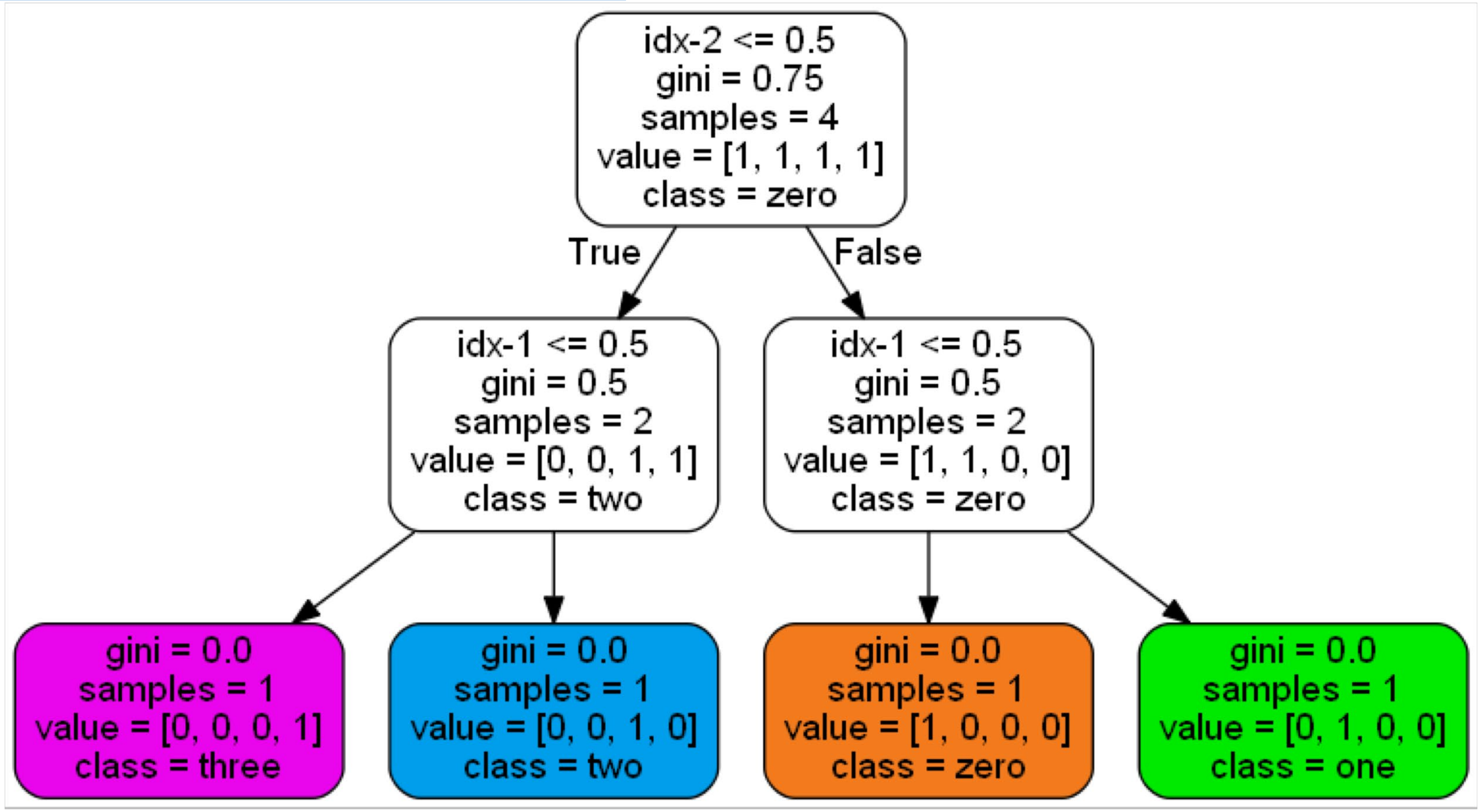
Boosting уменьшает **bias**

Альтернативное использование деревьев1



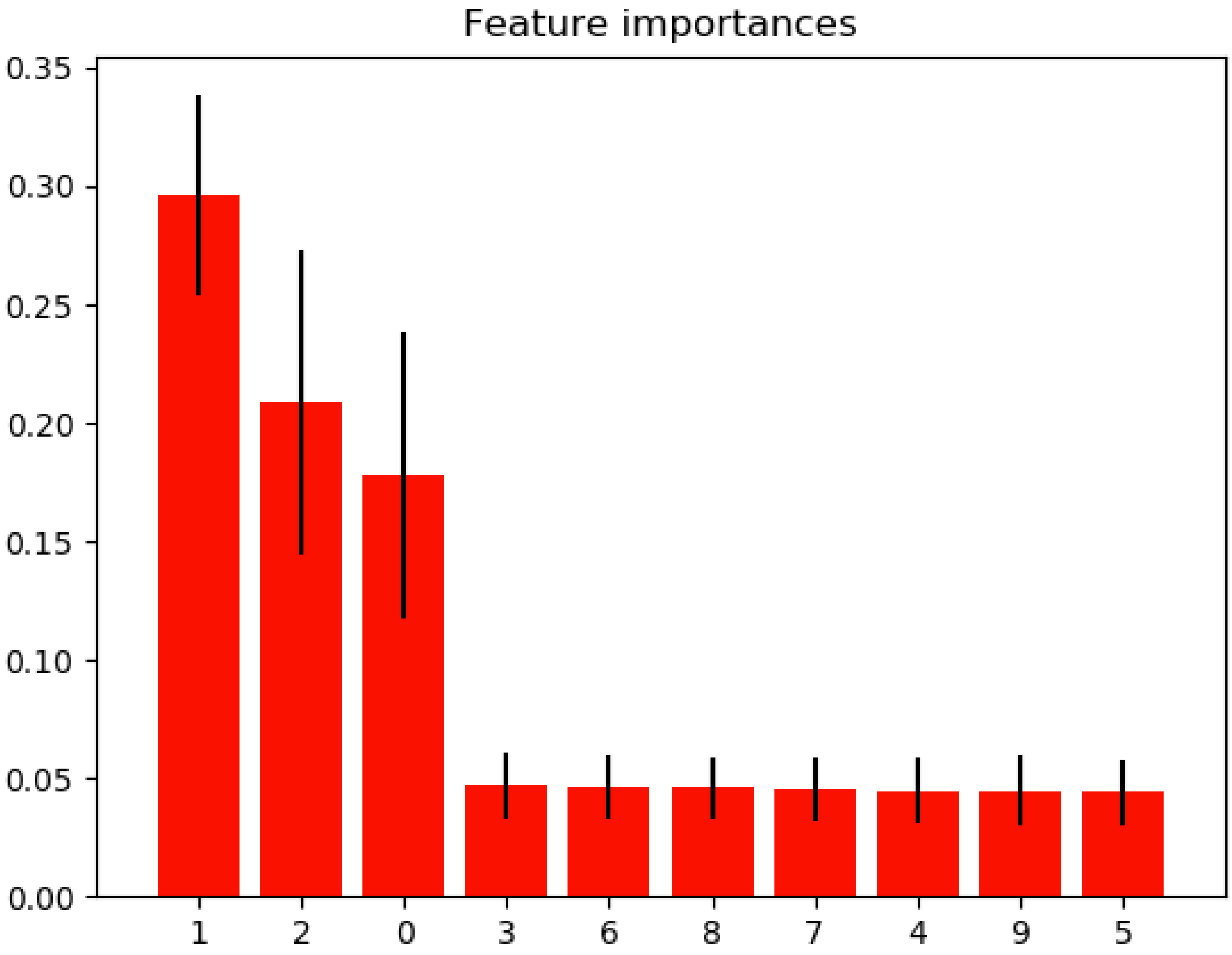
Альтернативное использование деревьев2

sklearn.ensemble.RandomTreesEmbedding



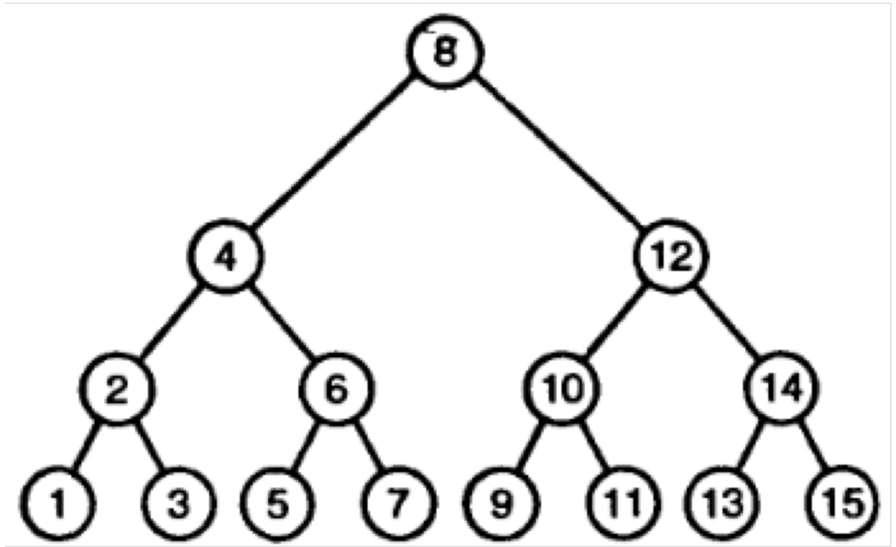
0 0 1 0

Альтернативное использование деревьев3



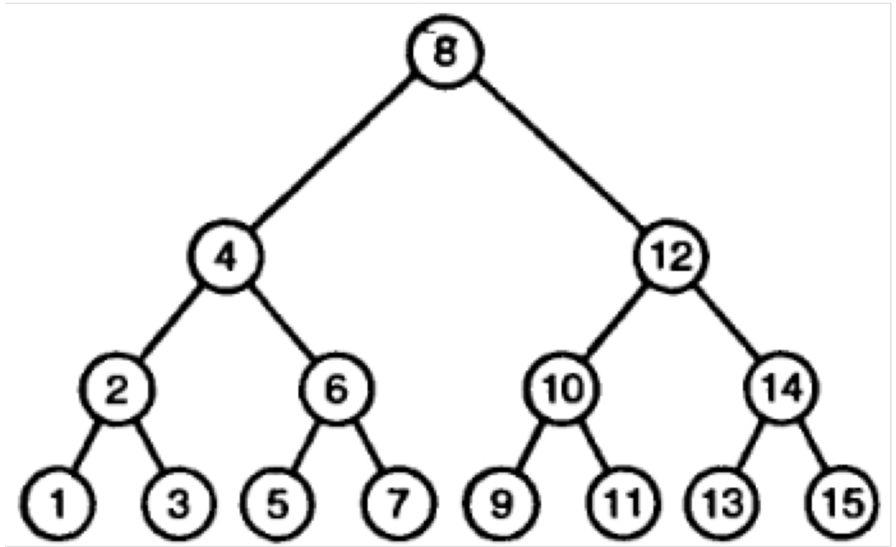
Накладные расходы

- есть полное бинарное дерево глубины h
- сколько в нём вершин?



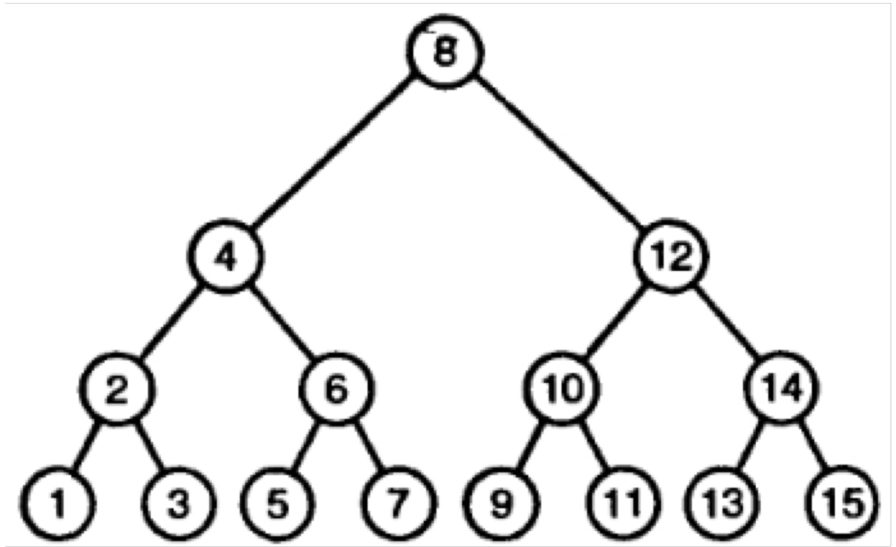
Накладные расходы

- есть полное бинарное дерево глубины h
- сколько в нём вершин? 2^h



Накладные расходы

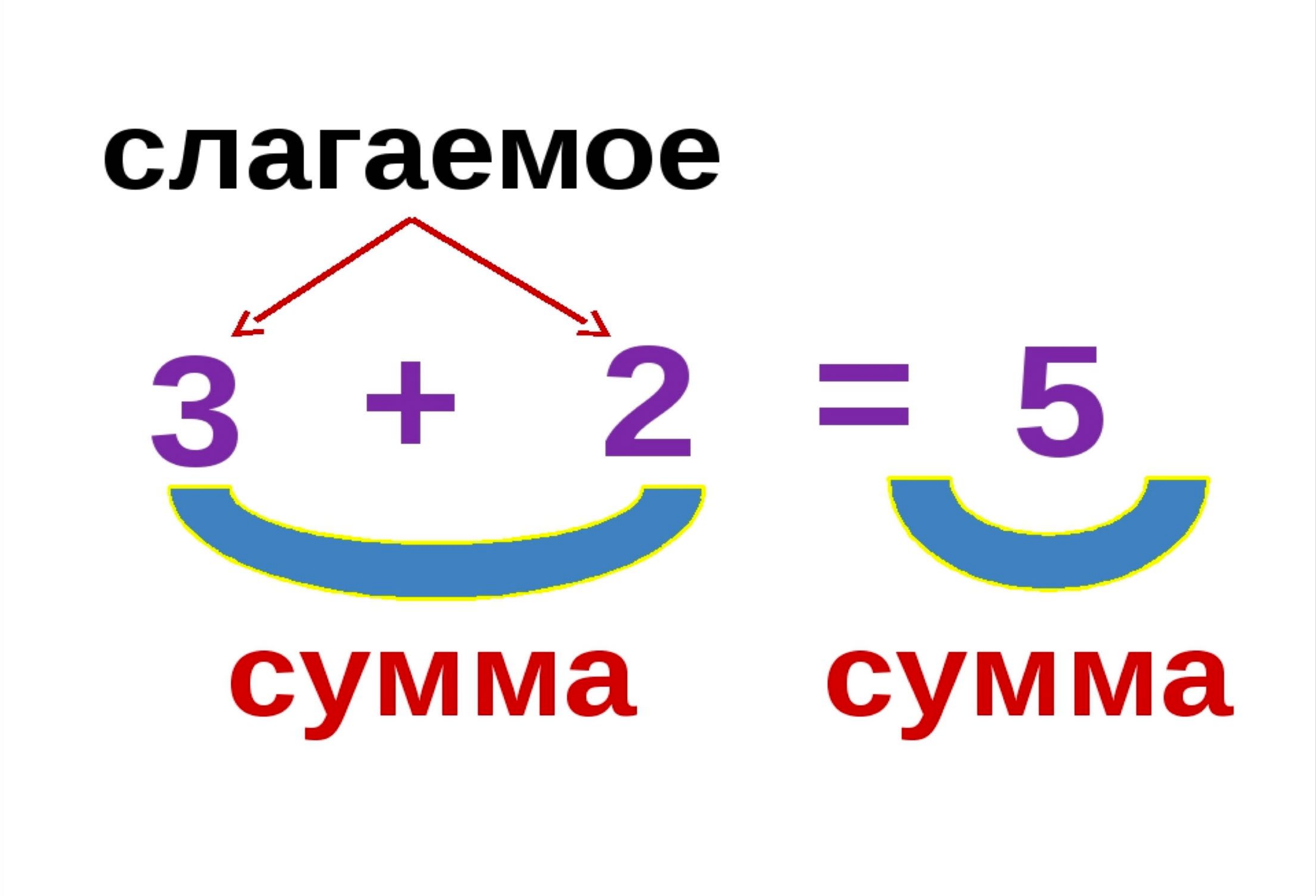
- есть полное бинарное дерево глубины h
- сколько в нём вершин? 2^h
- сколько памяти займёт лес на 1000 деревьев глубины 20?



3. ML Puzzles

Сумма

Может ли GB или RF выучить сумму двух чисел?



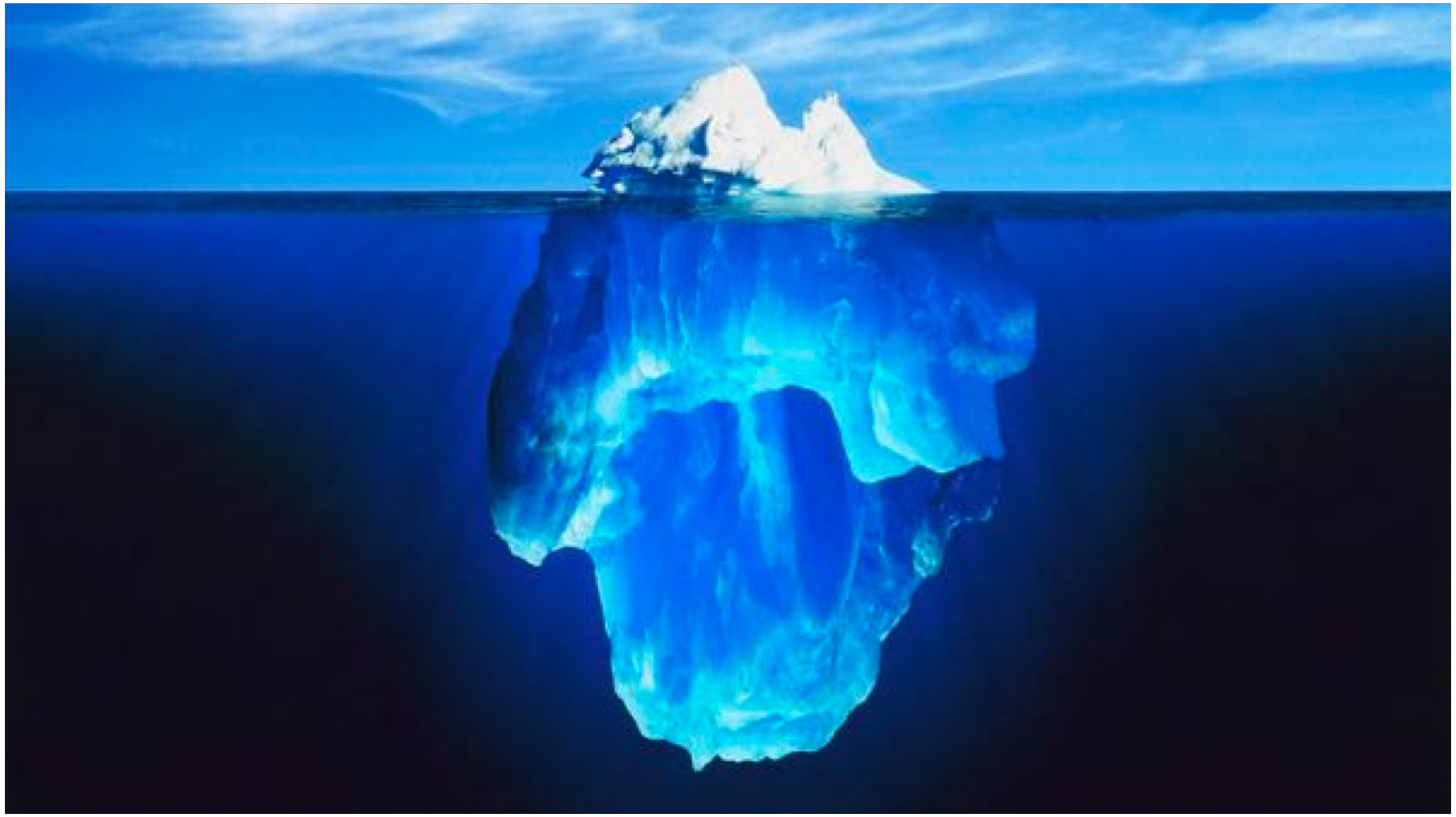
Отрицательные числа

В обучении есть положительные таргеты, а на тесте есть отрицательные предсказания:

- GB
- RF
- Linear models
- kNN

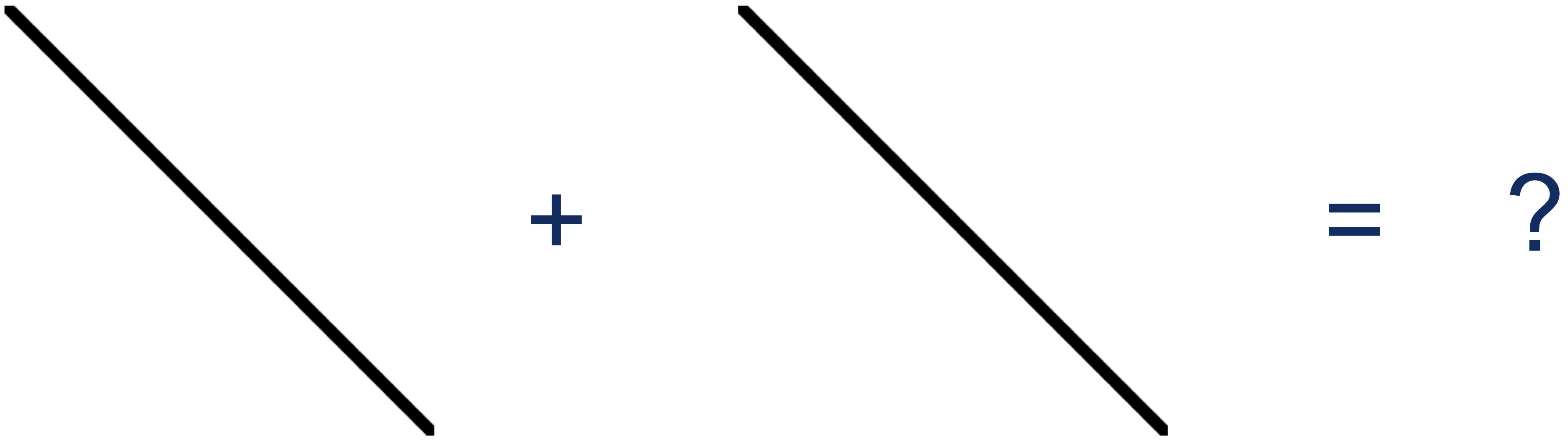
Глубина деревьев

Какая глубина дерева характернадля бустинга, а какая для бэггинга? И почему?



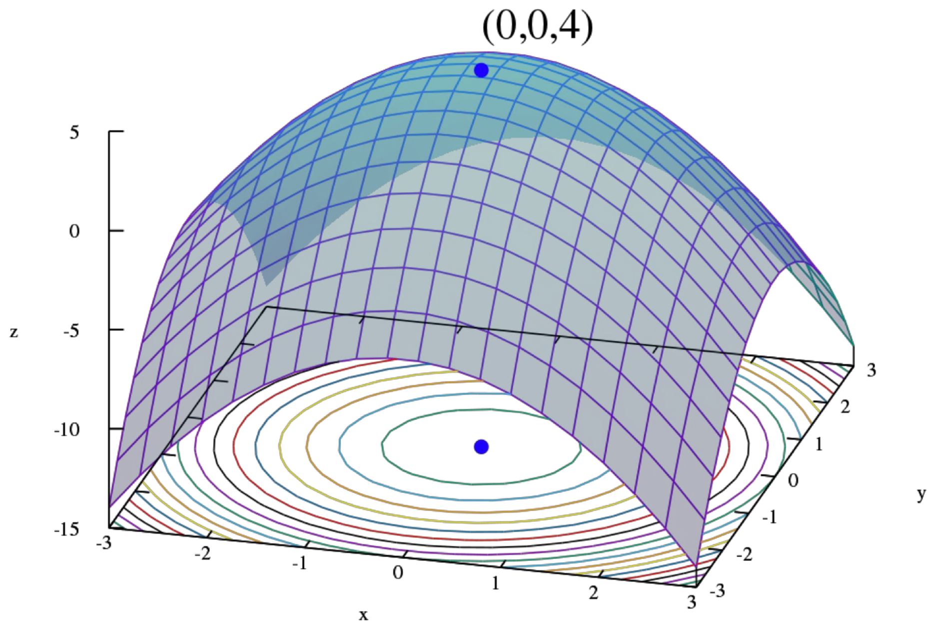
Немного о линейных моделях

Имеет ли смысл делать бустинг и бэггинг над линейными моделями?



О разном

Можно ли использовать GB модель для оптимизации фичей с целью максимизации таргета?



О параллелизме

Как распараллелить бэггинг? А бустинг?