

Ансамблем (Ensemble, Multiple Classifier System) называется алгоритм, **который состоит из нескольких алгоритмов машинного обучения**, а процесс построения ансамбля называется ансамблированием (ensemble learning).

Ансамбль алгоритмов используется с целью получения лучшей эффективности прогнозирования, чем можно было бы получить от каждого обучающего алгоритма по отдельности.

Простейший пример ансамбля в регрессии – усреднение нескольких алгоритмов:

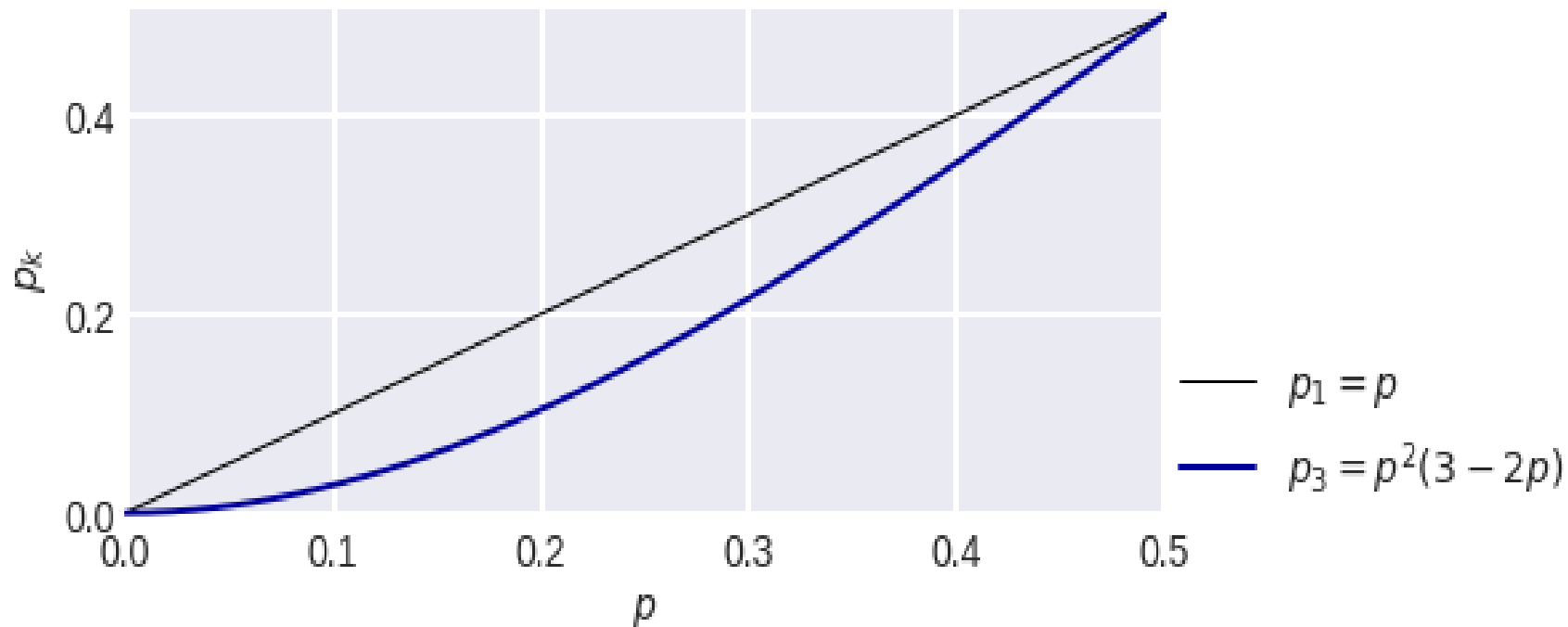
$$a(x) = \frac{1}{n} (b_1(x) + \dots + b_n(x))$$

Алгоритмы из которых состоит ансамбль b_i называются базовыми алгоритмами (base learners).

В задачах классификации простейший пример ансамбля – комитет большинства:

$$a(x) = \text{mode}(b_1(x), \dots, b_n(x)),$$

где mode – значение, которое встречается чаще других среди аргументов функции.



Если рассмотреть задачу классификации с двумя классами $\{0, 1\}$ и три алгоритма, каждый из которых ошибается с вероятностью p , то в предположении, что их ответы – независимые случайные величины, получаем, что комитет большинства этих трёх алгоритмов ошибается с вероятностью **$pp(3-2p)$** . Как видно на рис., это выражение может быть существенно меньше p (при $p=0.2$ почти в два раза), т.е. использование такого ансамбля уменьшает ошибку базовых алгоритмов.

Если рассмотреть большее число алгоритмов, то по неравенству Хёфдинга ошибка комитета большинства экспоненциально убывает с ростом числа базовых алгоритмов:

$$\sum_{t=0}^{\lfloor n/2 \rfloor} C_n^t (1-p)^t p^{n-t} \leq e^{-\frac{1}{2}n(2p-1)^2}.$$

Для того чтобы ошибки отдельных алгоритмов на отдельных объектах компенсировались корректной работой других алгоритмов, **классификаторы должны быть независимыми**. НО в реальной практической ситуации, это не так.

Поэтому большинство приёмов в ансамблировании направлено на то, чтобы сделать ансамбль «достаточно разнообразным», при этом:

- повышают качество базовых алгоритмов,
- повышают разнообразие (diversity) базовых алгоритмов.

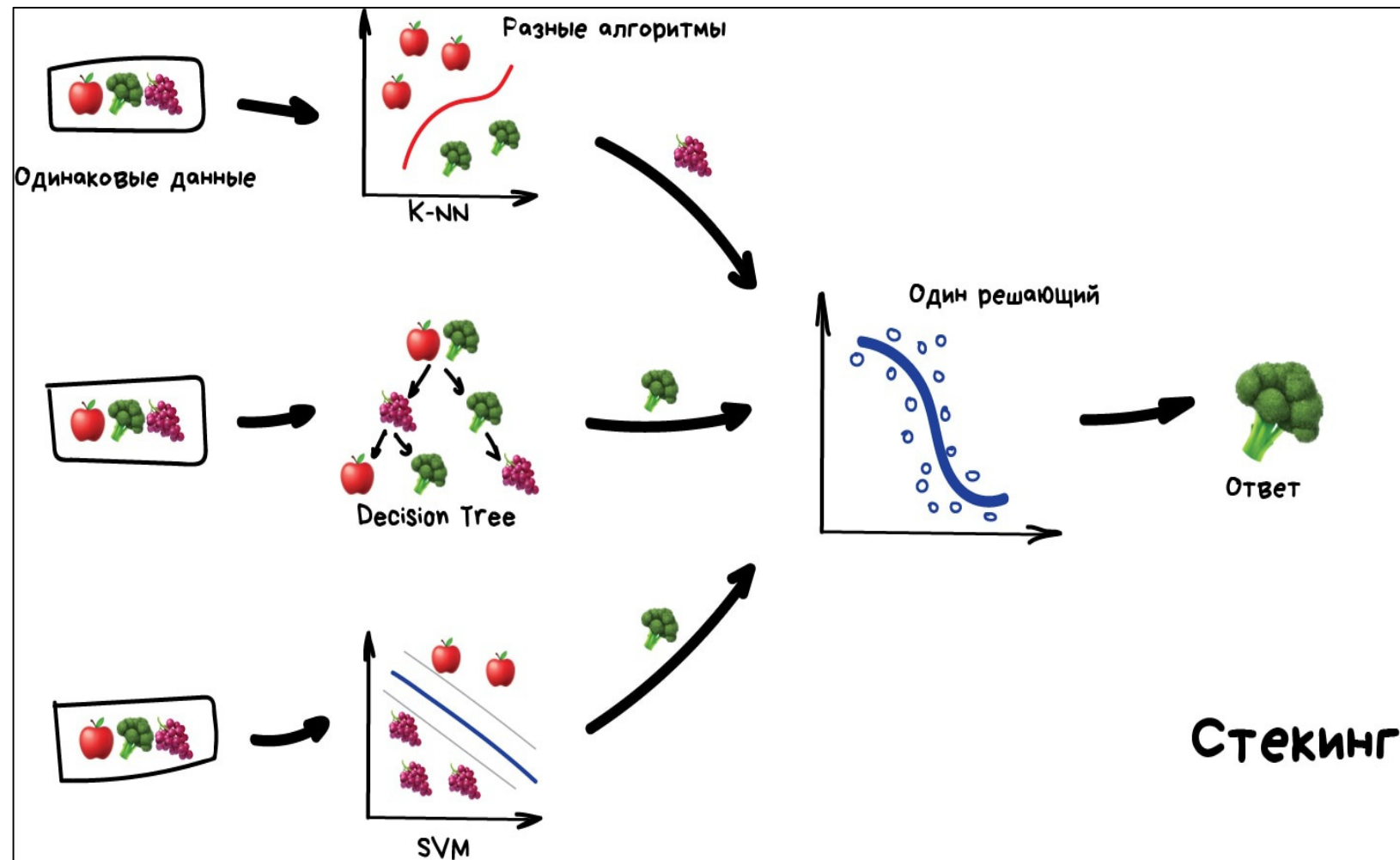
Разнообразие повышают за счёт «варьирования» обучающей выборки (**бэгинг**), «варьирования» моделей (**стэкинг**) и т.п.

Существуют различные **модели ансамблирования**. Рассмотрим три из них:

- стекинг
- бэггинг
- бустинг

Стекинг (Stacking). Обучаем несколько **разных алгоритмов** и передаём их результаты на вход последнему, который принимает итоговое решение.

Ключевое слово — **разных** алгоритмов, ведь один и тот же алгоритм, обученный на одних и тех же данных не имеет смысла. В качестве решающего алгоритма чаще берут регрессию.

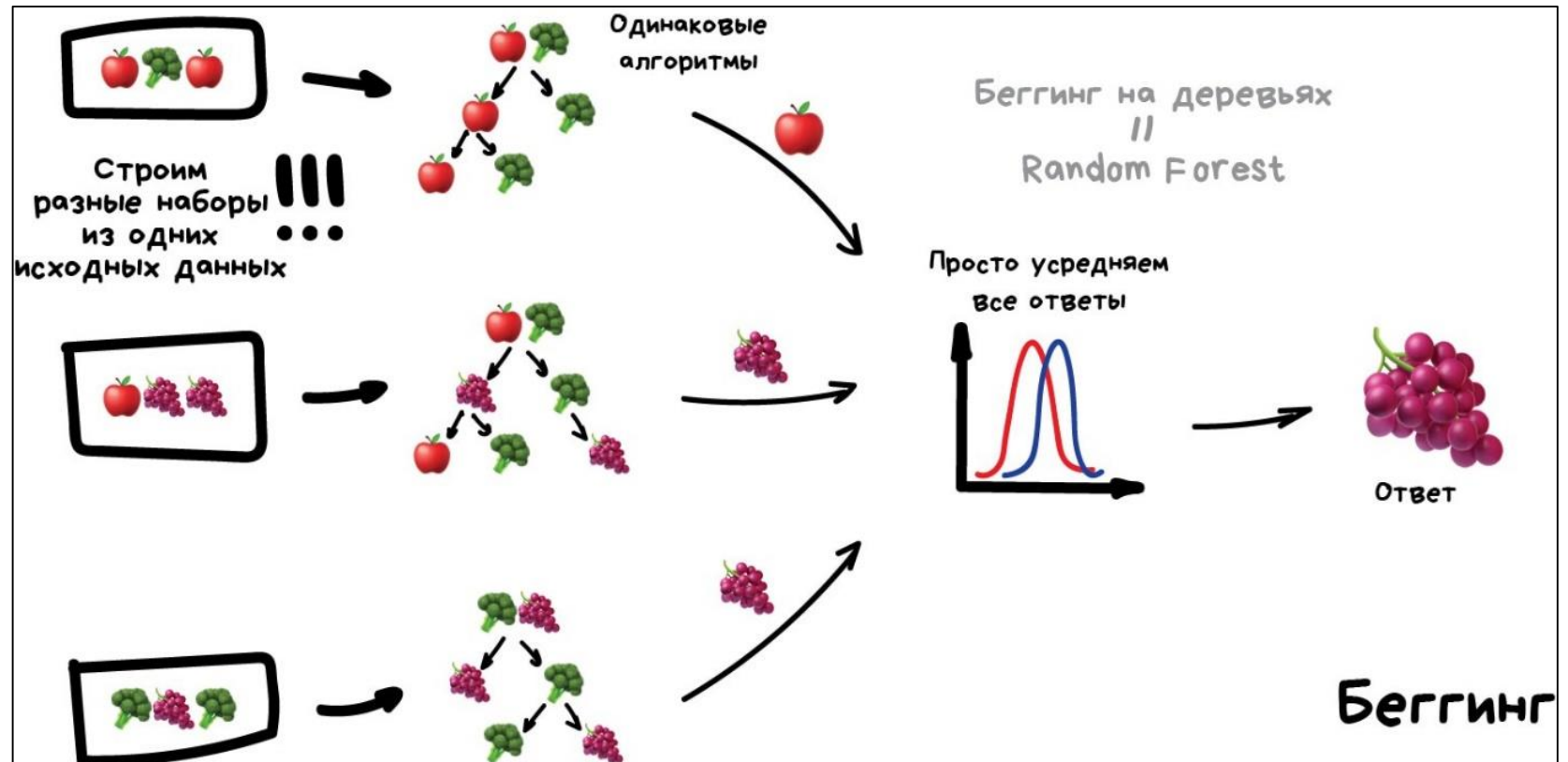


Бэггинг (Bagging). Обучаем **один алгоритм много раз на случайных выборках из исходных данных**. В самом конце усредняем ответы.

Данные в случайных выборках могут повторяться. То есть из набора 1-2-3 мы можем делать выборки 2-2-3, 1-2-2, 3-1-2 и т.д.. На них мы обучаем один и тот же алгоритм несколько раз, а в конце вычисляем ответ простым голосованием.

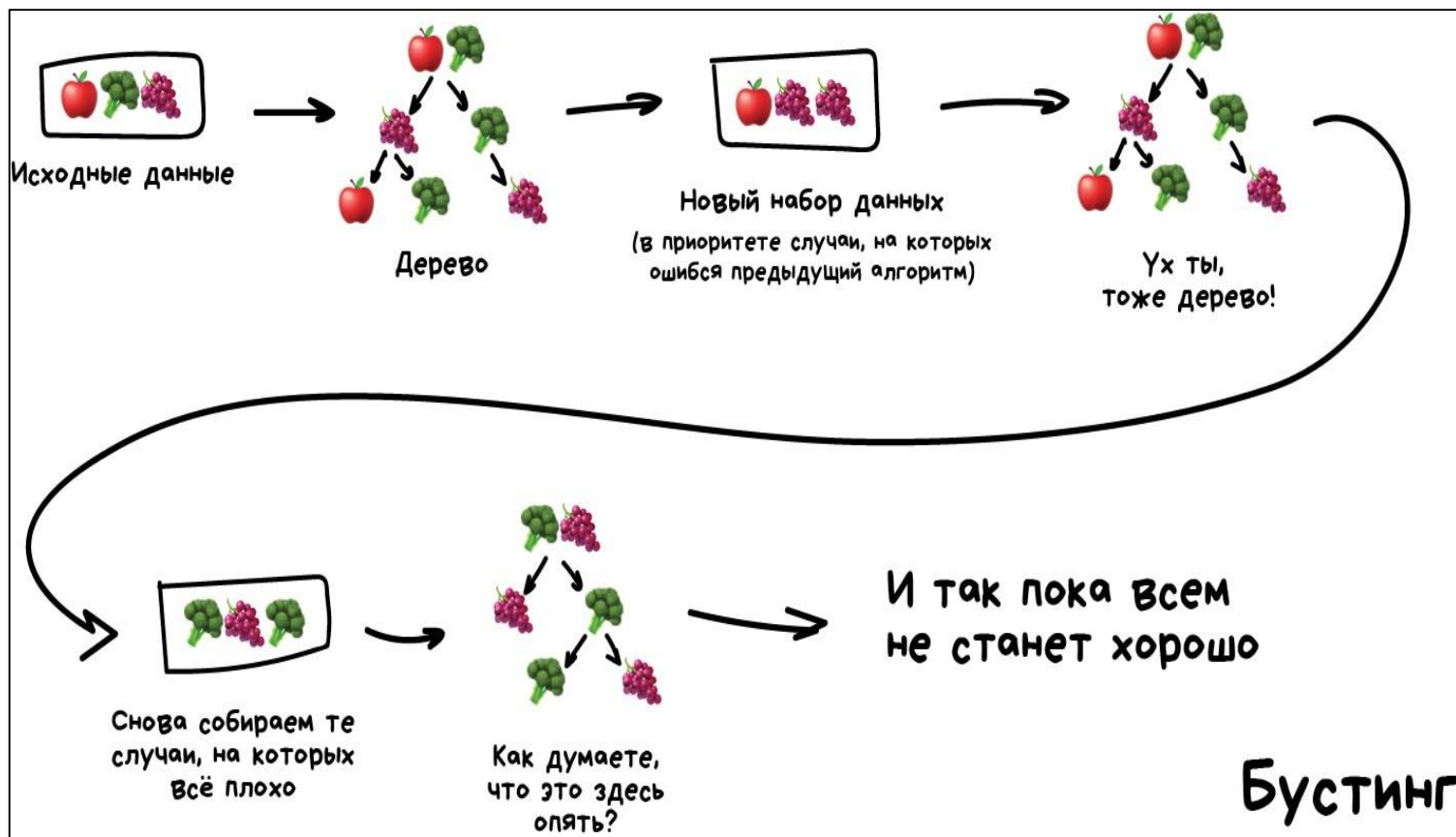
Самый популярный пример беггинга — алгоритм Random Forest, беггинг на деревьях.

Преимущество: скорость. Например, в реальном времени нейросеть будет слишком медлительна, а беггинг идеален, ведь он может считать свои деревья параллельно.



Бустинг (Boosting). Обучаем алгоритмы последовательно, каждый следующий уделяет особое внимание тем случаям, на которых ошибся предыдущий.

Как в беггинге, мы делаем выборки из исходных данных, но теперь не совсем случайно. В каждую новую выборку мы берём часть тех данных, на которых предыдущий алгоритм отработал неправильно. То есть как бы доучиваем новый алгоритм на ошибках предыдущего и повышал качество всего ансамбля.



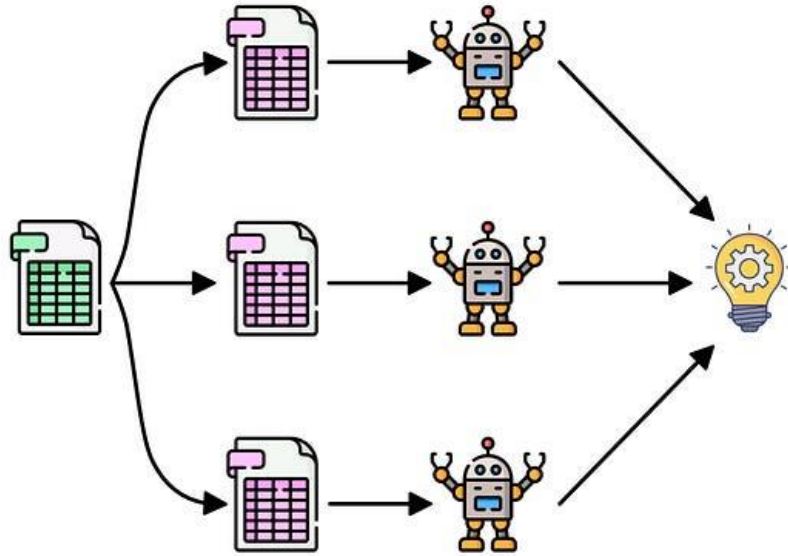
Как правило, бустинг показывает высокое качество над неглубокими деревьями (глубина от 3 до 6).

Плюсы — высокая точность классификации.

Минусы — не параллелится, НО, пока еще работает быстрее нейросетей.

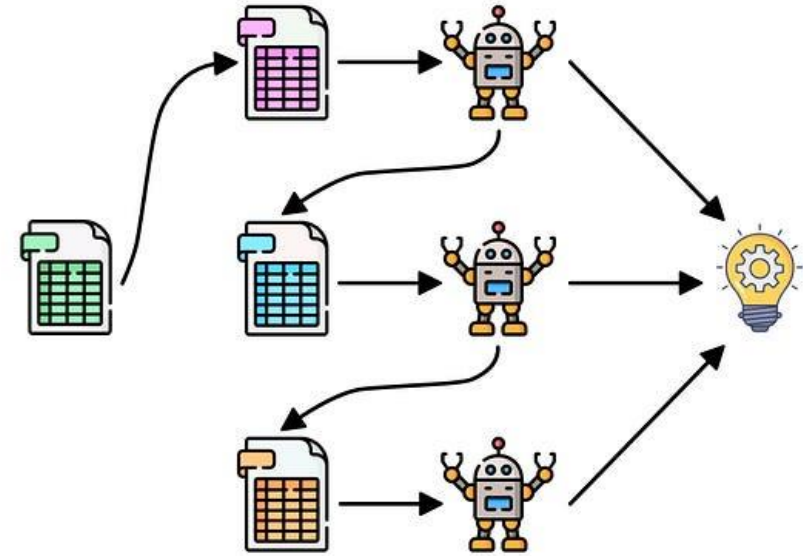
Реальный пример работы бустинга — поисковые системы Яндекса.

Bagging



Parallel

Boosting



Sequential