Лекция 5. **Деревья решений и ансамблевые методы машинного обучения**

Дисциплина: **Интеллектуальный анализ данных, текстов и изображений**
Лектор: к.т.н. **Буров Сергей Александрович**
burov-sa@ranepa.ru
29.04.2024

Вопросы лекции:

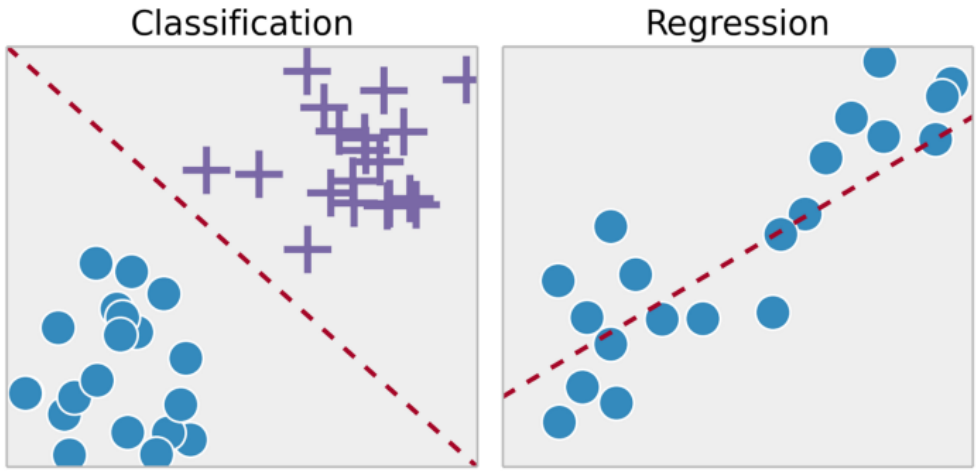
1. Классификация и регрессия с помощью деревьев решений. Алгоритм CART
3. Ансамблевые методы машинного обучения. Бутстреп (bootstrap), беггинг (bootstrap aggregating), бустинг (boosting)
4. Классификация и регрессия с помощью алгоритма случайного леса (Random Forest)

Введение

На прошлом занятии мы начали изучать основные алгоритмы машинного обучения. Давайте ещё раз вспомним и скажем, что такое машинное обучение и какие задачи может решать

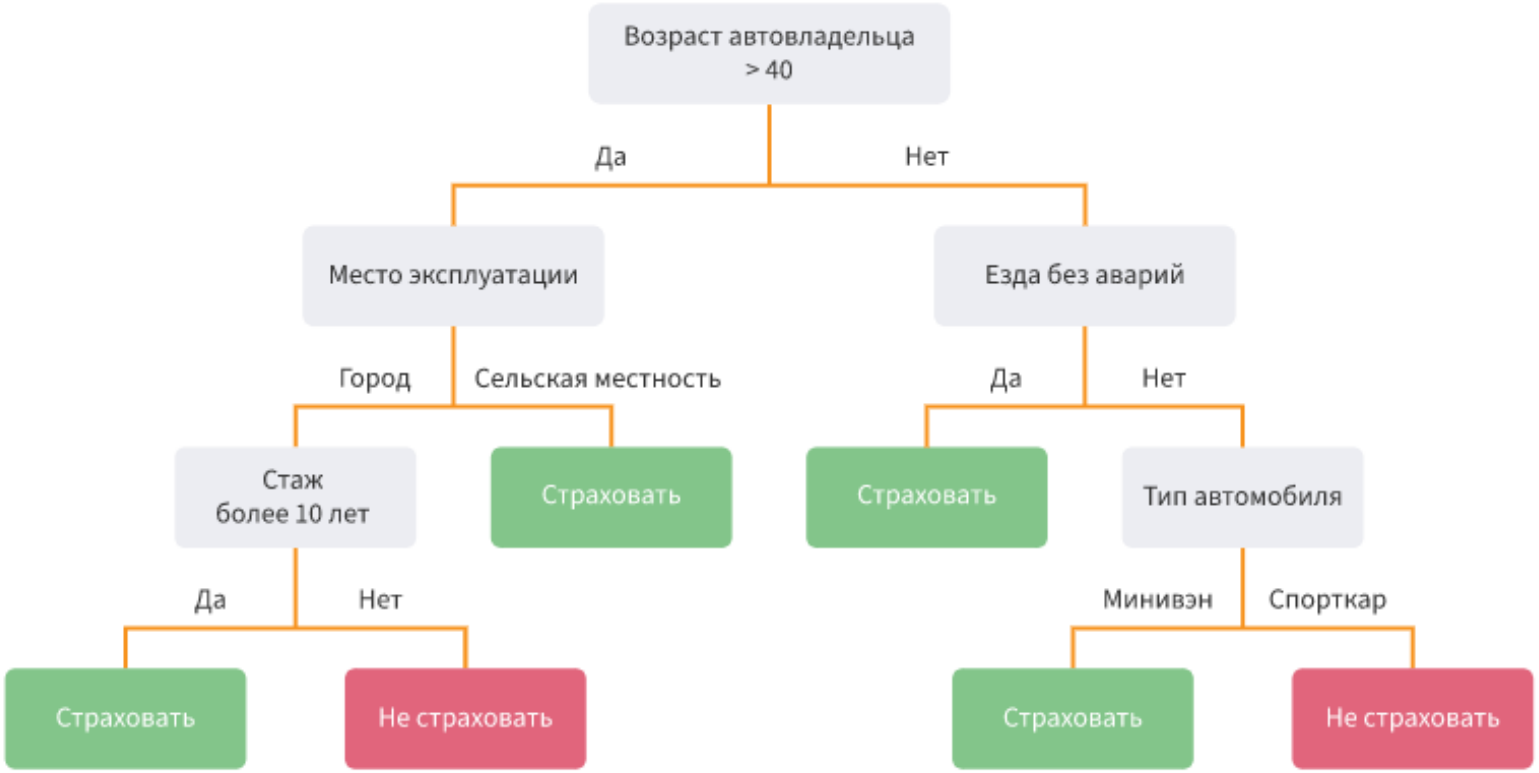
Введение

Модель (компьютерная программа) *обучается* при решении какой-то задачи из класса ***T***, если ее производительность, согласно метрике ***P***, улучшается при накоплении опыта ***E***



Введение

Сегодняшняя лекция посвящена **решающим деревьями** — семейством моделей, которые позволяют восстанавливать нелинейные зависимости произвольной сложности для решения задач регрессии и классификации



Понятие дерева решений

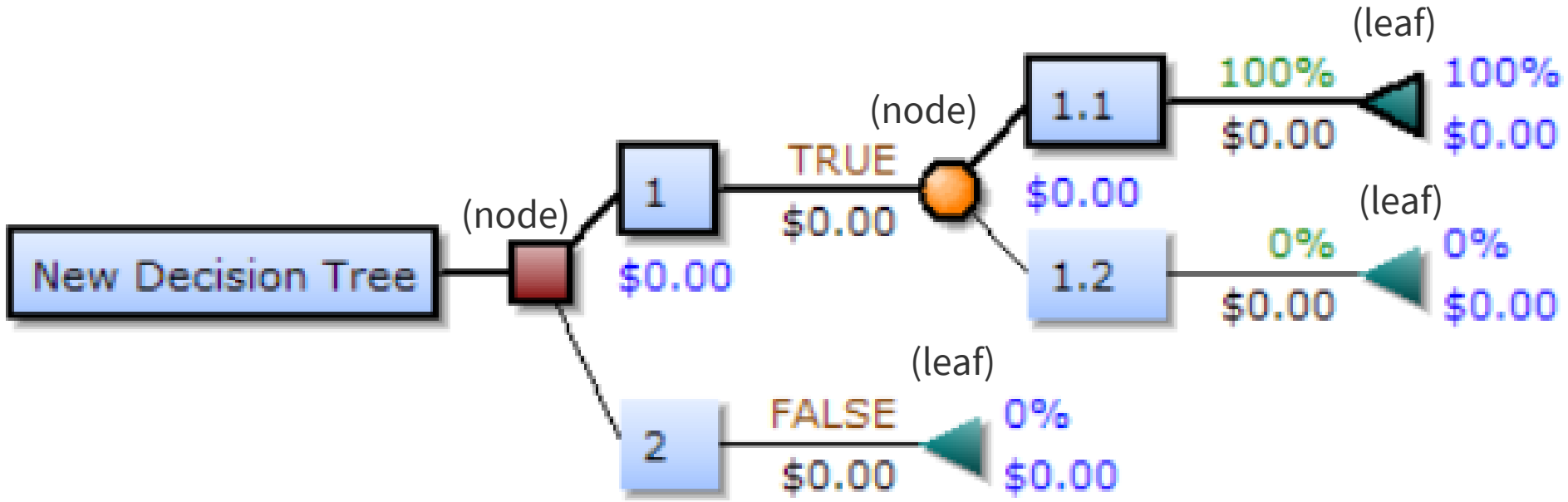
Дерево решений (дерево принятия решений, также называют **деревом классификации или регрессионным деревом,** *decision tree, DT*) — средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике.

Структура дерева представляет собой «листья» и «ветки».
На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Понятие дерева решений

Дерево решений состоит из трёх типов узлов:

- Узлы решения — обычно представлены квадратами
- Вероятностные узлы — представляются в виде круга
- Замыкающие узлы — представляются в виде треугольника

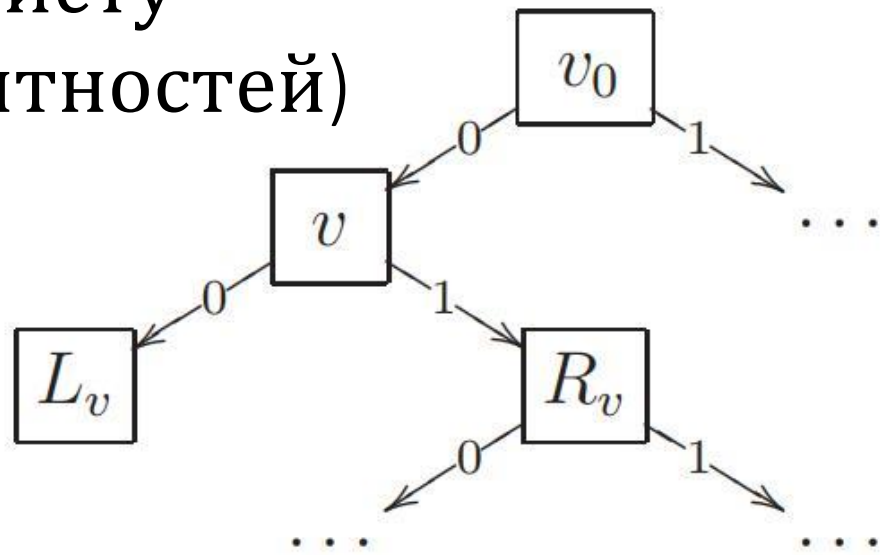


Определение бинарного дерева

Бинарные (двоичные) деревья — деревья, каждый узел которых при разбиении имеет только двух потомков.

Каждой внутренней вершине v приписана функция (или предикат) $\beta_v: X \to \{0,1\}$

Каждой листовой вершине v приписан прогноз $c_v \in Y$ (в случае с классификацией листу может быть приписан вектор вероятностей)



Основные этапы построения дерева решений

1. Выбор атрибута, по которому будет производиться разбиение в данном узле (атрибута разбиения).

2. Выбор критерия остановки обучения.

3. Выбор метода отсечения ветвей (упрощения).

4. Оценка точности построенного дерева.

Выбор атрибута разбиения

теоретико-информационный критерий

$$H = -\sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right)$$

где n — число классов в исходном подмножестве, N_i — число примеров i-го класса, N — общее число примеров в подмножестве.

H – информационная энтропия, при правильном выборе атрибута $H \to 0$

$$\mathrm{Gain}(A) = \mathrm{Info}(S) - \mathrm{Info}(S_A),$$

где $\mathrm{Info}(S)$ — информация, связанная с подмножеством S до разбиения, $\mathrm{Info}(S_A)$ — информация, связанная с подмножеством, полученными при разбиении по атрибуту A.

H – прирост информации, при правильном выборе атрибута ***Gain(A)*** $\to max$

Выбор атрибута разбиения
статистический подход

Основа подхода — индекс Джини (назван в честь итальянского статистика и экономиста Коррадо Джини) - насколько часто случайно выбранный пример обучающего множества будет распознан неправильно, при условии, что целевые значения в этом множестве были взяты из определённого статистического распределения.

Индекс Джини (Jini) фактически показывает расстояние между двумя распределениями — распределением целевых значений, и распределением предсказаний модели.

$$\mathrm{Gini}(Q) = 1 - \sum_{i=1}^{n} p_i^2,$$

где Q — результирующее множество, n — число классов в нём, p_i — вероятность i-го класса (выраженная как относительная частота примеров соответствующего класса).

Выбор критерия остановки алгоритма

1. **Ранняя остановка** — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров. Единственным преимуществом подхода является снижение времени обучения. Главным недостатком является то, что ранняя остановка всегда делается в ущерб точности дерева, поэтому многие авторы рекомендуют отдавать предпочтение отсечению ветвей.

2. **Ограничение глубины дерева** — задание максимального числа разбиений в ветвях, по достижении которого обучение останавливается. Данный метод также ведёт к снижению точности дерева.

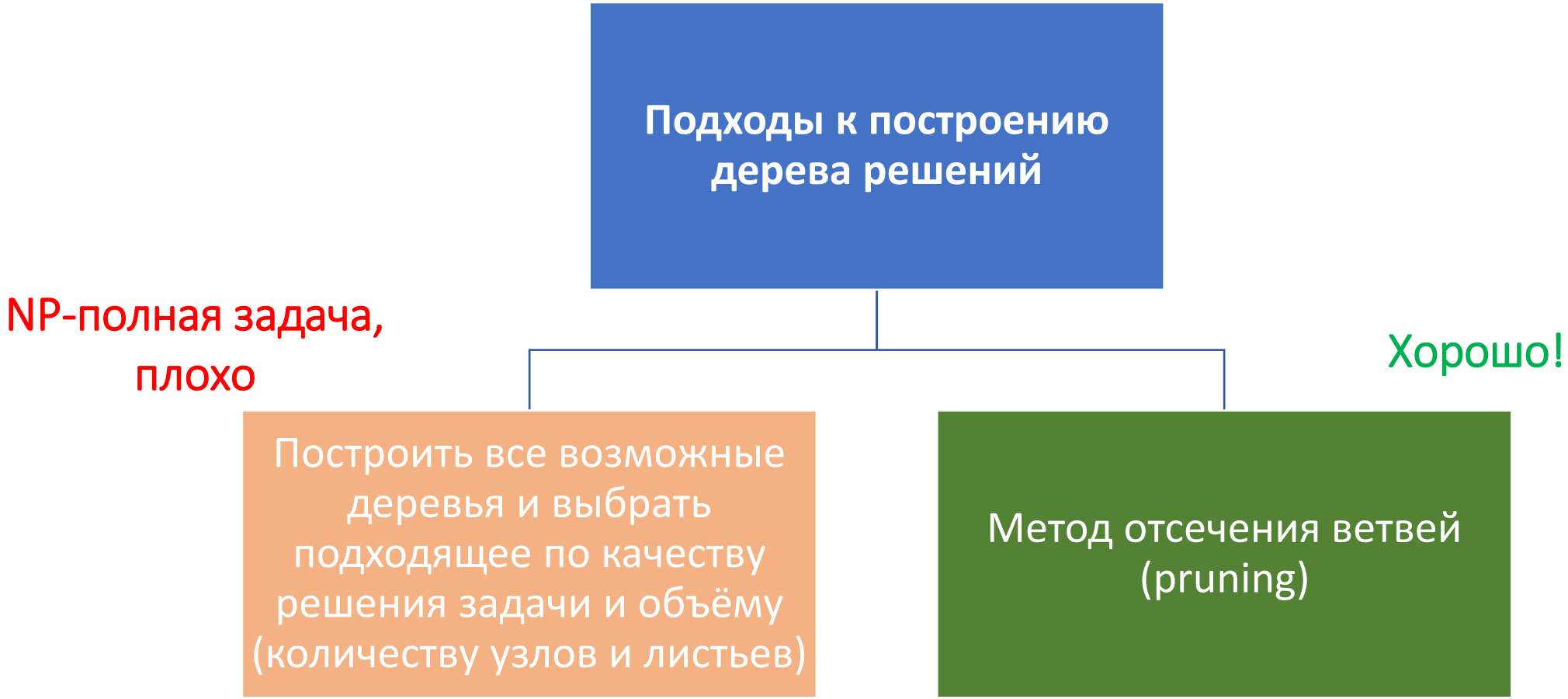
3. **Задание минимально допустимого** число примеров в узле — запретить алгоритму создавать узлы с числом примеров меньше заданного (например, 5). Это позволит избежать создания тривиальных разбиений и, соответственно, малозначимых правил.

Все критерии эвристические

Выбор метода отсечения ветвей

Большое количество узлов и листьев - плохо

Малое количество узлов и листьев – хорошо!



Выбор метода отсечения ветвей
Pruning

1. Построить полное дерево (чтобы все листья содержали примеры одного класса).

2. Определить два показателя: относительную точность модели — отношение числа правильно распознанных примеров к общему числу примеров, и **абсолютную ошибку** — число неправильно классифицированных примеров.

3. **Удалить** из дерева листья и узлы, отсечение которых не приведёт к значимому уменьшению точности модели или увеличению ошибки

Отсечение ветвей производится в направлении, противоположном направлению роста дерева, т.е. снизу вверх, путём последовательного преобразования узлов в листья.

Преимуществом отсечения ветвей по сравнению с ранней остановкой является возможность поиска оптимального соотношения между точностью и понятностью дерева.

Недостатком является большее время обучения из-за необходимости сначала построить полное дерево.

Достоинства метода дерева решений

- ✓ Прост в понимании и интерпретации.
- ✓ Не требует специальной подготовки данных, как например: нормализации данных добавления фиктивных переменных, а также удаления пропущенных данных.
- ✓ Способен работать как с категориальными, так и с интервальными переменными. (Прочие методы работают лишь с теми данными, где присутствует лишь один тип переменных. Например, метод отношений может быть применён только на номинальных переменных, а метод нейронных сетей только на переменных, измеренных по интервальной шкале.)
- ✓ Использует модель «белого ящика»,
- ✓ Позволяет оценить модель при помощи статистических тестов
- ✓ Метод хорошо работает даже в том случае, если были нарушены первоначальные предположения, включённые в модель.
- ✓ Позволяет работать с большим объёмом информации без специальных подготовительных процедур

Недостатки метода дерева решений

- ✓ Проблема получения оптимального дерева решений является NP-полной задачей, с точки зрения некоторых аспектов оптимальности даже для простых задач. Практическое применение алгоритма деревьев решений основано на эвристических алгоритмах, не способных обеспечить оптимальность всего дерева в целом.
- ✓ В процессе построения дерева решений могут создаваться слишком сложные конструкции, которые недостаточно полно представляют данные (переобучение).
- ✓ Для данных, которые включают категориальные переменные с большим набором уровней (закрытий), больший информационный вес присваивается тем признакам, которые имеют большее количество уровней.

Алгоритмы построения деревьев решений

В настоящее время разработано значительное число алгоритмов обучения деревья решений: ID3, CART, C4.5, C5.0, NewId, ITrule, CHAID, CN2 и т.д.

Наиболее популярные алгоритмы:

- **ID3 (Iterative Dichotomizer 3)** — алгоритм позволяет работать только с дискретной целевой переменной, поэтому деревья решений, построенные с помощью данного алгоритма, являются классифицирующими. Число потомков в узле дерева не ограничено. Не может работать с пропущенными данными.
- **C4.5** — усовершенствованная версия алгоритма ID3, в которую добавлена возможность работы с пропущенными значениями атрибутов (по версии издания Springer Science в 2008 году алгоритм занял 1-е место в топ-10 наиболее популярных алгоритмов Data Mining).
- **CART (Classification and Regression Tree)** — алгоритм обучения деревьев решений, позволяющий использовать как дискретную, так и непрерывную целевую переменную, то есть решать как задачи классификации, так и регрессии. Алгоритм строит деревья, которые в каждом узле имеют только два потомка.

Алгоритм CART

Алгоритм CART (Classification and Regression Tree), решает задачи классификации и регрессии построением дерева решений. Он разработан в 1974—1984 годах четырьмя профессорами статистики: Лео Брейманом (Беркли), Джеромом Фридманом (англ.) рус. (Стэнфорд), Чарлзом Стоуном (Charles Stone, Беркли) и Ричардом Олшеном (Richard A. Olshen, Стэнфорд).

Алгоритм CART предназначен для построения бинарного дерева решений. Для алгоритма CART «поведение» объектов выделенной группы означает долю модального (наиболее частого) значения выходного признака. Выделенные группы — те, для которых эта доля достаточно высока. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров на две части — часть, в которой выполняется правило (потомок — **right**) и часть, в которой правило не выполняется (потомок — **left**)

Алгоритм CART. Правила разбиения

- 1) Вектор, подаваемый на вход дерева может содержать как порядковые так и категориальные переменные.
- 2) В каждом узле разбиение идет только по *одной переменной*.

2.1) Если переменная числового типа, то в узле формируется правило вида $x_i <= c$. Где c – некоторый порог, который чаще всего выбирается как среднее арифметическое двух соседних *упорядоченных* значений переменной x_i обучающей выборки.

2.2) Если переменная категориального типа, то в узле формируется правило $x_i \in V(x_i)$, где $V(x_i)$ – некоторое непустое подмножество множества значений переменной x_i в обучающей выборке.

Следовательно, для n значений числового атрибута алгоритм сравнивает $n-1$ разбиений, а для категориального $(2^{n-1} – 1)$.

Достоинства алгоритма CART

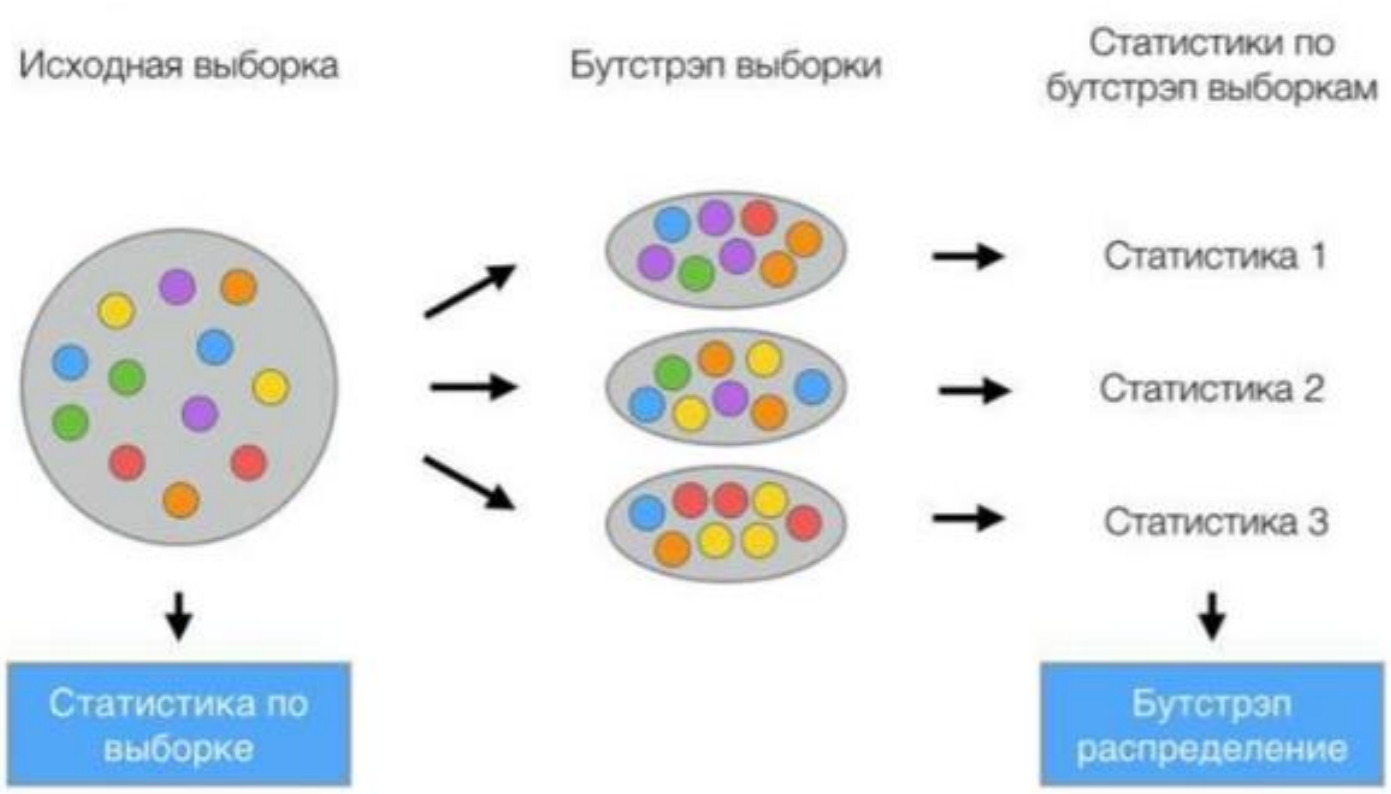
1. Данный метод является непараметрическим, это значит, что для его применения нет необходимости рассчитывать различные параметры вероятностного распределения.
2. Для применения алгоритма CART нет необходимости заранее выбирать переменные, который будут участвовать в анализе: переменные отбираются непосредственно во время проведения анализа на основании значения индекса Gini.
3. CART легко борется с выбросами: механизм «разбиения» (от англ. splitting), заложенный в алгоритме просто помещает «выбросы» в отдельный узел, что позволяет очистить имеющиеся данные от шумов.
4. Для применения этого алгоритма не надо принимать в расчет никаких предположений или допущений перед проведением анализа.
5. Большим преимуществом является скорость работы алгоритма.

Недостатки алгоритма CART

1. **Деревья решений, предложенные алгоритмом, не являются стабильными**: результат, полученный на одной выборке, бывает не воспроизводим на другой (дерево может увеличиваться, уменьшаться, включать другие предикторы и т.д.)
2. В случае, когда необходимо построить **дерево с более сложной структурой**, лучше использовать **другие алгоритмы**, так как CART может не идентифицировать правильную структуру данных.

Понятие бутстрэпа

Бутстрэп (Bootstrap) – разбиение исходной выборки на подвыборки (разбитие данных на части)

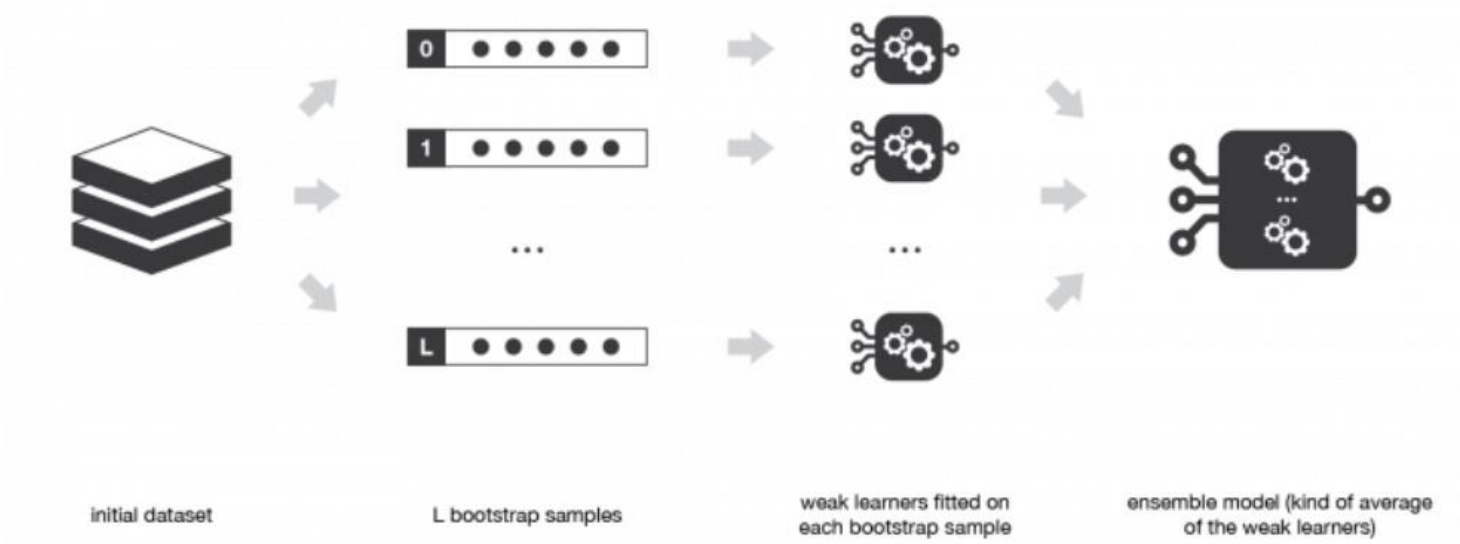


- ✓ Позволяет просто и быстро оценивать самые разные статистики (доверительные интервалы, дисперсию, корреляцию и т.д.) для сложных моделей.
- ✓ Бутстрэп-выборки оказываются очень эффективны в оценке распределений на маленьких датасетах.

Понятие бэггинг

Бэггинг (begging – bootstrap agregation) – ансамбль моделей, каждая из которых обучена отдельно.

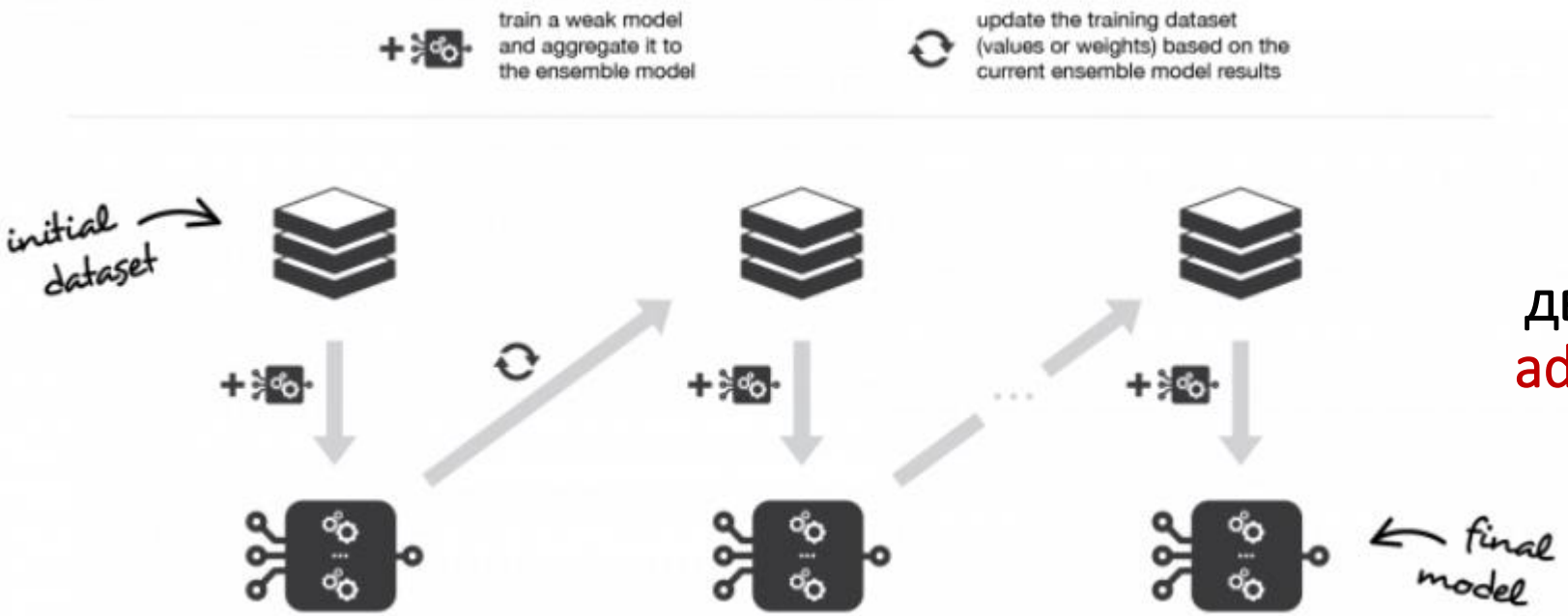
Основная идея – голосование - **усредняем** результаты нескольких моделей



Базовые алгоритмы, обученные на различных подвыборках, дают разные результаты. Бэггин позволяет компенсировать их ошибки

Понятие бустинга

Бустинг (boosting - усиление) – процедура последовательного гаправленного построения ансамбля моделей машинного обучения, когда каждый следующий алгоритм стремится компенсировать ошибки предыдущих алгоритмов.



два важных алгоритма бустинга: adaboost (адаптивный бустинг) и градиентный бустинг.
В данном курсе не рассматриваются

Алгоритм случайного леса (Random Forest)

Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев.

Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Алгоритм случайного леса (Random Forest)

В задаче регрессии – результаты деревьев решений усредняются
В задаче классификации – принимается решение голосованием по большинству.

Схема построения случайного леса:

1. **Бутстрэп.** Разбиваем исходный датасет на подвыборки.
2. **Бэггинг.** Для каждой подвыборки строится дерево (для каждого дерева — своя подвыборка).
3. Для построения каждого расщепления в дереве просматриваем max_features случайных признаков (для каждого нового расщепления — свои случайные признаки). Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.

Алгоритм случайного леса (Random Forest)

Достоинства

- ✓ Способность эффективно обрабатывать данные с большим числом признаков и классов.
- ✓ Нечувствительность к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков.
- ✓ Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки. Существуют методы построения деревьев по данным с пропущенными значениями признаков.
- ✓ Существуют методы оценивания значимости отдельных признаков в модели.
- ✓ Внутренняя оценка способности модели к обобщению (тест по неотобранным образцам).
- ✓ Высокая параллелизуемость и масштабируемость.

Алгоритм случайного леса (Random Forest)

Недостатки

Большой размер получающихся моделей.

Требуется O(K)} памяти для хранения модели, где K — число деревьев.

Заключение

Перерыв до 20.25