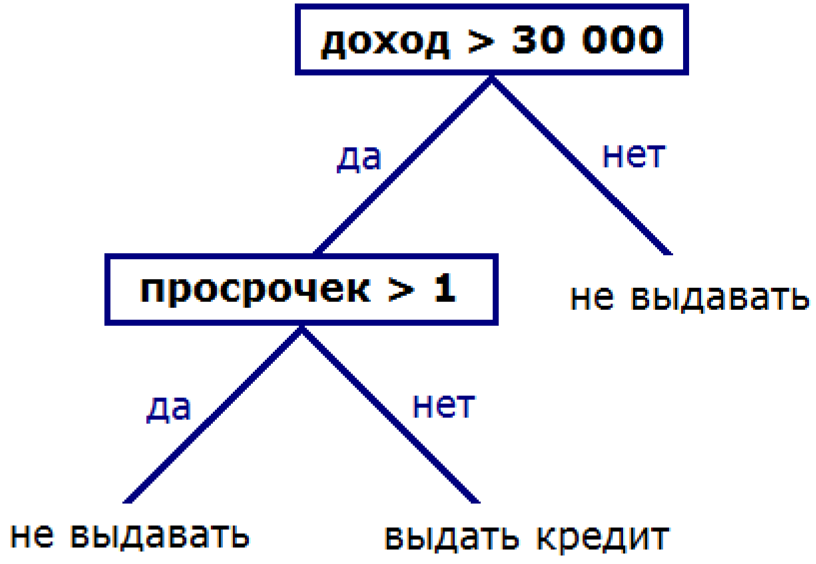
«Машинное обучение»

Решающие деревья

Александр Дьяконов

1 ноября 2021 года

Решающее дерево (Decision Tree)



лист или терминальная вершина (leaf / terminal node)
← метка (вероятности меток)

внутренняя вершина(internal node)
← ветвление, предикат (признак, порог)

дуга ← значение предиката

CART = Classification and Regression Trees

бинарные деревья (binary trees) – каждая вершина имеет двух потомков
дальше, в основном, рассматриваем их

Какие бывают предикаты / ветвления в бинарных деревьях

Предикат может быть любым…

проблема в построении оптимального дерева для конкретного предиката

Для вещественного признака обычно

$$P(x \mid i, \theta) = I[f_i(x) \le \theta]$$

Для категориального признака обычно

$$P(x \mid i, C) = I[f_i(x) \in C]$$

«косые деревья»

oblique decision trees / binary space partition trees (BSP trees)

$$P(x \mid \{w_i\}_i, \theta) = I\left[\sum_i w_i f_i(x) \le \theta\right]$$

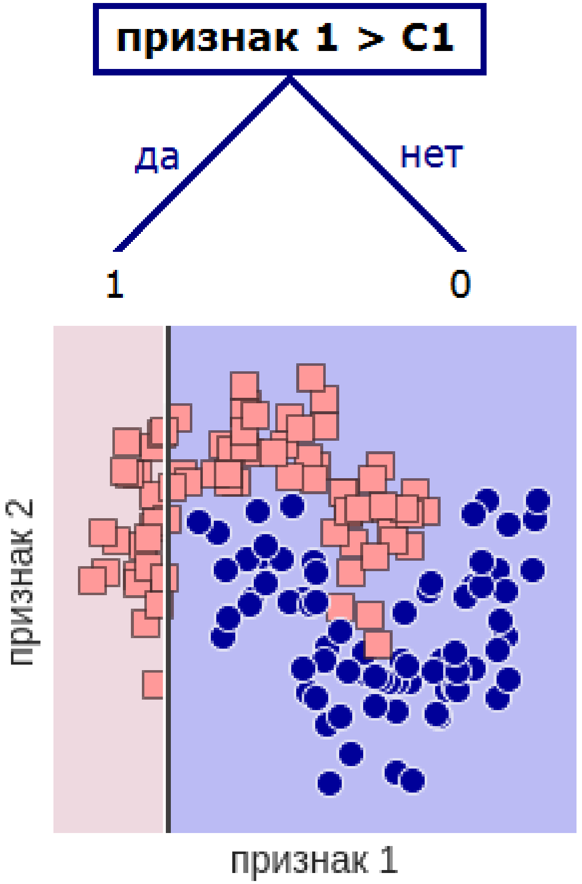
«сферические деревья»

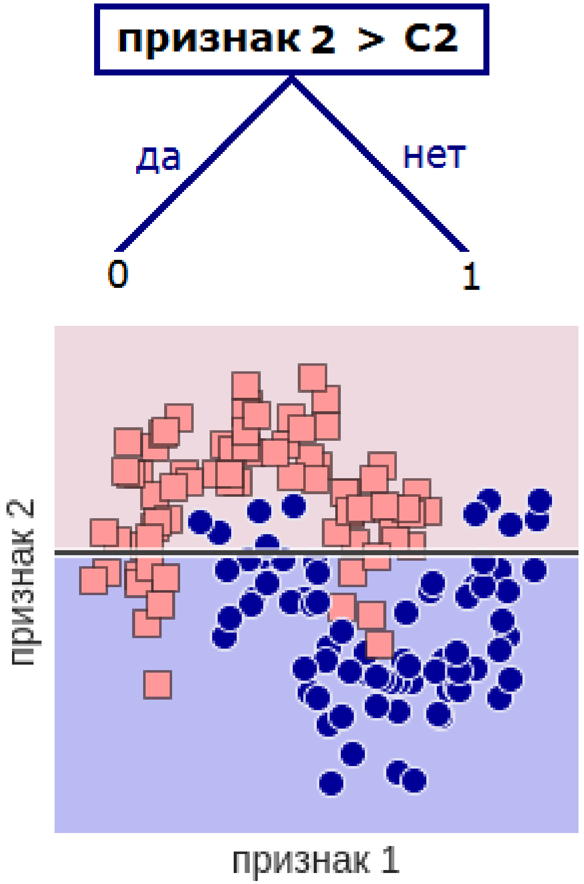
sphere trees

$$P(x \mid \{z_i\}_i, \theta) = I\left[\sum_i (f_i(x) - z_i)^2 \le \theta^2\right]$$

Разбиение на области

Расщепление по переменной (splitting) ⇒
разбиение (stratifying / segmenting) на области (регионы)





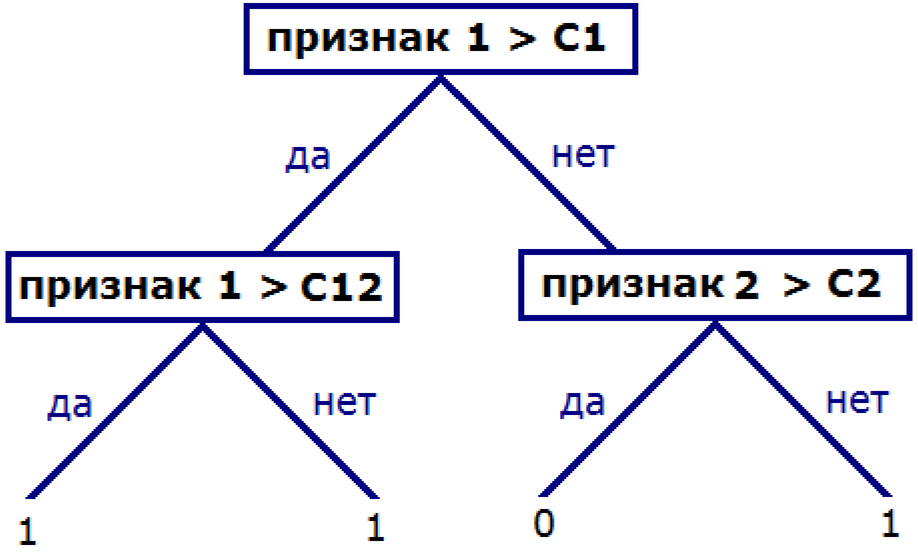
– это, кстати, «решающие пни» (decision stumps)

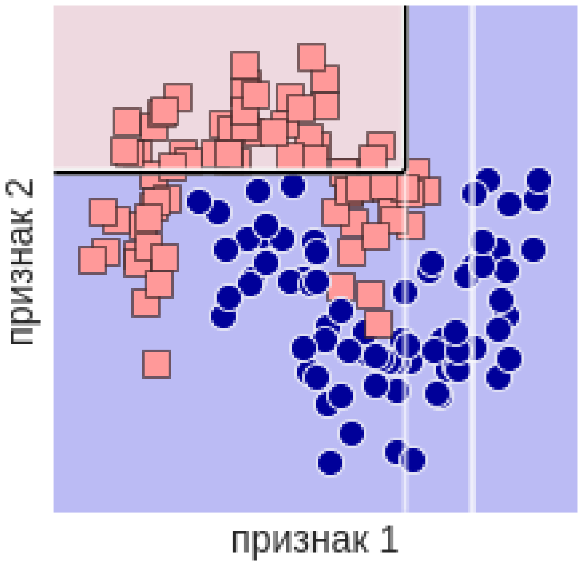
Решающее дерево как функция – кусочно-постоянная

$$a(x)=\sum_j a_{R_j} I[x\in R_j]$$

$$\bigcup_j R_j=\aleph$$

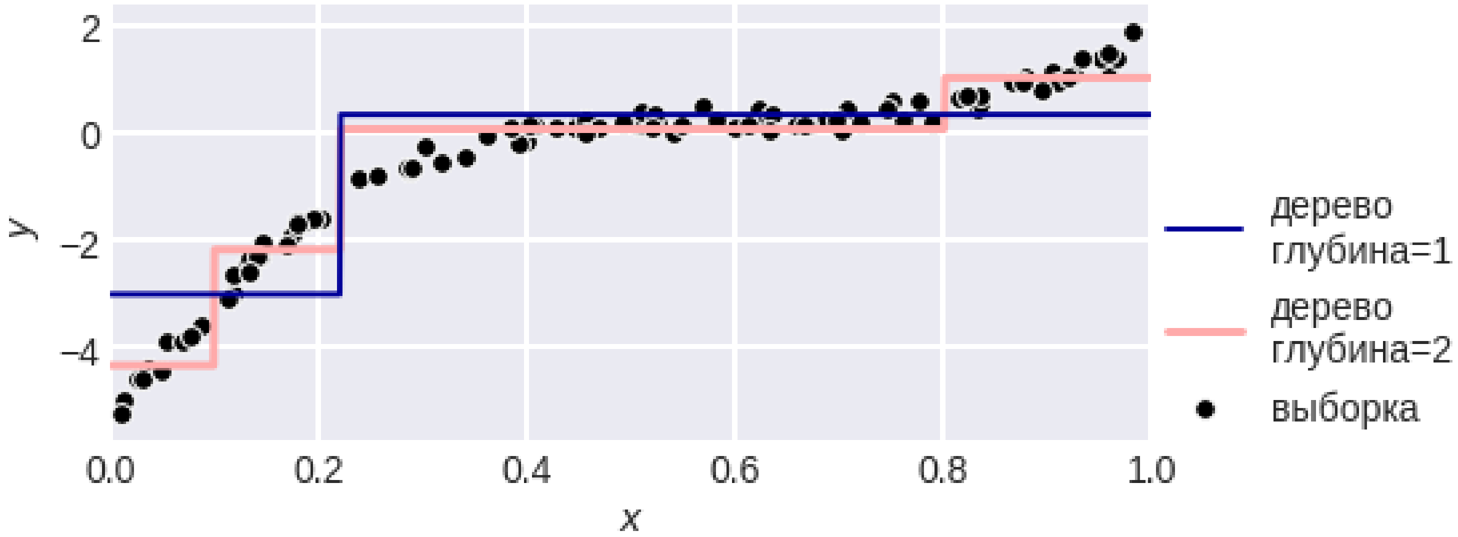
$$R_i\cap R_j=\varnothing\ \forall i\neq j$$

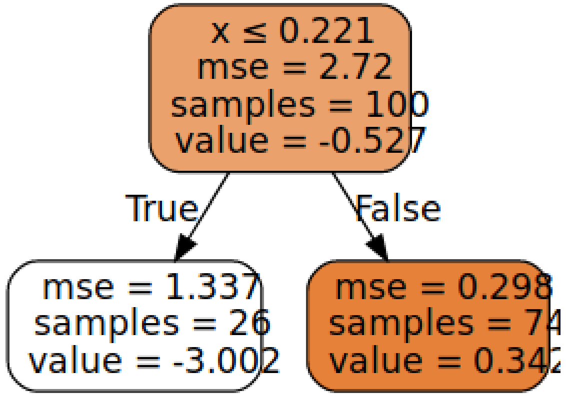


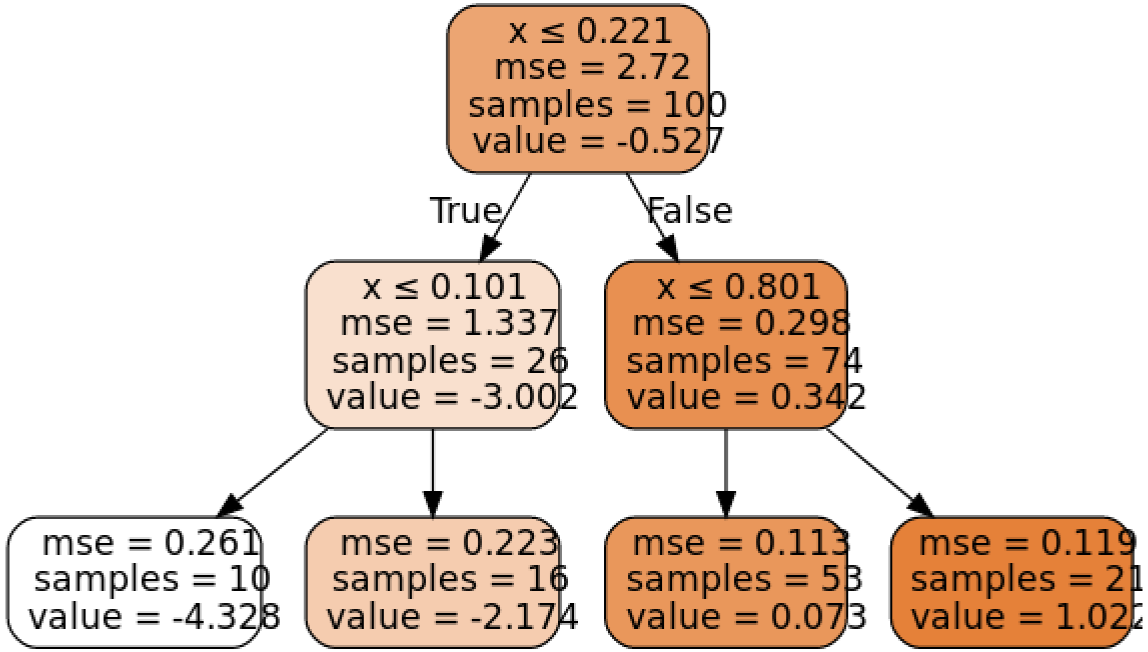


в соседних регионах метки могут быть одинаковые

Решающее дерево в задаче регрессии







$$y = -3.002 I[x \leq 0.221] + 0.342 I[x > 0.221]$$

Построение дерева: общая задача

В идеале в задаче регрессии с MSE
нужно решить такую задачу минимизации:

$$\sum_i \sum_j I[x_i \in R_j](y_i - a_{R_j})^2 \to \min,$$

– Residual Sum of Squares (RSS)

минимизация по всем разбиениям на области $\{R_j\}$

и по всем выборам a_{R_j}

Трудоёмко $\Rightarrow$ строим дерево «по уровням»,
последовательно минимизируя RSS
(top-down greedy approach)

Построение дерева: сверху-вниз

**Стартуя от дерева, состоящего из одной вершины,
можно проводить расщепления выбирая признак и порог так,
чтобы минимизировать RSS**

Сейчас уточним – что будем оптимизировать

**Расщепления производятся пока не выполнятся некоторые критерии останова
(ограничения на глубину дерева,
число объектов обучающей выборки в листьях,
на изменение RSS, см. дальше)**

Построение дерева: приписывание меток областям

Заметим, что <u>если зафиксировать область</u>,
то оптимальное значение метки для области
(в смысле минимизации эмпирического риска)

в задаче регрессии (MSE)

$$a_{R_j} = \frac{1}{|\{x_i : x_i \in R_j\}|} \sum_{x_i \in R_j} y_i$$

в задаче классификации (точность)

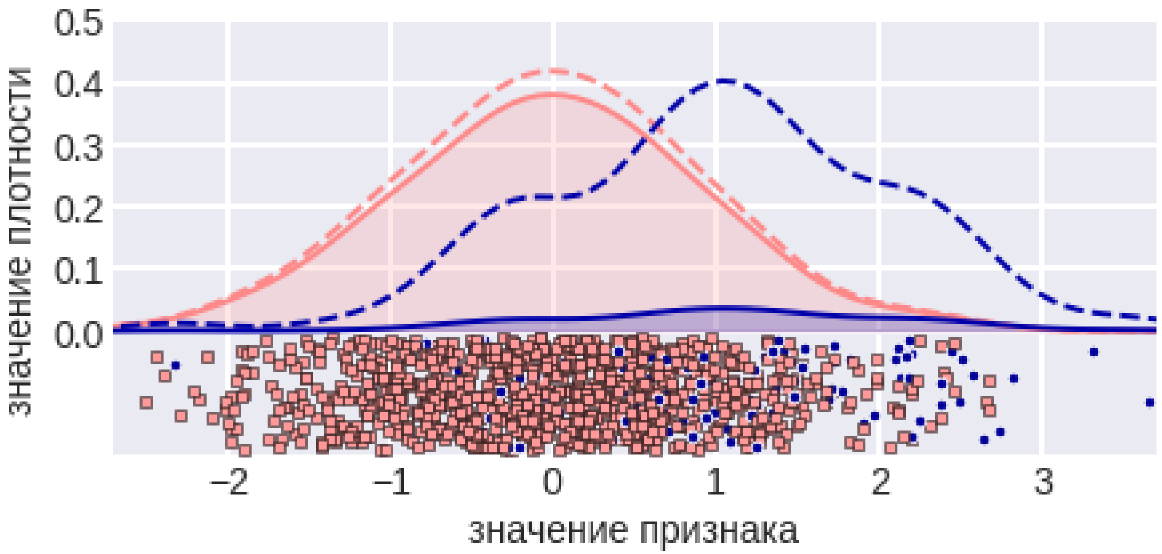
$$a_{R_j} = \mathrm{mode}(\{y_i : x_i \in R_j\})$$

поэтому именно так они и выбираются!

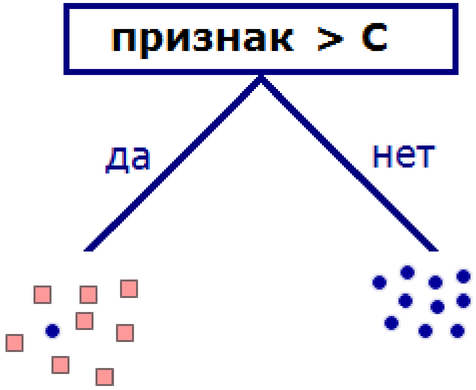
возможно другие значения для специальных функционалов качества

Построение дерева: как делать расщепления

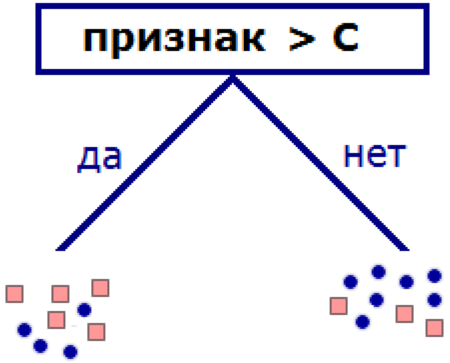
как выбрать порог для расщепления при построении дерева?



«хорошо»:



«плохо»:



Критерии расщепления в задачах классификации

Идея: ввести меру неоднородности / чистоты (impurity) множества $H(R)$

~ насколько в области «почти все объекты одного класса»

при расщеплении области R **на** R_{left} **и** R_{right} **оптимизировать**

$$Q(R,\theta) = H(R) - \frac{|R_{\text{left}}|}{|R|} H(R_{\text{left}}) - \frac{|R_{\text{right}}|}{|R|} H(R_{\text{right}})$$

Меры impurity (неоднородности / чистоты) в задачах классификации

Пусть есть область R

в ней доли объектов всех классов: $p_1,\ldots,p_l$

Missclassification criteria

$$H(R)=1-p_{\max}$$

энтропийный

$$H(R)=-\sum_j p_j\log_2 p_j$$

Джини

$$H(R)=\sum_j p_j(1-p_j)=1-\sum_j p_j^2$$

Мера неоднородности (impurity) минимальна (=0)
только если все объекты принадлежат одному классу

Критерии расщепления: частный случай двух классов

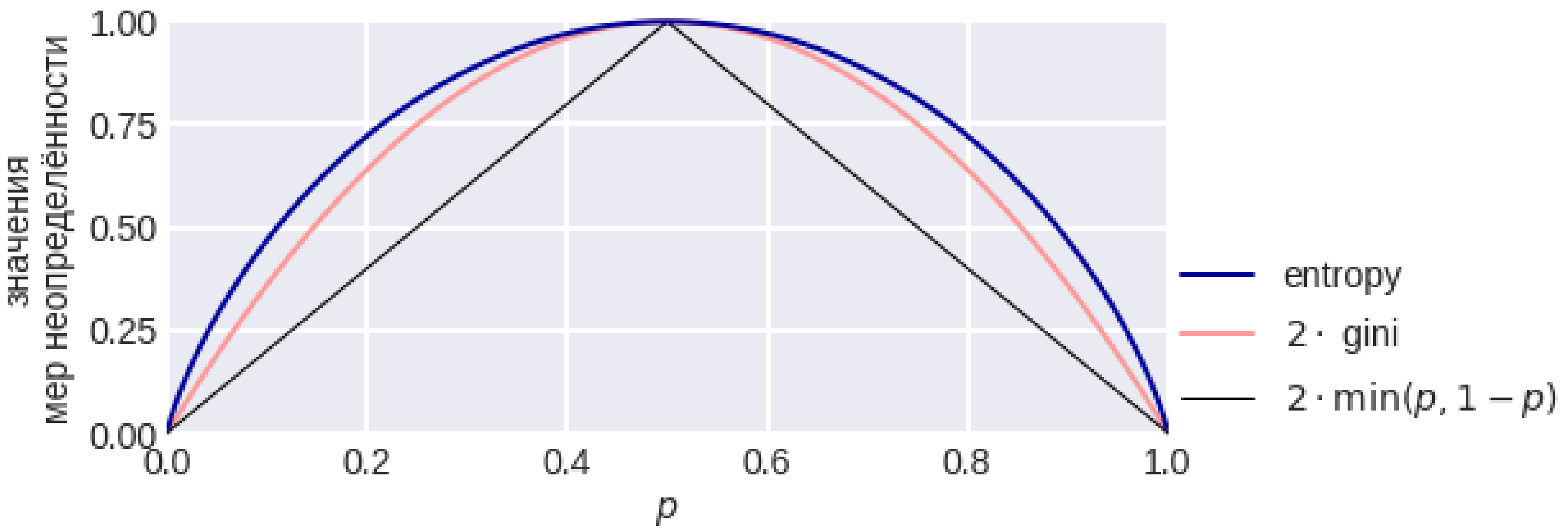
Пусть есть область R

в ней доли объектов всех классов: $p_1 = p, p_2 = 1 - p$

Missclassification criteria $H(R) = \min[p, 1-p]$

энтропийный $H(R) = -p\log_2 p - (1-p)\log_2(1-p)$

Джини $H(R) = 2p(1-p) = 1 - p^2 - (1-p)^2$



Критерии расщепления

4. AUC_ROC: будет в разделе «метрики качества»

$$Q(R,\theta)=\left|\frac{|R_{\text{right}}\cap K_0|}{|K_0|}-\frac{|R_{\text{right}}\cap K_1|}{|K_1|}\right|=$$

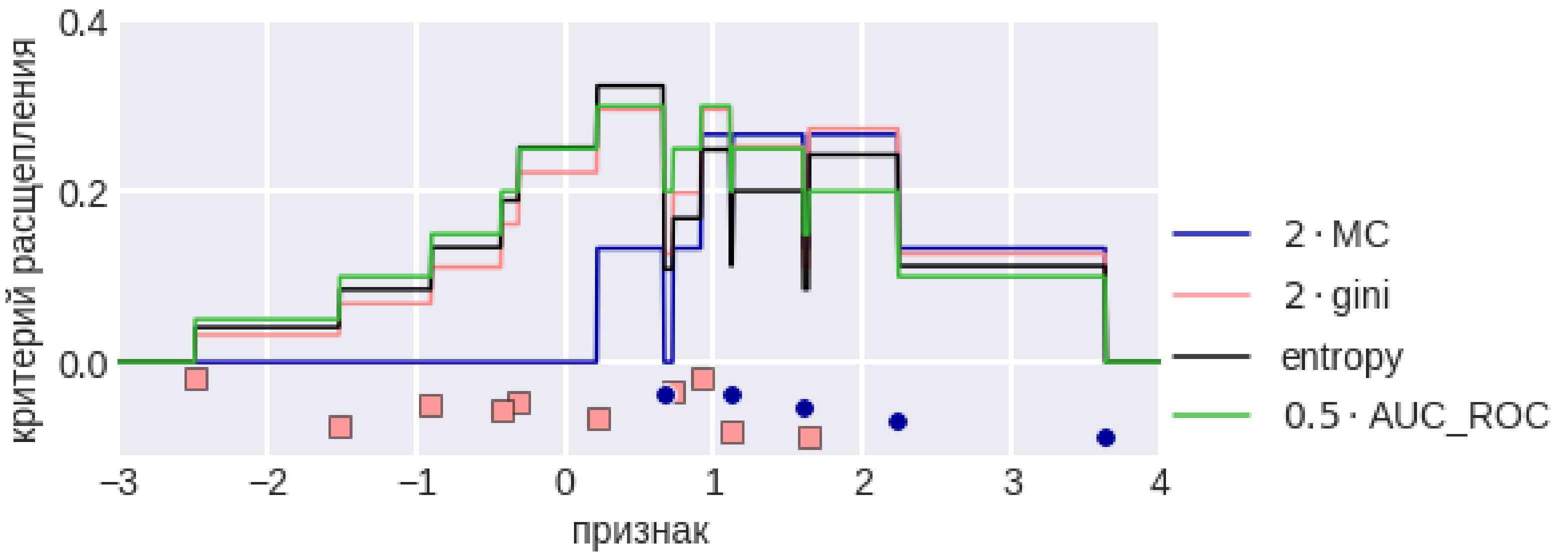
$$=\left|\frac{|R_{\text{left}}\cap K_0|}{|K_0|}-\frac{|R_{\text{left}}\cap K_1|}{|K_1|}\right|$$

5. Twoing

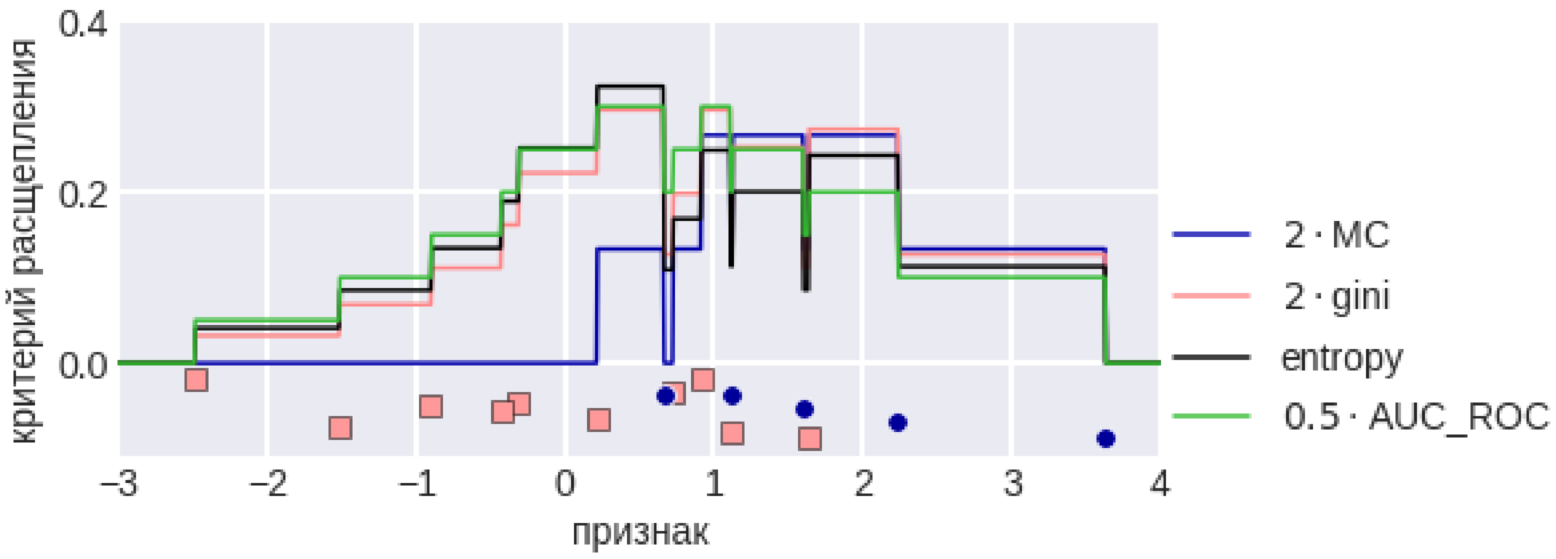
$$Q(R,\theta)=\frac{1}{4}\frac{|R_{\text{left}}|}{|R|}\frac{|R_{\text{right}}|}{|R|}\left(\sum_{j=1}^{l}\left|p_j(R_{\text{left}})-p_j(R_{\text{right}})\right|\right)^2$$

Для двух классов ~ Джини

Критерии расщепления: частный случай двух классов



Энтропия – мера неопределённости распределения



При пороге $\theta = 1$

$$\frac{|R_{\text{left}}|}{|R|} = \frac{9}{15}$$

$$\frac{|R_{\text{right}}|}{|R|} = \frac{6}{15}$$

$$H(R) = -(5/15)\log_2(5/15) - (10/15)\log_2(10/15) \approx 0.918$$

$$H(R_{\text{left}}) = -(1/9)\log_2(1/9) - (8/9)\log_2(8/9) \approx 0.503$$

$$H(R_{\text{right}}) = -(4/6)\log_2(4/6) - (2/6)\log_2(2/6) \approx 0.918$$

$$Q(R,\theta) \approx 0.918 - \frac{9}{15}0.503 - \frac{6}{15}0.918 \approx 0.249$$

Обоснование энтропийного критерия расщепления

	облачно	ясно
дождь	3	1
сухо	1	5

$$X = \{\text{дождь}, \text{сухо}\}$$
$$Y = \{\text{облачно}, \text{ясно}\}$$

Совместная энтропия

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y) =$$
$$= -\frac{3}{10}\log_2\frac{3}{10} - \frac{1}{10}\log_2\frac{1}{10} - \frac{1}{10}\log_2\frac{1}{10} - \frac{5}{10}\log_2\frac{5}{10}$$

Энтропия при условии

знаем, что дождь

$$H(Y \mid X = x) = -\sum_{x\in X}\sum_{y\in Y} p(y\mid x)\log_2 p(y\mid x) =$$
$$= -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$$

Обоснование энтропийного критерия расщепления

	облачно	ясно
дождь	3	1
сухо	1	5

$$X = \{\text{дождь}, \text{сухо}\}$$
$$Y = \{\text{облачно}, \text{ясно}\}$$

Ожидаемая условная энтропия

$$H(Y \mid X) = \sum_{x \in X} p(x) H(Y \mid X = x)$$

Энтропия при условии, что знаем X

Information Gain / Mutial Information

$$IG(Y \mid X) = H(Y) - H(Y \mid X)$$

Насколько знание X уменьшило неопределённость

Как раз это и считаем!

$$Q(R, \theta) = H(R) - \frac{|R_{\text{left}}|}{|R|} H(R_{\text{left}}) - \frac{|R_{\text{right}}|}{|R|} H(R_{\text{right}})$$

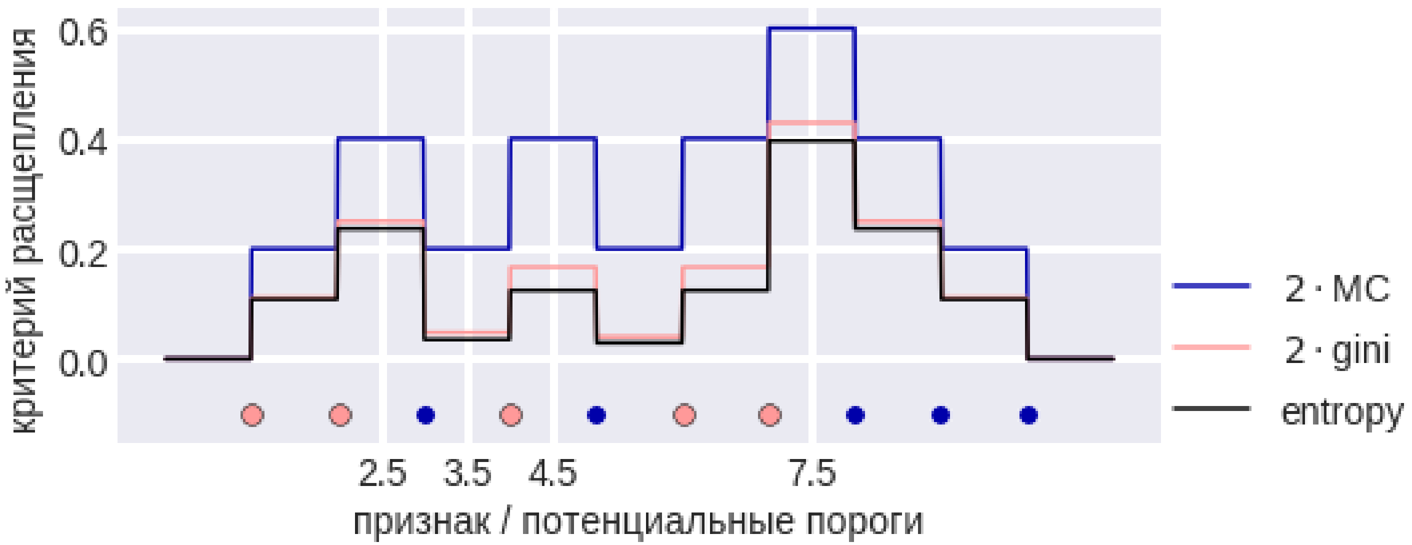
Критерии расщепления: тонкости

при выборе расщепления мы выбираем порог

- **достаточно рассматривать только «средние точки»**
- **достаточно рассматривать только «границы регионов»**

но в чём тут подвох?

для начала надо отсортировать все значения
есть проблема константных признаков



Критерии расщепления в задачах регрессии

аналогично… но тут «неоднородность» – дисперсия

$$H(R) = \mathrm{var}(\{y_i \mid y_i \in R\})$$

$$Q(R,\theta) = H(R) - \frac{|R_{\text{left}}|}{|R|} H(R_{\text{left}}) - \frac{|R_{\text{right}}|}{|R|} H(R_{\text{right}})$$

Делаем разбиение, которое максимизирует Q
Повторяем процедуру в листьях

Кстати, для бинарной случайной величины $y_i \in \{0,1\}$

$$p_1 = \overline{p},\, p_0 = 1 - \overline{p},$$
$$\mathrm{var}(\{y_i\}) = \overline{p}(1 - \overline{p})$$

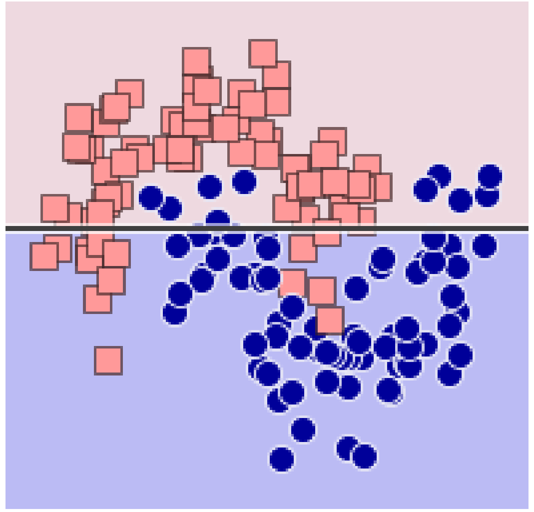
т.е. это ~ Джини!

Как долго строить дерево

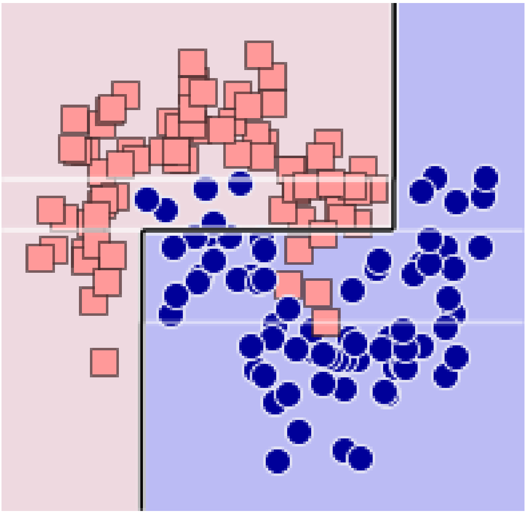
Критерии останова

- ограничение на глубину / на число листьев
- ограничение на число объектов в листьях / на число, когда делаем деление
- «естественное ограничение» (все объекты одного класса)

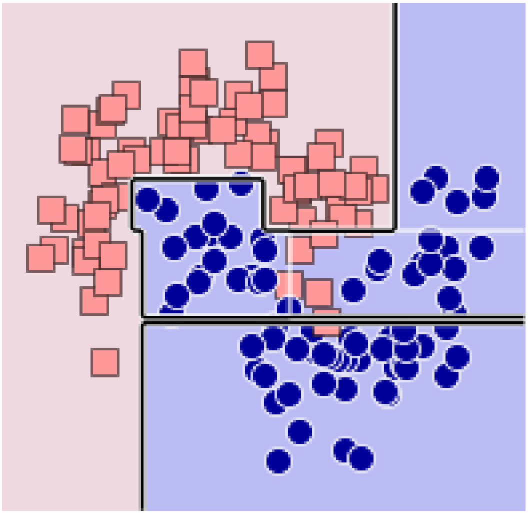
  обобщение: почти все объекты одного класса
- изменение impurity



max_depth=1

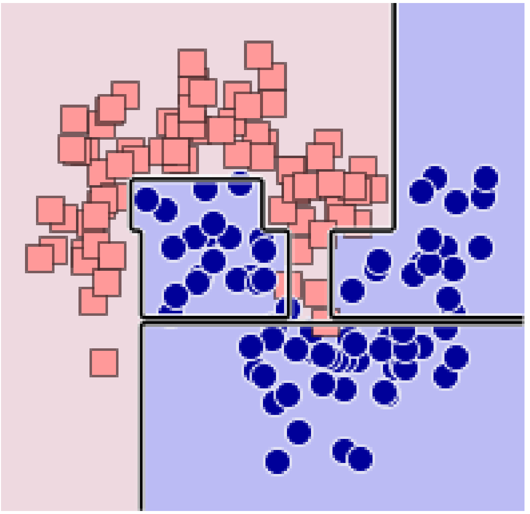


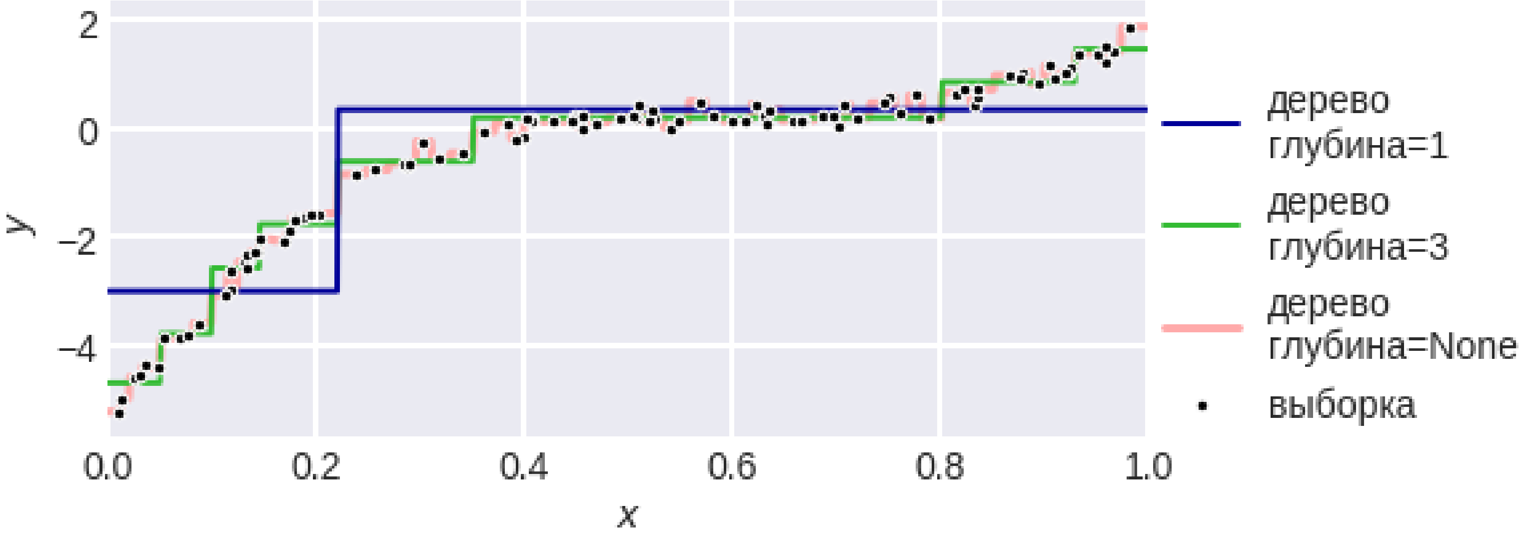
max_depth=3



max_depth=5

Минутка кода: «Решающее дерево»





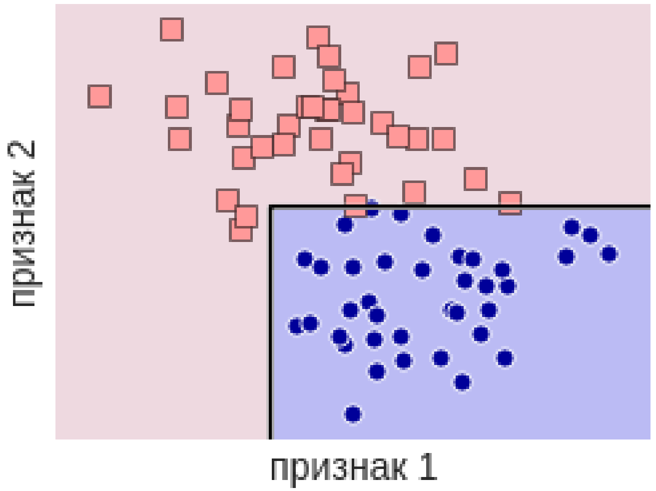

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(criterion='gini', splitter='best',
                               max_depth=None, min_samples_split=2,
                               min_samples_leaf=1,
                               min_weight_fraction_leaf=0.0,
                               max_features=None, random_state=3,
                               max_leaf_nodes=None,
                               min_impurity_decrease=0.0,
                               min_impurity_split=None, class_weight=None,
                               presort=False)
model.fit(X, y)
```

Минутка кода: «Решающее дерево»

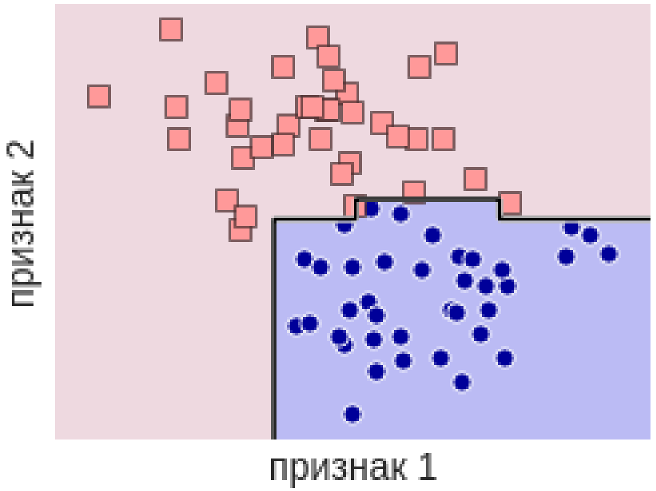
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(max_depth=10)
tree.fit(X_train, y_train)
```

max_depth=10



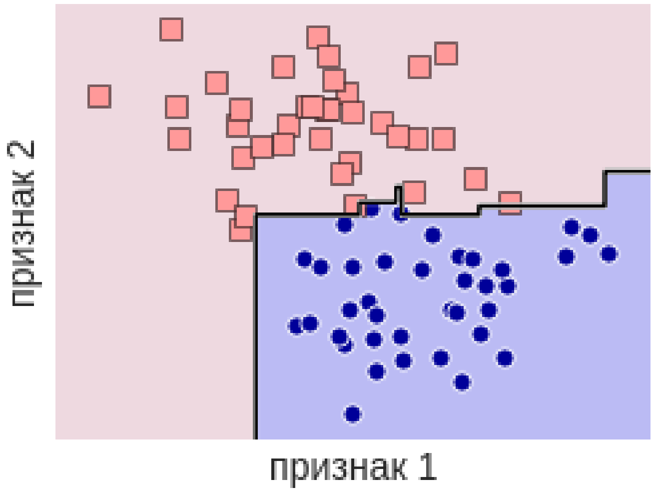
88%

splitter='random'



92%

splitter='random'
max_features=1



92%

Реализация в scikit-learn

```
sklearn.tree import DecisionTreeClassifier
```

criterion	критерий расщепления «gini» / «entropy»
splitter	разбиение «best» / «random»
max_depth	допустимая глубина
min_samples_split	минимальная выборка для разбиения
min_samples_leaf	минимальная мощность листа
min_weight_fraction_leaf	аналогично с весом
max_features	число признаков, которые смотрим для нахождения разбиения
random_state	инициализация генератора случайных чисел
max_leaf_nodes	допустимое число листьев
min_impurity_decrease	порог изменения «зашумлённости» для разбиения
min_impurity_split	порог «зашумлённости» для останова
class_weight	веса классов («balanced» или словарь, список словарей)

Особенности

Изменение impurity – порог на

$$\frac{|R|}{m}\left(H(R)-\frac{|R_{\text{left}}|}{|R|}H(R_{\text{left}})-\frac{|R_{\text{right}}|}{|R|}H(R_{\text{right}})\right)$$

Overfitting
Pre-pruning
Post-pruning

Проблема переобучения деревьев

**Глубокие деревья склонны к переобучению,
поскольку «затачиваются» на отдельные объекты**

**1. Прекращают построение достаточно рано
(см. критерии останова, stopping early)**
можно на отложенной выборке выбрать точку останова

2. Подрезают деревья (post-pruning)

3. Используют в ансамблях (например, в случайном лесе)
отдельная тема

Подрезка (post-pruning)

Сейчас используется редко – если задачу надо решить деревом
(или ансамблем из нескольких деревьев)

Раньше…

- **использовали отложенный контроль**
(удаляли листья, на которых плохое качество)

- **MDL (Minimum Description Length)**

$$\sum_{j}\sum_{x_i\in R_j}(y_i-a_{R_j})^2+\alpha\,|\{R_j\}|\to\min$$

оптимальное значение α находят с помощью скользящего контроля,
потом с этим значением параметра дерево перестраивается по всей выборке

α **регулирует баланс между стремлением обучиться и получить небольшое дерево**

Классические алгоритмы

название	ID3 = Iternative Dichotomizer Quinlan, 1986	C4.5 Quinlan, 1993	CART Breiman, Friedman, Stone, Olshen 1974-84	CHAID Kass, 1980	QUEST Loh, Shih, 1997
Критерий расщепления	Information Gain	Gain ratio, Information Gain м.б. небинарное разделение	Gini (Twoing)	Chi-square м.б. небинарное разделение	Chi-square (кат) J-way ANOVA (вещ)
Признаки	Категориальные	Вещественные и категориальные	Вещественные и категориальные		Вещественные и категориальные
Целевой признак	Категориальный	Вещественный и категориальный	Вещественный и категориальный	категориальный	категориальный
Пропуски	Не поддерживает	+	+		
Подрезка	–	+ Error Based prunning	+ Cost-complexity prunning	+	+ Post-pruning
	Остановка, когда все объекты листа одному классу или information gain <= 0	Ограничение на число объектов для расщепления	Бинарные деревья		

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf

Итог: решающие деревья

возможности

- способны обучиться на любой (непротиворечивой) выборке (при возможности построения неограниченного дерева)
- можно использовать при признаках разных типов (+ пропуски)
- можно сделать устойчивыми к выбросам
- универсальный метод – для всех типов задач машинного обучения
- встроенный отбор признаков
- нелинейный метод!

качество

- не очень высокое качество решения задачи / переобучение
- хороши в ансамблях будет в ансамблировании

Итог: решающие деревья

эффективность / стабильность

- достаточно быстро строятся
- нет ограничений на распределения признаков
- «неустойчивый алгоритм» (меняется при небольшом изменении выборки)
- плох для больших / изменяющихся данных

понимание, интерпретация и анализ

- просто объяснить неспециалисту
- ближе к человеческой логики принятия решения
- можно изобразить (на слайде)
- нет красивой аналитической формулы для модели

Итог: решающие деревья

особенности

- не использует геометрию (нет расстояний, неметрический)
- устойчив к масштабированию
- устойчив к дубликатам признаков, зависимостям в признаках и т.п.
- автоматическое решение проблемы пропусков
- неспособен к экстраполяции
- использует мало признаков!!!

Важно: эвристическое жадное обучение

(т.к. построение оптимального дерева очень сложная – NP-полная – задача)

если категориальные признаки с большим число категорий – всё сваливается на них…

Важности признаков

вспомним формулу

$$Q(R,\theta)=H(R)-\frac{|R_{\text{left}}|}{|R|}H(R_{\text{left}})-\frac{|R_{\text{right}}|}{|R|}H(R_{\text{right}})$$

– это уменьшение неоднородности при выборе такого расщепления!

Идея: чем больше признак уменьшает неоднородность, тем он важнее!

Важность признака = сумма уменьшений однородностей с помощью этого признака при построении дерева (иногда умножается на $|R|$ ~ sklearn)

Это только один из способов… (хорошо, что важность учитывается в совокупности)

- **коэффициенты в моделях**
- **OOB-оценки**
- **корреляции / функциональные зависимости и т.п.**

Деревья: проблема пропусков (Missing Values)

- удалить
- заменить (средним)
- рассматривать как отдельную категорию
- ○ **пронести в обе ветви дерева**
- ○ **выбрать наиболее подходящую ветвь дерева**

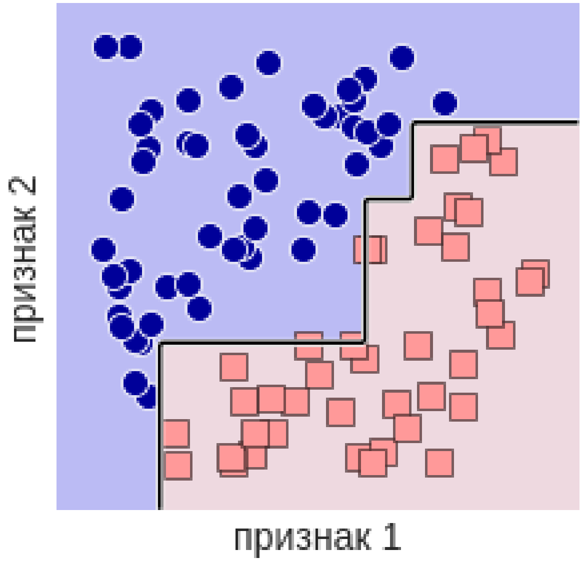
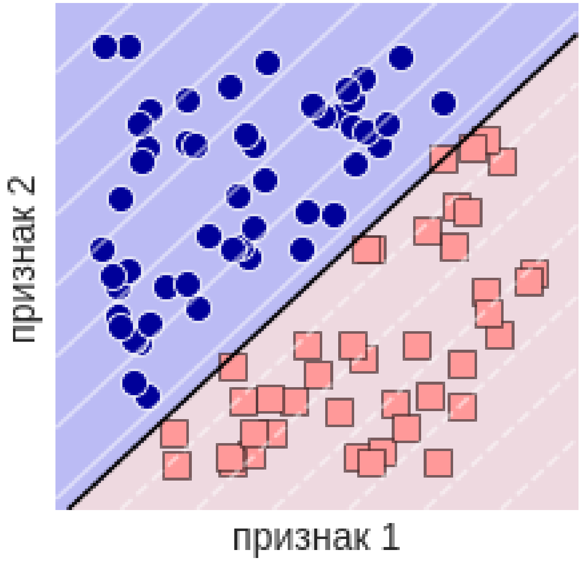
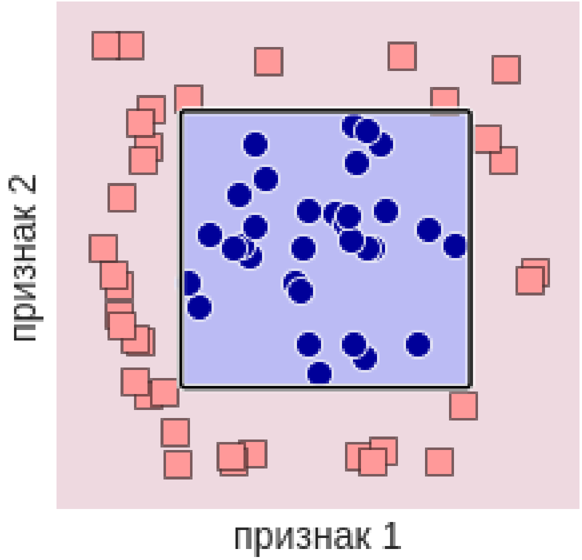
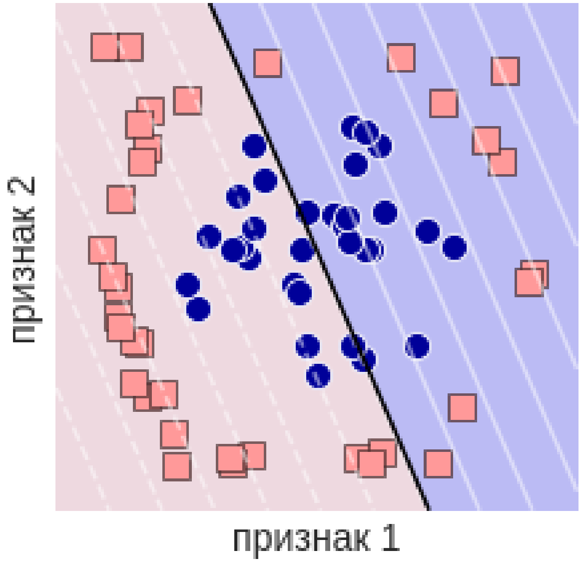
Деревья: категориальные признаки

формально при расщеплении должны рассмотреть
все подмножества множества категорий

Реально (в задаче бинарной классификации):
упорядочиваем по вероятности класса 1,
каждая категория → номер по порядку
находим для полученного числового признака оптимальное разбиение

переобучение для мелких категорий

Деревья vs линейные модели

	дерево	гиперплоскость
Линейная зависимость		
Нелинейная зависимость		

Генерация признаков с помощью деревьев

$$a(x) = \sum_j a_{R_j} I[x \in R_j]$$

нелинейный признак

$$f_{\text{new}}(x) = I[x \in R_j]$$

Литература

Ritschard, G. (2013). CHAID and Earlier Supervised Tree Methods. In J.J. McArdle & G. Ritschard (eds),Contemporary Issues in Exploratory DataMining in Behavioral Sciences, Routeledge, New York, pages 48–74

https://www.researchgate.net/profile/Gilbert_Ritschard/publication/315476407_CHAID_and_Earlier_Supervised_Tree_Methods/links/58d156a0a6fdcc3fe78521a8/CHAID-and-Earlier-Supervised-Tree-Methods.pdf

«Прогнозное моделирование в IBM SPSS Statistics, R и Python. Деревья решений и случайный лес» вышла в издательстве «ДМК-Пресс»

https://dmkpress.com/catalog/computer/data/978-5-97060-539-4