Машинное обучение и анализ данных

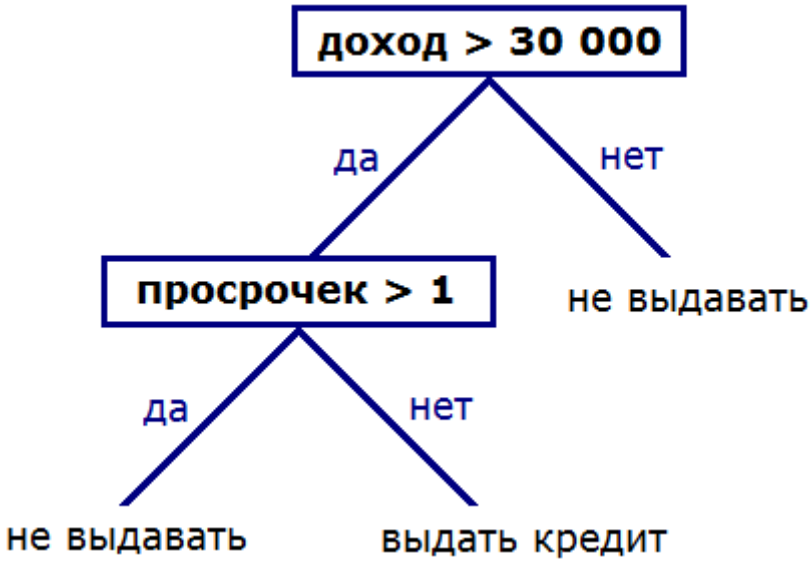
Решающие деревья

Дьяконов А.Г.

Московский государственный университет
имени М.В. Ломоносова (Москва, Россия)

Решающие деревья (Decision Trees)

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

Какие бывают предикаты / ветвления

Мы рассмотрим бинарные деревья (binary trees)

– каждая вершина имеет двух потомков

Для вещественного признака обычно

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

Для категориального признака обычно

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

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

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

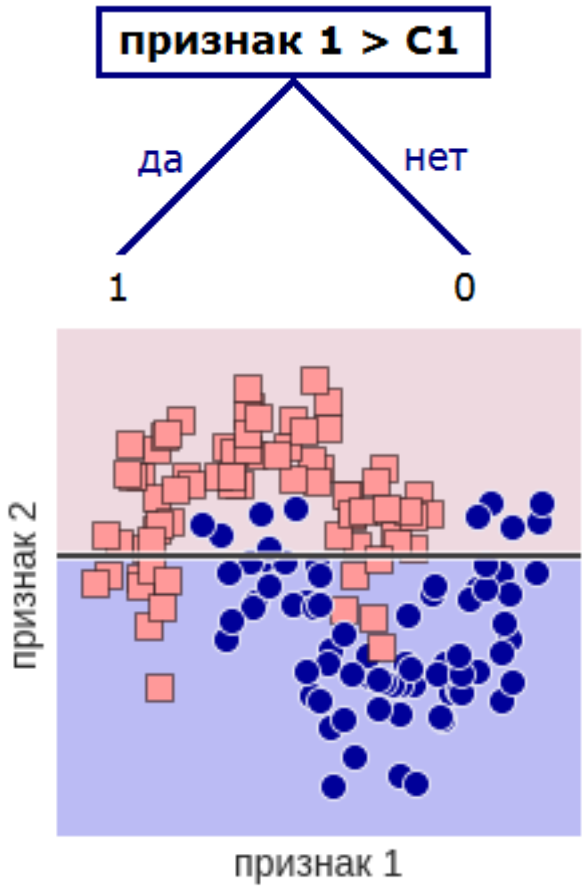
sphere trees

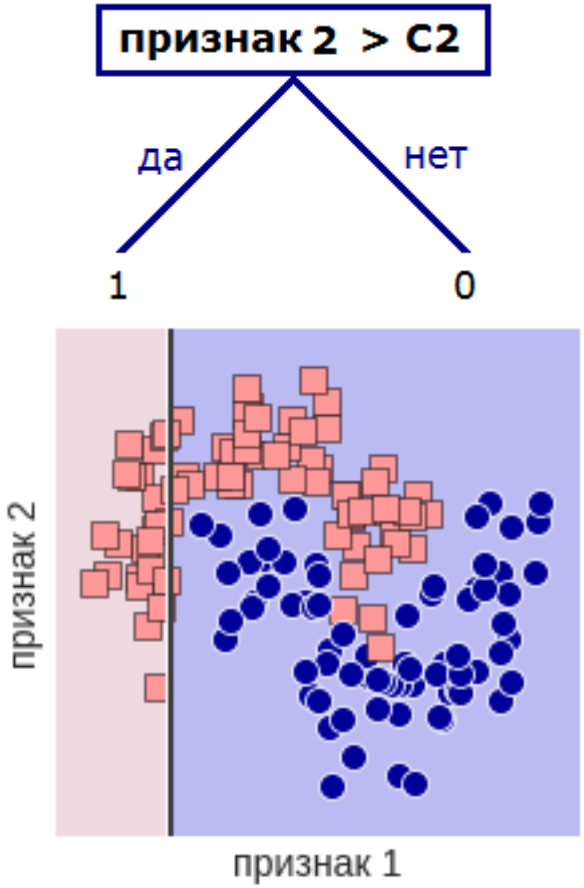
$$P(x \mid \{z_i\}_i, \theta) = I[\sum_i (f_i(x) - z_i)^2 \le \theta^2]$$

Предикат может быть любым… проблема в построении оптимального дерева для конкретного предиката.

Разбиение на области

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



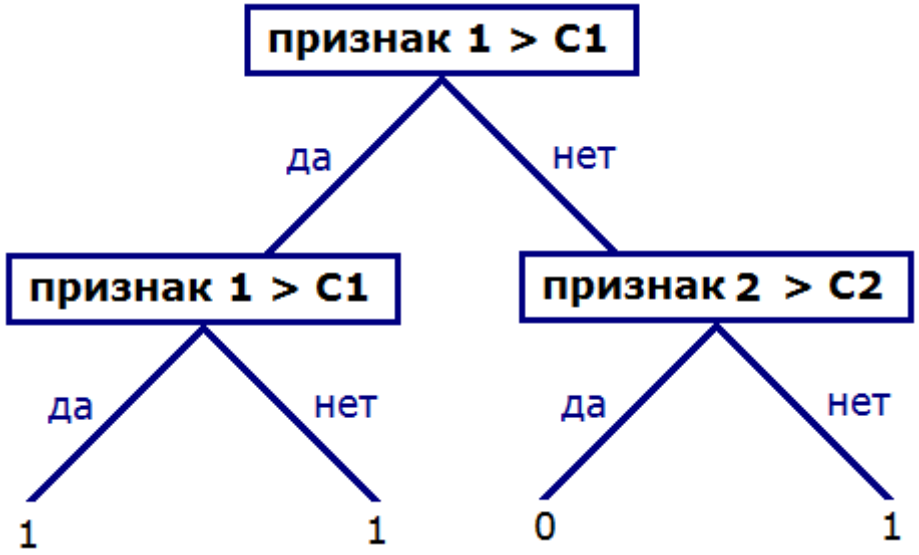


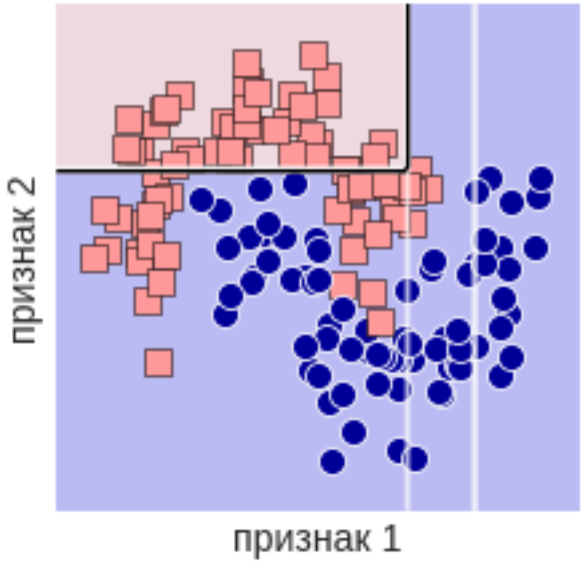
Решающая поверхность дерева – кусочно-постоянна

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

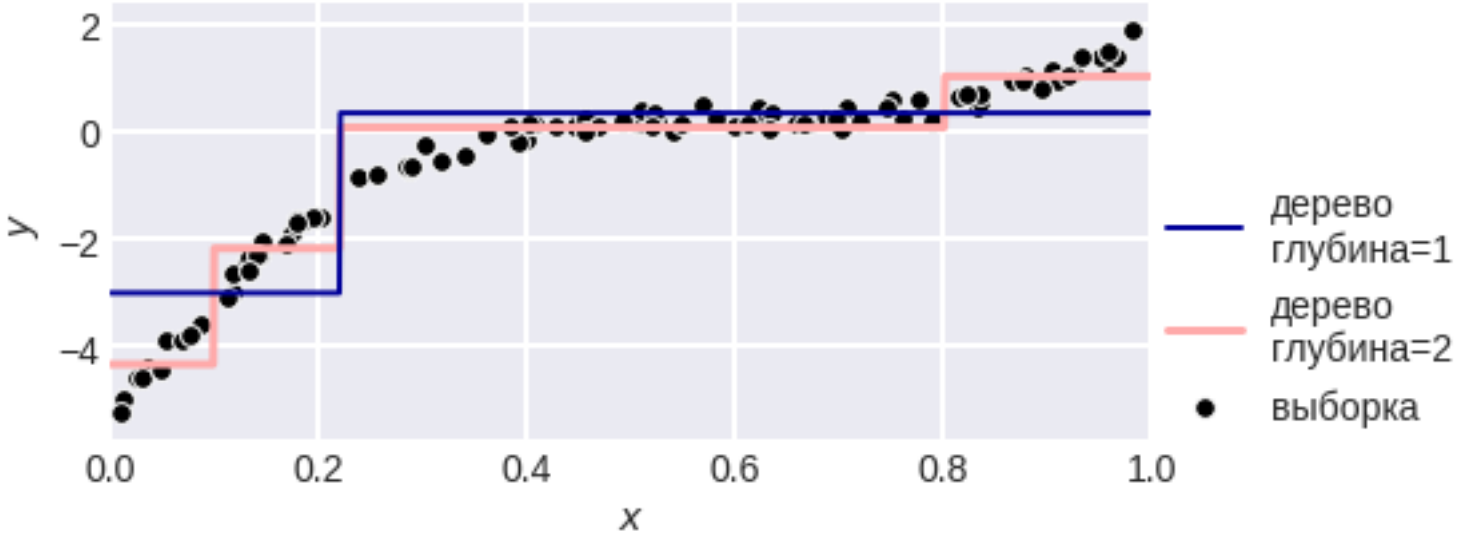
$$\bigcup_j R_j = \aleph$$

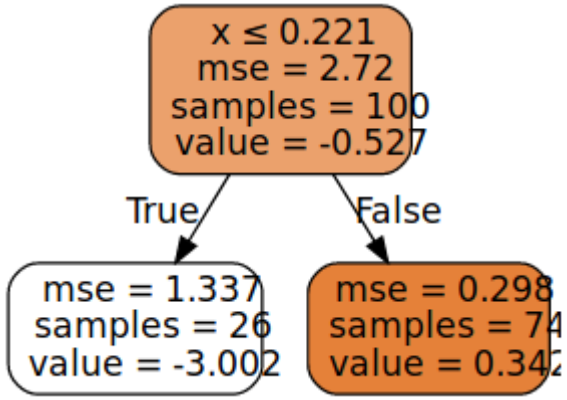
$$R_i \cap R_j = \varnothing \; \forall i \neq j$$



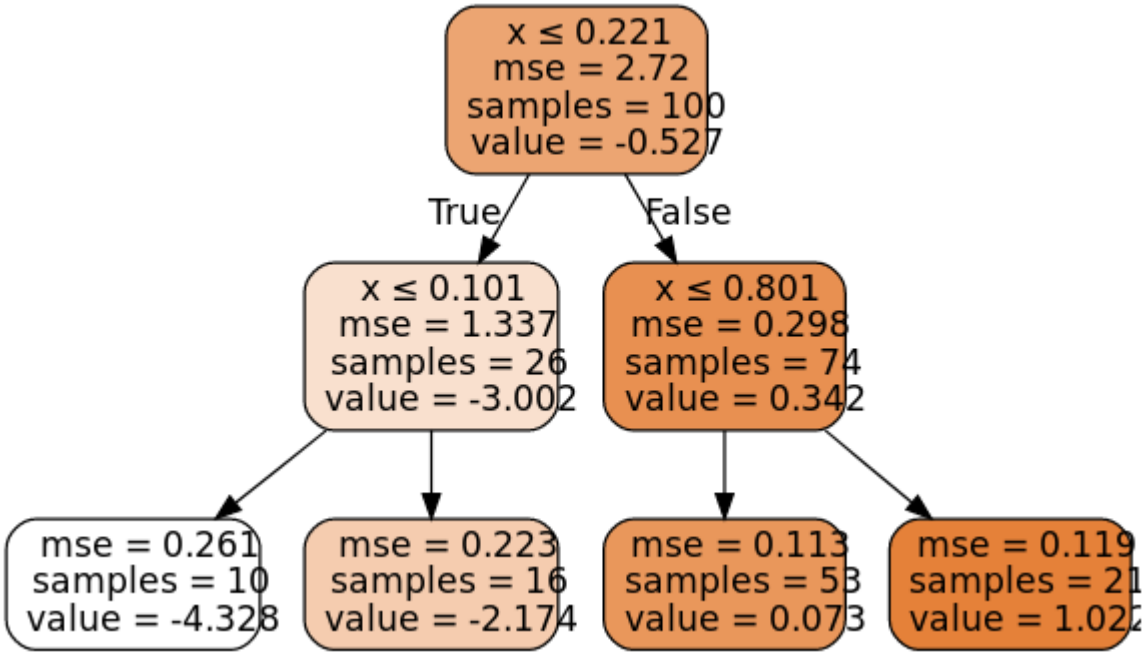


Решающее дерево в задаче регрессии





$$y = \begin{array}{l} -3.002 I[x \le 0.221] + \\ +0.342 I[x > 0.221] \end{array}$$



Построение дерева

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

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

где минимизация проводится по всем разбиениям на области $\{R_j\}$ и по всем выборам a_{R_j}.

Но это трудоёмко, поэтому последовательно минимизируем в листьях (top-down greedy approach).

Построение дерева

Заметим, что если зафиксировать области, то тут оптимальные значения

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

Ответы дерева

по объектам в листьях

В задаче регрессии

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

В задаче классификации

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

возможно другие значения для специальных функционалов качества

Построение дерева

Стартуя от дерева, состоящего из одной вершины, можно проводить расщепления выбирая признак и порог так, чтобы минимизировать RSS (формулу).

Сейчас уточним – что будем оптимизировать.

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

Как делать расщепления

как выбрать порог для расщепления при построении дерева?

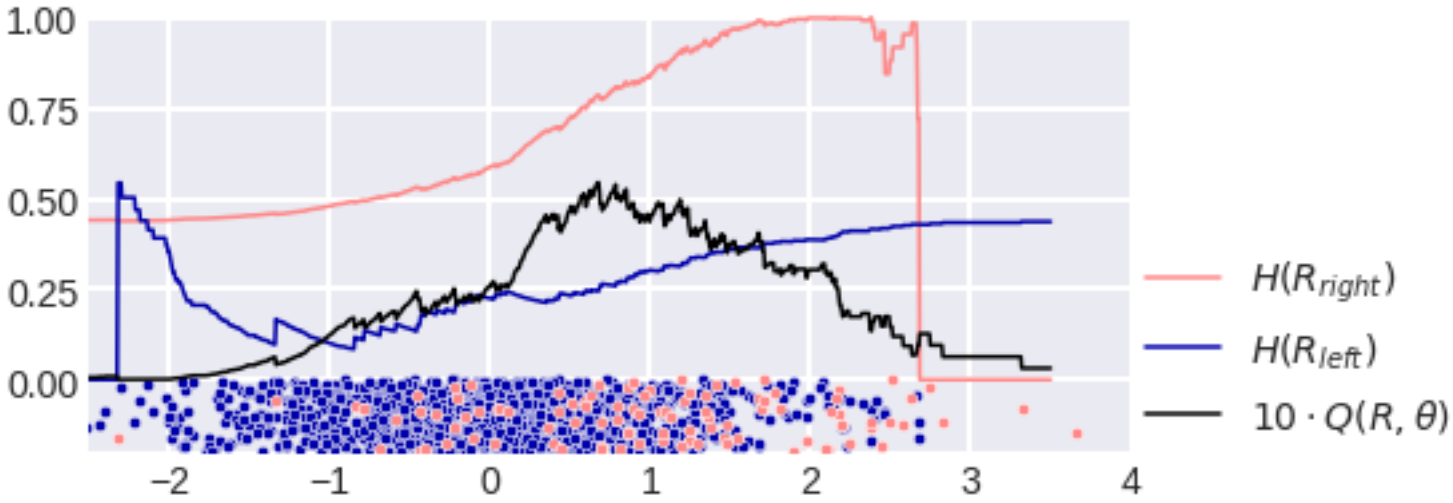
Критерии расщепления в задачах классификации

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

~ насколько в области «почти все объекты одного класса»

тогда при расщеплении области R на две подобласти R_{left} и R_{right} можно оптимизировать весовое усреднение мер неоднородностей

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



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

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

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

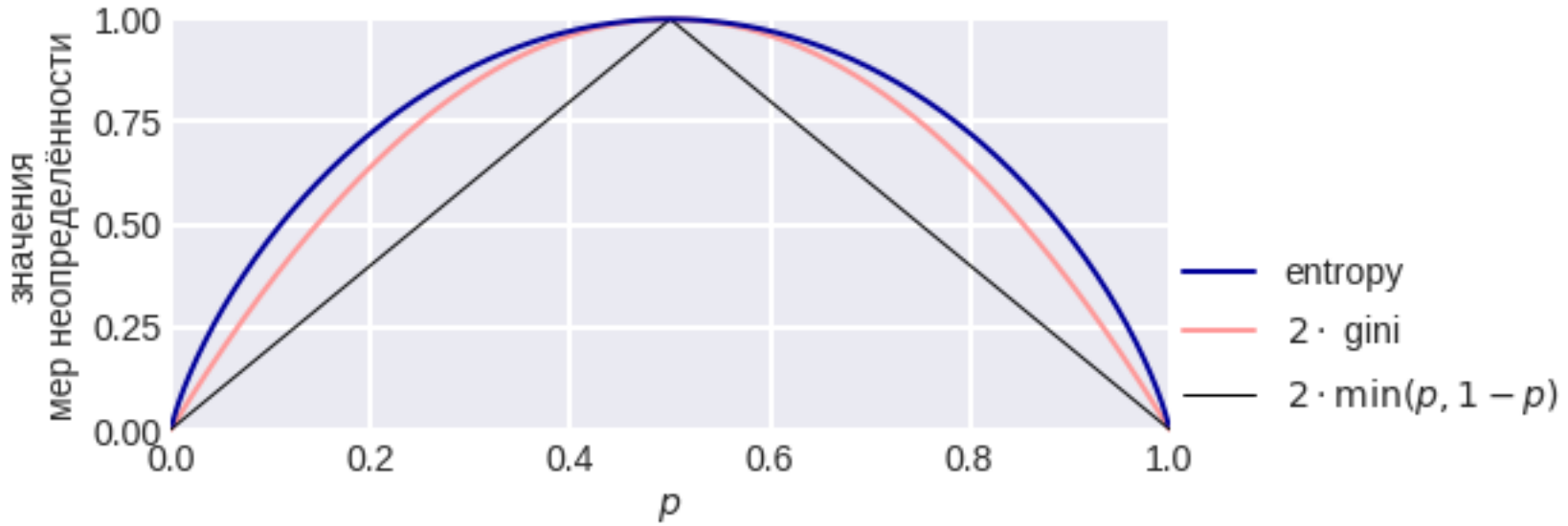
$$H(R) = \min[p, 1-p]$$

энтропийный

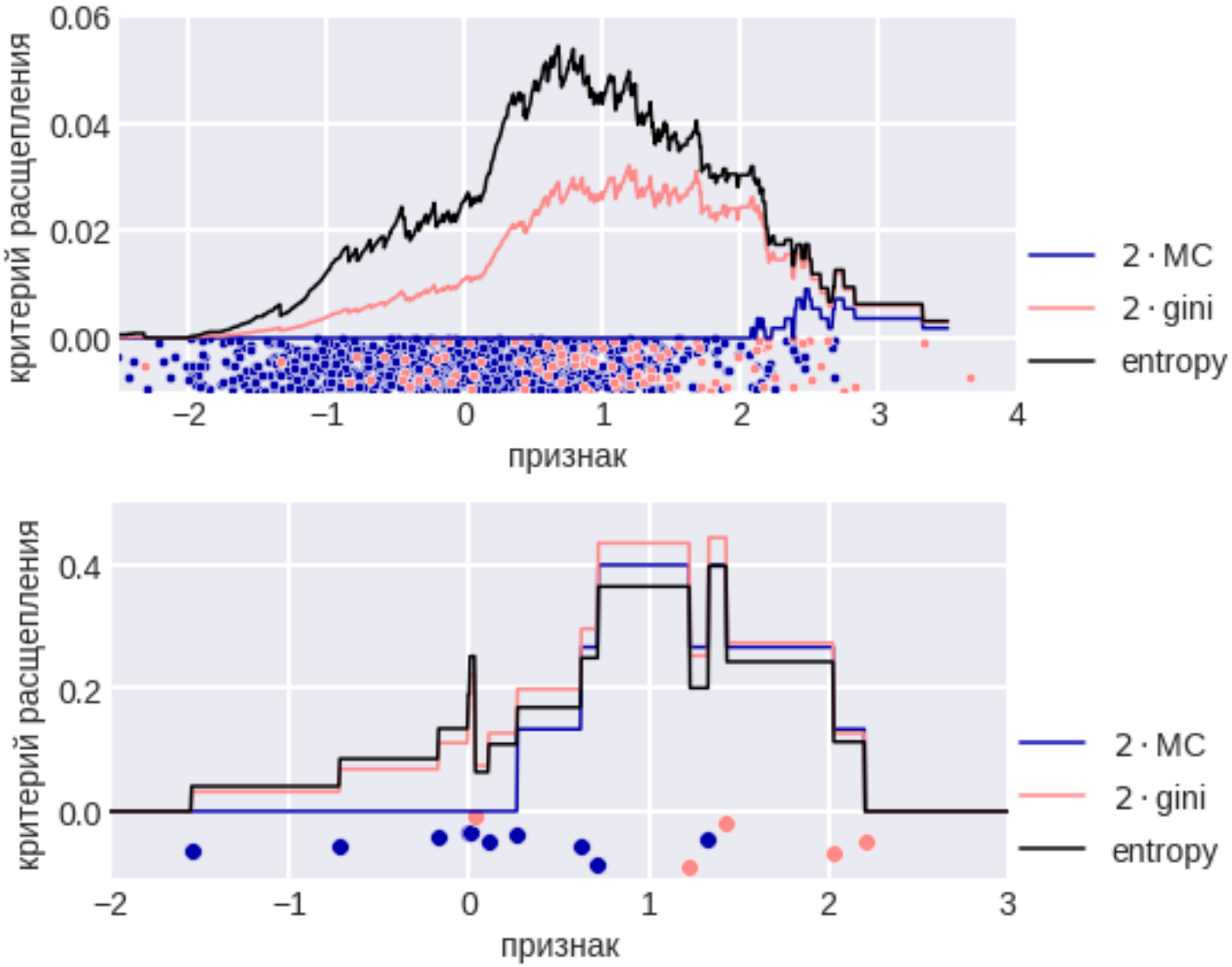
$$H(R) = -p \ln p - (1-p)\ln(1-p)$$

Джини

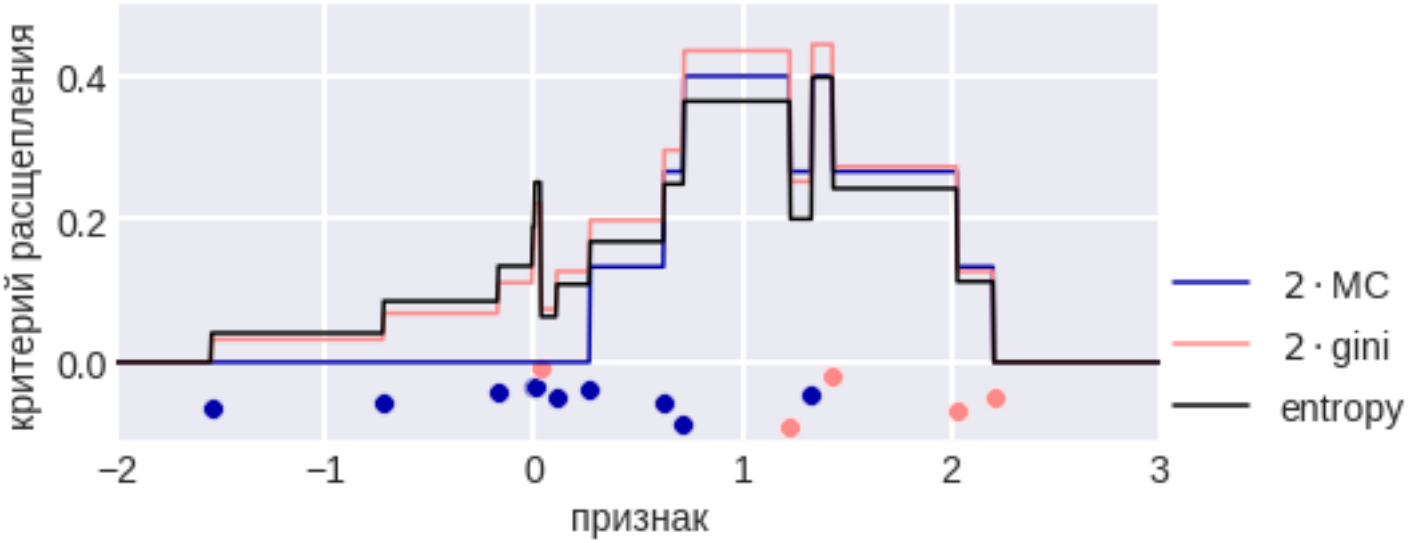
$$H(R) = 2p(1-p) = 1 - p^2 - (1-p)^2$$



Критерии расщепления: частный случай двух классов



Энтропия – мера неопределённости распределения



При пороге $\theta = 1$

$$\frac{|R_{\text{left}}|}{|R|} = \frac{10}{15} = \frac{2}{3}$$

$$\frac{|R_{\text{right}}|}{|R|} = \frac{5}{15} = \frac{1}{3}$$

$$H(R) = -(1/3)\log(1/3) - (2/3)\log(2/3) \approx 0.918$$

$$H(R_{\text{left}}) = -(1/10)\log(1/10) - (9/10)\log(9/10) \approx 0.469$$

$$H(R_{\text{right}}) = -(4/5)\log(4/5) - (1/5)\log(1/5) \approx 0.722$$

$$Q(R,\theta) \approx 0.918 - \frac{2}{3}0.469 - \frac{1}{3}0.722 \approx 0.365$$

Критерии расщепления в задачах регрессии

аналогично… но тут «неоднородность» – дисперсия

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

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

Чем меньше разброс – меньше значение H

Ищем разбиение … которое минимизирует Q

Делаем разбиения

Повторяем процедуру в листьях

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

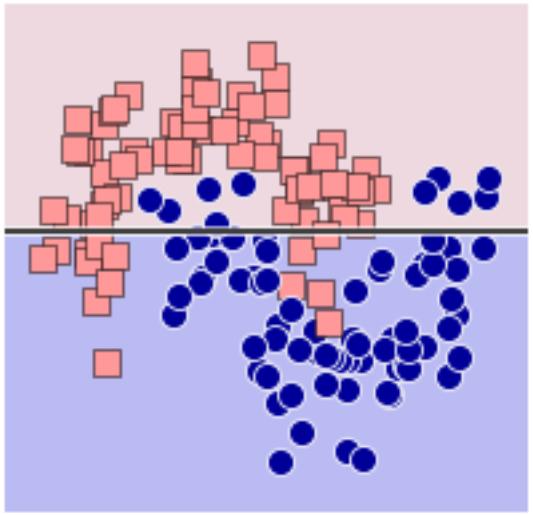
Как долго строить дерево

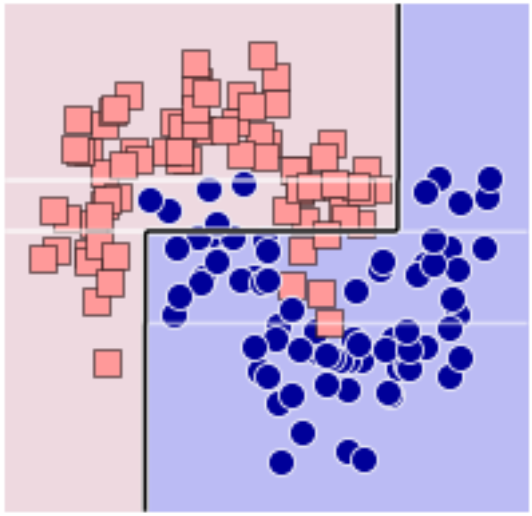
Критерии останова

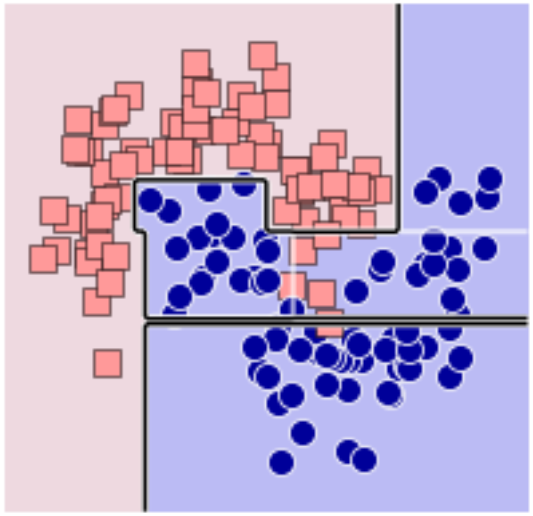
- **ограничение на глубину / на число листьев**
- **ограничение на число объектов в листьях / на число, когда делаем деление**
- **«естественное ограничение» (все объекты одного класса)**

обобщение: почти все объекты одного класса

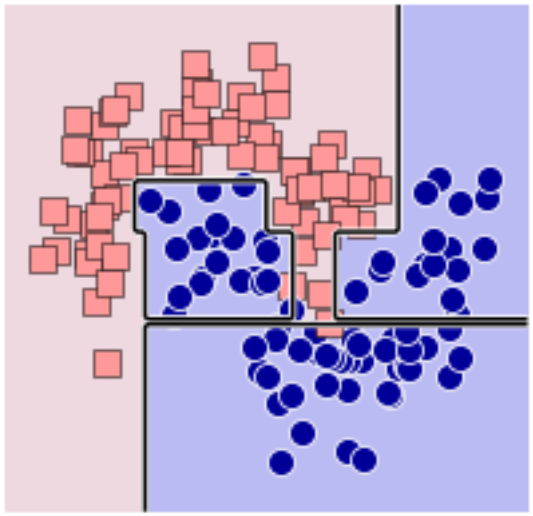
- **изменение impurity**

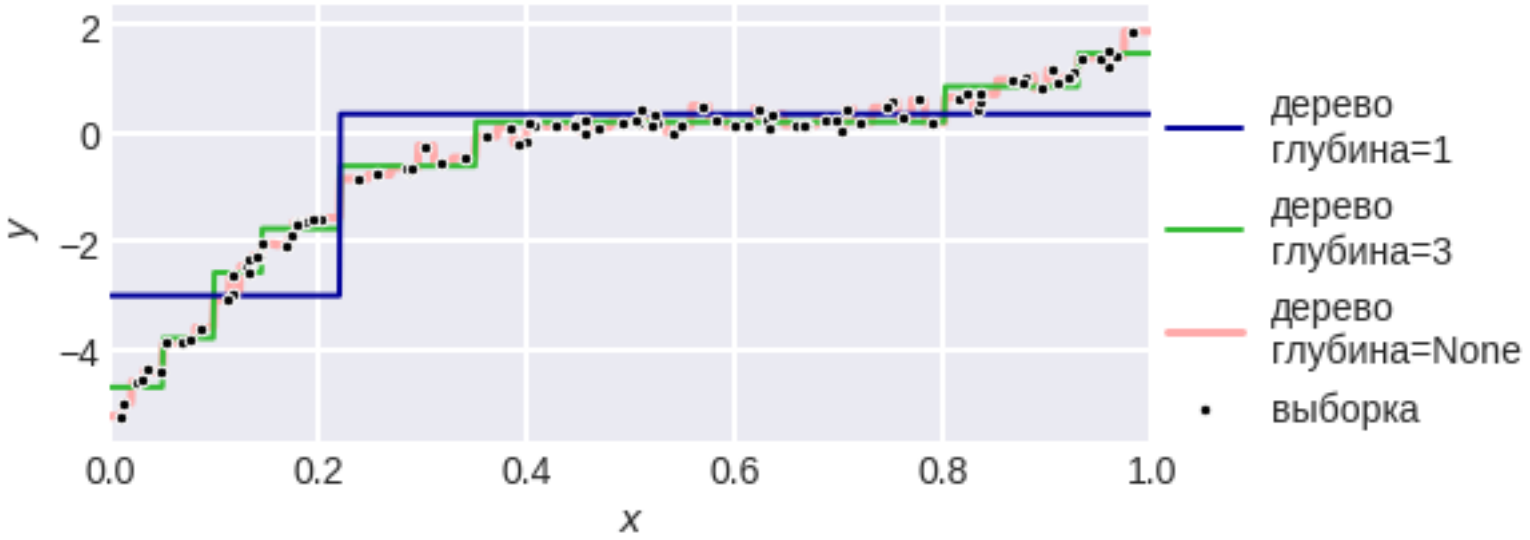

max_depth=1


max_depth=3


max_depth=5

Минутка кода: «Решающее дерево»


max_depth=None



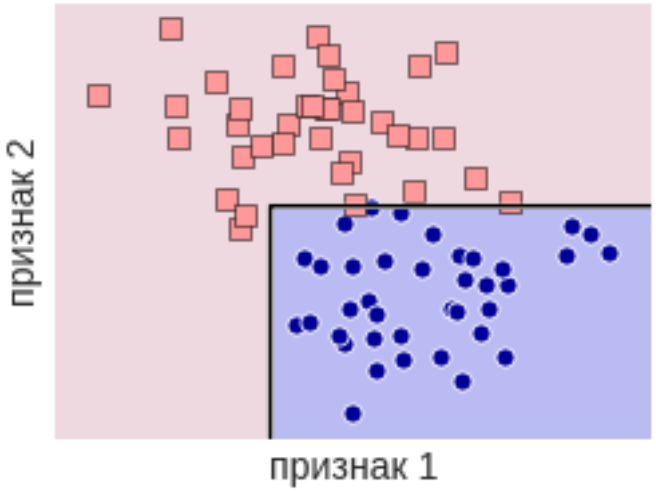

```
from sklearn.tree import DecisionTreeClassifier

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»

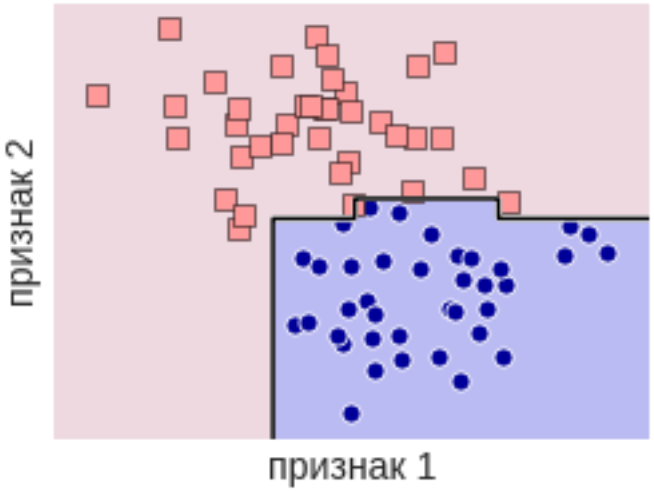
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(max_depth=10)
tree.fit(X_train, y_train)
```

max_depth=10



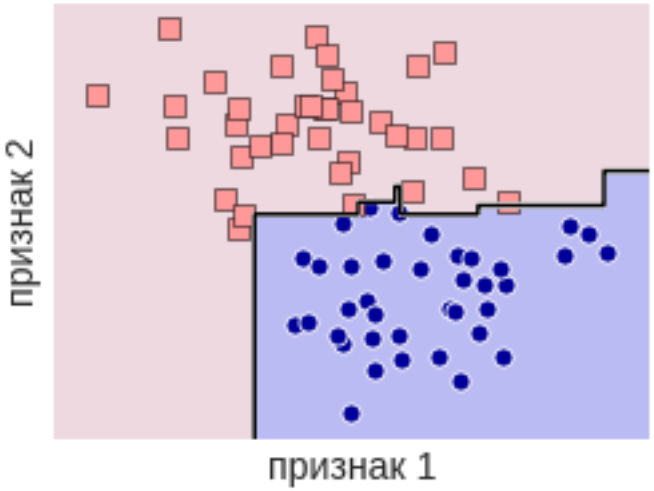
88%

splitter='random'



92%

splitter='random'
max_features=1



92%

Минутка кода: «Решающее дерево»

`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 «зашумлённости» для разбиения**

`min_impurity_split` – **порог «зашумлённости» для останова**

`class_weight` – **веса классов («balanced» или словарь, список словарей)**

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)

Подрезка (post-pruning)

Сейчас подрезка используется крайне редко.
Только в случаях, если задачу действительно пытаются решить одним деревом (или ансамблем из нескольких)

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

ID3

Энтропийный критерий

Остановка, когда все объекты листа одному классу или information gain <= 0

C5.0

Gain ratio

Ограничение на число объектов для расщепления

Подрезка

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 в выбросам
- универсальный метод – для всех типов задач машинного обучения
- встроенные 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» – может существенно измениться при небольшом изменении выборки **картинка 10% от 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**

Важно:

сразу превращаем в эвристику (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)

кратко:

- 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):

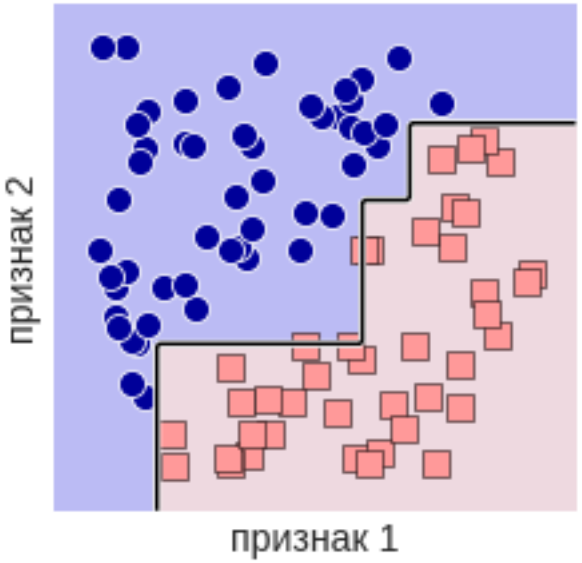
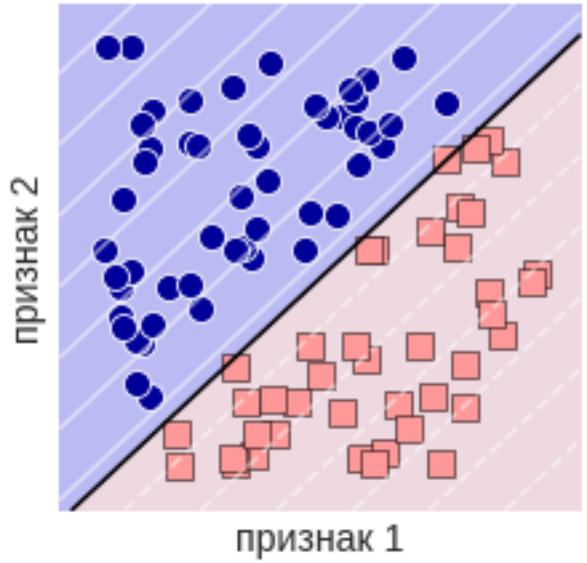
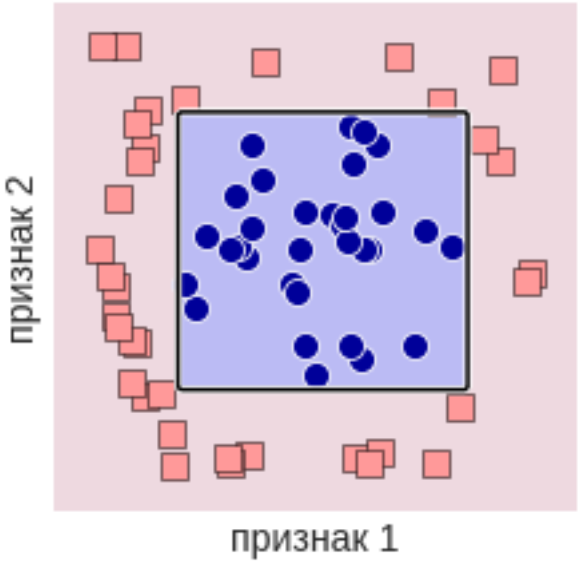
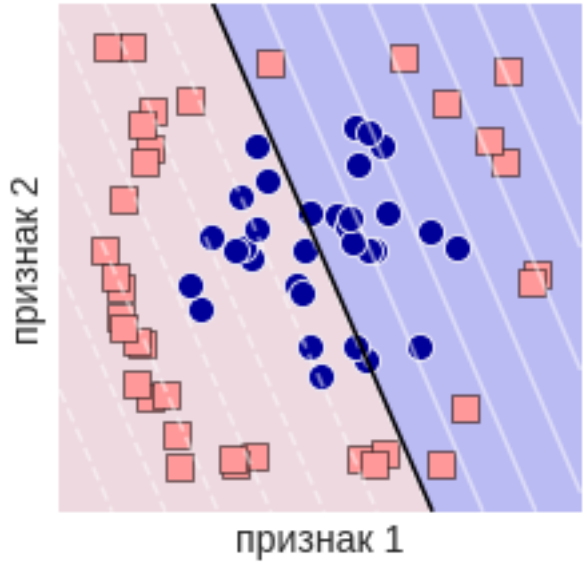
упорядочиваем по вероятности класса 1,

каждая категория $\rightarrow$ 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

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