



# Phystech @ DataScience

Решающие деревья

Случайный лес

# Решающие деревья

## Свойства линейных моделей

- ▶ Легко обучаются с помощью градиентного спуска
- ▶ Востанавливают только простые зависимости  
мало степеней свободы:  
обычно число параметров  $\approx$  количество признаков
- ▶ Не всегда отражают то, как люди принимают решения.  
Иногда не логично расставлять коэффициенты перед признаками.

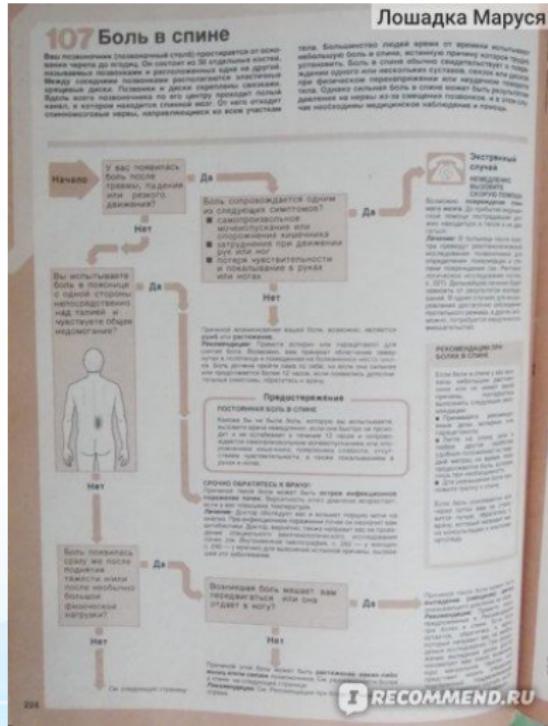
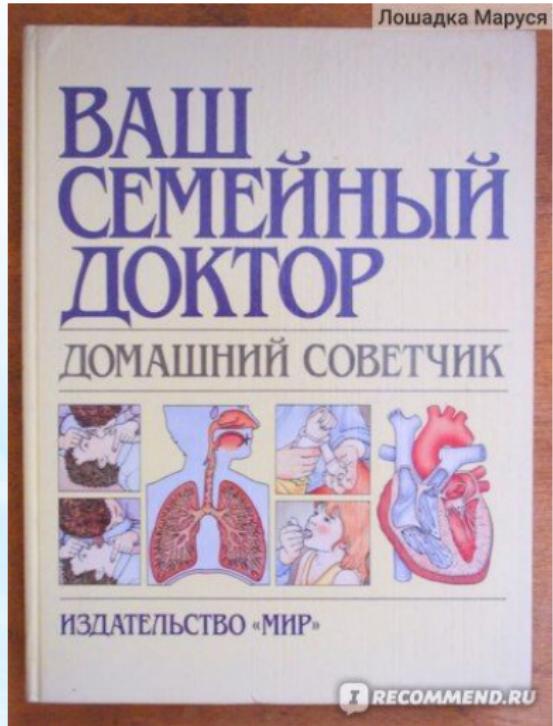


# Как люди принимают решения?





## Пример решающего дерева





# Решающие деревья

Общий случай дерева

Построение дерева

Критерий останова

Ответ в листе

Плюсы и минусы деревьев



# Решающие деревья

Общий случай дерева

Построение дерева

Критерий останова

Ответ в листе

Плюсы и минусы деревьев



## Решающее дерево:

- ▶ Бинарное дерево.
- ▶ В каждой вершине записано некоторое условие.
- ▶ В зависимости от условия идем в правую или левую вершину.
- ▶ В листьях дерева — предсказания.

*Замечание:* Существуют и не бинарные решающие деревья, однако в основном используются именно бинарные

## Пример

### Классификация котиков



котик

?

порода

?

рост

50 см

шерсть

да

## Пример

### Классификация котиков



котик



порода

Саванна

рост

50 см

шерсть

да

## Какие условия в вершинах?

Пусть  $x^{(j)}$  -  $j$ -ый признак объекта.

$t$  - некоторый порог.

Самый популярный подход - делать разбиение в вершине по правилу вида  $I\{x^{(j)} < t\}$ .

Оптимальные значения  $t$  и  $j$  подбираются по некоторому критерию, об этом позже.

# Регрессионное дерево

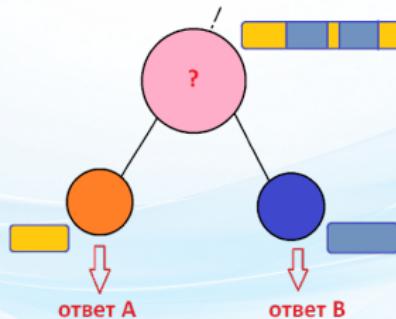
Пусть имеем выборку  $(x_1, Y_1), \dots, (x_n, Y_n)$ .

Строим бинарное дерево по следующему принципу:

**Начало:** один лист с меткой  $\bar{Y}$ , к нему относятся все объекты.

**Деление листа:** Пусть  $X_{leaf} \subset X$  — множество объектов в листе.

Хотим создать **правило**, по которому объекты обучающей выборки в вершине разделяются на две части.



# Регрессионное дерево

Пусть имеем выборку  $(x_1, Y_1), \dots, (x_n, Y_n)$ .

Строим бинарное дерево по следующему принципу:

**Начало:** один лист с меткой  $\bar{Y}$ , к нему относятся все объекты.

**Деление листа:**

Пусть  $X_{leaf} \subset X$  — множество объектов в листе.

Принцип деления: наилучшее приближение двумя константами в дочерних листах:

$$\sum_{x_i \in X_l} (Y_i - y_l)^2 + \sum_{x_i \in X_r} (Y_i - y_r)^2 \longrightarrow \min_{X_l, X_r : X_l \sqcup X_r = X_{leaf}}$$

Какие  $y_l$  и  $y_r$  выбрать для наилучшего приближения по MSE?

$$y_l = \frac{1}{|X_l|} \sum_{x_i \in X_l} y_i, \quad y_r = \frac{1}{|X_r|} \sum_{x_i \in X_r} y_i \quad \text{— метки в новых листах}$$

# Регрессионное дерево

Как разделить выборку  $X_m$  на  $X_l$  и  $X_r$ ?

Пусть  $x^{(j)}$  —  $j$ -ый признак  $x$ .

Обычно правило выглядит так:

- ▶  $x^{(j)} < t \Rightarrow$  объект  $x$  попадает в левое поддерево  $X_l$ .
- ▶  $x^{(j)} \geq t \Rightarrow$  объект  $x$  попадает в правое поддерево  $X_r$ .

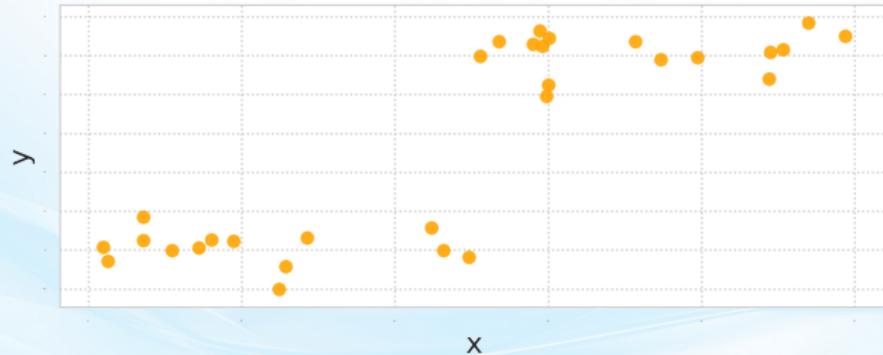
Для нахождения оптимальных  $j$  и  $t$  перебираются все возможные их значения.

Зам Для перебора  $t$  достаточно перебрать все значения признака.

# Регрессионное дерево

Оценка отклика для объекта — метка листа, в который попадет объект.  
Т.е. строится кусочно-постоянная функция.

Пример 1:

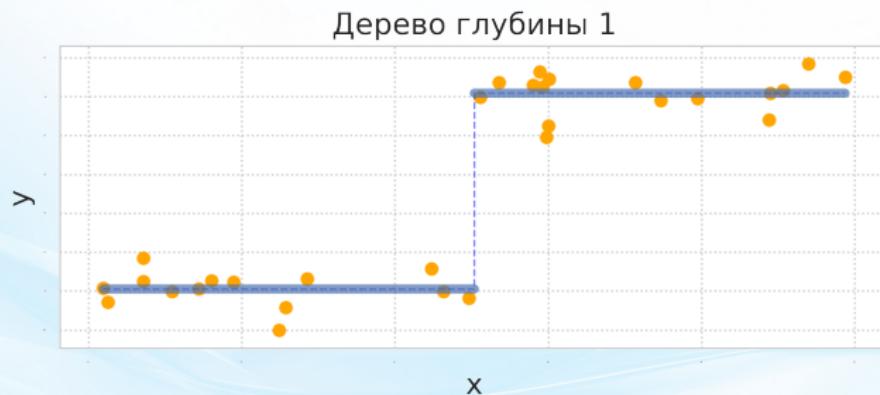


# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 1:

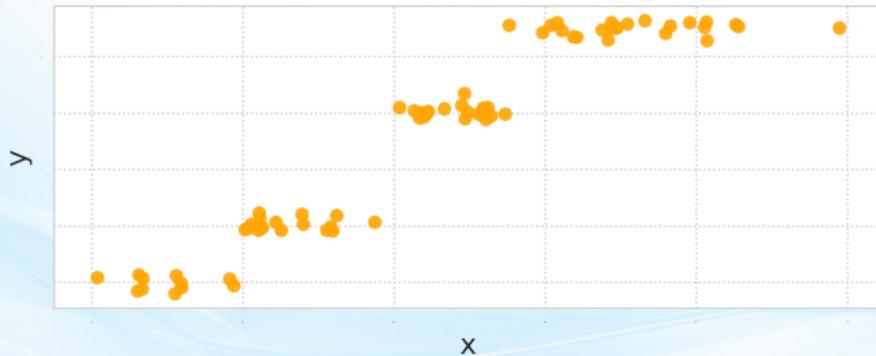


# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 2:

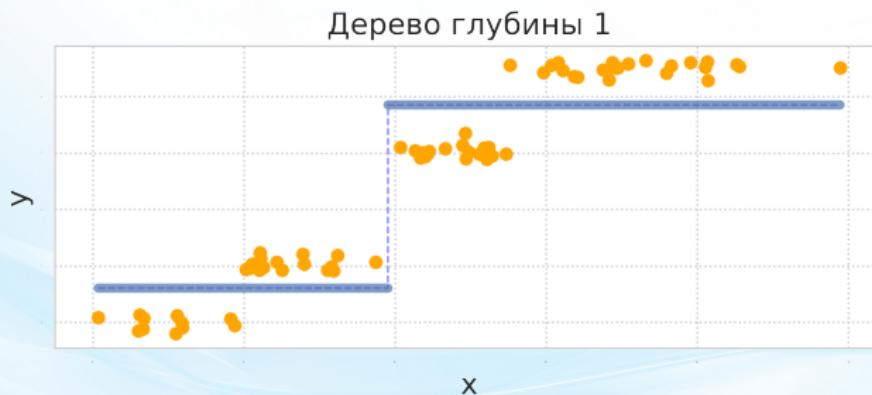


# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 2:



# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 2:

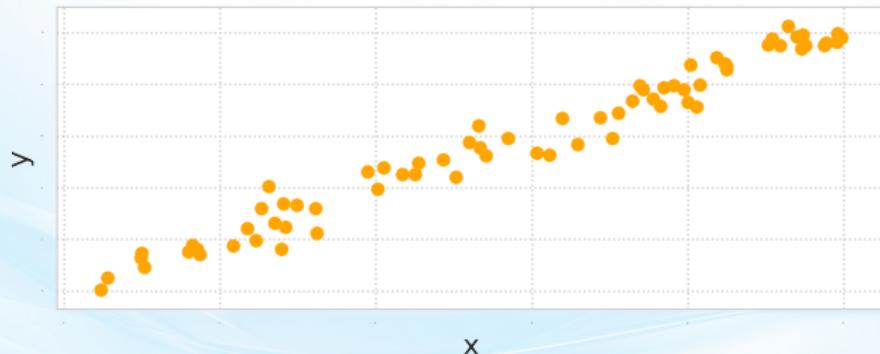


# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 3:

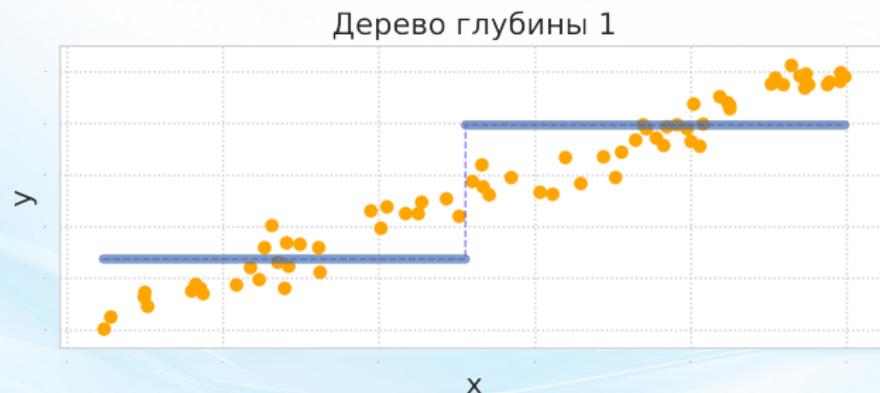


# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 3:



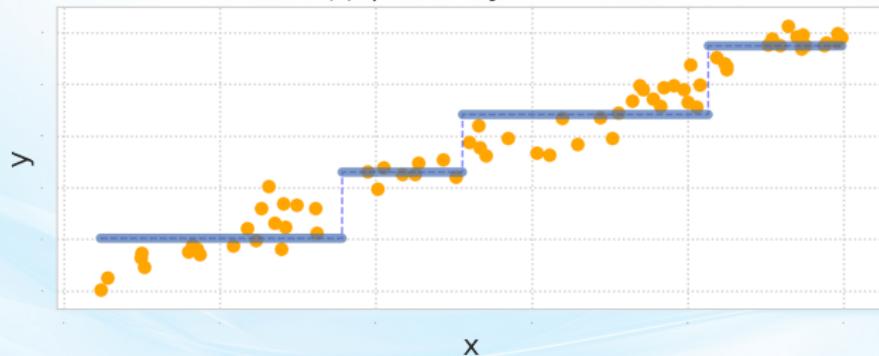
# Регрессионное дерево

Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 3:

Дерево глубины 2

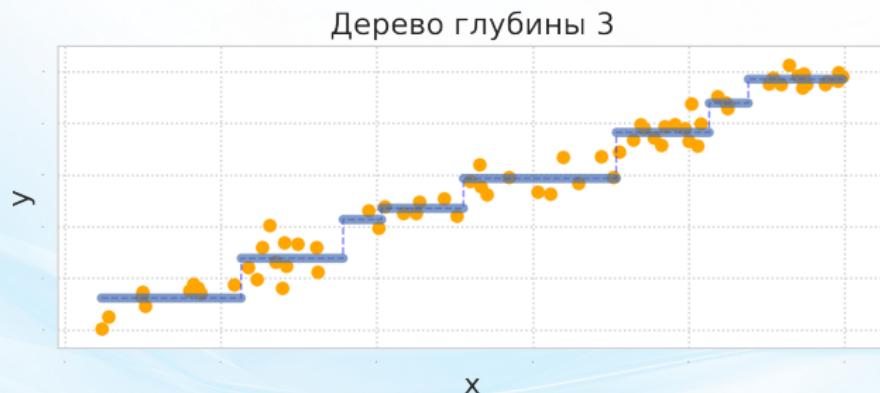


# Регрессионное дерево

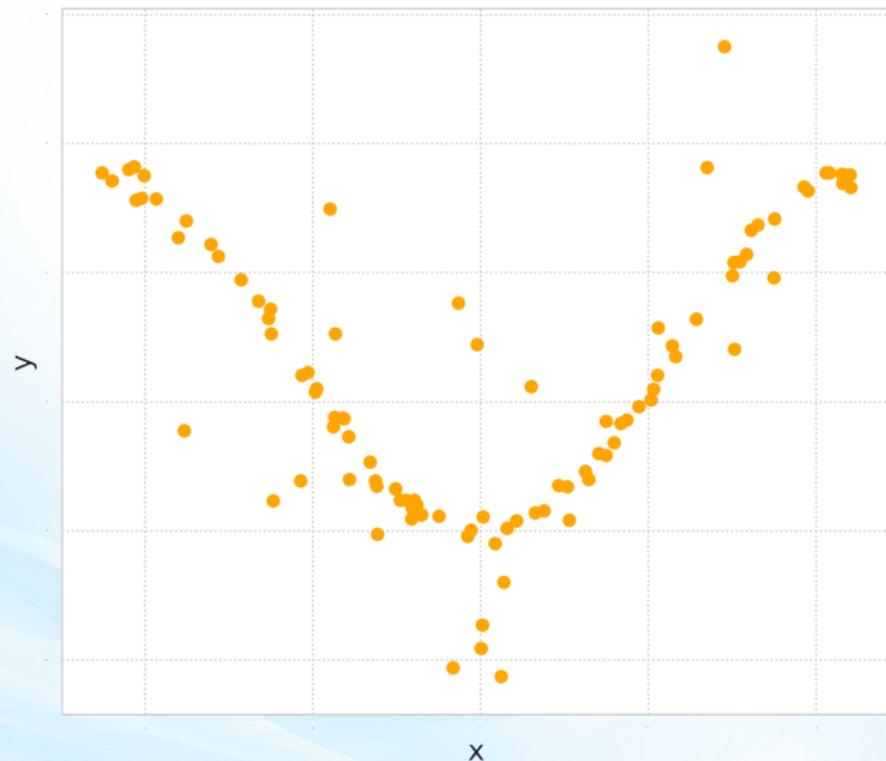
Оценка отклика — метка листа, в который попадет объект.

Т.е. строится кусочно-постоянная функция.

Пример 3:

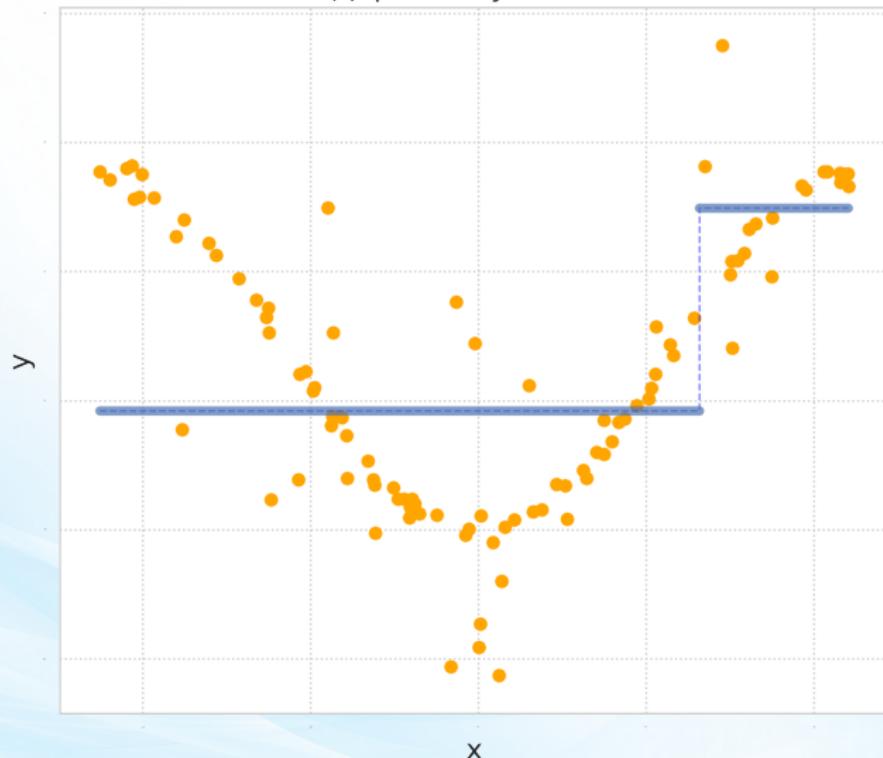


# Регрессионное дерево и выбросы



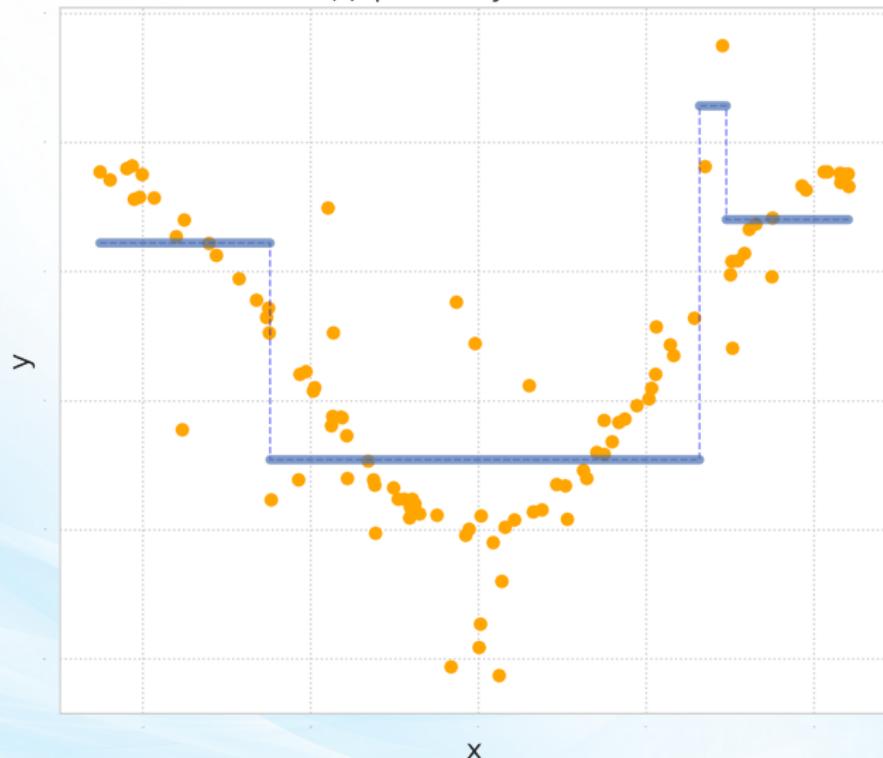
# Регрессионное дерево и выбросы

Дерево глубины 1



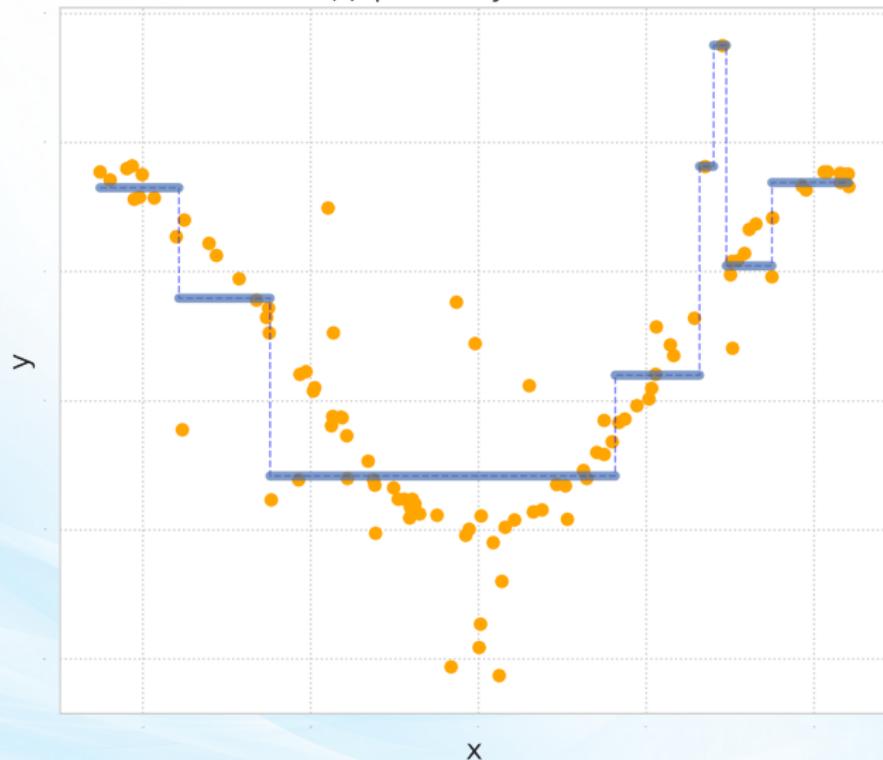
# Регрессионное дерево и выбросы

Дерево глубины 2



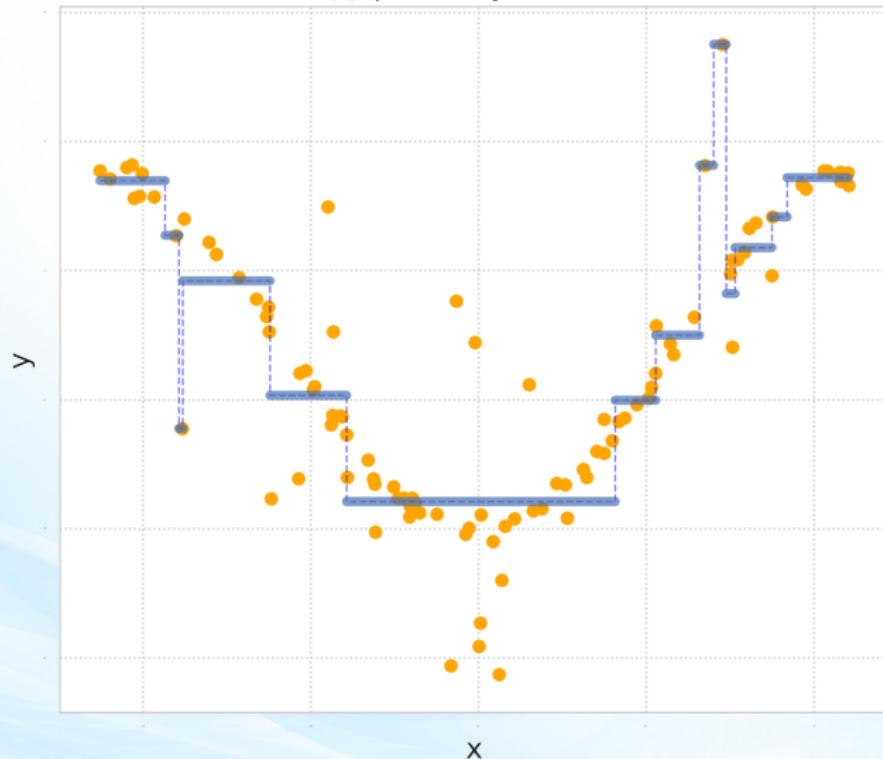
# Регрессионное дерево и выбросы

Дерево глубины 3



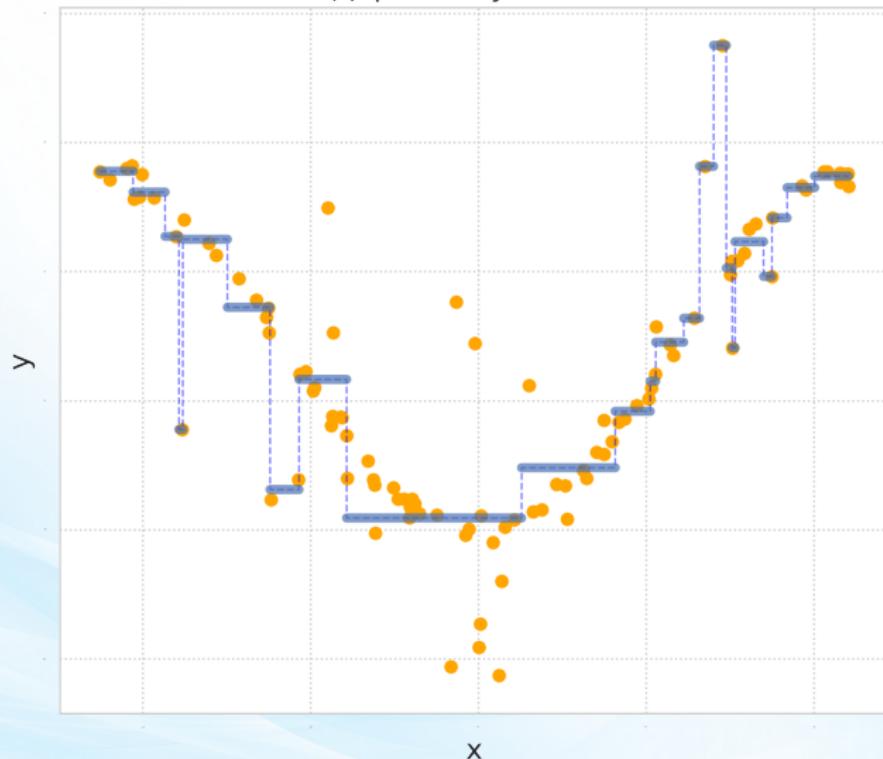
# Регрессионное дерево и выбросы

Дерево глубины 4



# Регрессионное дерево и выбросы

Дерево глубины 5



# Переобучение

## Утверждение

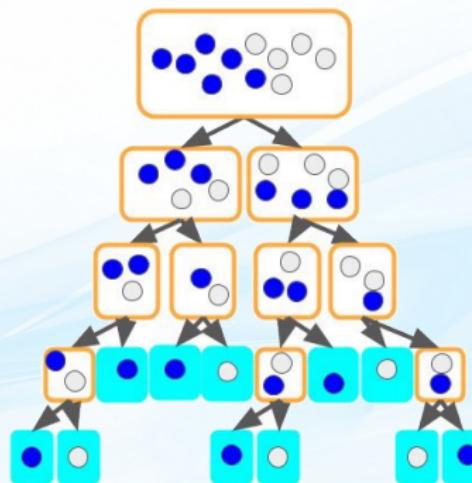
Для любой обучающей выборки можно построить решающее дерево с нулевой ошибкой на обучении.

## Доказательство:

Построим решающее дерево, в котором каждый лист содержит только один объект.

Ответ в листе определяется только этим одним объектом.

⇒ предсказание для обучающей выборки не содержит ошибок.





# Решающие деревья

Регрессионное дерево

Общий случай дерева

Построение дерева

Критерий останова

Ответ в листе

Плюсы и минусы деревьев

# Как строится дерево?

Пусть  $X$  - обучающая выборка.

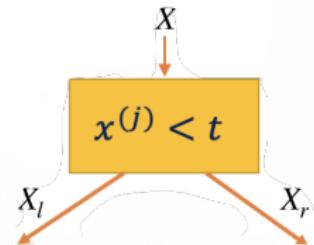
## Деление

Найдем правило  $I\{x^{(j)} < t\}$

оптимальное по некоторому критерию.

Данное правило разобъет  $X$

на 2 части:  $X_l$  и  $X_r$ .



# Как строится дерево?

Пусть  $X$  - обучающая выборка.

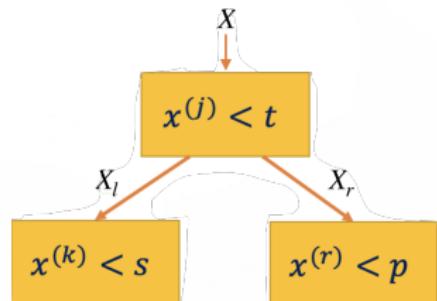
## Деление

Найдем правило  $I\{x^{(j)} < t\}$

оптимальное по некоторому критерию.

Данное правило разбьет  $X$

на 2 части:  $X_l$  и  $X_r$ .



## Итерации

Процедура вызывается рекурсивно от двух дочерних вершин с обучающими выборками  $X_l$  и  $X_r$  соответственно.

Если выполнен некий критерий останова —  
не делить текущую вершину.

# Выбор разбиения

Как выбрать оптимальное разбиение  $I\{x^{(j)} < t\}$ ?

Пусть в вершине  $m$  оказалась выборка  $X_m$ .

Пусть  $Q(X_m, j, t)$  - критерий ошибки условия  $I\{x^{(j)} < t\}$ .

Ищем лучшие параметры (номер признака  $j$  и порог  $t$ ) перебором<sup>(\*)</sup> :

$$Q(X_m, j, t) \rightarrow \min_{j,t}$$

(\*) — Перебираем все возможные признаки, для каждого все возможные его пороги.

# Выбор разбиения

**Вид критерия:**

$$Q(X_m, j, t) = \frac{|X_l|}{|X_m|} \cdot H(X_l) + \frac{|X_r|}{|X_m|} \cdot H(X_r)$$

$H(X)$  - критерий информативности (impurity).

Показывает разброс ответов в вершине, т.е. качество подвыборки  $X$ .

Чем меньше разброс ответов в вершине, тем меньше значение  $H(X)$ .

Хорошее разбиение: после него больше уверены в ответе в вершине.

Т.е. хотим разбить вершину на две так,

чтобы полученные две вершины были более однородны по ответам.

# Выбор разбиения

$$Q(X_m, j, t) = \frac{|X_l|}{|X_m|} \cdot H(X_l) + \frac{|X_r|}{|X_m|} \cdot H(X_r)$$

$H(X_l)$  и  $H(X_r)$  нормируются на доли объектов, которые идут вправо и влево.

Зачем это нужно?

Пусть  $|X_m| = 1000$ ,  $|X_l| = 990$ ,  $|X_r| = 10$

В  $X_l$  все объекты имеют один класс  $\Rightarrow H(X_l)$  маленький.

$X_r$  содержит объекты всех возможных классов  $\Rightarrow H(X_r)$  большой.

Не так страшно, что  $X_r$  получилось плохим, при том, что 990 попали в правильную вершину.

# Критерий информативности

## Неформальное определение:

- ▶  $H(X)$  зависит от меток в выборке  $X$
- ▶ Показывает разброс ответов в  $X$
- ▶ Чем меньше разброс ответов в  $X$ , тем меньше  $H(X)$

## Задача регрессии:

Возьмем выборочную дисперсию ответов в качестве  $H(X)$ :

$$H(X) = \frac{1}{|X|} \sum_{x_i \in X} (y_i - \bar{y})^2$$

# Критерий информативности: задача классификации

Пусть решаем задачу классификации на  $K$  классов.

$p_1, \dots, p_k$  - доли объектов классов  $1, \dots, K$  в  $X$ :

$$p_j = \frac{\sum_{x_i \in X} I\{y_i = j\}}{|X|}$$

## Энтропийный критерий

$$H(X) = - \sum_{k=1}^K p_k \ln p_k$$

Считаем, что  $0 \ln 0 = 0$

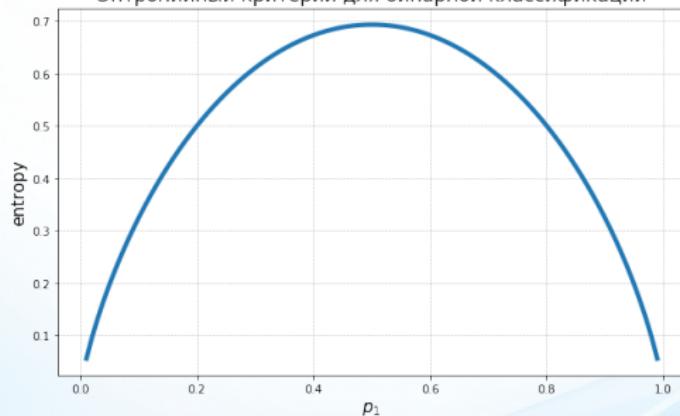
- ▶  $H(X) \geq 0$
- ▶ При  $p_1 = 1, p_2 = 0, \dots, p_K = 0 : H(X) = 0$

# Критерий информативности: задача классификации

## Энтропийный критерий

$$H(X) = - \sum_{k=1}^K p_k \ln p_k$$

Энтропийный критерий для бинарной классификации



Интерпретация:

Мера отличия распределения классов от вырожденного.

Вырожденное — всегда знаем, что получим,  
равномерное — непредсказуемо

# Критерий информативности: задача классификации

Пусть решаем задачу классификации на  $K$  классов.

$p_1, \dots, p_k$  - доли объектов классов  $1, \dots, K$  в  $X$ :

$$p_j = \frac{\sum_{x_i \in X} I\{y_i = j\}}{|X|}$$

## Критерий Джини

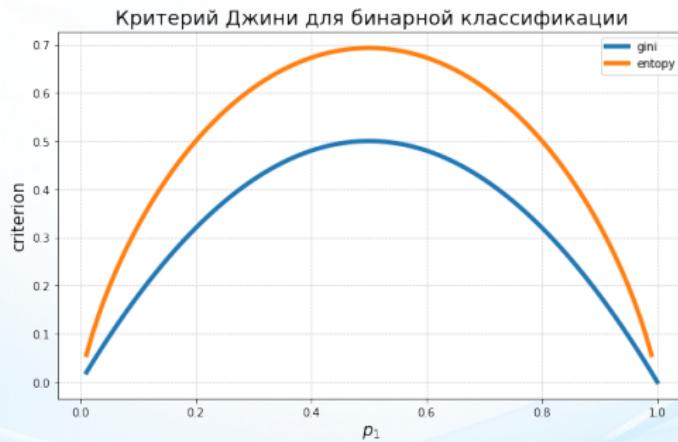
$$H(X) = \sum_{k=1}^K p_k(1 - p_k)$$

- ▶  $H(X) \geq 0$
- ▶ При  $p_1 = 1, p_2 = 0, \dots, p_K = 0 : H(X) = 0$

# Критерий информативности: задача классификации

## Критерий Джини

$$H(X) = \sum_{k=1}^K p_k(1 - p_k)$$



Интерпретация:

Вероятность ошибки случайного классификатора, который выдает ответы пропорционально  $p_k$ .



# Решающие деревья

Регрессионное дерево

Общий случай дерева

Построение дерева

Критерий останова

Ответ в листе

Плюсы и минусы деревьев

# Критерий останова

Как понять, разбивать ли вершину далее или сделать ее листовой?

- ▶ **Все объекты в вершине одного класса**

Простой критерий, хорош для простых выборок.

Если выборка сложная, то сработает только когда в вершине осталось 1-2 объекта.

- ▶ **В вершину попало  $\leq k$  объектов.**

$k$  - гиперпараметр. Нужно выбирать таким, что по  $k$  объектам в листе можно построить надежный прогноз.

- ▶ **Глубина дерева превысила порог.**

Грубый критерий, не зависит ни от распределения классов, ни от числа объектов.

Хорошо работает в композициях  
(много моделей объединяются в одну сложную модель).

# Критерий останова

Как понять, разбивать ли вершину далее или сделать ее листовой?

- ▶ Число листьев в дереве превысило порог.
- ▶ Функционал ошибки при делении не уменьшился.

Если лучшее из разбиений приводит к росту функционала ошибки, не разбиваем эту вершину.

- ▶ Функционал ошибки при делении уменьшился на  $< s\%$ .

Если лучшее из разбиений не уменьшает функционал на  $s\%$ , не разбиваем эту вершину.



# Решающие деревья

Регрессионное дерево

Общий случай дерева

Построение дерева

Критерий останова

**Ответ в листе**

Плюсы и минусы деревьев

## Ответ в листе

Дерево построено.

Какое предсказание выдавать для объекта, попавшего в лист?

### ► Классификация:

1. Самый популярный класс в листе:

$$\widehat{y}_m = \arg \max_{y \in \mathbb{Y}} \sum_{i \in X_m} I\{Y_i = y\}$$

2. Оценки вероятности классов  $\widehat{p}_m = [\widehat{p}_m^1, \dots, \widehat{p}_m^K]$  , где

$$\widehat{p}_m^k = \frac{1}{|X_m|} \sum_{i \in X_m} I\{Y_i = k\}$$

### ► Регрессия с MSE:

$$\widehat{y}_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i$$



# Решающие деревья

Регрессионное дерево

Общий случай дерева

Построение дерева

Критерий останова

Ответ в листе

Плюсы и минусы деревьев

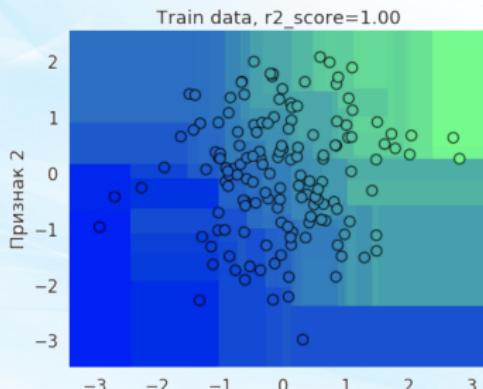
# Основные свойства решающих деревьев

## Плюсы

- ▶ Восстанавливают сложные закономерности
- ▶ Интерпретируемая структура
- ▶ Не требуют нормализации и масштабирования

## Минусы

- ▶ Очень легко переобучаются.  
Неустойчивы к малейшим изменениям в данных.
- ▶ Восстанавливаемая зависимость довольно ужасна.



⇒ Сами деревья не очень хороши.

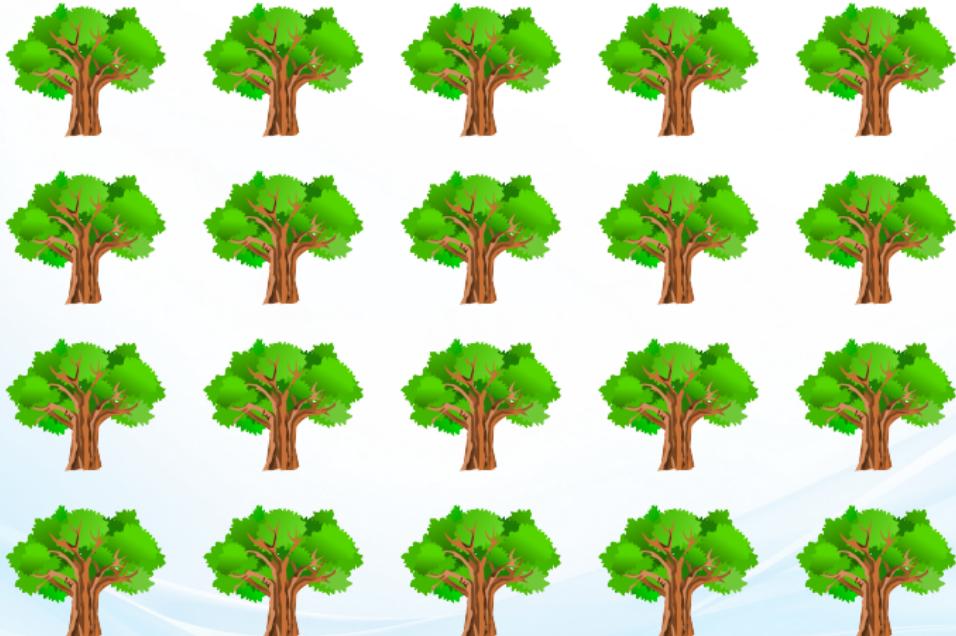
# Случайный лес

# Идея



Один в поле не воин...

# Идея



Лес - много деревьев

# Идея

А есть ли смысл брать деревья одинаковыми?

Нужны разные деревья



"Танцующий лес", нац. парк Куршская коса, Калининградская обл.

# Идея

Возьмем композицию вида:

$$f = \frac{1}{T} \sum_{t=1}^T b_t,$$

где  $b_t$  — решающее дерево.

Чтобы сделать деревья  $b_t$  разными:

- ▶  $b_t$  обучаем на **случайной подвыборке**.

*Случайная подвыборка*: упорядоченный выбор с возвращением

- ▶  $b_t$  обучаем на **случайном подпространстве признаков**.

Зам Деревья строятся **сильно переобученные** — так они меньше похожи друг на друга.

## Случайный лес

Пусть  $d$  — количество признаков,  $d_0$  — количество случайно выбираемых признаков при разбиении.

*Рекомендации:*

- ▶ В задаче классификации

Взять  $d_0 = \left\lfloor \sqrt{d} \right\rfloor$ .

Строить каждое дерево до тех пор,

пока в каждом листе не окажется по 1 объекту.

- ▶ В задаче регрессии

Взять  $d_0 = \left\lfloor d/3 \right\rfloor$ .

Строить каждое дерево до тех пор,

пока в каждом листе не окажется по 5 объектов.

Theta



BCE!