



Machine Learning

my TRACKER

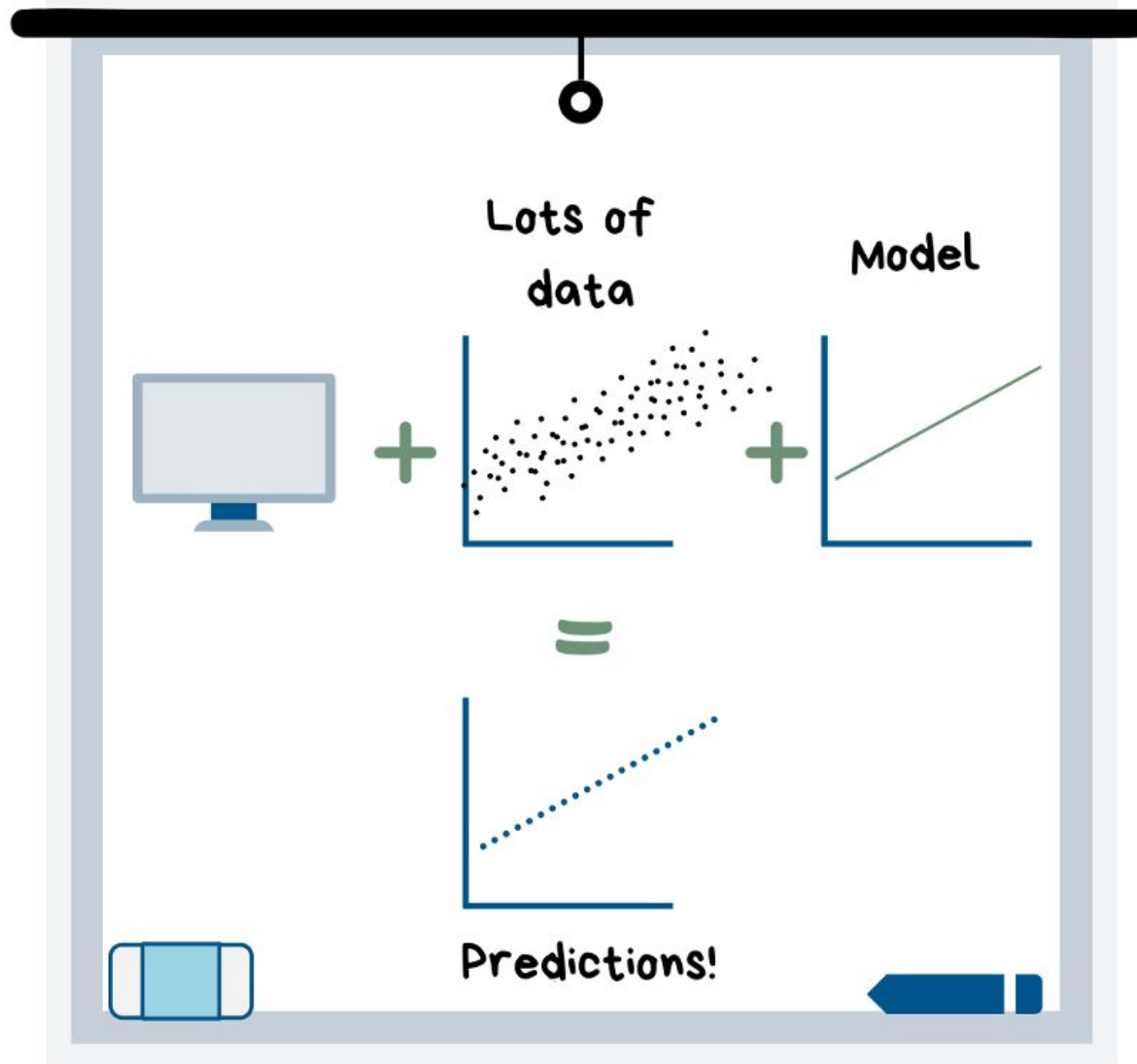


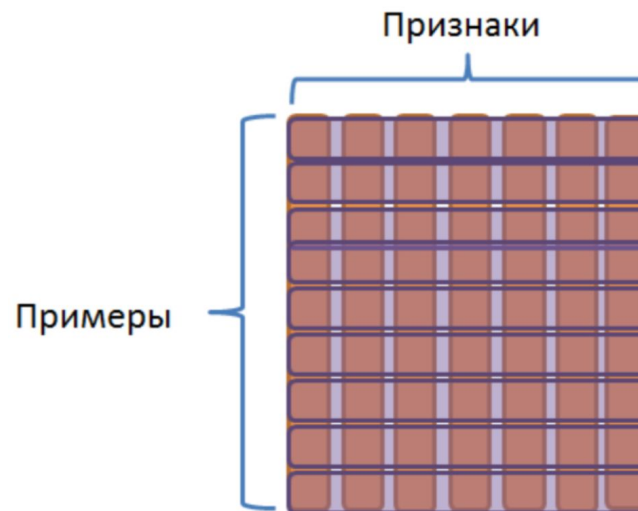
@ mail.ru
group

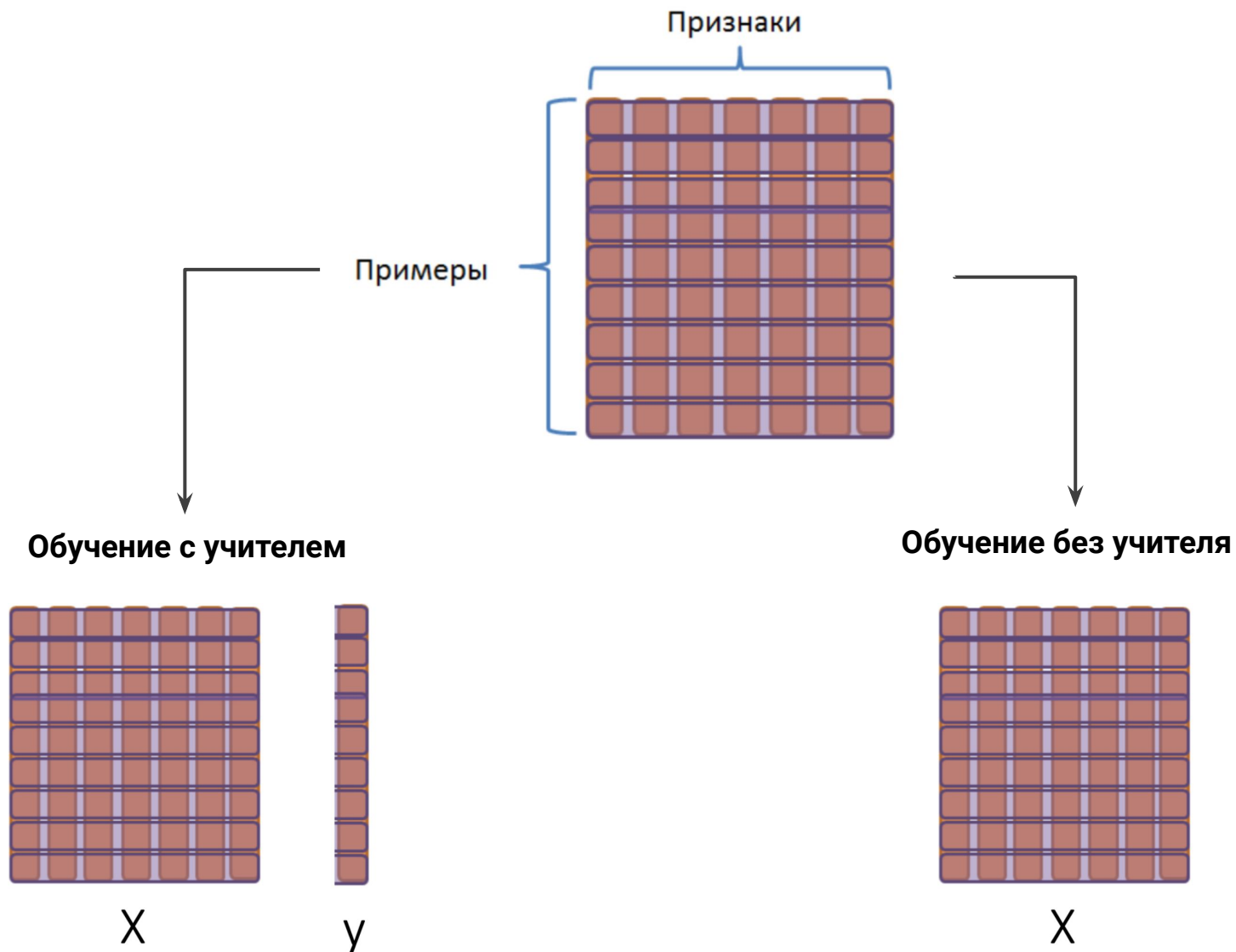


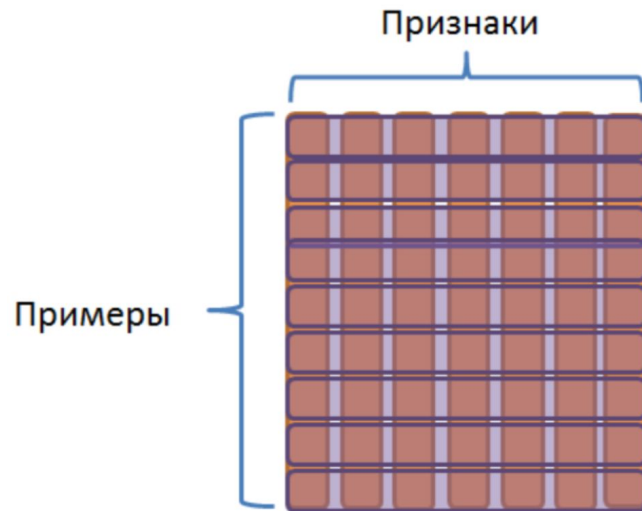
Деревья решений







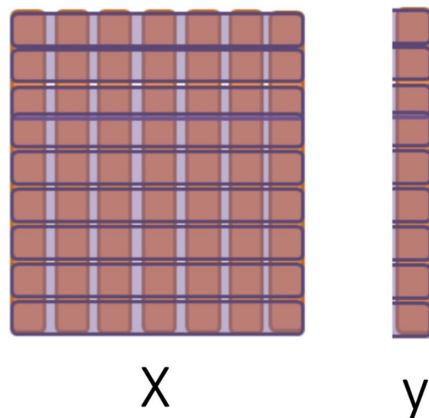




Классификация

Кошка	1	0.13
Собака	0	0.19
Собака	0	0.98
Собака	0	0.14
Рыбка	2	0.57
Собака	0	0.42
Слон	3	0.39
Слон	3	0.02

Регрессия



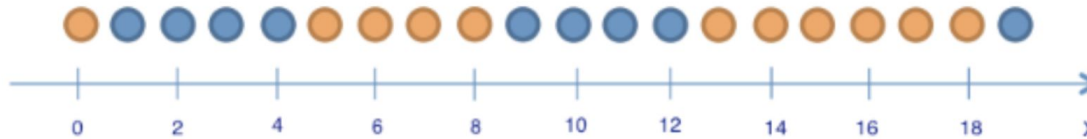
30000	- 0.13	7
45000	0.19	57
45000	0.98	19
37500	0.14	24
18000	- 0.57	21
30000	0.42	35
180000	0.39	39
95000	0.02	27

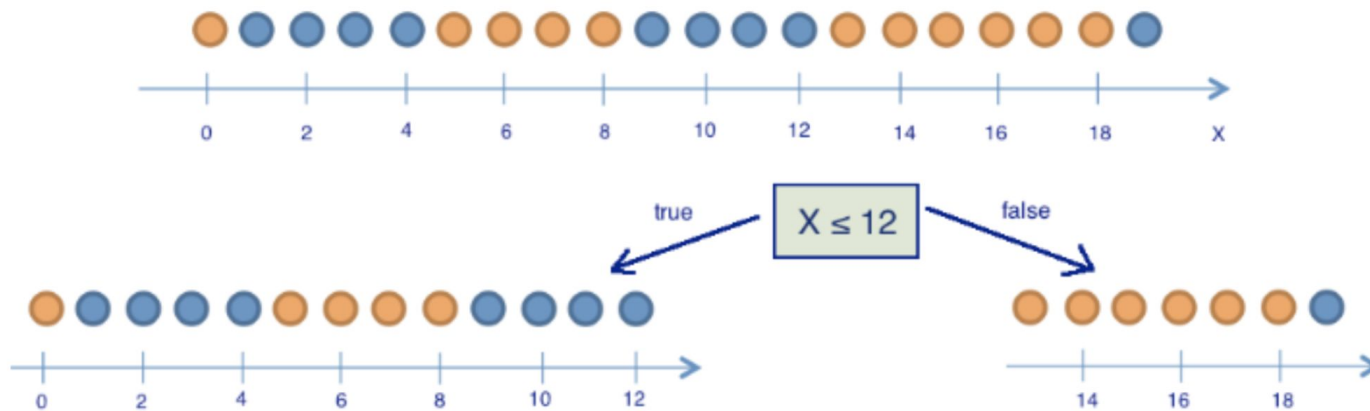
Пусть дан набор объектов $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i \in 1, \dots, N$, полученный из неизвестной закономерности $y = f(\mathbf{x})$. Необходимо построить такую $h(\mathbf{x})$, которая наиболее точно аппроксимирует $f(\mathbf{x})$.

Будем искать неизвестную

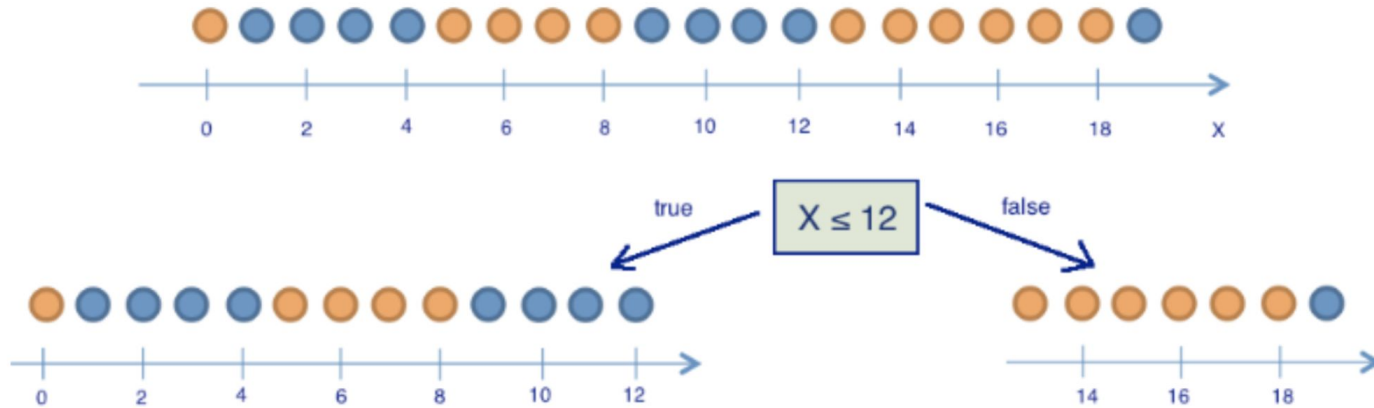
$$h(\mathbf{x}) = h(a_1, \dots, a_T)$$



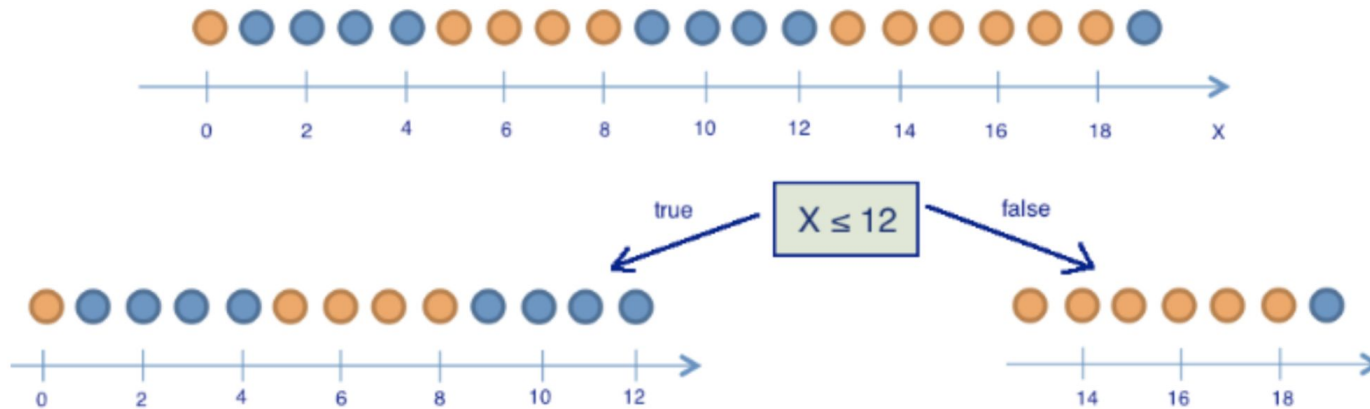




Оранжевые: $p_1 = \frac{9}{20}$ Синие: $p_2 = \frac{11}{20}$



Оранжевые: $p_1 = \frac{9}{20}$ Синие: $p_2 = \frac{11}{20}$



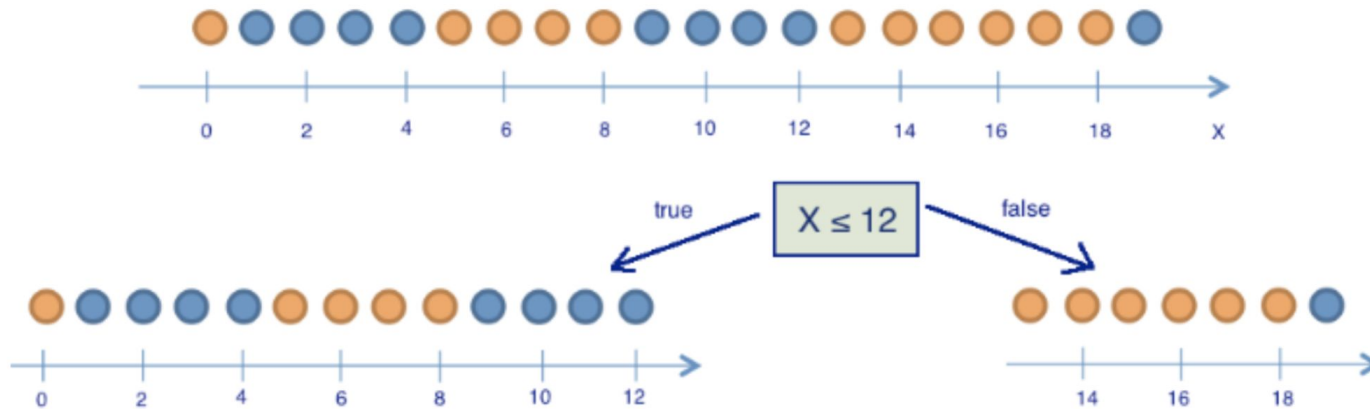
Энтропия Шеннона

$$-\sum_{i=1}^N p_i \log_2(p_i)$$

Индекс Джини

$$1 - \sum_{i=1}^N p_i^2$$

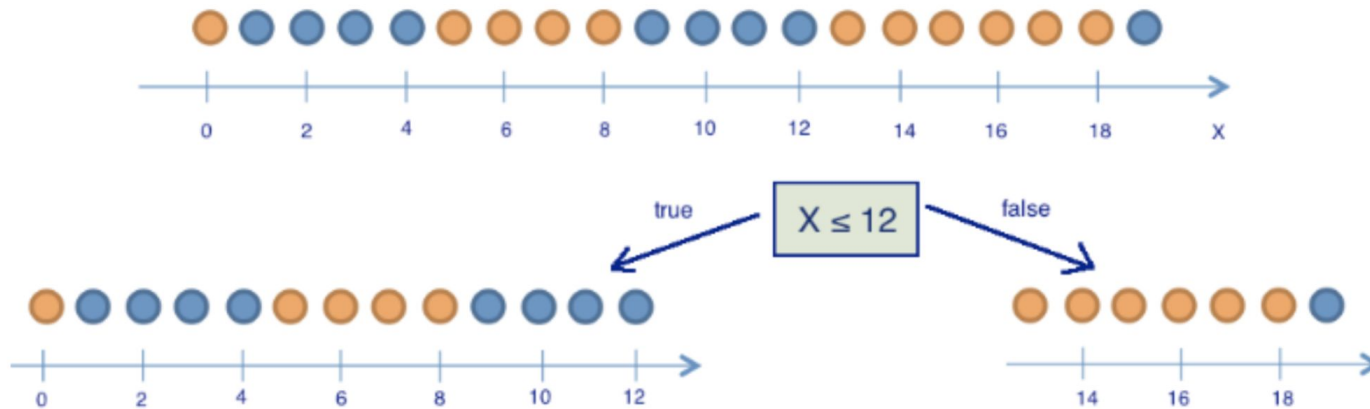
Оранжевые: $p_1 = \frac{9}{20}$ Синие: $p_2 = \frac{11}{20}$ $S_0 = -\frac{9}{20} \log_2\left(\frac{9}{20}\right) - \frac{11}{20} \log_2\left(\frac{11}{20}\right) = 0.993$



$$S_1 = -\frac{5}{13} \log_2\left(\frac{5}{13}\right) - \frac{8}{13} \log_2\left(\frac{8}{13}\right) = 0.96$$

$$S_2 = -\frac{1}{7} \log_2\left(\frac{1}{7}\right) - \frac{6}{7} \log_2\left(\frac{6}{7}\right) = 0.59$$

Оранжевые: $p_1 = \frac{9}{20}$ Синие: $p_2 = \frac{11}{20}$ $S_0 = -\frac{9}{20} \log_2\left(\frac{9}{20}\right) - \frac{11}{20} \log_2\left(\frac{11}{20}\right) = 0.993$



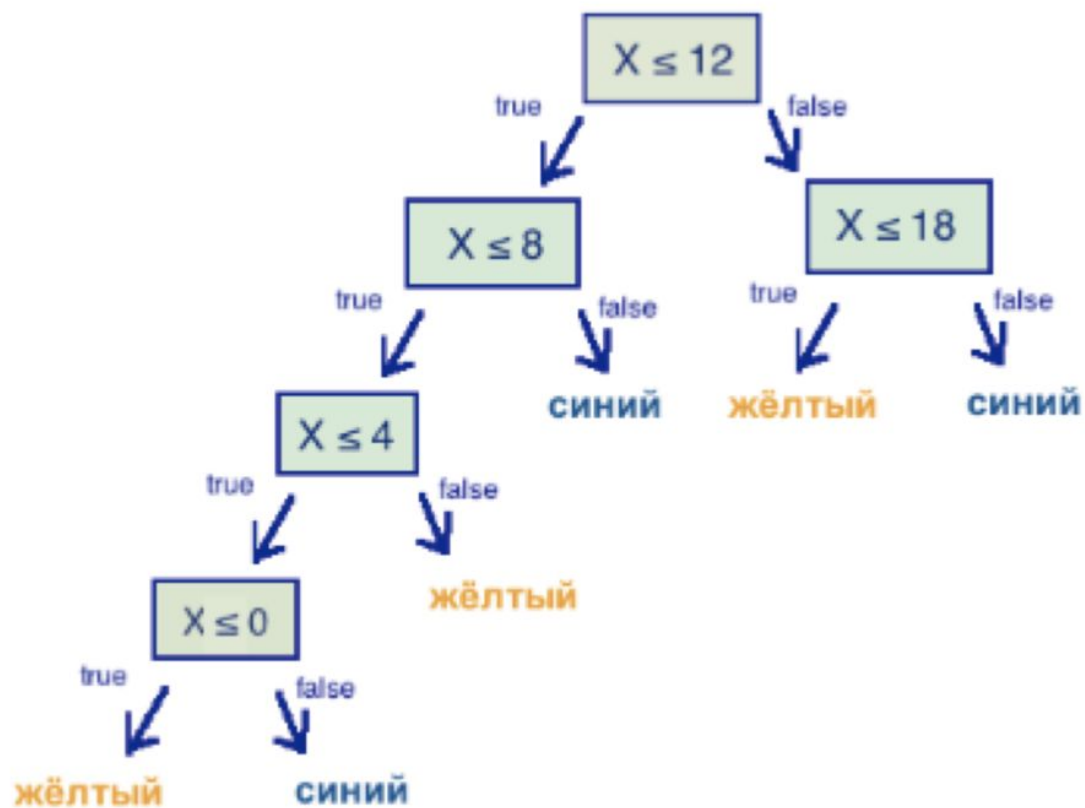
$$S_1 = -\frac{5}{13} \log_2\left(\frac{5}{13}\right) - \frac{8}{13} \log_2\left(\frac{8}{13}\right) = 0.96$$

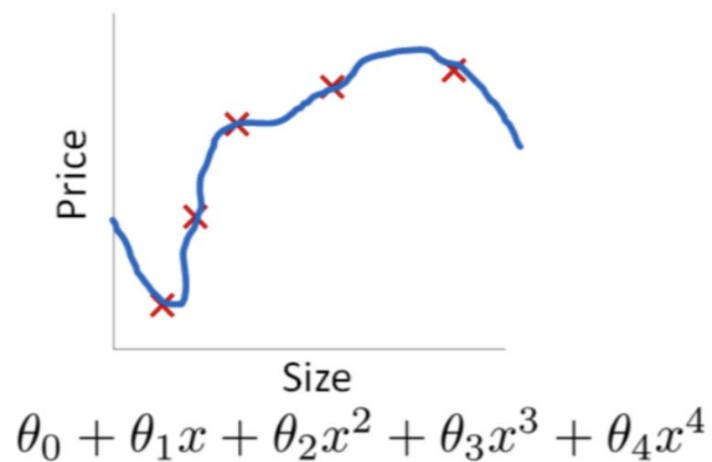
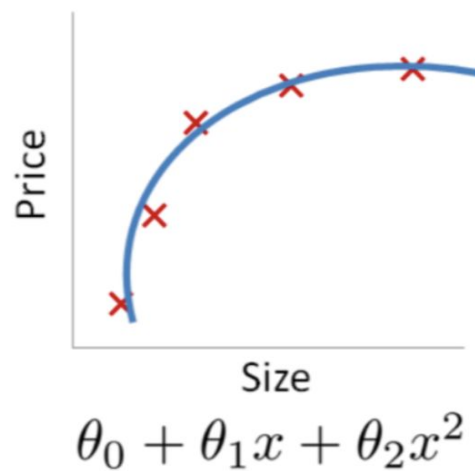
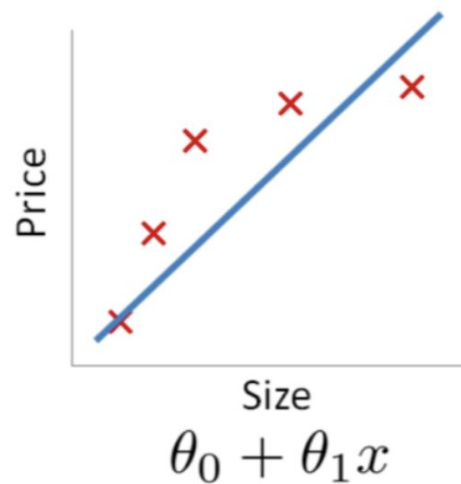
$$S_2 = -\frac{1}{7} \log_2\left(\frac{1}{7}\right) - \frac{6}{7} \log_2\left(\frac{6}{7}\right) = 0.59$$

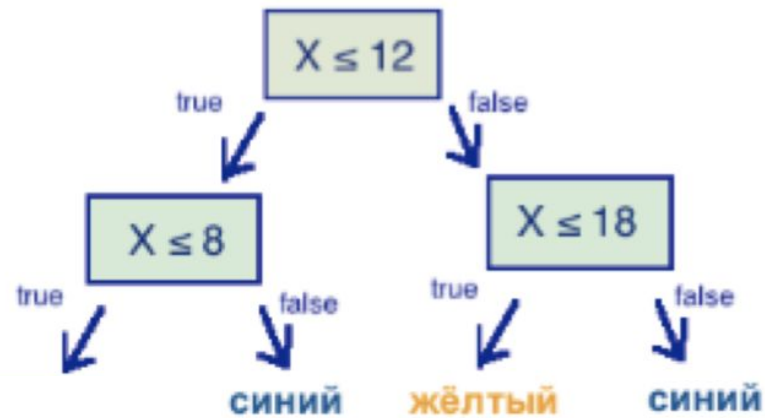
$$IG("X \leq 12") = S_0 - \frac{13}{20} S_1 - \frac{7}{20} S_2 = 0.163$$

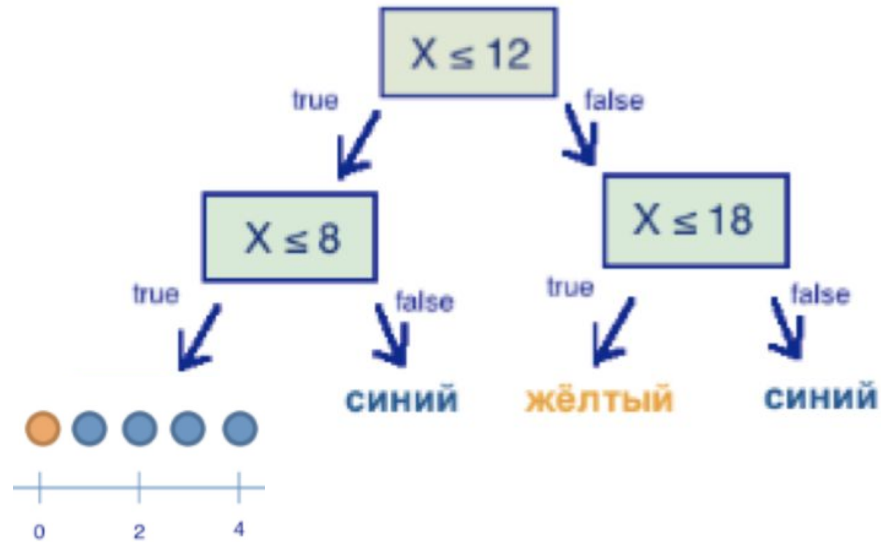
Алгоритм

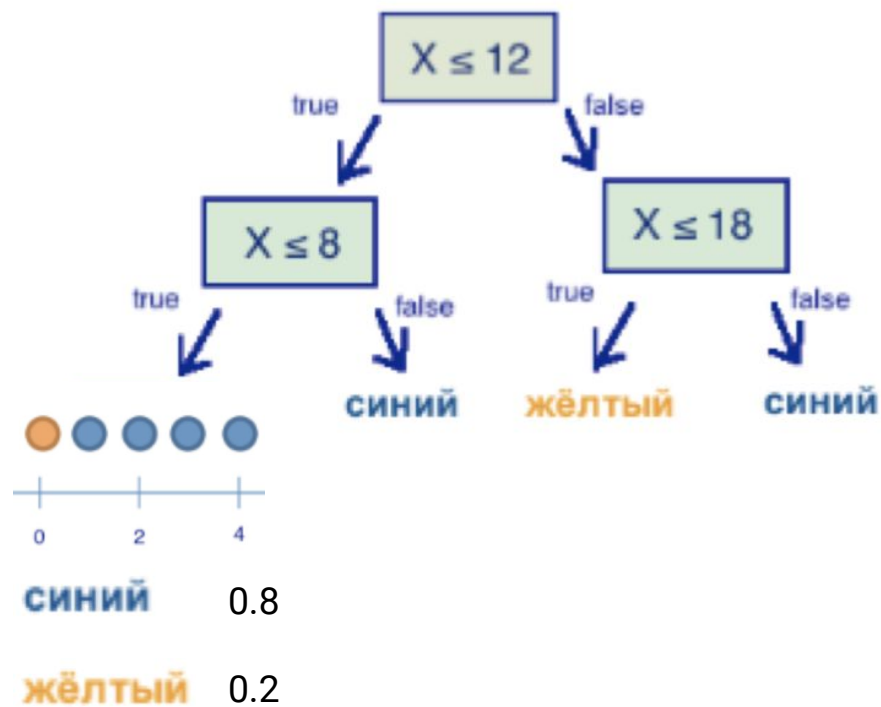
Обходим все варианты и находим разбиение с наибольшим Information Gain (IG). После того повторяем операцию для каждого из разбиений, пока все объекты из разбиения не будут одного класса.



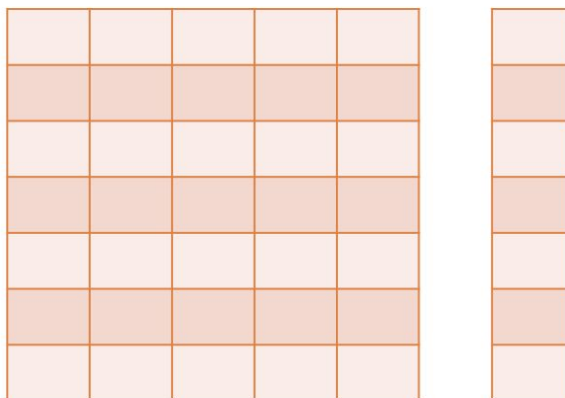




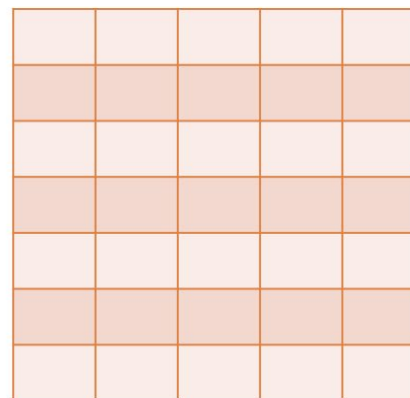




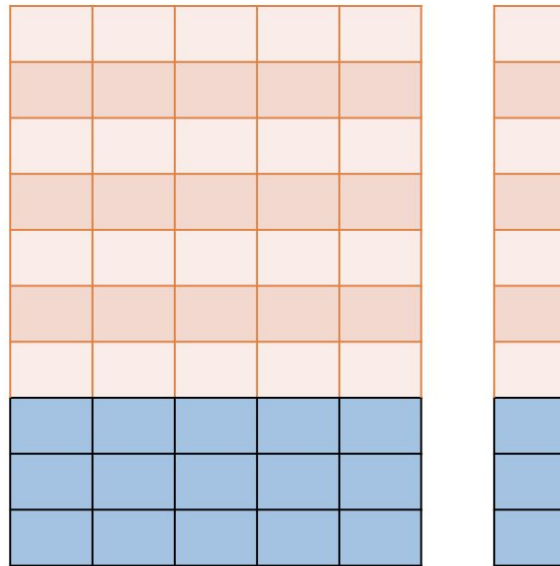
```
def build(L):  
    create node t  
    if the stopping criterion is True:  
        assign a predictive model to t  
    else:  
        Find the best binary split  $L = L_{\text{left}} + L_{\text{right}}$   
        t.left = build(L_left)  
        t.right = build(L_right)  
    return t
```



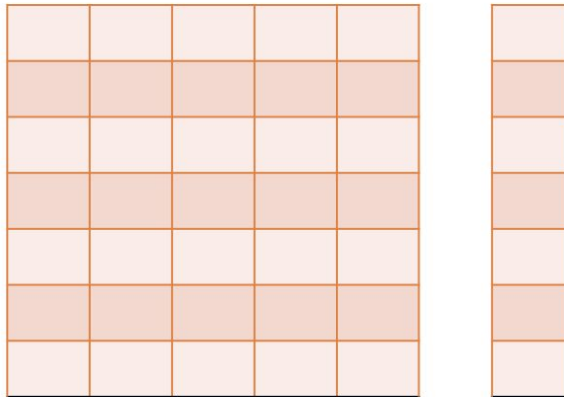
Обучающая выборка



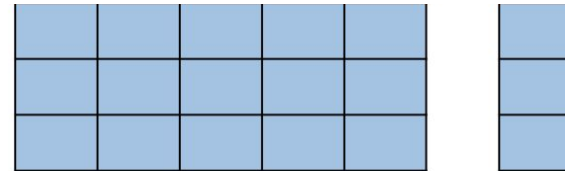
Тестовая выборка



Обучающая выборка



Обучающая выборка
 $(X_{train}; y_{train})$



Тестовая выборка
 $(X_{test}; y_{test})$

Весь датасет



Разбиваем на 5 частей

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Запуск 1					
Запуск 2					
Запуск 3					
Запуск 4					
Запуск 5					

	Обучение
	Валидация

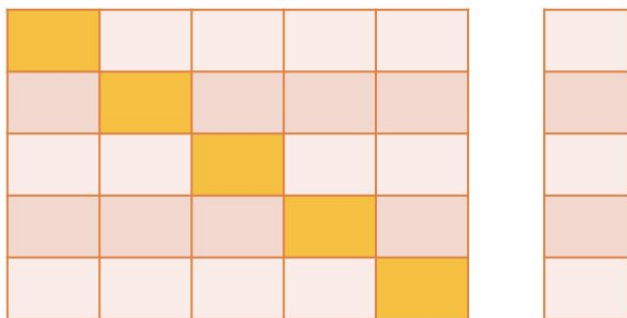
Весь датасет



Разбиваем на 5 частей

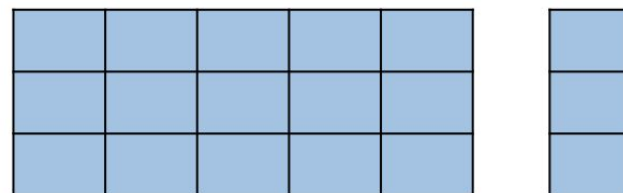
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Запуск 1					
Запуск 2					
Запуск 3					
Запуск 4					

	Обучение
	Валидация
	Не используем



Кросс-валидация на
обучающей выборке

+



Результат работы на
отложенной выборке

Плюсы

- Интерпретируемость
- Отлично подходит как базовый алгоритм для ансамбля моделей
- Мало чувствителен к выбросам
- Высокая скорость работы
- Не требует сложной предобработки данных
- Можно оценить модель с помощью статистических тестов

Минусы

- Очень склонен к переобучению
- Изменяет всю структуру дерева от небольших изменений в выборке
- Алгоритм построен на эвристиках
- Слабый алгоритм