



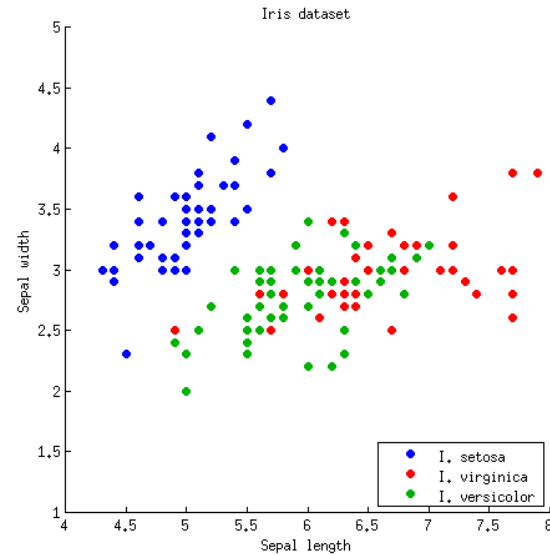
Open  
Data  
Science



Тема 4.  
Виды решающих деревьев и метрические  
алгоритмы.

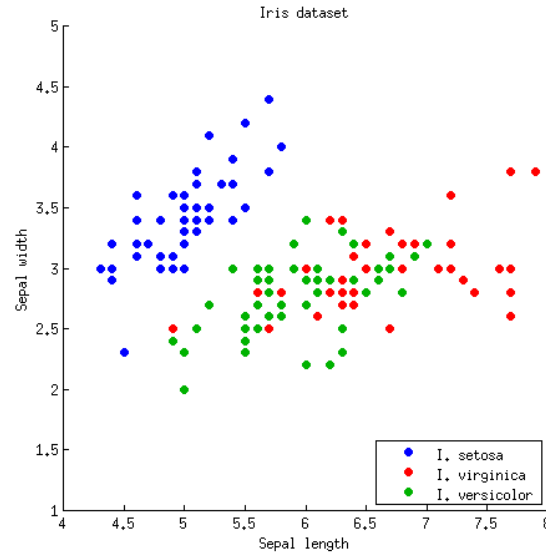


# Perfect random trees ensembles



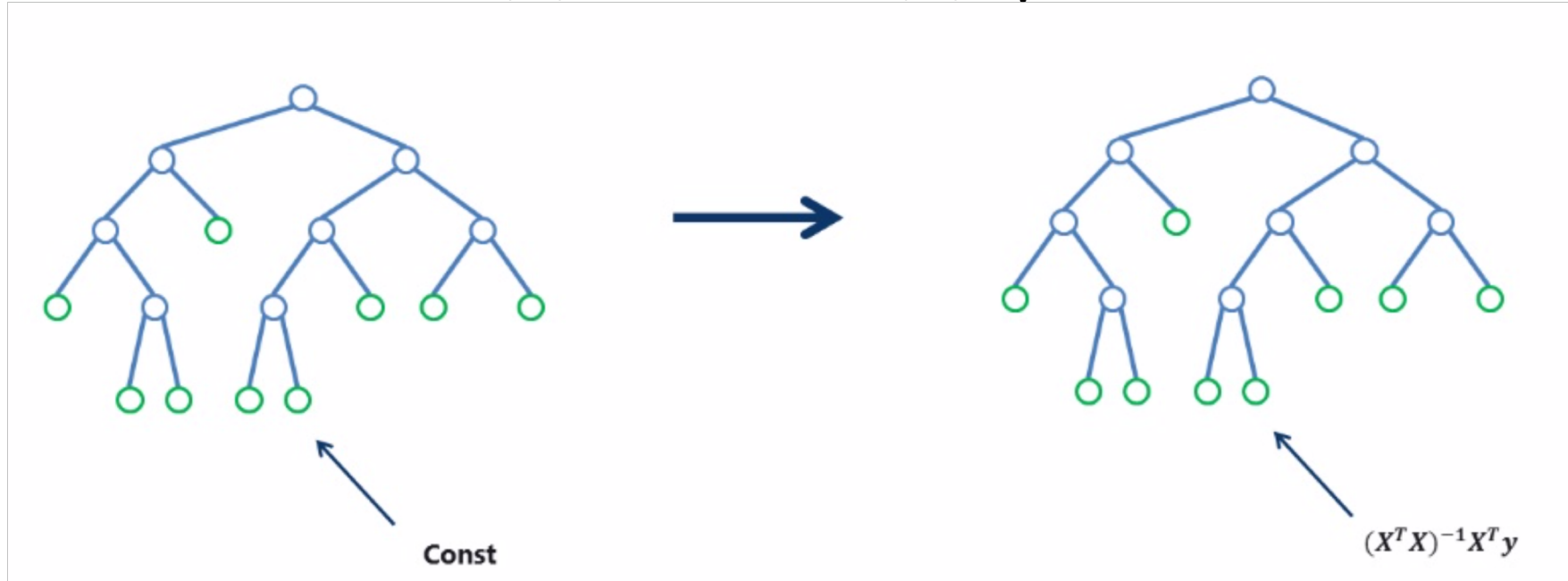
- Быстрое построение деревьев
- Качество сопоставимо с Random Forest
- Работает только с задачами классификации

# Дерево решений с линейными признаками (вариант с сеткой)



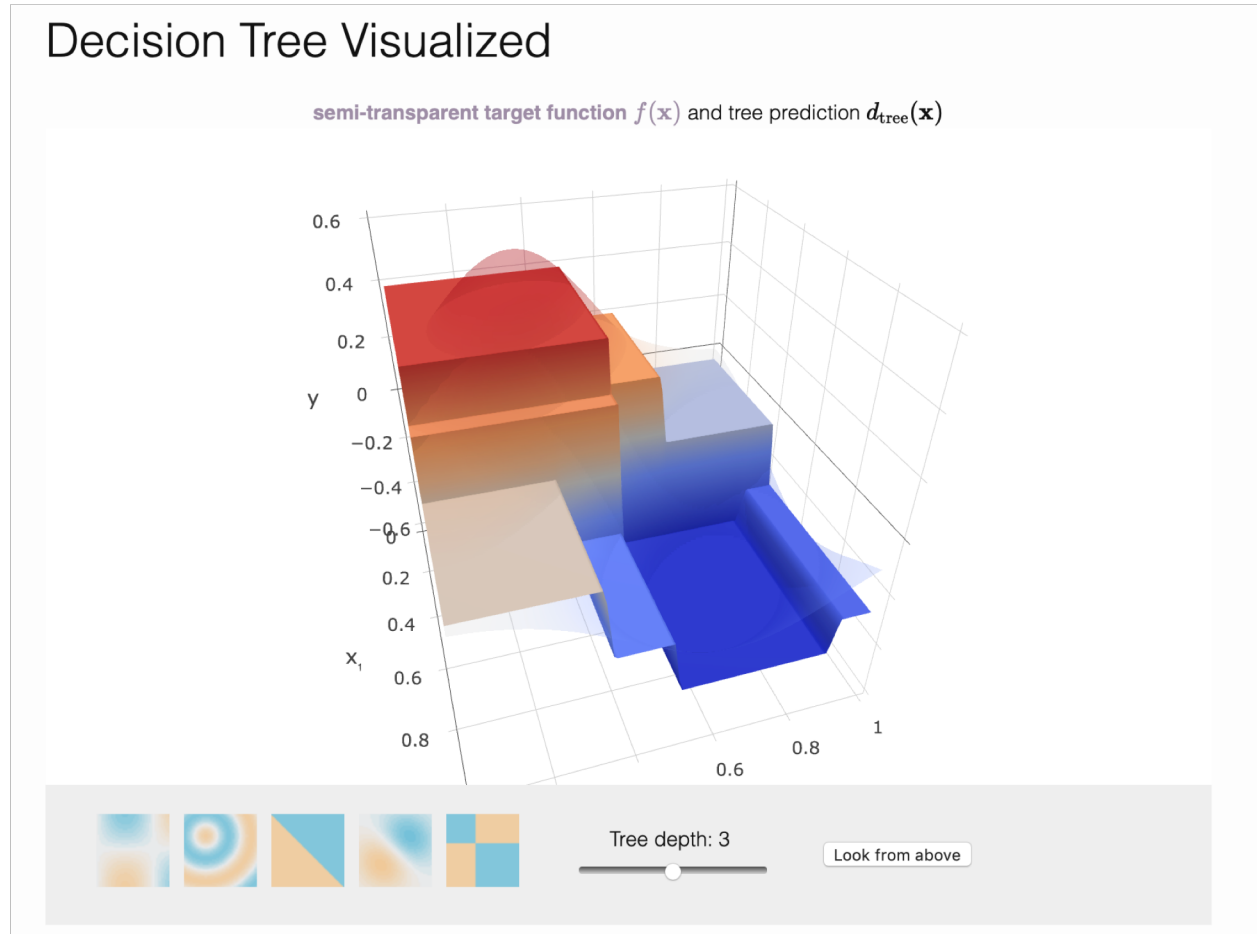
- Внутри узла перебираем не только существующие признаки, но и добавляем новые линейные признаки, построенные по сетке

# Модельное дерево



- Поместим в узлы дерева решений другой алгоритм вместо констант

# Поиграться с деревьями



[http://arogozhnikov.github.io/2016/06/24/gradient\\_boosting\\_explained.html](http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html)

# Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Структуры белков

# Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Структуры белков

+

Умеем сравнивать  
объекты между собой

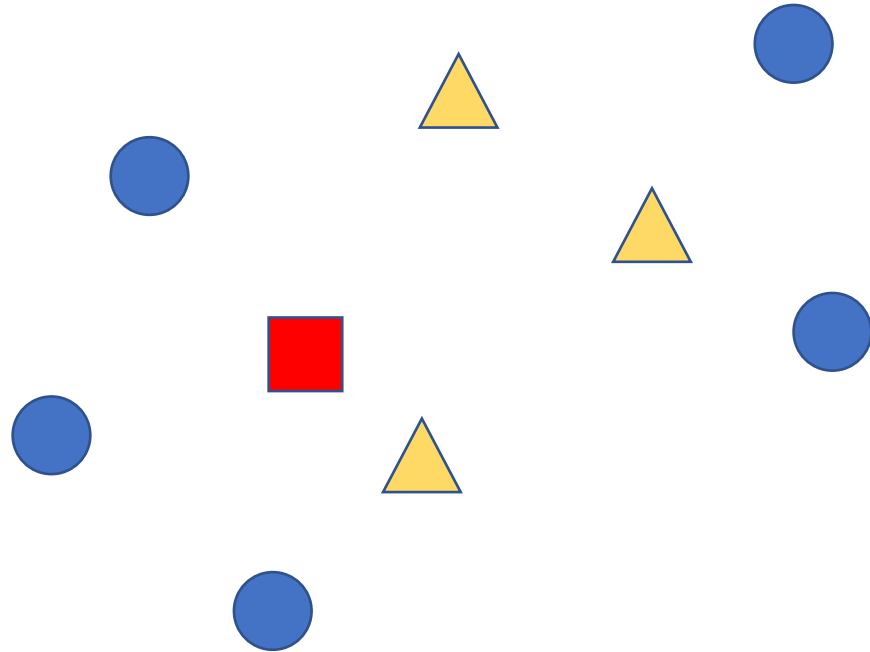
## Введем меру сходства объектов



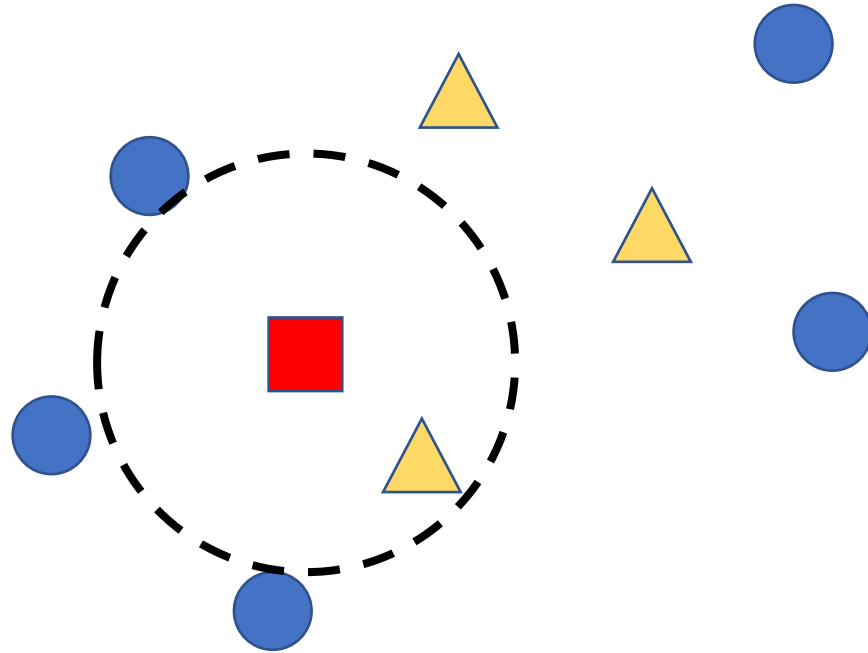
# Метрика

Пусть на множестве объектов  $X$  задана функция расстояния  $\rho: X \times X \rightarrow [0; \infty)$ . Существует целевая зависимость  $f: X \rightarrow Y$ , значения которой известны только на объектах обучающей выборки  $X_{train} = (x_i; y_i)_{i=1}^{train\ size}$ ,  $y_i = f(x_i)$ . Множество классов  $Y$  конечно. Требуется построить алгоритм классификации  $p: X \rightarrow Y$  а  $: X \rightarrow Y$ , аппроксимирующий целевую зависимость  $y^*(x)$  на всём множестве  $X$ .

# Метод ближайшего соседа



# Метод ближайшего соседа



# Метод ближайшего соседа

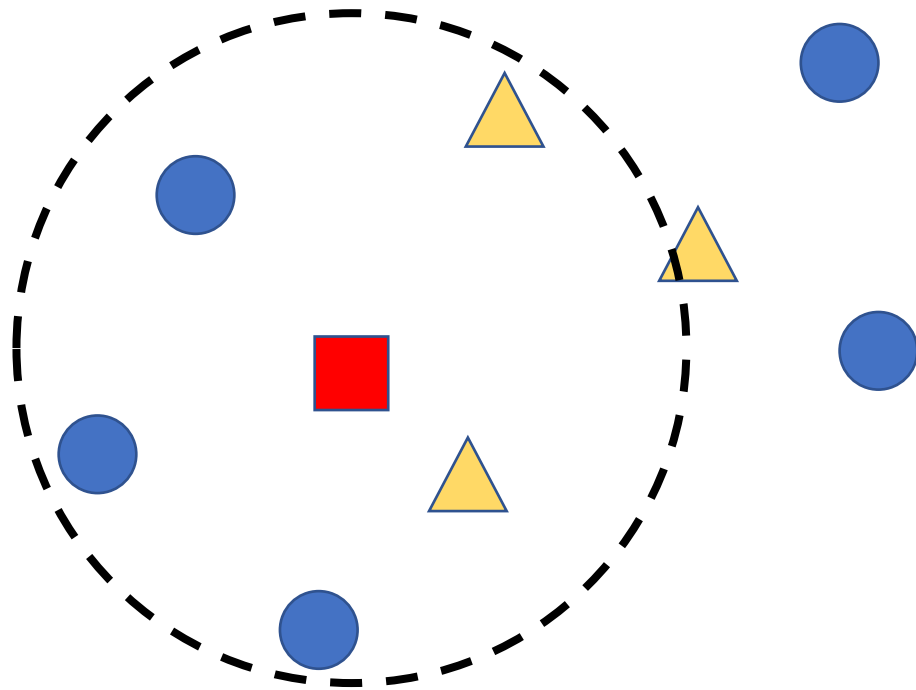
## Плюсы

- Простая реализация
- Интерпретируемость

## Минусы

- Неустойчивость к выбросам
- Мало гиперпараметров
- Низкое качество
- Надо хранить обучающую выборку

# Метод ближайших соседей



# Метод ближайших соседей

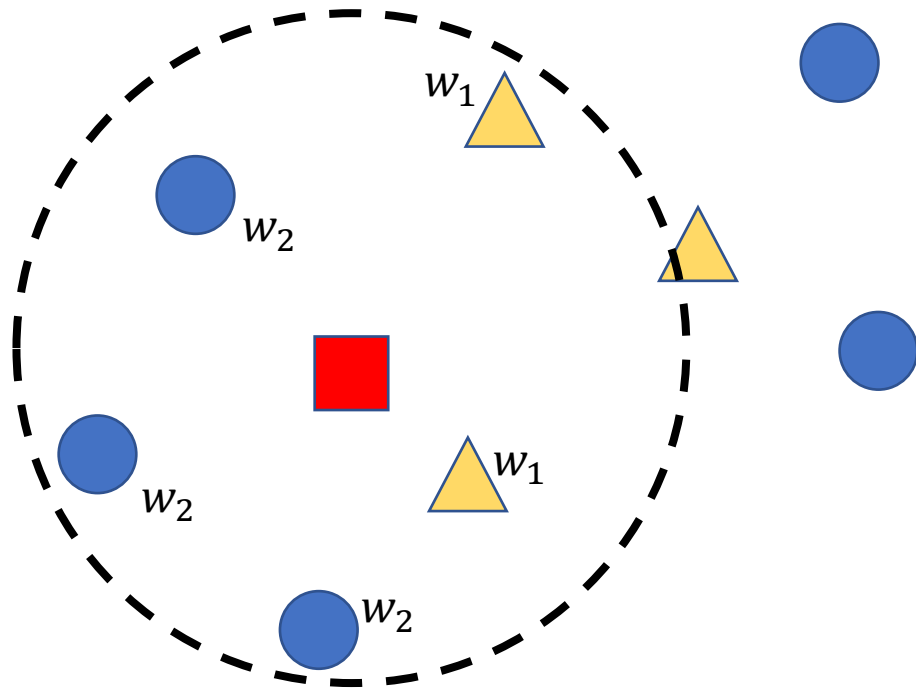
## Плюсы

- Простая реализация
- Интерпретируемость

## Минусы

- Проблемы с выбором класса
- Проблемы с несбалансированной выборкой
- Большой расход памяти
- Потенциально долгий перебор объектов

# Метод ближайших соседей (с весами)



# Метод ближайшего соседа (с весами)

## Плюсы

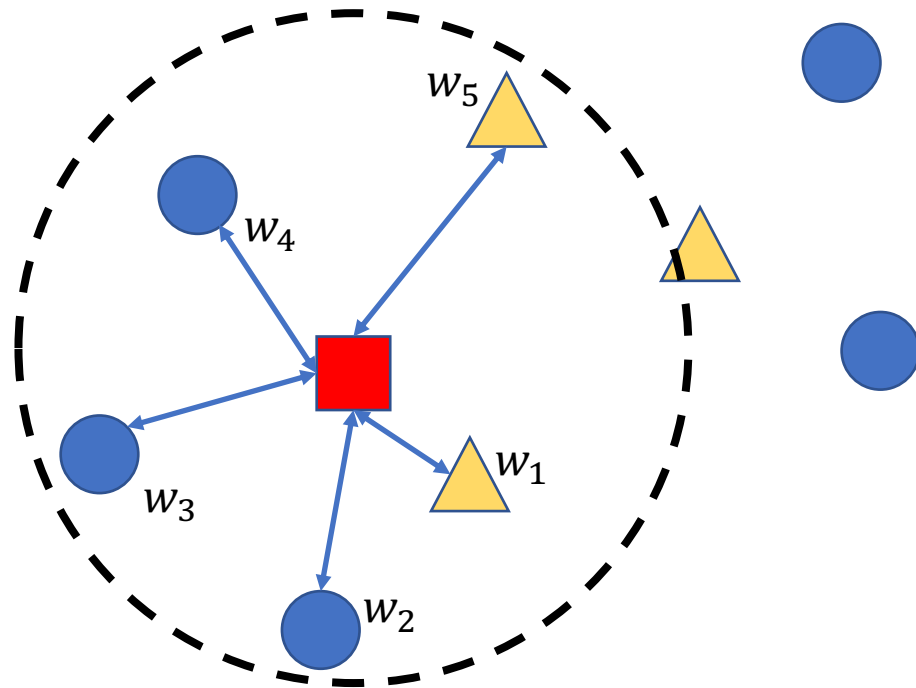
- Простая реализация
- Интерпретируемость

## Минусы

- Большой расход памяти
- Потенциально долгий перебор объектов



# Метод парзеновского окна



# Метод парзеновского окна

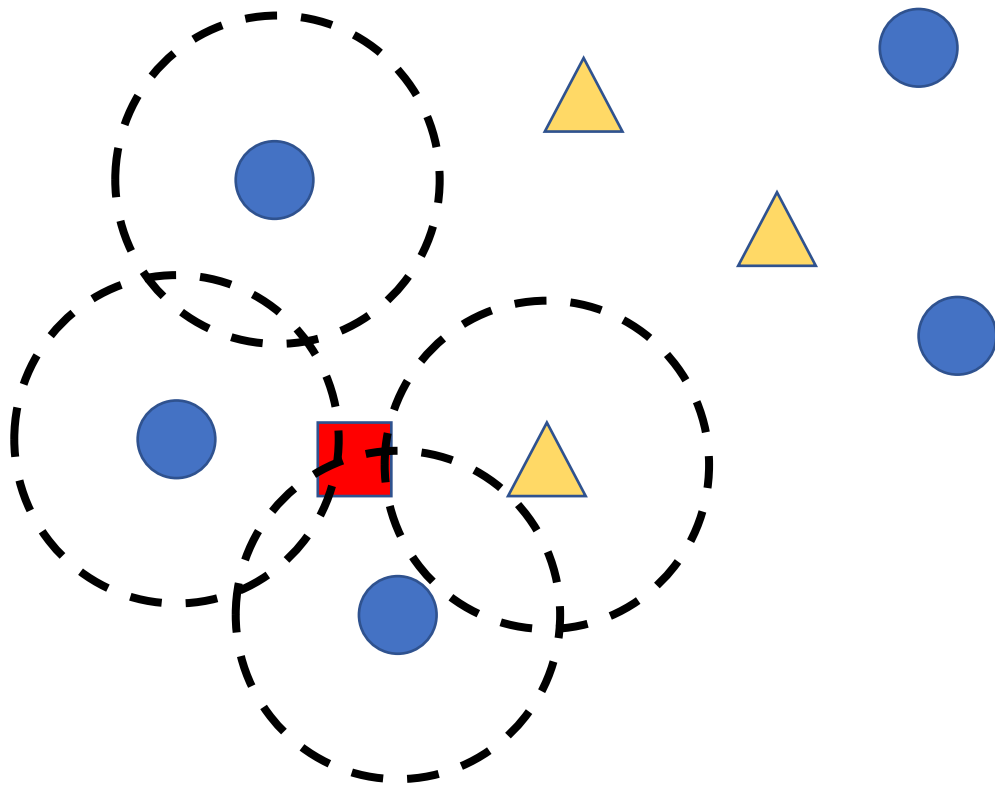
**Окно постоянной ширины:**

$$p(u; X_{train}; h; K) = \arg \max_{y \in Y} \sum_{i=1}^{train\ size} [y_{\blacksquare}^{(i)} = y] K\left(\frac{\rho(\blacksquare; x_{\blacksquare}^{(i)})}{h}\right)$$

**Окно переменной ширины:**

$$p(u; X_{train}; k; K) = \arg \max_{y \in Y} \sum_{i=1}^{train\ size} [y_{\blacksquare}^{(i)} = y] K\left(\frac{\rho(\blacksquare; x_{\blacksquare}^{(i)})}{\rho(\blacksquare; x_{\blacksquare}^{(k+1)})}\right)$$

# Метод потенциальных функций



# Отступ объекта

**Отступом** объекта  $x_i \in X_{train}$  относительно алгоритма классификации, имеющего вид  $p(u) = \arg \max_{y \in Y} \Phi_y(u)$ , называется величина:

$$M(x_i) = \Phi_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Phi_y(x_i)$$

• Эталонные	$M \gg 0$	Все слишком хорошо
• Неинформативные	$M > 0$	Стандартная ситуация
• Пограничные	$M \sim 0$	Плохая метрика
• Ошибочные	$M < 0$	Плохой алгоритм
• Шумовые	$M \ll 0$	Плохая выборка

# Виды алгоритмов классификации

## Метрические алгоритмы

- KNN
- Ядерные  
алгоритмы

## Алгоритмы, построенные на деревьях

- Деревья решений
- Random Forest
- Бустинг

## Линейные алгоритмы

- Линейные модели
- Логистрическая  
регрессия

# Сложный случай для метрических алгоритмов

X1	X2	X3	X4	X5	Y
2	11	12	17	17	2
1	10	14	6	5	1
1	16	10	15	9	1
3	2	19	17	18	3

- Есть признак X1, который очень хорошо аппроксимирует целевую функцию
- Остальные признаки являются шумом
- При всех стандартных метриках будет большая ошибка

Классная работа



[bit.ly/2lQem5C](https://bit.ly/2lQem5C)