IT-Academy

# *Алгоритм «Дерево решений»*

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ

# Вопросы занятия

1. Задача классификации: постановка и примеры
2. Дерево решений: как его построить?
3. Достоинства и недостатки деревьев решений.
4. Визуализируем принятие решений и предсказания алгоритма.

# *В конце занятия научимся:*

- применять *алгоритм классификации*, принятие решений которого *можно проинтерпретировать;*

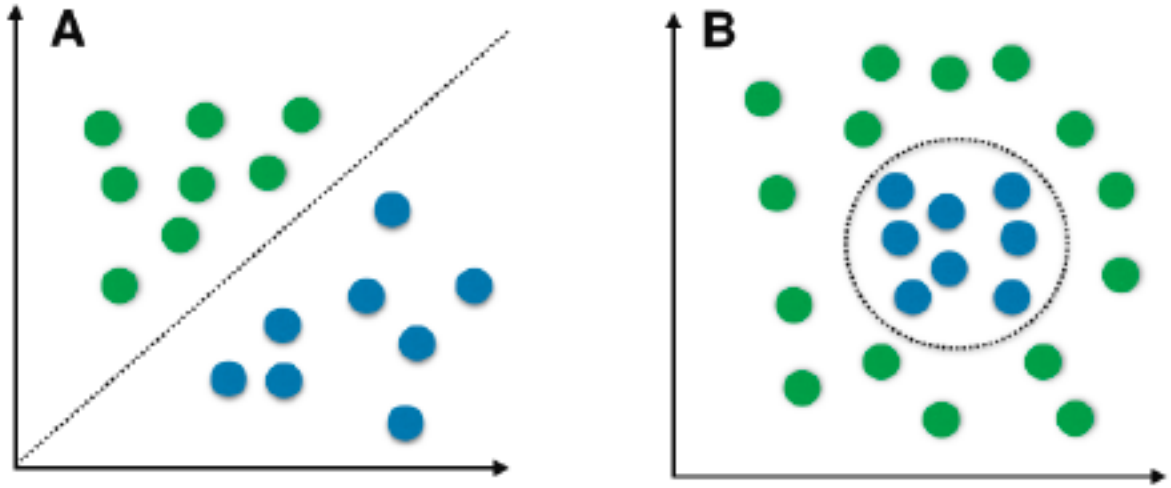
- *измерять качество* решений в задачах классификации;

- *оценивать важность* фичей.

IT-Academy

# ТИПЫ ЗАДАЧ

- классификация
- ранжирование
- регрессия
- кластеризация



# 1. ЗАДАЧА КЛАССИФИКАЦИИ

## ПРИМЕРЫ ЗАДАЧ КЛАССИФИКАЦИИ

***Скоринг.*** Оценка риска выдачи клиенту кредита? (banking, insurance)

***Отток.*** Перестанет ли пользоваться клиент услугами компании? Перестанет ли, если дать ему бонус? (marketing)

***Intent recognition.*** О чём говорит пользователь в своём обращении? (может быть несколько intent'ов, может быть древовидная структура)

***Image recognition.*** Распознавание образов

IT-Academy

# ПОСТАВНОВКА ЗАДАЧИ

Задача восстановления зависимости $y: X \to Y$, $|Y| < \infty$
по точкам *обучающей выборки* $(x_i, y_i)$, $i = 1, \ldots, \ell$.

**Дано:** векторы $x_i = (x_i^1, \ldots, x_i^n)$ — объекты обучающей выборки,
$y_i = y(x_i)$ — классификации, ответы учителя, $i = 1, \ldots, \ell$:

$$\begin{pmatrix} x_1^1 & \ldots & x_1^n \\ \ldots & \ldots & \ldots \\ x_\ell^1 & \ldots & x_\ell^n \end{pmatrix} \xrightarrow{y^*} \begin{pmatrix} y_1 \\ \ldots \\ y_\ell \end{pmatrix}$$

**Найти:** функцию $a(x)$, способную классифицировать объекты
произвольной *тестовой выборки* $\tilde{x}_i = (\tilde{x}_i^1, \ldots, \tilde{x}_i^n)$, $i = 1, \ldots, k$:

$$\begin{pmatrix} \tilde{x}_1^1 & \ldots & \tilde{x}_1^n \\ \ldots & \ldots & \ldots \\ \tilde{x}_k^1 & \ldots & \tilde{x}_k^n \end{pmatrix} \xrightarrow{a?} \begin{pmatrix} a(\tilde{x}_1) \\ \ldots \\ a(\tilde{x}_k) \end{pmatrix}$$

# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

## ЦВЕТКИ ИРИСА: ЗАДАЧА

Ирис щетинистый
(*Iris setosa*)

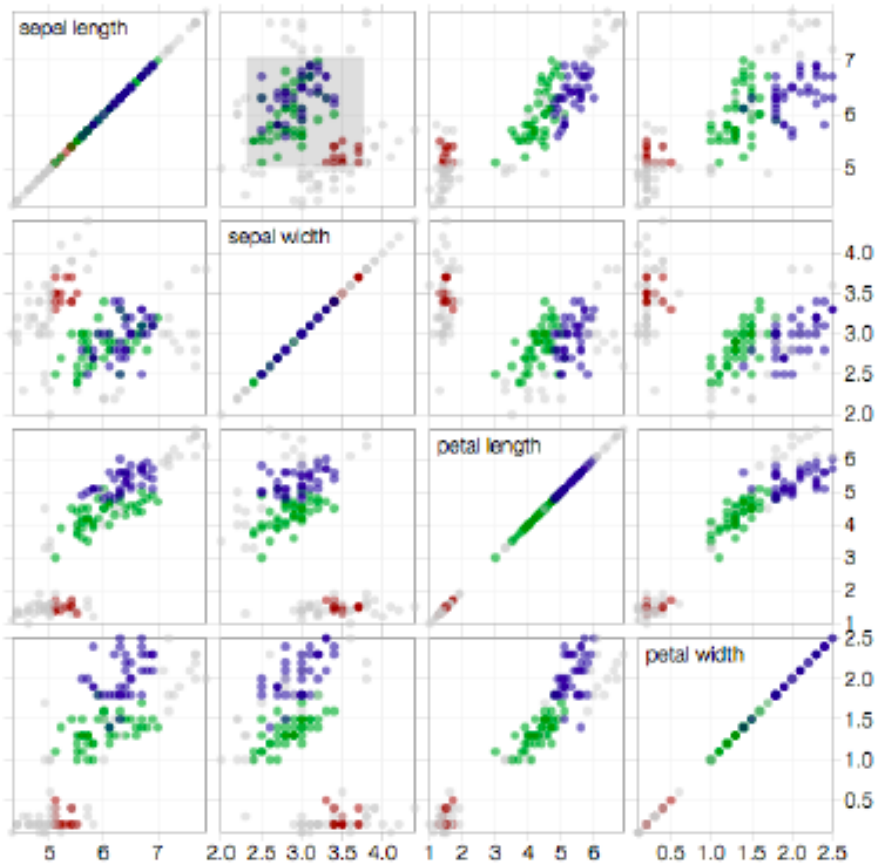
Ирис разноцветный
(*Iris versicolor*)

Ирис виргинский
(*Iris virginica*)

# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

IT-Academy

## ЦВЕТКИ ИРИСА: ДАННЫЕ




**Дано:**

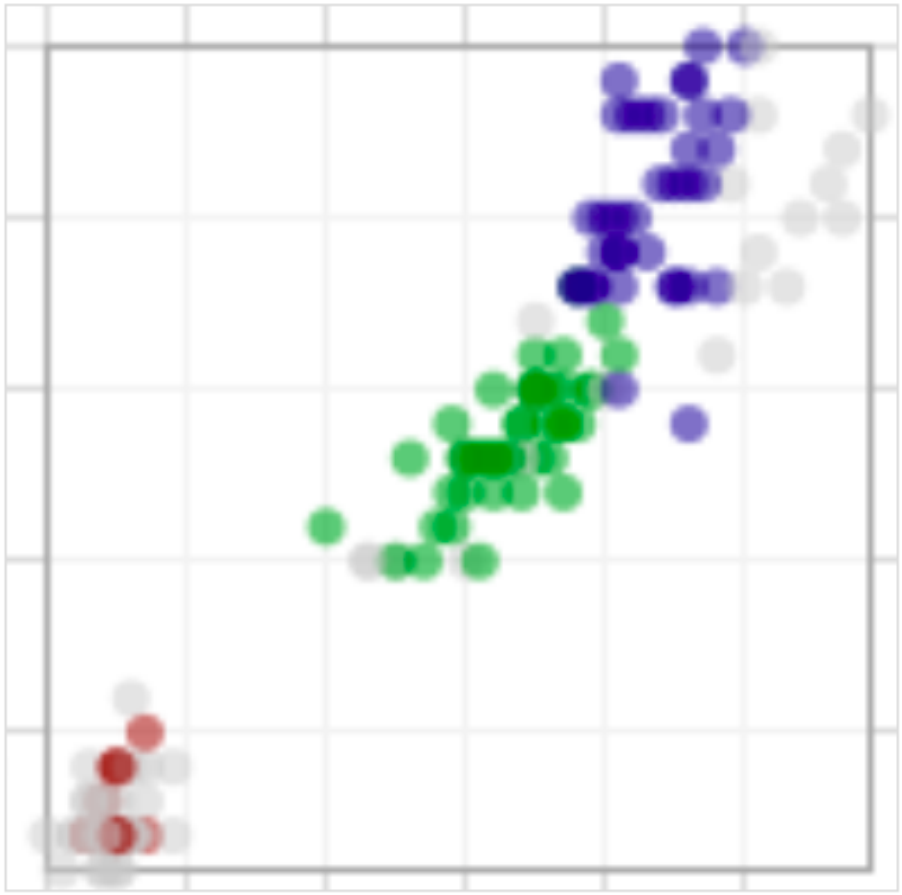
- 3 вида цветков ириса
- 4 параметра: 2 длины и 2 ширины листа
- по 50 наборов значений на каждый вид

**Найти:**

- тип цветка по 4 параметрам

# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

## ЦВЕТКИ ИРИСА: ДАННЫЕ



**Дано:**

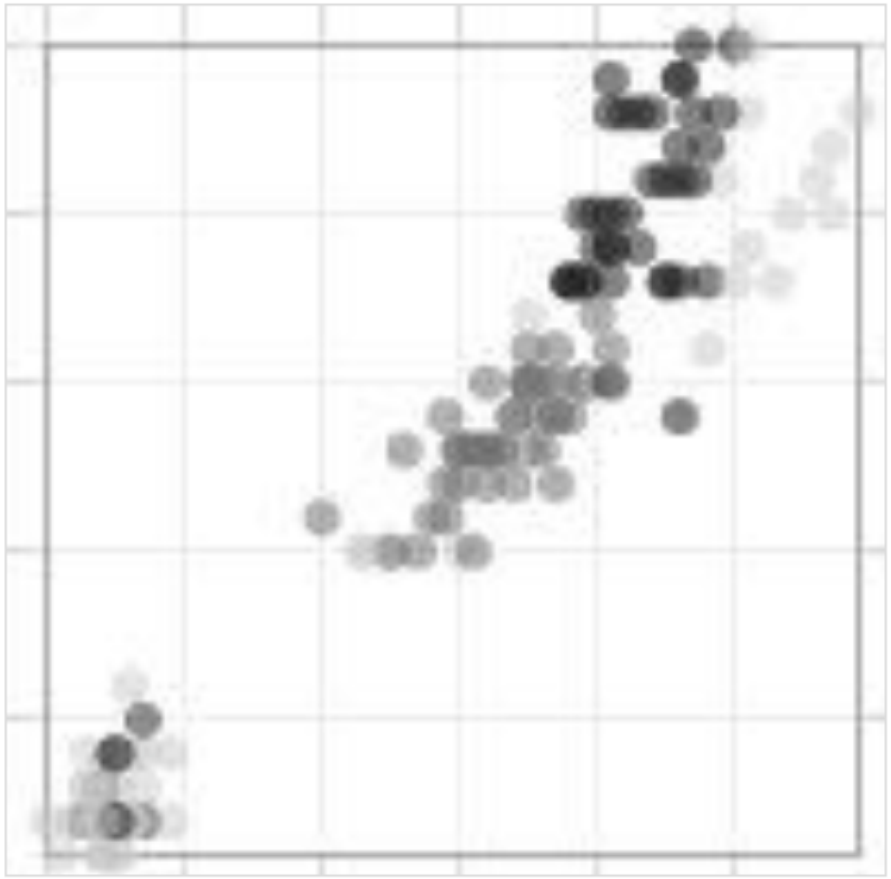
- 3 вида цветков ириса
- 4 параметра: 2 длины и 2 ширины листа
- по 50 наборов значений на каждый вид

**Найти:**

- тип цветка по 4 параметрам

# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

## ЦВЕТКИ ИРИСА: ДАННЫЕ



**Дано:**

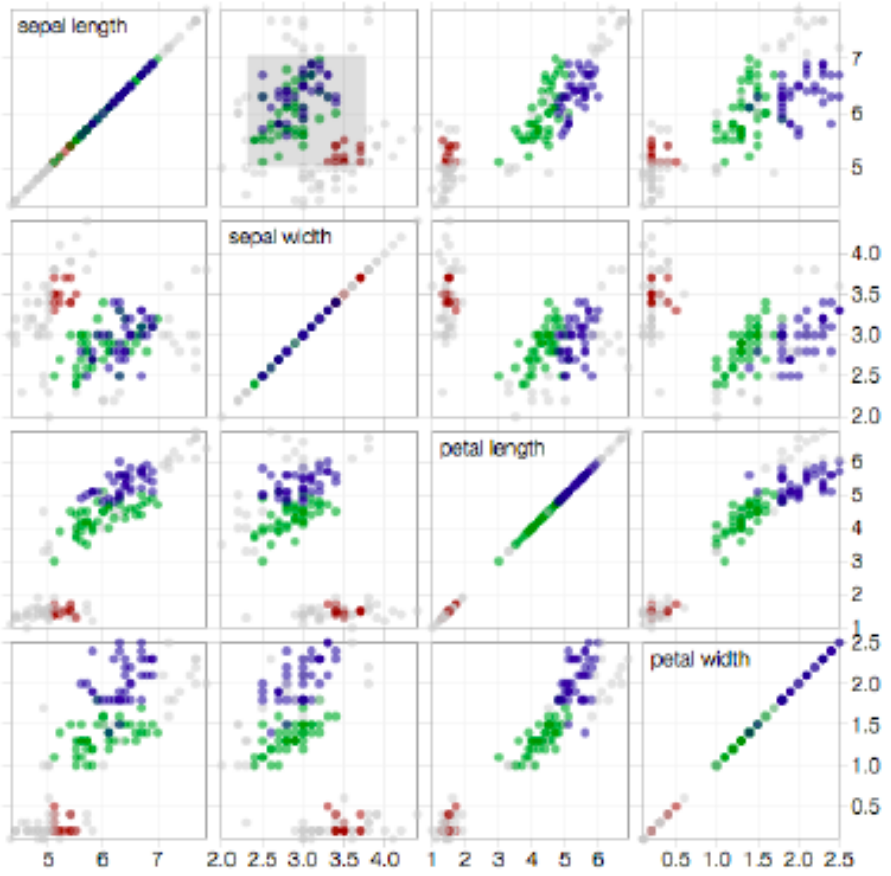
- 3 вида цветков ириса
- 4 параметра: 2 длины и 2 ширины листа
- по 50 наборов значений на каждый вид

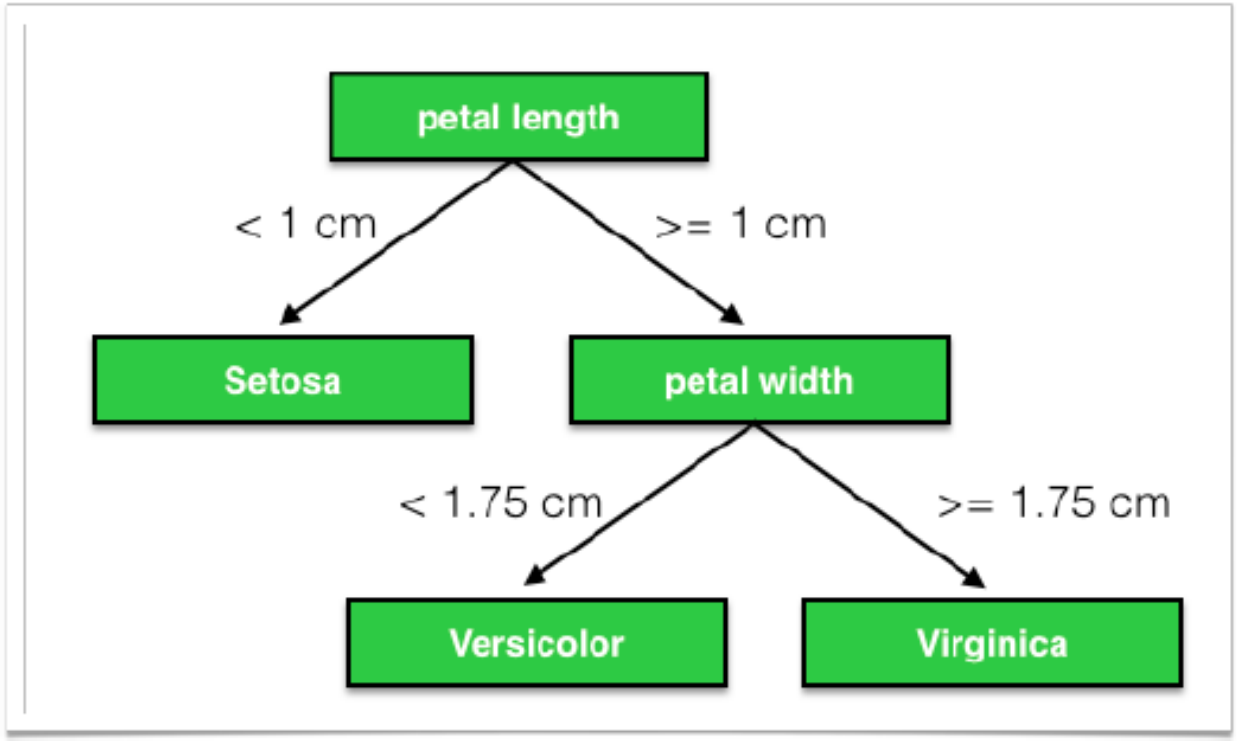
**Найти:**

- тип цветка по 4 параметрам

# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

## ЦВЕТКИ ИРИСА: РЕШАЮЩЕЕ ДЕРЕВО



IT-Academy

# ПОСТРОЕНИЕ ДЕРЕВА

**Определить:**

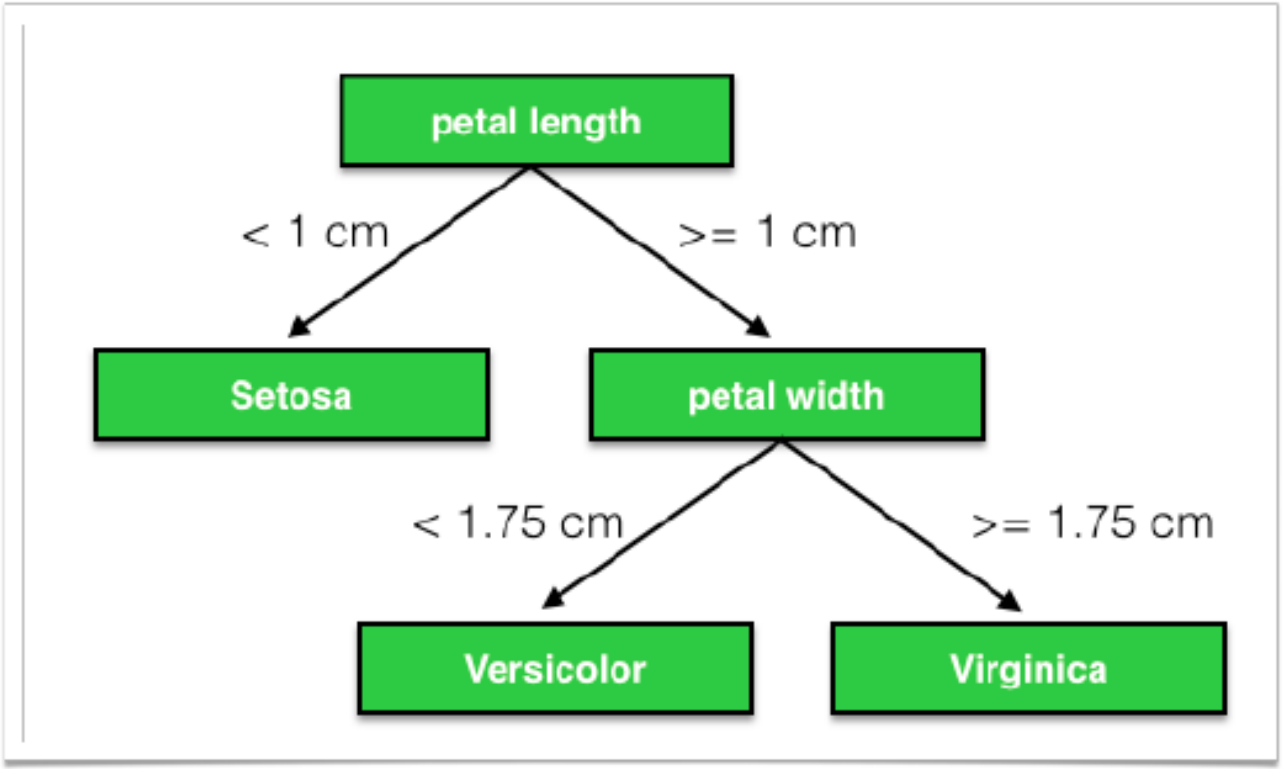
-вид правила разбиения

-критерий информативности разбиения

-критерий останова

-метод стрижки

- обработка пропусков



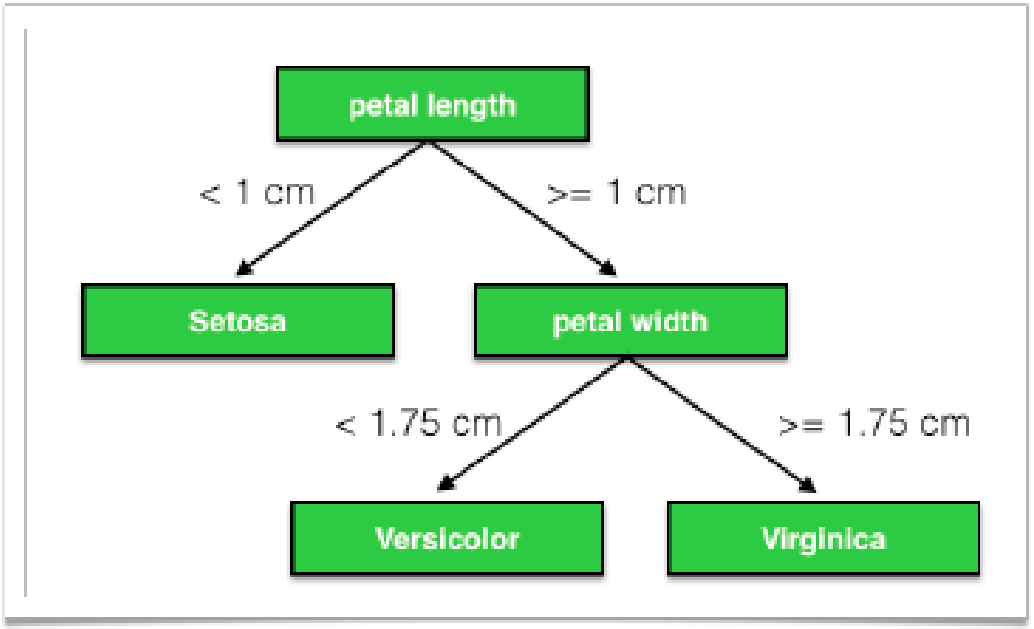
# 2. ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

## ВИД ПРАВИЛА РАЗБИЕНИЯ

**- одномерное**:
сравнивается значение одной фичи вектора x

**- линейное**:
сравнивается линейная комбинация фичей x

**- метрическое**:
расстояние до точки признакового пространства




*здесь используется одномерный предикат: сравнение идёт лишь по одной фиче из вектора признаков*

IT-Academy

# ФУНКЦИОНАЛ КАЧЕСТВА РАЗБИЕНИЯ

*Идея:*

- взять признак
- отсортировать его по возрастанию
- в зависимости от целевой переменной установить порог разделения выборки на две, максимально снижая численно выражаемый разброс внутри каждой из 2 групп
- подобрать лучшее с точки зрения улучшения разбиение

*Вопрос:* как измерить улучшение?

IT-Academy

# ИЗМЕРЕНИЕ ПОЭТАПНОГО УЛУЧШЕНИЯ


Есть 1 группа, в ней 2 класса

Пусть H(R) - «критерии информативности» группы,
больше разнообразия - больше H(R) - хуже для классификатора

Будем измерять улучшение разбиения по функционалу вида:
$IG(R) = H(R) - q_{left}*H(R_{left}) - q_{right}*H(R_{right})$, где $q_{left}$ и $q_{right}$ - доли объектов, попавших в левый или правый класс соответственно

IT-Academy

# ИЗМЕРЕНИЕ ПОЭТАПНОГО УЛУЧШЕНИЯ


$$IG(R) = H(R) - q_{left} * H(R_{left}) - q_{right} * H(R_{right})$$

$$H(R) = x > 0$$
$$H(R_{left}) = 0$$
$$H(R_{right}) = 0$$

$$IG(R) = x - 5/9 * 0 - 4/9 * 0 = x > 0$$

IT-Academy

# КРИТЕРИЙ ДЖИНИ


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

*K - количество классов*
*$p_k$ - доля класса в выборке*

$$IG(R) = H(R) - q_{left}*H(R_{left}) - q_{right}*H(R_{right})$$

$$H(R) = 4/9*(1-4/9) + 5/9*(1-5/9) = 0.494$$
$$H(R_{left}) = 3/4*(1-3/4) + 1/4*(1-1/4) = 0.375$$
$$H(R_{right}) = 1/5*(1-1/5) + 4/5*(1-4/5) = 0.32$$

$$IG(R) = 0.494 - 4/9*0.375 - 5/9*0.32 = \mathbf{0.15}$$

IT-Academy

# ЭНТРОПИЙНЫЙ КРИТЕРИЙ


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

*K - количество классов*
*$p_k$ - доля класса в выборке*

$$IG(R) = H(R) - q_{left} * H(R_{left}) - q_{right} * H(R_{right})$$

$$H(R) = -4/9*\log2(4/9) - 5/9*\log2(5/9) = 0.991$$
$$H(R_{left}) = -3/4*\log2(3/4) - 1/4*\log2(1/4) = 0.811$$
$$H(R_{right}) = -1/5*\log2(1/5) - 4/5*\log2(4/5) = 0.722$$

$$IG(R) = 0.991 - 4/9*0.811 - 5/9*0.722 = \mathbf{0.229}$$

# ПРАКТИЧЕСКИЕ ЗАДАНИЯ

IT-Academy

1. Построить критерии информативности: джини и энтропийный

IT-Academy

# КРИТЕРИИ ОСТАНОВА

-останов, когда в каждом листе объекты только одного класса

-ограничение max глубины дерева

-ограничение min число объектов в листьях

- требование улучшения функционала качества при дроблении не менее, чем x или на x%

IT-Academy

# ПРОБЛЕМА ПРОПУСКОВ

- удалить объекты с пропусками из обучающей;

- замена на значения вне средние, медианные;

- заменить на значения вне области значений фич;

- модифицировать алгоритм построения и работы дерева: включать элементы с пропусками в обе ветки дерева, но взвешивать качество разбиения по объёму пропусков

IT-Academy

# ДОСТОИНСТВА

- легко интерпретировать, визуализировать, «белый ящик»;

- простота подготовки данных: не требуется нормализация, dummy переменные, возможны пропуски;

- скорость работы.

IT-Academy

# НЕДОСТАТКИ

- острая проблема переобучения;

- неустойчивость;

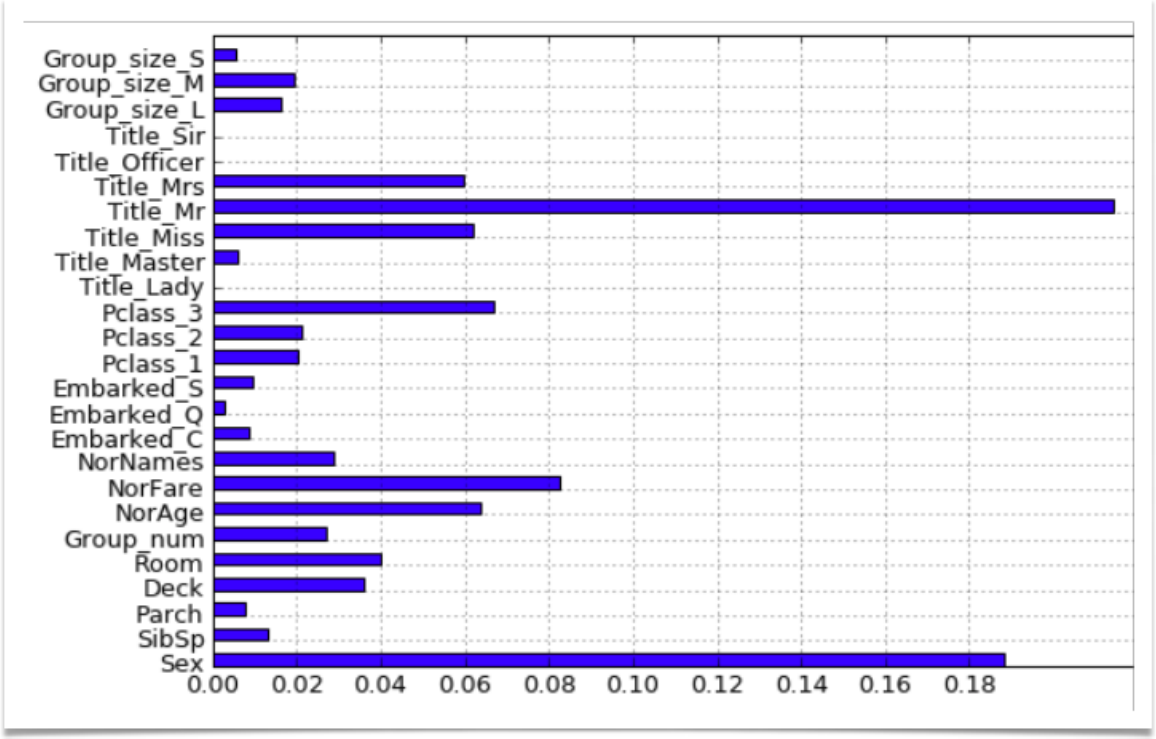
- не учитывает нелинейные зависимости или даже простые линейные, которые идут не по осям координат;

- чувствителен к несбалансированным классам;

- хорошо интерполирует, плохо экстраполирует.

# РЕАЛИЗАЦИЯ В SKLEARN

Деревья могут оценивать важность фичей. Метод: feature_importances_




Например, судя по решению, на выживаемость на Титанике сильнее всего влияли:

* наличие в обращении «Mr.»
* пол
* уровень дохода
* проживание в 3 классе
* возраст
* наличие в обращении «Mrs» / «Miss»

# ПРАКТИЧЕСКИЕ ЗАДАНИЯ

IT-Academy

2. Обучить дерево решений на цветках ириса

3. Нарисовать дерево принятий решений

4. Оценить важность фичей

# ЧТО МЫ СЕГОДНЯ УЗНАЛИ

IT-Academy

1. Деревья решений, объединённые в «лес», составляют одни из наиболее сильных алгоритмов. По одиночке же они являются слабыми, зато очень легко интерпретируемыми и визуализируемыми алгоритмами.

2. Деревья позволяют оценивать важность признаков.

IT-Academy

# *Алгоритм «Дерево решений»*

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ