

ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ № 06

Древовидные модели. Решающие деревья

СОСТАВИТЕЛЬ: КАНД. ТЕХН. НАУК БЫКАДОР В.С.

Общие сведения о решающих деревьях

Решающие деревья (деревья решений, decision trees) являются моделями машинного обучения, которые очень часто используются в системах поддержки принятия решений (СППР).

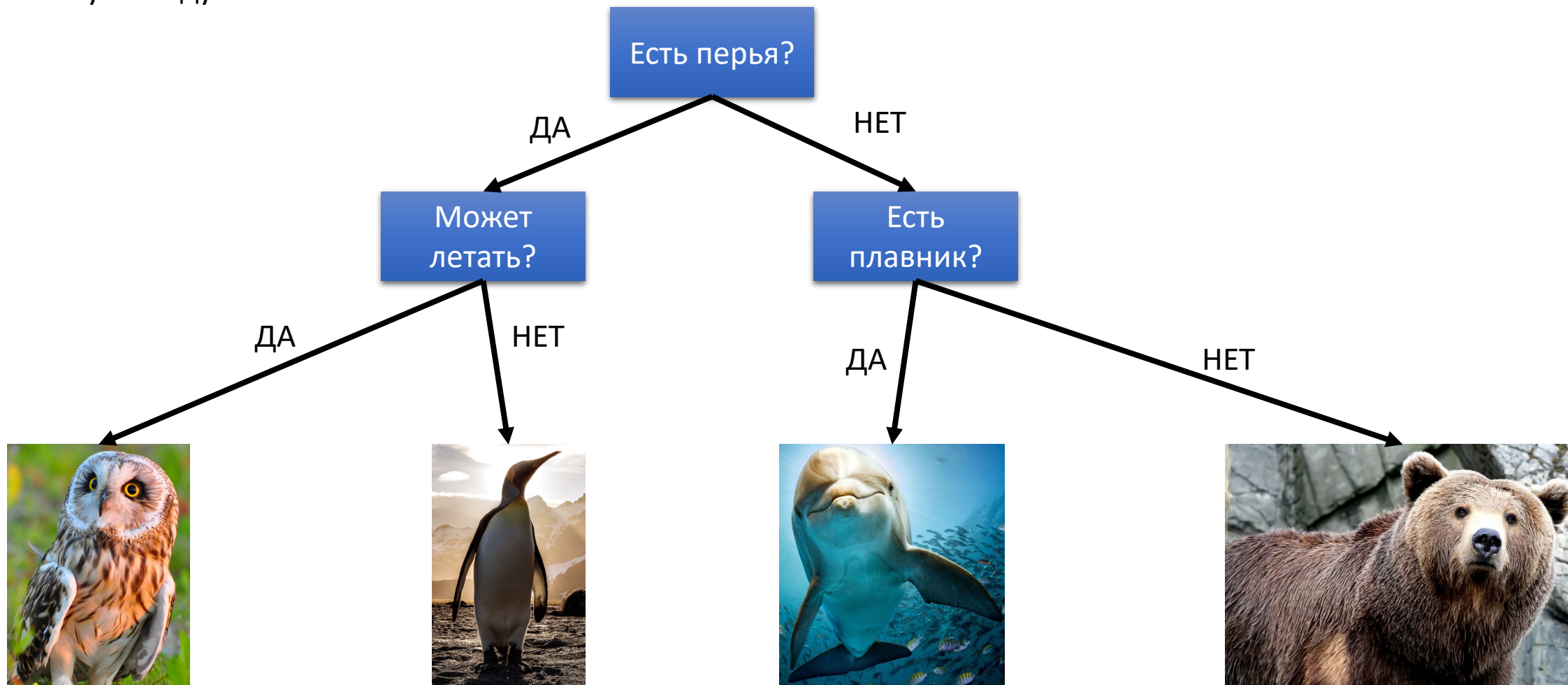
Решающие деревья могут быть деревьями классификации, то есть используются для отнесения объекта к одному из известных классов, деревьями регрессии, то есть используются для прогнозирования значения какой-либо величины. Но так же деревья позволяют решать и другие задачи машинного обучения, в том числе и задачи ранжирования, кластеризации, оценивания вероятностей.

Деревья решений часто используют в следующих сферах деятельности:

- **производство** – оценка продукта соответствующим критерия качества;
- **банковское дело** – оценка заёмщика денежных средств;
- **медицина** – постановка диагноза;
- **торговля** – классификация товаров и покупателей.

Элементарное представление о решающем дереве

Решающее дерево представляет собой структурированный набор вопросов, ответы на которые приводят к тому или иному выводу.



Источники фото животных:

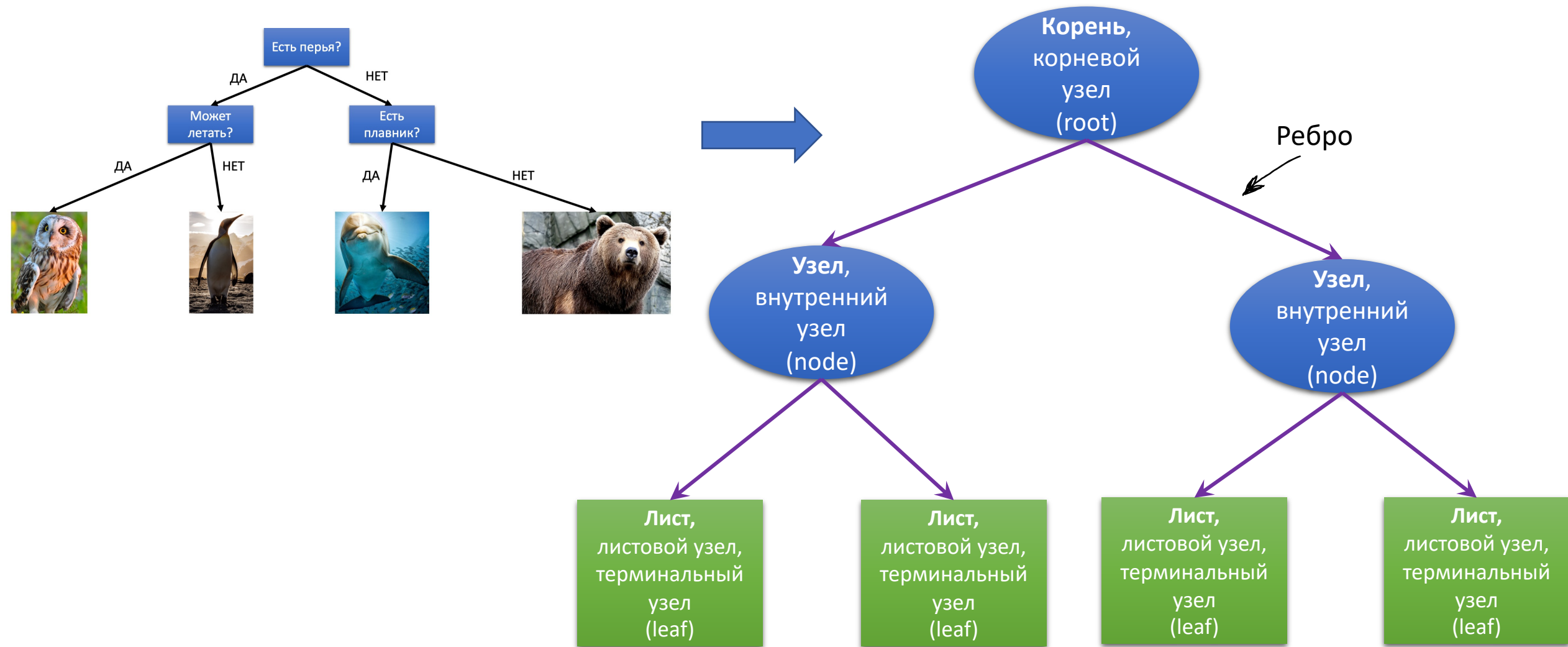
сова - https://akspic.ru/image/140291-ptica-snezhnaya_sova-nazemnye_zhivotnye-sova-zivotnoe

пингвин - https://akspic.ru/image/125711-neletayushhaya_ptica-ptica-zhivaya_priroda-zoopark-pingvin

дельфин - https://akspic.ru/image/138898-podzemnye_vody-delfin-zhivaya_priroda-afalina-kitoobraznyh

медведь - https://akspic.ru/image/98237-nazemnye_zhivotnye-zhivaya_priroda-grizli-morda-medved

Терминология деревьев



Признаки, классы и их кодирование в моделях решающих деревьев

Для обобщенных математических рассуждений и программной реализации моделей машинного обучения термины «Есть перья?» и «Дельфин» не подойдут. Потребуется определить что есть признак, что есть класс в терминах машинного обучения и дать какие-то формальные обозначения этим терминам.

Для начала совсем немного вспомним некоторые обозначения и операции из булевой алгебры:

Логические выражения вида: **ПРИЗНАК = ЗНАЧЕНИЕ** называется *литералом*.

Составные булевы выражения строятся при помощи логических операций:

- конъюнкции (логическое умножение, операция «И») \wedge ;
- дизъюнкции (логическое сложение, операция «ИЛИ») \vee ;
- отрицания (операция «НЕ») \neg ;
- импликации (\approx операция «ЕСЛИ ... ТО ...») \rightarrow ;

Признаки, классы и их кодирование в моделях решающих деревьев

Каждый из вопросов дерева классификации животных по сути является признаком, который может принимать значения из множества: {ДА, НЕТ} или {1, 0}, где 1 = ДА, а 0 = НЕТ.

Обозначим вопросы из дерева классификации животных как признаки:

| | | |
|-----------------|--|----------------------------|
| «Есть перья?» | = T (от англ. «Fea T her» - перо) | $\Rightarrow T = \{1, 0\}$ |
| «Может летать?» | = L (от англ. «F L y» - летать) | $\Rightarrow L = \{1, 0\}$ |
| «Есть Плавник?» | = N (от англ. «Fi N » - плавник) | $\Rightarrow N = \{1, 0\}$ |

Обозначим животных как классы:



$$\Rightarrow \hat{c}(L) = O$$

Owl



$$\Rightarrow \hat{c}(L) = P$$

Penguin



$$\Rightarrow \hat{c}(L) = D$$

Dolphin



$$\Rightarrow \hat{c}(L) = B$$

Bear

Для программной реализации каждый класс тоже можно закодировать:

$$O = 1$$

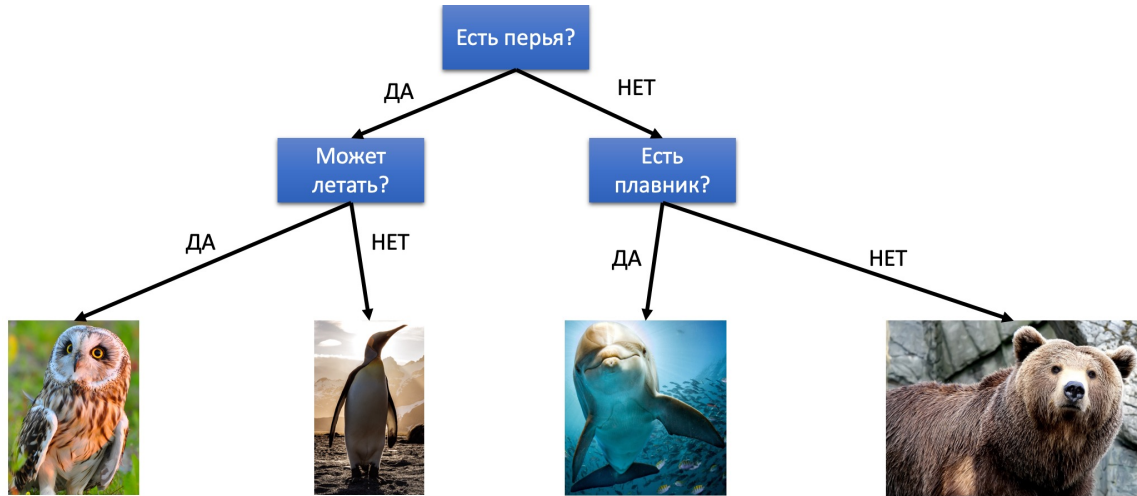
$$P = 2$$

$$D = 3$$

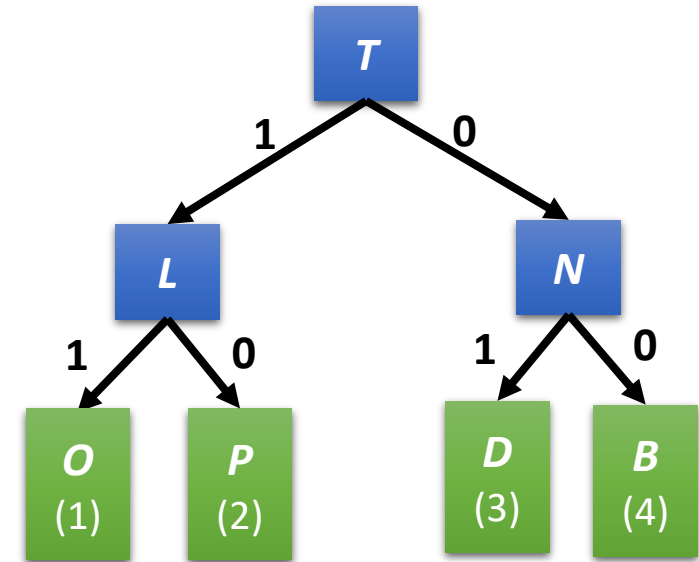
$$B = 4$$

Признаки, классы и их кодирование в моделях решающих деревьев

Дерево решений в описательной форме



Дерево решений в закодированной форме



L в выражениях классификатора вида $\hat{c}(L) = P$ означает логическое выражение. Тогда применяя это логическое выражение можно получить соответствующих класс, то есть отнести объект к одному из уже известных классов.

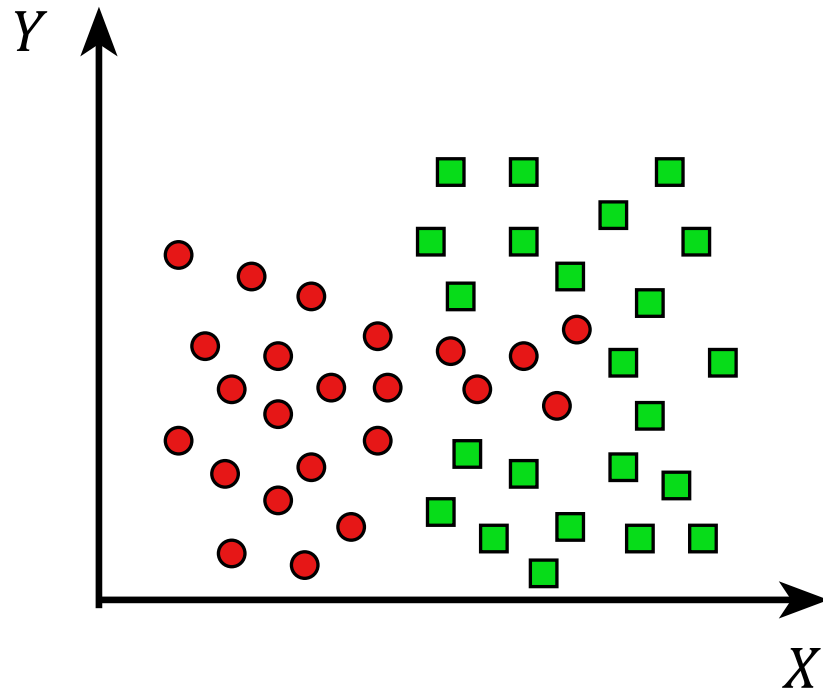
$$L = (T = 1 \wedge L = 1) \rightarrow O$$

$$L = (T = 1 \wedge L = 0) \rightarrow P$$

$$L = (T = 0 \wedge L = 1) \rightarrow D$$

$$L = (T = 0 \wedge L = 0) \rightarrow B$$

Признаки, классы и их кодирование в моделях решающих деревьев



Непрерывные значения признаков:

$$X \in \mathbb{R}$$

$$Y \in \mathbb{R}$$



Два класса:

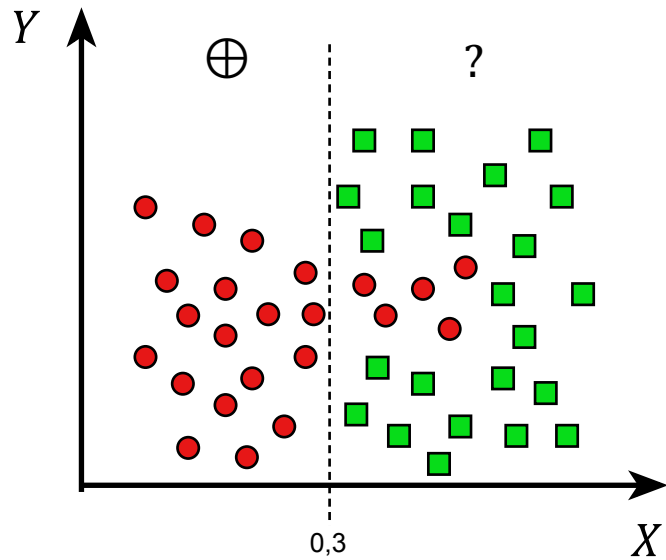


$\Rightarrow \oplus = +1$ («кружки», «положительный», +1)



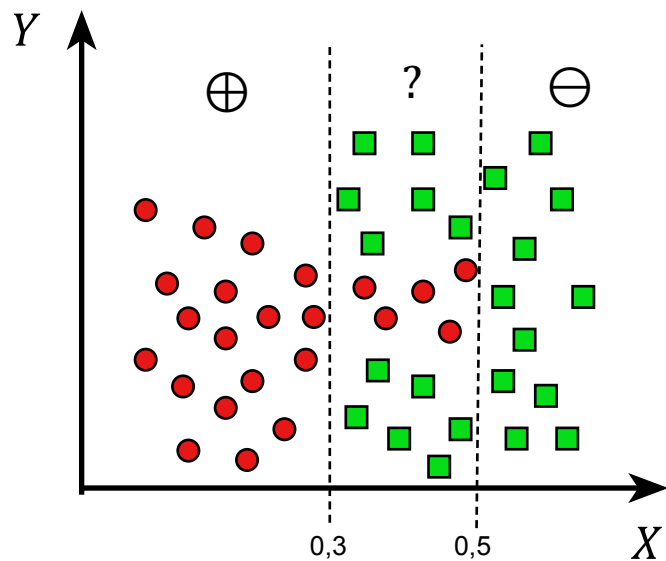
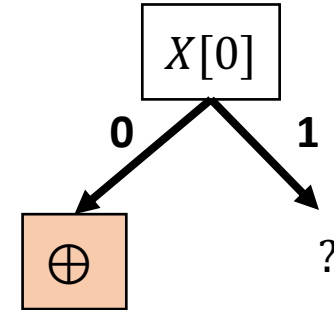
$\Rightarrow \ominus = -1$ («квадратики», «отрицательный», -1)

Признаки, классы и их кодирование в моделях решающих деревьев



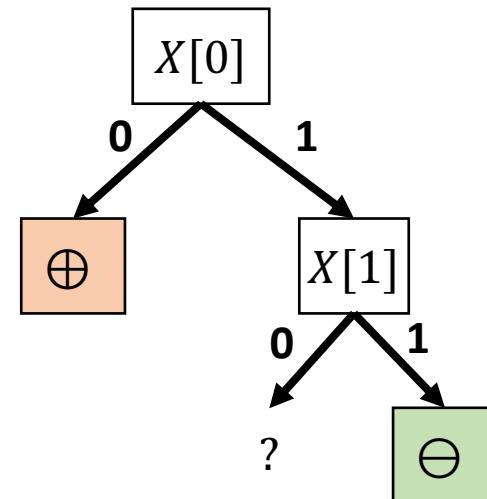
$$X < 0,3 \Rightarrow X[0] = 0 \Rightarrow \hat{c}(L) = \oplus$$

$$X \geq 0,3 \Rightarrow X[0] = 1 \Rightarrow \hat{c}(L) = ?$$

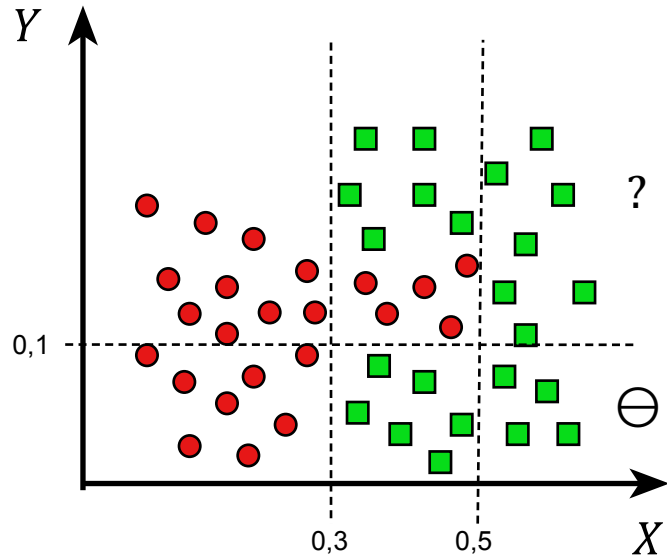


$$X < 0,5 \Rightarrow X[1] = 0 \Rightarrow \hat{c}(L) = ?$$

$$X \geq 0,5 \Rightarrow X[1] = 1 \Rightarrow \hat{c}(L) = \ominus$$

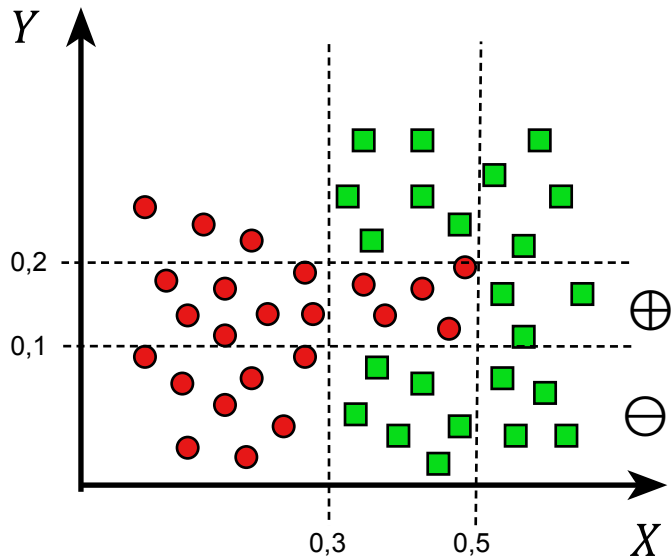


Признаки, классы и их кодирование в моделях решающих деревьев



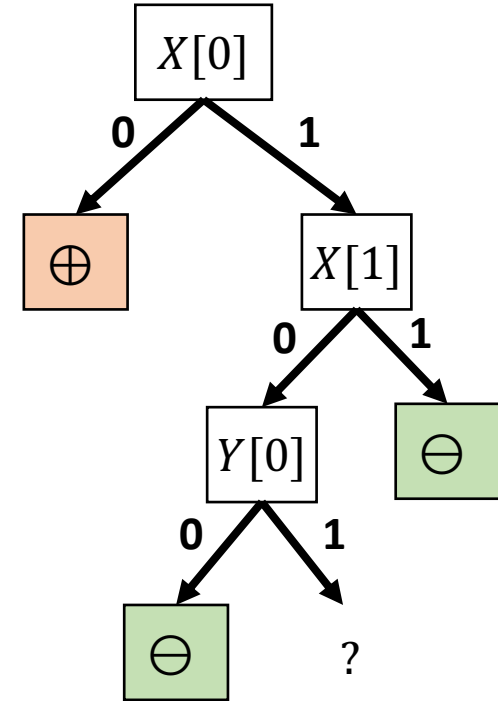
$$Y < 0,1 \Rightarrow Y[0] = 0 \Rightarrow \hat{c}(L) = \Theta$$

$$Y \geq 0,1 \Rightarrow Y[0] = 1 \Rightarrow \hat{c}(L) = ?$$

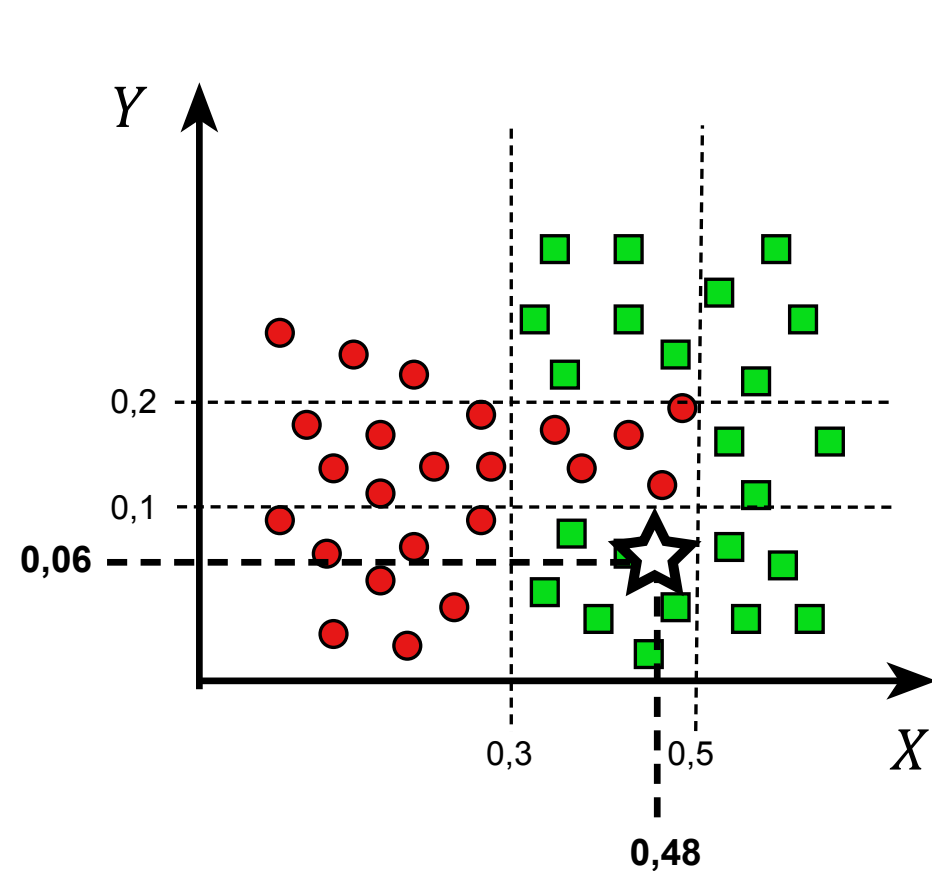


$$Y < 0,2 \Rightarrow Y[1] = 0 \Rightarrow \hat{c}(L) = \oplus$$

$$Y \geq 0,2 \Rightarrow Y[1] = 1 \Rightarrow \hat{c}(L) = \Theta$$



Признаки, классы и их кодирование в моделях решающих деревьев



$X = 0.48$
 $Y = 0.06$

$$(X = 0.48) \Rightarrow (X \geq 0.3) \Rightarrow (X[0] = 1)$$

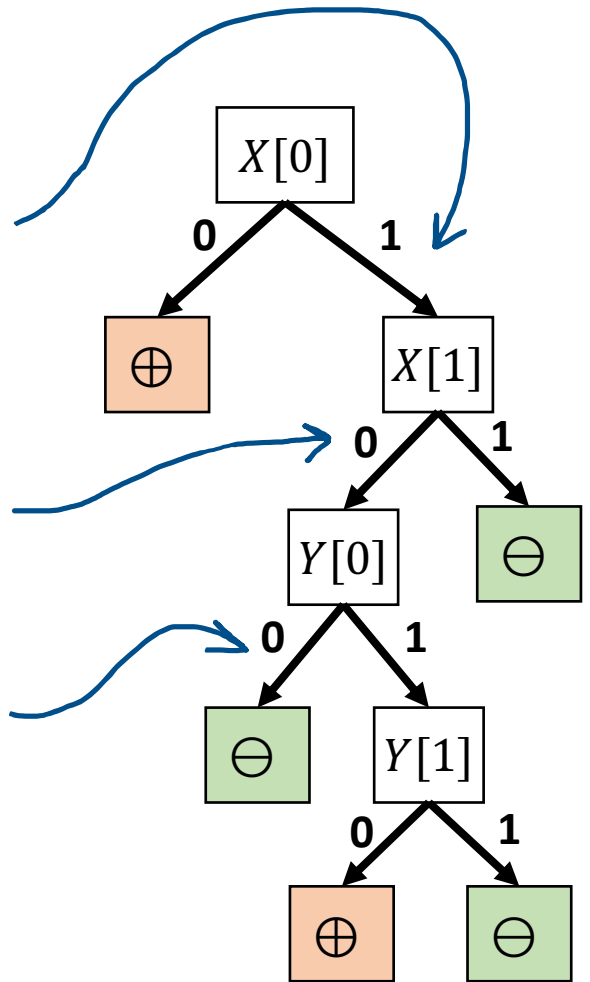
$$(X = 0.48) \Rightarrow (X < 0.5) \Rightarrow (X[1] = 0)$$

$$(Y = 0.06) \Rightarrow (Y < 0.1) \Rightarrow (Y[0] = 0)$$

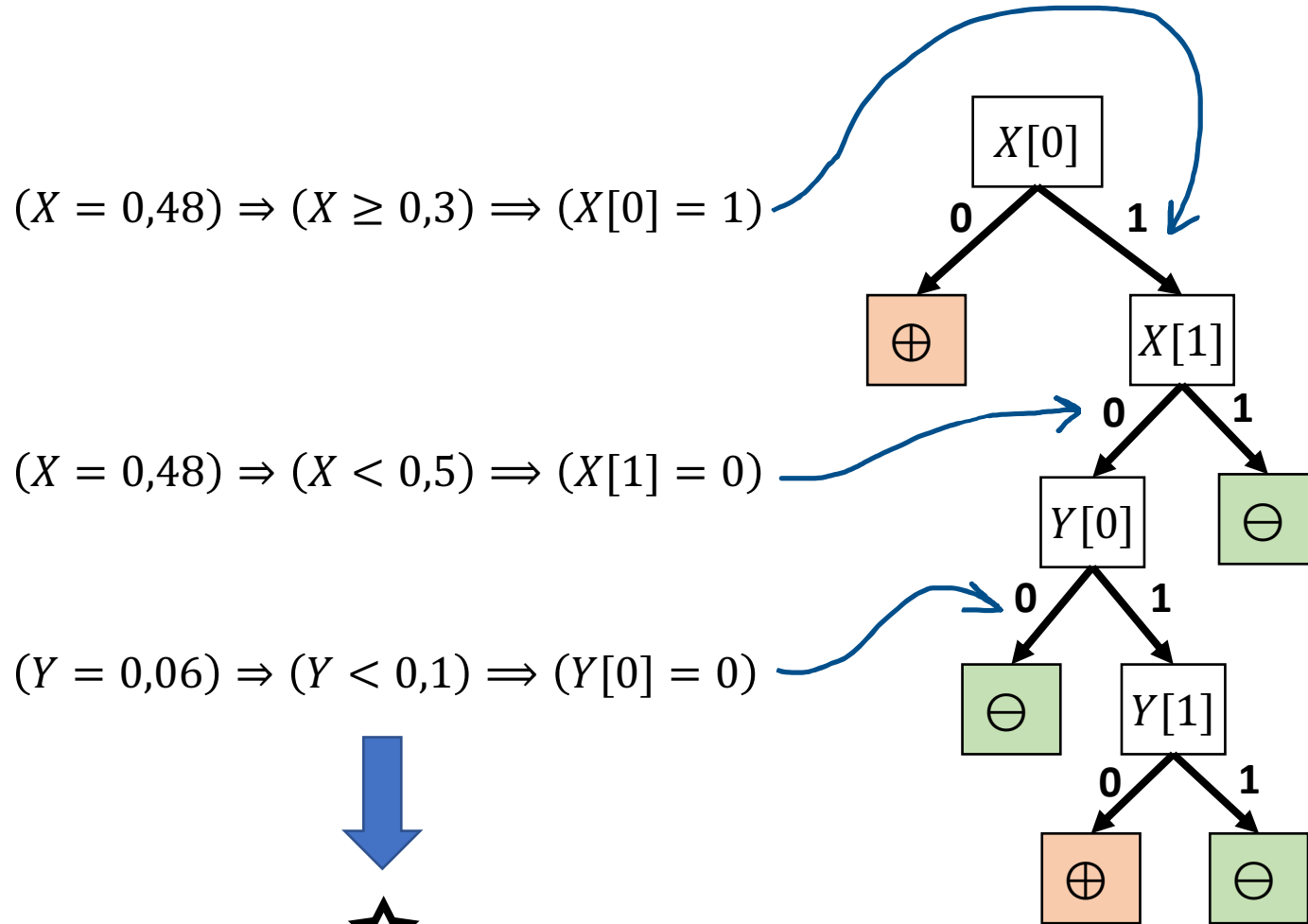


$$\star \Rightarrow \hat{c}(L) = \ominus$$

(«квадратики», «отрицательный», -1)



Признаки, классы и их кодирование в моделях решающих деревьев



$\star \Rightarrow \hat{c}(L) = \ominus$

Тогда для \star можно записать логическое выражение:

$$L = (X[0] = 1 \wedge X[1] = 0 \wedge Y[0] = 0) \rightarrow \ominus$$

Тогда можно составить полные логические выражение для классов \ominus и \oplus обходя дерево решений:

$$\begin{aligned}
 L_{\ominus} = & (X[0] = 1 \wedge X[1] = 1) \vee \\
 & \vee (X[0] = 1 \wedge X[1] = 0 \wedge Y[0] = 0) \vee \\
 & \vee (X[0] = 1 \wedge X[1] = 0 \wedge Y[0] = 1 \wedge Y[1] = 1)
 \end{aligned}$$

$$\begin{aligned}
 L_{\oplus} = & (X[0] = 0) \vee (X[0] = 1 \wedge X[1] = 1) \vee \\
 & \vee (X[0] = 1 \wedge X[1] = 0 \wedge Y[0] = 1 \wedge Y[1] = 0)
 \end{aligned}$$

Данные логические выражения программно могут быть реализованы через логические операции с применением (или без) оператора ветвления (If).

Формальные определения

Так как деревья решений строятся на основе признаков, то их ещё называют деревья признаков. Тогда можно дать Формальное определение деревьев признаков.

Определение

Деревом признаков называется такое дерево, в котором каждый узел помечен признаком, а каждое ребро, исходящее из узла, помечено литералом.

Определение

Множество литералов в узле называется **разделением**.

Определение

Каждый лист дерева представляет **логическое выражение**, являющееся **конъюнкцией литералов**, встречающихся на пути от корня дерева к этому листу.

Обучение дерева решений

Пример обучающих данных

| Возраст | Пол | Город | Образование | Уровень дохода | | Класс клиента |
|---------|-------|--------------|-------------|----------------|-------|---------------|
| 23 | Ж | Москва | высшее | высокий | | 2 |
| 44 | М | Пермь | среднее | средний | | 1 |
| 54 | М | С. Петербург | среднее | средний | | 1 |
| 19 | Ж | Ростов н/Д | высшее | низкий | | 2 |
| 86 | М | Воронеж | высшее | средний | | 1 |
| 55 | М | Владивосток | высшее | средний | | 0 |
| 35 | М | Дербент | высшее | средний | | 0 |
| 34 | Ж | С. Петербург | высшее | высокий | | 1 |
| 27 | Ж | Ростов н/Д | среднее | средний | | 2 |
| | | | | | | |
| 32 | М | Москва | отсутствует | средний | | 1 |
| 53 | М | Астрахань | высшее | низкий | | 2 |
| 89 | Ж | Мурманск | среднее | средний | | 3 |

Обучение дерева решений

Обучить дерево решений означает автоматически построить его иерархическую структуру.

Проблема существующие при обучении дерева решений: **нужно правильно выбирать последовательность узлов**, какой узел будет корневым, какой будет вторым и так далее. При различных порядках узлов получаются **деревья разной структуры**, разной глубины. В начале нужно выбирать такие признаки, которые максимально позволяют отделить классы объектов, затем более специфичные. Из-за большого количества информации «в ручную» определить правильную последовательность признаков бывает практически невозможно.

Обучение дерева решений

Рассмотрим ещё один пример.

Пусть мы собираемся разработать классификатор животных и собираем результаты измерений всех имеющихся особей по следующим признакам:

| | |
|-----------------------------|--------------|
| Длина = {1, 2, 3} | => {1, 2, 3} |
| Когти = {длинные, короткие} | => {1, 0} |
| Зубы = {мало, много} | => {0, 1} |

Допустим у нас есть 10 особей животных и они уже размечены на два класса:

«положительный» – \oplus ;
«отрицательный» – \ominus .

Обучение дерева решений

Результаты измерений признаков двух классов у 10-ти животных.

Длина = {1, 2, 3}

Когти = {длинные, короткие}

Зубы = {мало, много}

[3+, 1-] [1+, 0] [1+, 4-]

[0+, 2-] [3+, 5-]

[2+, 3-] [1+, 4-]

Запись [3+, 1-] означает, что из всех 10-ти измеренных животных 4-ре животных имели длину тела равную 1-му метру, при этом 3-ое животных относились к классу \oplus , а 1-но животное относилось к классу \ominus .

3 из 4 относились к классу \oplus

1 из 4 относились к классу \ominus

[3+, 1-]

4 из 10 животных

Обучение дерева решений

Нечистота разделения обучающих примеров

$[3+, 1-]$ - это разделение

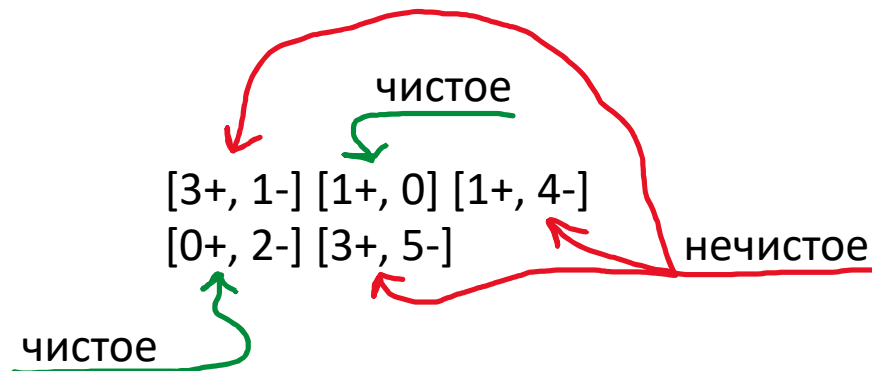
Формально разделение можно записать как: $[n^{\oplus}, n^{\ominus}]$

где n^{\oplus} - количество положительных примеров, примеров относящихся к классу \oplus ;
 n^{\ominus} - количество отрицательных примеров, примеров относящихся к классу \ominus .

Если разделение состоит из примеров, только одного класса, то такое разделение называется **чистым**, иначе - **нечистым**.

$[n^{\oplus}, 0]$ и $[0, n^{\ominus}]$ – чистые разделения.

Длина = {1, 2, 3}
Когти = {длинные, короткие}



Обучение дерева решений

Нечистота разделения обучающих примеров

Чем чище разделение для соответствующего признака, тем лучше данный признак позволяет разделить классы. В идеале, дерево решений должно однозначно отделять одни классы от других, правда это может увеличить глубину дерева, что не очень хорошо.

При обучении дерева первыми узлами выбирают признаки дающие наиболее чистые разделения.

Признак с самыми чистыми разделениями является корневым.

В разделении может быть разное количество положительных и отрицательных примеров, более того один и тоже признак для различных значений имеет несколько разделений (Длина = {1, 2, 3} [3+, 1-] [1+, 0] [1+, 4-]).

**Как численно оценить нечистоту всех разделений для признака
или как ещё говорят – как оценить нечистоту узла?**



Обучение дерева решений

Оценка нечистоты разделения (или листа)

Индекс Джини

$$G_j = 2\dot{p}(1 - \dot{p})$$

Математическое ожидание ошибки при разделении примеров. Чем чище узел, тем меньше индекс Джини, для чистого узла $G_j = 0$.

Энтропия

$$H_j = -\dot{p} \cdot \log_2(\dot{p}) - (1 - \dot{p}) \cdot \log_2(1 - \dot{p})$$

Ожидаемый объём информации в битах. Чем чище узел, тем меньше информации ожидается об этом узле, для чистого узла $H_j = 0^*$).

Миноритарный класс

$$M_j = \min(\dot{p}, (1 - \dot{p}))$$

Частота ошибок при неправильном разделении примеров. Чем чище узел, тем меньше ошибок будет.

где $\dot{p} = \frac{n^{\oplus}}{n^{\oplus} + n^{\ominus}}$ - эмпирическая вероятность.

$$1 - \dot{p} = \frac{n^{\ominus}}{n^{\oplus} + n^{\ominus}}$$

*) Считается, что $0 \cdot \log_2(0) = 0$ (<https://bigenc.ru/c/entropiia-v-teorii-informatsii-8e42df>)

Обучение дерева решений

Оценка нечистоты узла (совокупности листов одного узла)

$$\text{Imp}(\{D_1, D_2, \dots, D_m\}) = \sum_{j=1}^m \frac{|D_j|}{|D|} \text{Imp}(D_j)$$

где $|D_j|$ - количество примеров в j -ом разделении (листе);

$|D|$ - количество примеров во всём обучающем наборе;

$\text{Imp}(D_j)$ – оценка нечистоты j -го разделения (листа);

Обучение дерева решений

Оценка нечистоты узла

Вычислим чистоту узлов (признаков) для примера, на основе индекса Джини:

Длина = {1, 2, 3} [3+, 1-] [1+, 0] [1+, 4-]



$$[3+, 1-] \Rightarrow n^{\oplus} = 3 \text{ и } n^{\ominus} = 1$$


$$|D_j| = |D_1| = n^{\oplus} + n^{\ominus} = 4$$

$$|D| = 10$$

$$\text{Imp}(D_j) = \text{Imp}(D_1) = G_j = 2\dot{p}(1 - \dot{p}) = 2 \cdot \frac{3}{4} \cdot \frac{1}{4}$$

$$\dot{p} = \frac{n^{\oplus}}{n^{\oplus} + n^{\ominus}} = \frac{3}{3 + 1} = \frac{3}{4}$$

$$1 - \dot{p} = \frac{n^{\ominus}}{n^{\oplus} + n^{\ominus}} = \frac{1}{4}$$


$$\frac{|D_1|}{|D|} \text{Imp}(D_1) = \frac{4}{10} \cdot \left(2 \cdot \frac{3}{4} \cdot \frac{1}{4} \right)$$

Обучение дерева решений

Оценка нечистоты узла

Вычислим чистоту узлов (признаков) для примера, на основе индекса Джини:

Длина = {1, 2, 3} [3+, 1-] [1+, 0] [1+, 4-]

$$[3+, 1-] \rightarrow \frac{|D_1|}{|D|} \text{Imp}(D_1) = \frac{4}{10} \cdot \left(2 \cdot \frac{3}{4} \cdot \frac{1}{4} \right)$$

$$[1+, 0] \rightarrow \frac{|D_2|}{|D|} \text{Imp}(D_2) = \frac{1}{10} \cdot \left(2 \cdot \frac{1}{1} \cdot \frac{0}{1} \right)$$

$$[1+, 4-] \rightarrow \frac{|D_3|}{|D|} \text{Imp}(D_3) = \frac{5}{10} \cdot \left(2 \cdot \frac{1}{6} \cdot \frac{4}{6} \right)$$

$$\text{Imp}(\{D_1, D_2, D_3\}) = \frac{4}{10} \cdot \left(2 \cdot \frac{3}{4} \cdot \frac{1}{4} \right) + \frac{1}{10} \cdot \left(2 \cdot \frac{1}{1} \cdot \frac{0}{1} \right) + \frac{5}{10} \cdot \left(2 \cdot \frac{1}{6} \cdot \frac{4}{6} \right) = 0,310$$

Обучение дерева решений

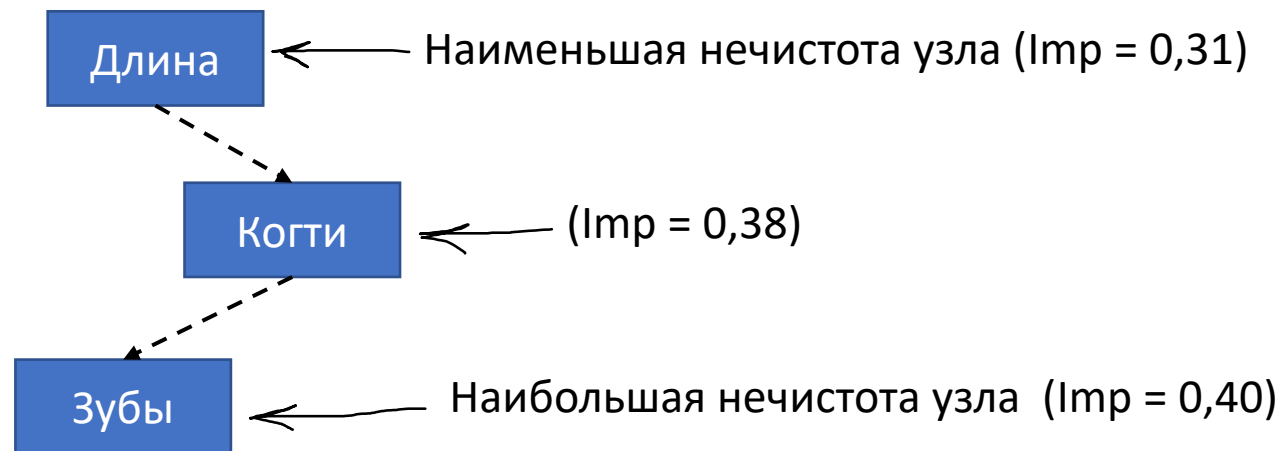
Оценка нечистоты узла

Вычислим чистоту узлов (признаков) для примера, на основе индекса Джини:

Длина = {1, 2, 3} [3+, 1-] [1+, 0] [1+, 4-] $\Rightarrow \text{Imp}(\{D_1, D_2, D_3\}) = \frac{4}{10} \cdot \left(2 \cdot \frac{3}{4} \cdot \frac{1}{4}\right) + \frac{1}{10} \cdot \left(2 \cdot \frac{1}{1} \cdot \frac{0}{1}\right) + \frac{5}{10} \cdot \left(2 \cdot \frac{1}{6} \cdot \frac{4}{6}\right) = 0,31$

Когти = {дл., кор.} [0+, 2-] [3+, 5-] $\Rightarrow \text{Imp}(\{D_1, D_2\}) = \frac{2}{10} \cdot \left(2 \cdot \frac{0}{2} \cdot \frac{2}{2}\right) + \frac{8}{10} \cdot \left(2 \cdot \frac{3}{8} \cdot \frac{5}{8}\right) = 0,38$

Зубы = {мл., мн.} [2+, 3-] [1+, 4-] $\Rightarrow \text{Imp}(\{D_1, D_2\}) = \frac{5}{10} \cdot \left(2 \cdot \frac{2}{5} \cdot \frac{3}{5}\right) + \frac{5}{10} \cdot \left(2 \cdot \frac{1}{5} \cdot \frac{4}{5}\right) = 0,40$



Алгоритм обучения (построения) дерева решений

Сигнатура функции: *GrowTree(D, F)*

Вход: помеченные входные данные D , множество признаков F .

Выход: дерево признаков (узлов) T с помеченными листьями L .

```
 $T \leftarrow \text{linkedlist}(\quad);$  /*связанный список структур*/  
 $L \leftarrow \text{Label}(D);$  /*возвращаем класс объектов для множества*/  
if  $L = \text{None}$  then /*класс для множества не определён*/  
     $S \leftarrow \text{BestSplit}(D, F);$  /*возвращает наиболее чистый признак для разделения множества  $D$ */  
    if  $S = \text{None}$  then return; /*все признаки уже рассмотрены => ошибка и выход */  
  
     $T \leftarrow \text{linkedlist}(S);$  /*добавили узел в дерево к родительскому узлу*/  
  
    разделяем множество  $D$  на подмножества  $D_i$  в соответствии с литералами в  $S$ ; /*требуется реализовать*/  
  
    for  $D_i$  in  $D$  do  
         $\text{GrowTree}(D_i, F);$  /*рекурсивный вызов функции*/  
    end  
  
    else  
         $T \leftarrow \text{linkedlist}(L);$  /*добавили лист к родительскому узлу*/  
    end  
  
return  $T$ ; /*дерево построено*/
```

Алгоритм выбора наилучшего признака

Сигнатура функции: *BestSplit(D, F)*

Вход: помеченные входные данные D , множество признаков F .

Выход: признак f , по которому выполнять разделение.

$I_{min} \leftarrow 1;$

$f_{best} \leftarrow \text{None};$

for f **in** F **do**

разделить D на подмножества D_1, \dots, D_m согласно значениям v_j признака f ;

*/*т. е. необходимо для каждого значения признака найти разделения*

$f = [v_1, v_2, v_3] \Rightarrow [n_1+, n_1-] [n_2+, n_2-] [n_3+, n_3-]$

(Длина = {1, 2, 3} \Rightarrow [3+, 1-] [1+, 0] [1+, 4-])

**/*

if $\text{Imp}(\{D_1, D_2, \dots, D_m\}) < I_{min}$ **then**

$I_{min} \leftarrow \text{Imp}(\{D_1, D_2, \dots, D_m\});$

$f_{best} \leftarrow f;$

end

end

return $f_{best};$

*/**

т.к. необходимо сохранять признаки по которым ранее было выполнено разделение, то могут быть значения типа

$n(+) = 0, n(-) = 0 \Rightarrow n(+) + n(-) = 0 \Rightarrow$ деление на нуль

в таких случаях требуется пропуск итерации

**/*

Алгоритм определения класса множества

Сигнатура функции: *Label(D)*

Вход: помеченные входные данные D .

Выход: возвращает один из классов или *None*.

```
for  $D_i$  in  $D$  do
    /*проверяем класс каждого  $D_i$  объекта с классов первого объекта  $D_1$ */
    if  $\hat{c}(D_i) \neq \hat{c}(D_1)$  then
        return None; /*есть хотя бы одни  $D_i$  с классом отличным от других объектов из  $D$ */
    end
end
return  $\hat{c}(D_1)$ ; /*класс для объектов всего множества определён*/
```

Приведенный алгоритм достаточно «жёстко» определяет класс множества – все объекты множества должны иметь один и тоже класс.

В общем случае, алгоритм может возвращать мажоритарный класс объектов, то есть класс тех объектов, которых больше всего во множестве. Если мажоритарных классов более одного, то возвращается любой из них (случайный выбор из мажоритарных классов).

Что ещё следует знать о решающих деревьях

В существует ряд широко распространённых алгоритмов построения деревьев решений, эти алгоритмы обычно используются в различных программных библиотеках машинного обучения: **CART, ID3, C4.5**.

Выше приведенный алгоритм построения решающего дерева приведет к тому, что в конечном итоге будет построено дерево со всеми чистыми листьями. В это случае говорят, что дерево достигает нулевой ошибки на обучающем наборе данных.

Нулевая ошибка на обучающем наборе данных



Глубокое дерево



**Машинная модель
переобучена**

Чтобы предотвратить переобучение решающего дерева используют два подхода:

- ранняя остановка построения дерева (дереву не дают достроиться до конца), называется предварительной обрезкой (pre-pruning);
- дереву дают достичь нулевой ошибки, а потом выполняют удаление малоинформативных узлов и объединение получившихся поддеревьев, такой метод называется пост-обрезкой (post-pruning).

Много преимуществ и два недостатка решающих деревьев

Преимущества:

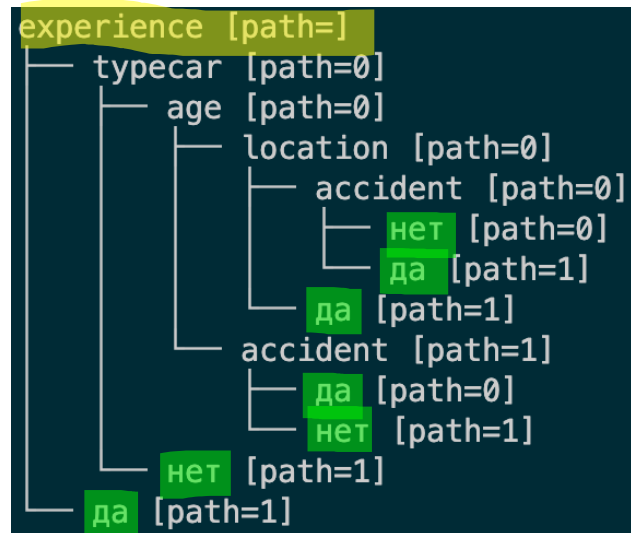
- легко поддаются интерпретации;
- позволяют обрабатывать смесь дискретных и непрерывных признаков;
- нет необходимости в стандартизации данных;
- автоматически производят обор признаков;
- относительно робастны к выбросам;
- быстро обучаются и масштабируются на большие объёмы данных.

Недостатки:

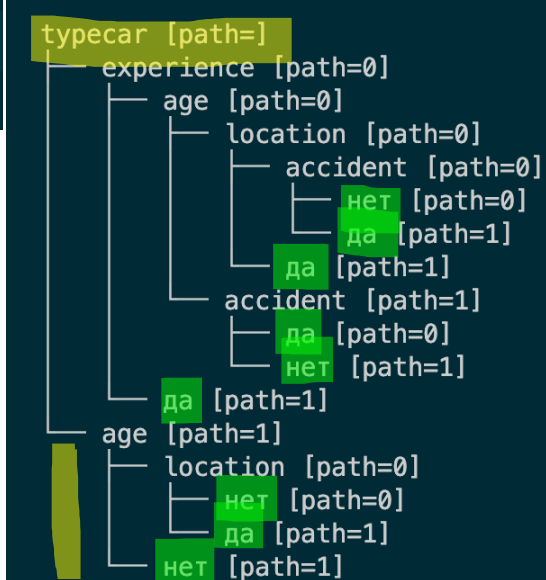
- прогнозы деревьев не очень точны по сравнению с другими видами моделей машинного обучения;
- неустойчивость деревьев: небольшие изменения входных данных оказывают существенное влияние на структуру всего дерева, что связано с его иерархичной сущностью, то есть ошибки верхних уровней распространяются и на все нижние уровни.

Пример неустойчивости решающих деревьев

| | | | | | | |
|----|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 |
| 7 | 1 | 0 | 1 | 1 | 0 | 1 |
| 8 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 1 | 1 | 0 | 1 | 1 | 1 |
| 10 | 0 | 0 | 1 | 0 | 0 | 1 |



| | | | | | | |
|----|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 |
| 7 | 1 | 0 | 1 | 1 | 0 | 1 |
| 8 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 1 | 1 | 0 | 1 | 1 | 0 |
| 10 | 0 | 0 | 1 | 0 | 0 | 1 |



Замечание.

Для повышения устойчивости деревьев и их точности применяют ансамбли деревьев решений – строят несколько (много) деревьев, которые хотя бы немного отличаются друг от друга, получают результаты от каждого дерева и затем все результаты усредняют (ансамбль «Случайный лес»).

Использованные информационные источники

1. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.: ил.
2. Мэрфи К. П. Вероятностное машинное обучение: введение / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2022. – 990 с.: ил.
3. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными.: Пер. с англ. - СПб.: ООО "Альфа-книга", 2017. - 480 с.: ил.