Разбор летучки

Лекция 4

Деревья принятия решений

Екатерина Тузова

Мотивирующий пример

Мотивирующий пример

Pokémon for Data Mining and Machine Learning (https://www.kaggle.com/alopez247/pokemon)

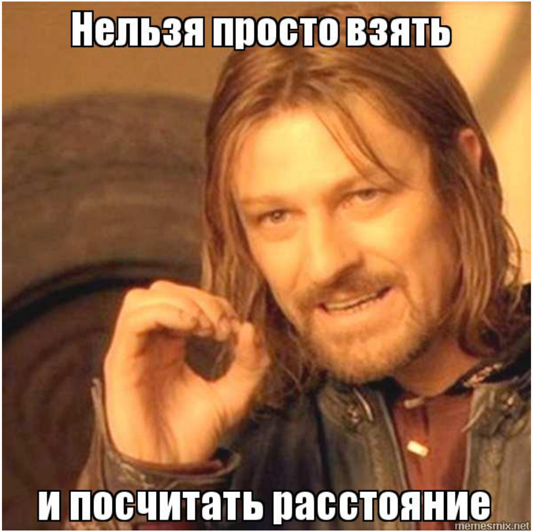
Датасет

In [10]:
```
pokemons.head()
```

Out[10]:

	Type_1	Type_2	isLegendary	Color	hasGender	Egg_Group_1	Egg_Group_2	hasMegaEvolution	Body_Style
0	Grass	Poison	False	Green	True	Monster	Grass	False	quadruped
1	Grass	Poison	False	Green	True	Monster	Grass	False	quadruped
2	Grass	Poison	False	Green	True	Monster	Grass	True	quadruped
3	Fire	NaN	False	Red	True	Monster	Dragon	False	bipedal_tailed
4	Fire	NaN	False	Red	True	Monster	Dragon	False	bipedal_tailed

Мотивация



Логические закономерности

$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \to \{0, 1\}$, который удовлетворяет двум требованиям:

Логические закономерности

$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \to \{0, 1\}$, который удовлетворяет двум требованиям:

1. Интерпретируемость

Логические закономерности

$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \to \{0, 1\}$, который удовлетворяет двум требованиям:

1. Интерпретируемость
2. Информативность относительно одного из классов $c \in Y$

Логические закономерности

$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка.

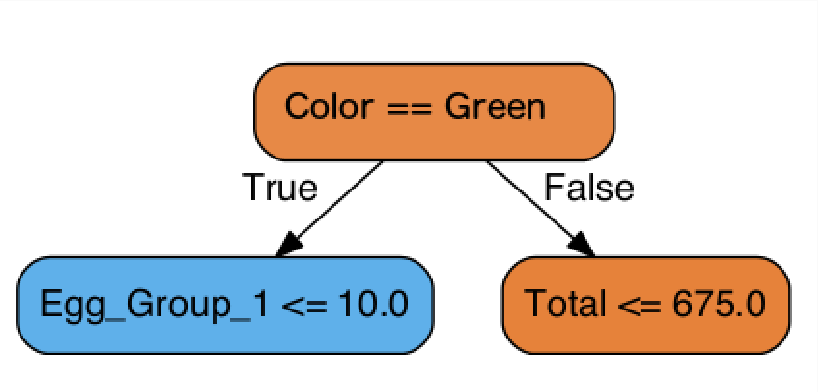
Предикат $\beta : X \to \{0, 1\}$

Задача: Найти множество логических закономерностей $\mathscr{B}$ по X^l. Построить алгоритм $a(X, \mathscr{B}) \to y$, способный классифицировать произвольный объект $x \in X$.

Интерпретируемость

1. Записывается на естественном языке
2. Зависит от небольшого числа признаков

Интерпретируемость



Информативность

Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

Информативность

Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

$$tp(\beta) = \#\left\{x_i : \beta(x_i) = 1, y_i = c\right\} \to \max$$

Информативность

Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

$$tp(\beta) = \#\{x_i : \beta(x_i) = 1, y_i = c\} \to \max$$

$$fp(\beta) = \#\{x_i : \beta(x_i) = 1, y_i \neq c\} \to \min$$

Основные вопросы

1. Какого вида закономерности $\beta(x)$ нужны?
2. Как определять информативность?
3. Как выбирать закономерности?
4. Как объединять закономерности в алгоритм?

Виды правил

Виды правил

- Пороговое условие
  $\beta(x) = \left[x^j \leq a_j\right]$ или $\left[a_j \leq x^j \leq b_j\right]$

Виды правил

- Пороговое условие
  $\beta(x) = \left[x^j \leq a_j\right]$ или $\left[a_j \leq x^j \leq b_j\right]$
- Конъюнкция из J пороговых условий
  $\beta(x) = \bigwedge\limits_{j \in J} \left[a_j \leq x^j \leq b_j\right]$

Виды правил

- Пороговое условие
  $\beta(x) = \left[x^j \le a_j\right]$ или $\left[a_j \le x^j \le b_j\right]$
- Конъюнкция из J пороговых условий
  $\beta(x) = \bigwedge\limits_{j \in J} \left[a_j \le x^j \le b_j\right]$
- Синдром – выполнение не менее d условий из J
  $$\beta(x) = \left[\sum_{j \in J} \left[a_j \le x^j \le b_j\right] \ge d\right]$$

Как собрать классификатор из закономерностей?

Решающий список

Идея:

Возьмем $\beta_1(x), \beta_2(x), \ldots, \beta_T(x)$ закономерностей и будем по порядку применять на объекте. Как только предикат β_i сработал – вернем соответствующий класс c_i.

Решающий список

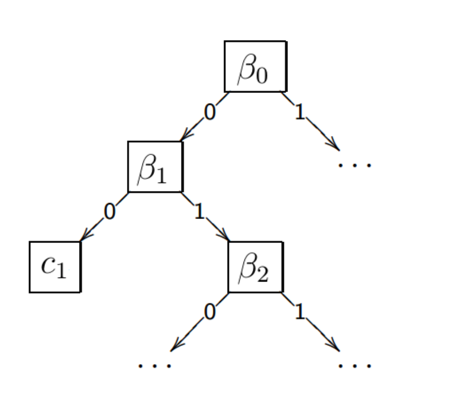
Идея:
Возьмем $\beta_1(x), \beta_2(x), \dots, \beta_T(x)$ закономерностей и будем по порядку применять на объекте. Как только предикат β_i сработал – вернем соответствующий класс c_i.

Каждое правило принимает окончательное решение $\Rightarrow$ ошибка правила равна ошибке всего алгоритма

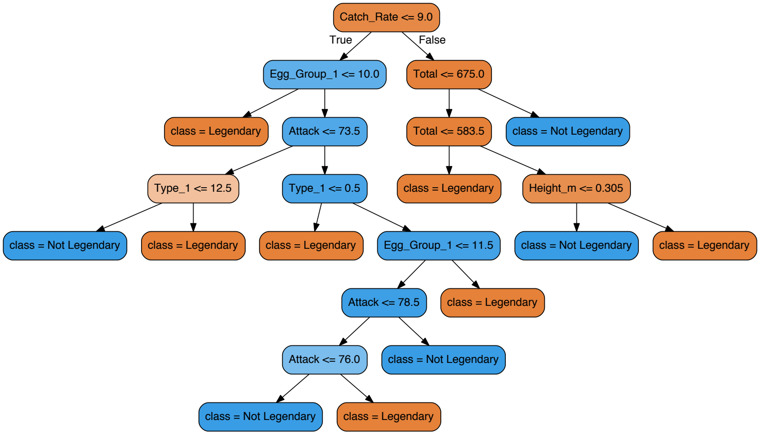
Бинарное решающее дерево

Бинарное решающее дерево – алгоритм классификации $a(x, \beta)$, задающийся бинарным деревом:

- $\forall v \in V_{inner} \to \beta_v : X \to \{0, 1\}$, $\beta \in \mathscr{B}$
- $\forall v \in V_{leaf} \to$ имя класса $c_v \in Y$



Пример решающего дерева



Алгоритм построения ID3

function $\textsc{LearnID3}(U, \mathscr{B})$

  if все объекты из U лежат в одном классе $c \in Y$ **then**

    return новый лист v, $c_v = c$

  $\beta^* = \max\limits_{\beta \in \mathscr{B}} I(\beta, U)$

  $U_{left} = \{x \in U : \beta^*(x) = 0\}$

  $U_{right} = \{x \in U : \beta^*(x) = 1\}$

  if $U_{left} = \varnothing$ или $U_{right} = \varnothing$ **then**

    return v, c_v = Majority(U)

  Создать новую внутреннюю вершину v: $\beta_v = \beta^*$

  L_v = LearnID3 (U_{left}, $\mathscr{B}$)

  R_v = LearnID3 (U_{right}, $\mathscr{B}$)

  return v

Критерии информативности

Оценивание информативности

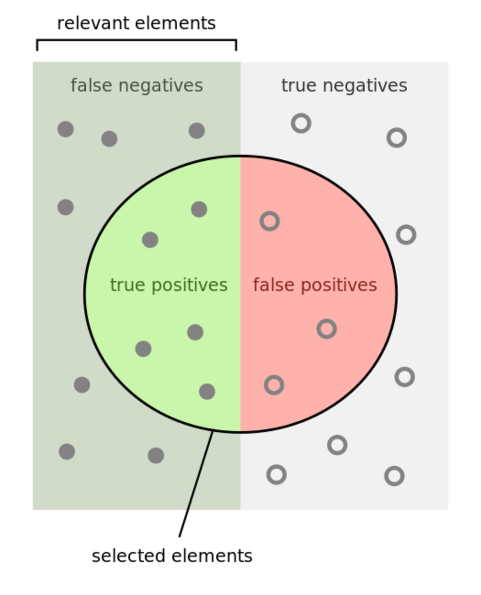
$$tp(\beta) = \#\{x_i : \beta(x_i) = 1, y_i = c\} \to \max$$
$$tn(\beta) = \#\{x_i : \beta(x_i) = 0, y_i \neq c\} \to \max$$
$$fp(\beta) = \#\{x_i : \beta(x_i) = 1, y_i \neq c\} \to \min$$
$$fn(\beta) = \#\{x_i : \beta(x_i) = 0, y_i = c\} \to \min$$

Оценивание информативности

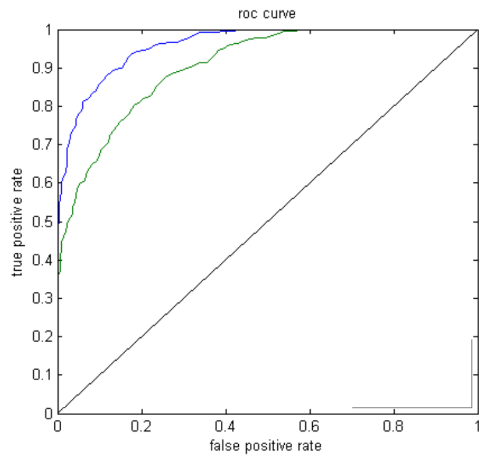
$$tp(\beta) = \#\left\{x_i : \beta(x_i) = 1, y_i = c\right\} \to \max$$
$$tn(\beta) = \#\left\{x_i : \beta(x_i) = 0, y_i \neq c\right\} \to \max$$
$$fp(\beta) = \#\left\{x_i : \beta(x_i) = 1, y_i \neq c\right\} \to \min$$
$$fn(\beta) = \#\left\{x_i : \beta(x_i) = 0, y_i = c\right\} \to \min$$

$Precision = \frac{tp}{tp+fp}$
$Recall = TPR = \frac{tp}{tp+fn}$
$FPR = \frac{fp}{fp+tn}$

Precision-Recall



ROC-кривая



AUC – Area under the ROC curve

Пример свертки двух критериев

Пусть число примеров искомого класса 200 и число остальных объектов 100

tp	fp	$tp - fp$	$tp - 5fp$	$Precision$
50	0	50	50	1
100	50	50	-150	0.6
50	9	41	5	0.84
5	0	5	5	1

Критерий Джини

$$I(\beta, X^l) = \#\left\{(x_i, x_j) : \beta(x_i) = \beta(x_j), y_i = y_j\right\}$$

Энтропия Шеннона

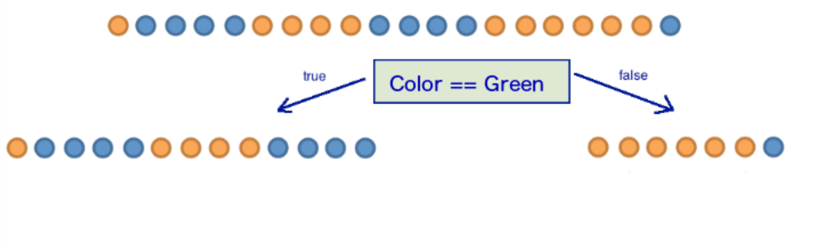
$$H(U) = -\sum_{i=1}^{C} p_i \log_2 p_i$$

p_i – процентное соотношение объектов класса i в выборке U

Энтропия Шеннона

$$H(U) = -\sum_{i=1}^{C} p_i \log_2 p_i$$

p_i – процентное соотношение объектов класса i в выборке U



Критерий информационного выигрыша

Прирост информации – уменьшение энтропии.

$$H = -\sum_{i=1}^{C} p_i \log_2 p_i$$

$$IGain(U, x^j) = H(U) - \sum_{v} \frac{|U_v|}{|U|} H(U_v)$$

$$v \in values(x^j) \qquad U_v = \{x \in U | x^j = v\}$$

Плюсы

\+ Интерпретируемость и простота классификации

Плюсы

+ Интерпретируемость и простота классификации
+ Допустимы разнотипные данные и данные с пропусками

Плюсы

+ Интерпретируемость и простота классификации
+ Допустимы разнотипные данные и данные с пропусками
+ Не бывает отказов от классификации

Плюсы

+ Интерпретируемость и простота классификации
+ Допустимы разнотипные данные и данные с пропусками
+ Не бывает отказов от классификации
+ Трудоёмкость линейна по длине выборки

Минусы

– Жадный ID3 сильно переобучается

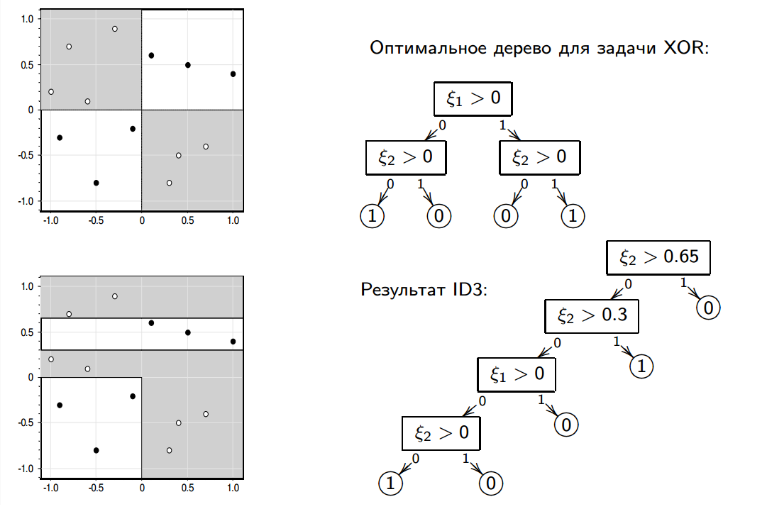
Минусы

– Жадный ID3 сильно переобучается
– Высокая чувствительность к шуму, к составу выборки, к критерию информативности

Минусы

- Жадный ID3 сильно переобучается
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности
- Чем дальше v от корня, тем меньше надёжность выбора β_v , c_v

Переобучение



Подрезание дерева C4.5

X^k – независимая контрольная выборка, $k \approx 0.5l$

Для всех $v \in V_{inner}$:

  $U_v =$ подмножество объектов X^k, дошедших до v

  Если $U_v = \emptyset$:

    Вернуть новый лист v, $c_v =$ Majority(U)

  Вычислить число ошибок четырьмя способами:

    $r(v)$ – поддеревом, растущим из вершины v

    $r_L(v)$ – левой дочерней вершины L_v

    $r_R(v)$ – правой дочерней вершины R_v

    $r_c(v)$ – к классу $c \in Y$

  В зависимости от того, какое из них минимально:

    Сохранить поддерево v

    Заменить поддерево v поддеревом L_v

    Заменить поддерево v поддеревом R_v

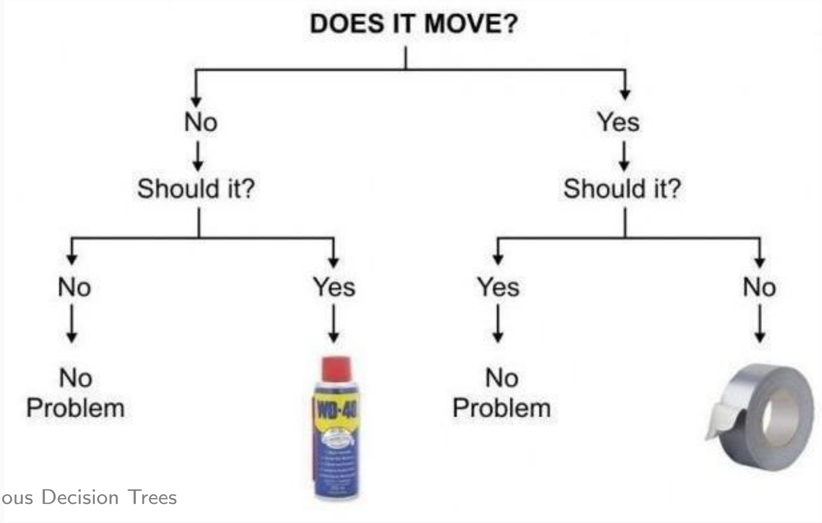
    Заменить v листом $c_v = \min\limits_{c \in Y} r_c(v)$

Небрежные решающие деревья

Идея: Сделаем дерево сбалансированным. Для этого нужно для всех узлов одного уровня использовать одинаковое условие ветвления.

Небрежные решающие деревья

Идея: Сделаем дерево сбалансированным. Для этого нужно для всех узлов одного уровня использовать одинаковое условие ветвления.



Композиция деревьев

Идея: Можно использовать результаты нескольких алгоритмов, а не одного.

Random forest

Голосование деревьев классификации, $Y = \{-1, +1\}$
$a(t) = Majority(b_t(x))$

- Каждое дерево $b_t(x)$ обучается по случайной выборке с повторениями
- В каждой вершине предикат выбирается из случайного подмножества n предикатов

Вопросы?

Что почитать по этой лекции

- G. James, D. Witten, T. Hastie, R. Tibshirani "An Introduction to Statistical Learning" Chapter 8
- Воронцов "Логические алгоритмы классификации"

На следующей лекции

- Байесовские методы классификации
- Вероятностная постановка задачи
- Оптимальный Байесов классификатор
- Наивность
- Максимальное правдоподобие
- Разные распределения