

Классификация: начало

Маша Шеянова, masha.shejanova@gmail.com

Популярные алгоритмы классификации

- k ближайших соседей (kNN)
- **наивный Байес**
- деревья решений
- логистическая регрессия
- метод опорных векторов (SVM)

Метрики качества

Any ideas?

Accuracy

Метрика accuracy — самая простая оценка классификации: поделить все правильные ответы классификатора на количество всех ответов.

Достоинства: простота.

Недостатки: плохо работает, когда данные сильно перекошены.

(Например, если мы ищем у человека редкую болезнь, модель, которая про всех будет говорить “здоров”, будет права в 99% случаев)ю

Confusion matrix

А вот более подробная информация про то, какие ошибки делает модель.

TP: true positive (верно сказали да)

FP: false positive (сказали да, а надо было нет)

TN: true negative (верно сказали нет)

FN: false negative (сказали нет, а надо было да)


The diagram illustrates a confusion matrix for a classification model. It features a central 2x2 grid with 'Actual' on the top axis and 'Predicted' on the left axis. The grid cells contain the counts for True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Marginal totals are provided for both axes, and the overall sample size (Total n=165) is noted in the top left. The counts are: TP=100, FP=10, FN=5, and TN=50. The marginal totals are 110 for 'Actual Yes' and 55 for 'Actual No' on the right, and 105 for 'Predicted Yes' and 60 for 'Predicted No' on the bottom.

		Actual		
		Yes	No	
Predicted	Yes	TP = 100	FP = 10	110
	No	FN = 5	TN = 50	55
		105	60	


Total n=165

Точность и полнота

How many selected items are relevant?

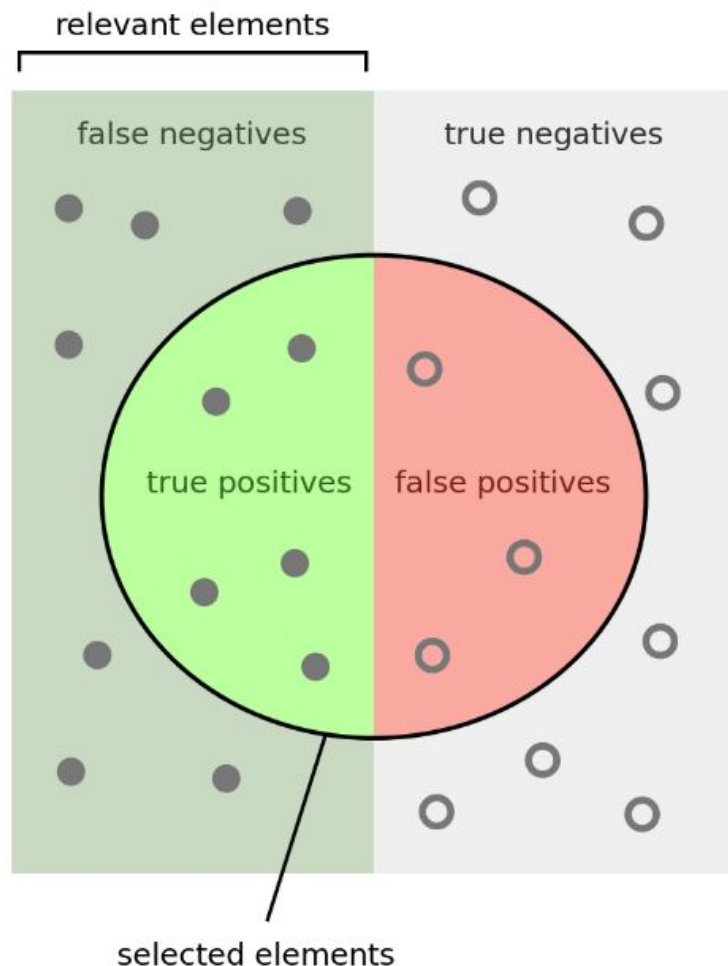
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$


How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$


Precision (точность): не отхватить лишнего

Recall (полнота): ничего не упустить



Точность и полнота

Когда важна точность (не отхватить случайно ничего лишнего):

- спам-фильтр (человек не хочет потерять важные письма)

Когда важна полнота (ничего не забыть):

- диагностика болезней
- поиск террористов
- ???

f1-мера

Но что, если важно и то, и то?

Можно было бы просто посчитать среднее между точностью и полнотой. Но тогда очень плохие результаты будут get away.

f1 — среднее гармоническое точности и полноты

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Наивный Байес

Города и сёла

Возьмём 1000 случайных человек и спросим их, где они живут: в городе или в селе.

В городе живёт больше людей, поэтому в нашей выборке таких оказалось 900, а из села — всего 100.



город: 900 человек

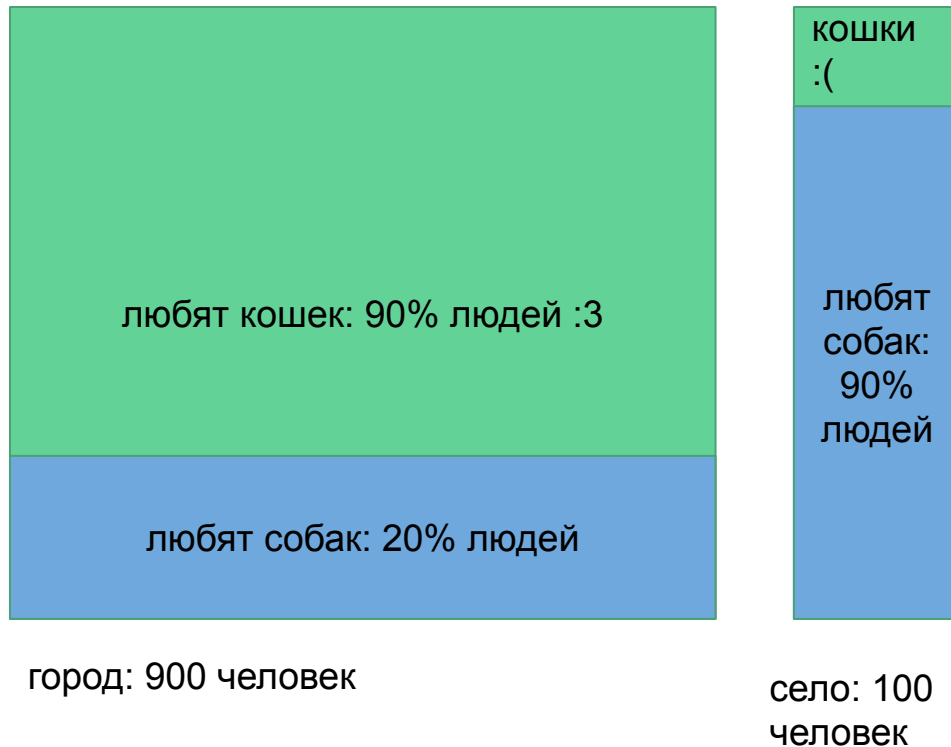
село:
100
человек

Кошки и собаки

А теперь спросим их, кого они бы предпочли в качестве домашнего питомца: кошек или собак.

Оказалось, что в сёлах собак любят гораздо больше, а в городе предпочитают кошек.

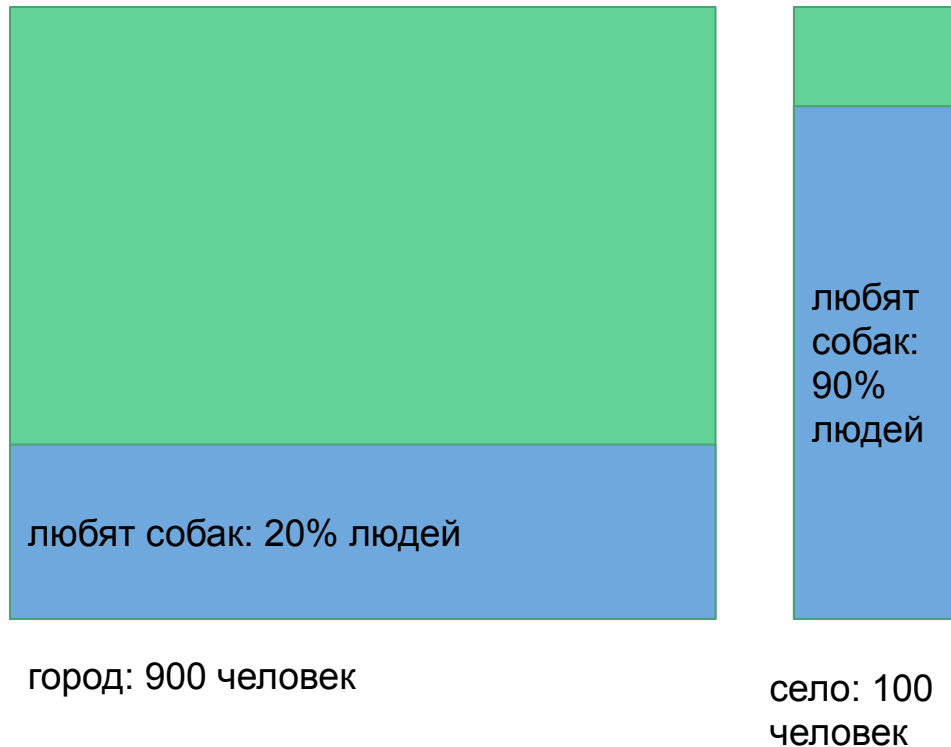
Пусть событие А = человек из села. Событие В = человек любит собак.



Вероятность события

Мы встретили случайного человека из нашей тысячи.

С какой вероятностью он любит собак?



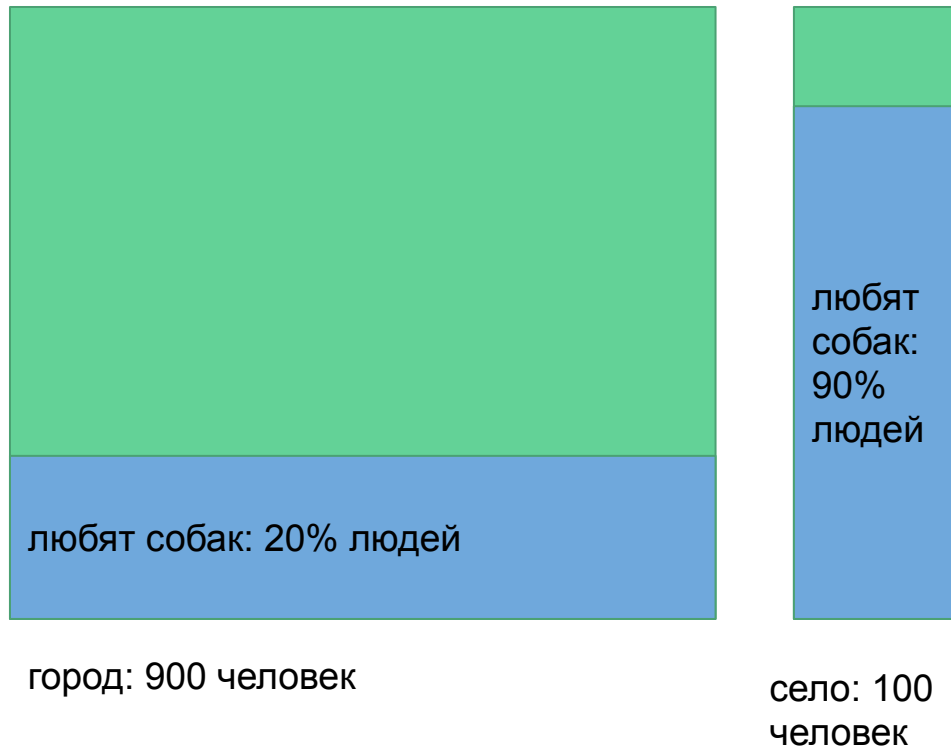
Вероятность события В

Мы встретили случайного человека из нашей тысячи. С какой вероятностью он любит собак?

$$P(\text{городской и собачник}) = 900/1000 * 0.1 = 0.9 * 0.2 = 0.18$$

$$P(\text{сельский и собачник}) = 100/1000 * 0.1 = 0.1 * 0.9 = 0.09$$

$$P(\text{собачник}) = 0.18 + 0.09 = 0.27$$



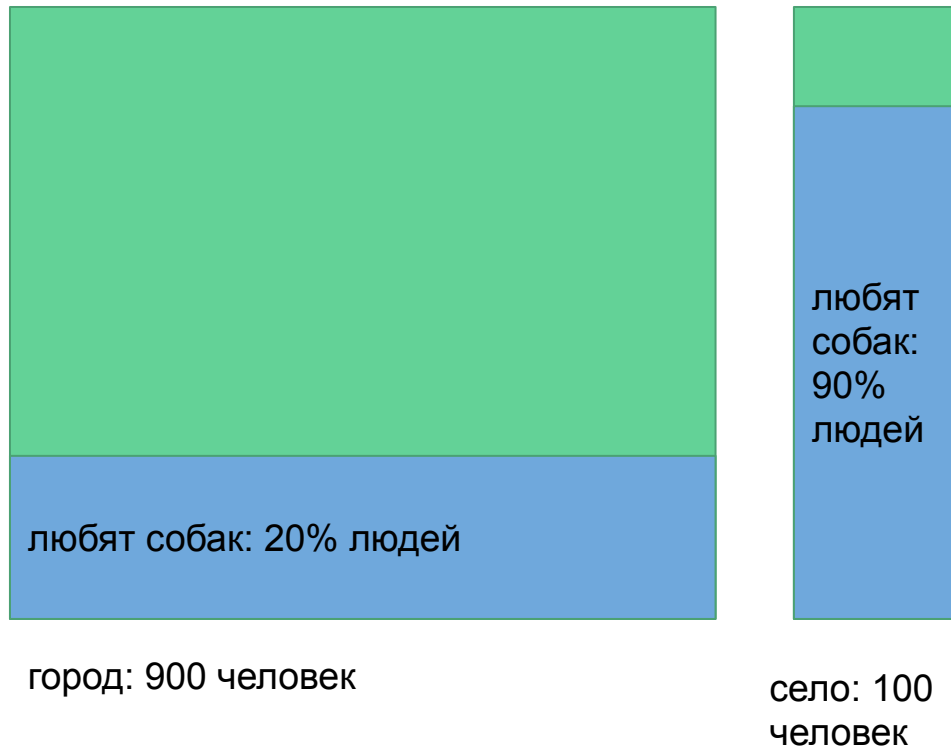
Вероятность события В

Мы встретили случайного человека из нашей тысячи. С какой вероятностью он любит собак?

$$P(\text{городской и собачник}) = 900/1000 * 0.1 = 0.9 * 0.2 = 0.18$$

$$P(\text{сельский и собачник}) = 100/1000 * 0.1 = 0.1 * 0.9 = 0.09$$

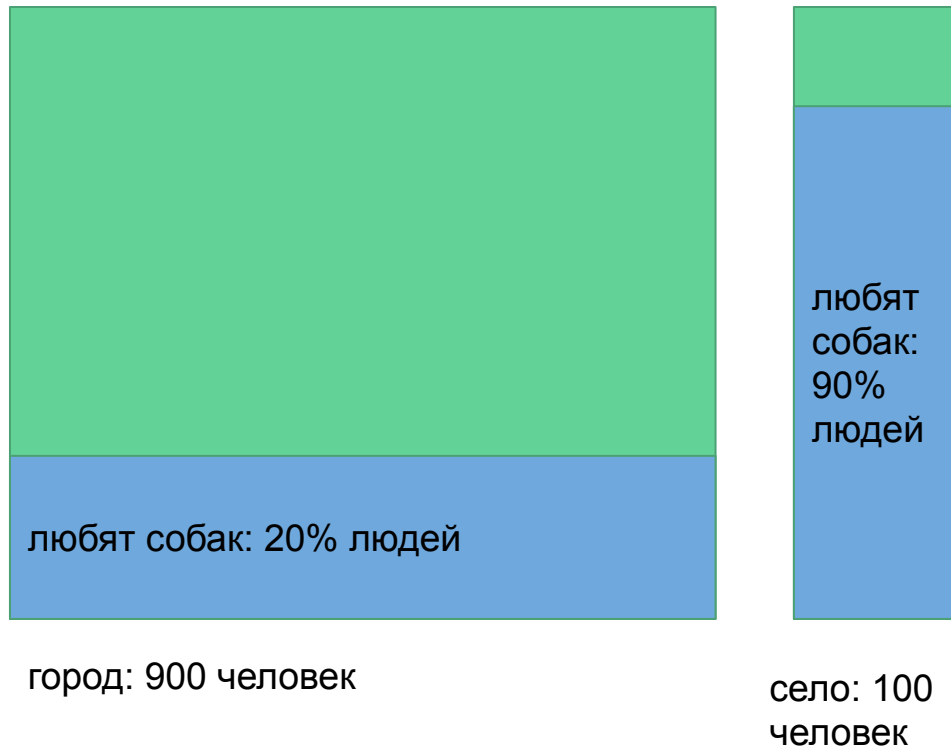
$$P(\text{собачник}) = 0.18 + 0.09 = 0.27$$



Вероятность события А

Мы встретили случайного человека из нашей тысячи, и оказалось, что он любит собак.

С какой вероятностью он живёт в селе?



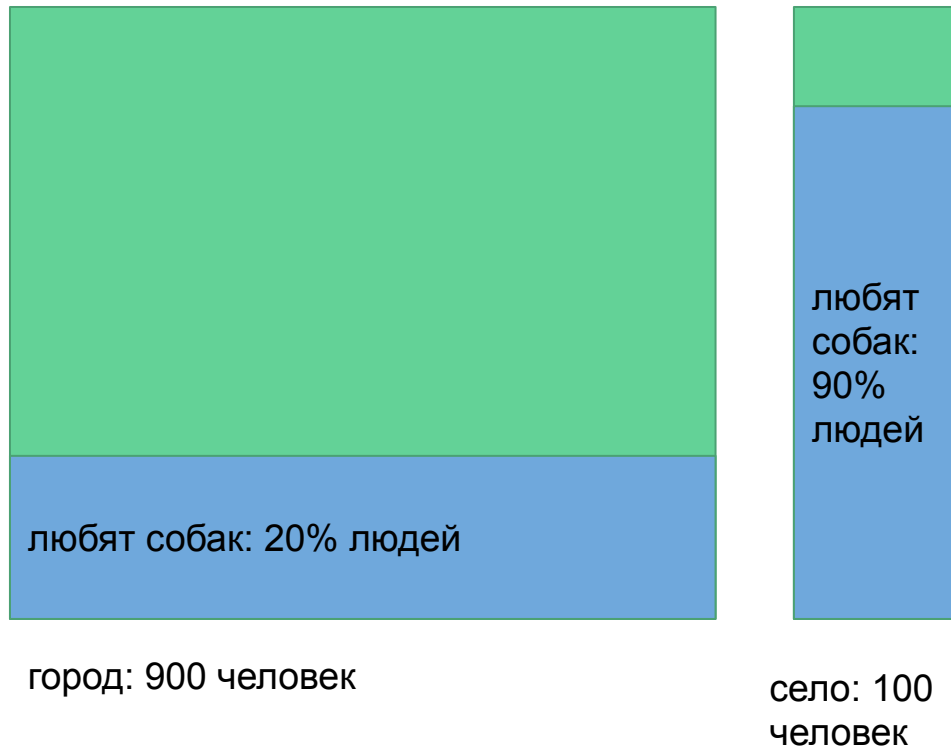
Вероятность события А

Случайный человек из нашей тысячи любит собак. С какой вероятностью он из села?

$$P(\text{сельский и собачник}) = 100/1000 * 0.1 = 0.1 * 0.9 = 0.09$$

$$P(\text{собачник}) = 0.18 + 0.09 = 0.27$$

$$P(\text{сельский} \mid \text{собачник}) = P(\text{сельский и собачник}) / P(\text{собачник}) = 0.09/0.27 = 1/3$$



Формула Байеса

вероятность того, что событие B
истинно, если событие A истинно



вероятность того, что
событие A истинно



$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

вероятность того, что
событие A истинно, если
событие B истинно



вероятность того, что
событие B истинно

([Источник картинки](#))

если $P(B)$ не дано
изначально, его можно
расписать как

$$P(B|A)*P(A) + P(B|\text{не } A)*P(\text{не } A)$$

Определение спама

На почту студента Вани в 7 % случаев приходит спам. В обычных входящих письмах для Вани слово *вклады* встречается в 5% писем. В спаме, который приходит к Ване, это слово встречается в 60% случаев.

К Ване пришло письмо, в котором есть слово *вклады*. С какой вероятностью это нормальное письмо (не спам)?

Определение спама

На почту студента Вани в 7 % случаев приходит спам. В обычных входящих письмах для Вани слово *вклады* встречается в 5% писем. В спаме, который приходит к Ване, это слово встречается в 60% случаев.

К Ване пришло письмо, в котором есть слово *вклады*. С какой вероятностью это нормальное письмо (не спам)?

$$P(\text{норм} \mid \text{вклады}) = P(\text{вклады} \mid \text{норм}) * P(\text{норм}) / P(\text{вклады})$$

Определение спама

На почту студента Вани в 7 % случаев приходит спам. В обычных входящих письмах для Вани слово *вклады* встречается в 5% писем. В спаме, который приходит к Ване, это слово встречается в 60% случаев.

К Ване пришло письмо, в котором есть слово *вклады*. С какой вероятностью это нормальное письмо (не спам)?

$$P(\text{вклады}) = P(\text{вклады} \mid \text{норм}) * P(\text{норм}) + P(\text{вклады} \mid \text{спам}) * P(\text{спам}) = 0.05 * 0.93 + 0.6 * 0.07 = 0.0465 + 0.042 = 0.0885$$

$$P(\text{норм} \mid \text{вклады}) = P(\text{вклады} \mid \text{норм}) * P(\text{норм}) / P(\text{вклады}) = 0.0465 / 0.0885 = 0.525423729$$

Наивный Байесовский классификатор

вероятность того, что событие B истинно, если событие A истинно

↓

вероятность того, что событие A истинно

↘

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

↑

вероятность того, что событие A истинно, если событие B истинно

↙

вероятность того, что событие B истинно

Итак, формула Байеса — это про то, **насколько вероятно наше предположение о мире при условии того, что мы видим.**

На этой идее и держится наивный Байесовский классификатор: насколько вероятно, что событие A (класс) произошло при условии события B (признака).

Наивный Байес \neq байесовские методы!

Наивный Байес — самый простой классификатор в машинном обучении.

Байесовские методы — нестандартный подход к нейросетям, по которому защищают диссертации и ведут целые курсы!

Векторизаторы

One-hot encoding

Упорядочиваем значения категориальной переменной... и превращаем её в столько бинарных переменных, сколько у неё было значений.

значение		is_french	is_english	is_italian	is_russian
french		1	0	0	0
english	→	0	1	0	0
italian		0	0	1	0
russian		0	0	0	1

Теперь можно подбирать коэффициенты!

Мешок слов

А теперь сделаем такой вектор для каждого слова в тексте — и сложим.

пара $\rightarrow (1, 0)$; мой $\rightarrow (0, 1)$

“пара моя, пара” $\rightarrow (2, 1)$

Такой подход называется моделью “мешок слов”. Почему? Нам не важен порядок!

А ещё вместо количества вхождений слова можно использовать TF-IDF. Сейчас объясню, что это.

Для чего нужен TF-IDF?

Это способ понять, **насколько слово специфично для документа**. Или, насколько большую роль слово играет в характеристике документа.

Допустим, у нас есть текст научной статьи. Слово *описание* встретилось в ней 7 раз. А слово *кварк* — всего 3 раза.

Тем не менее, первое слово мало что сказало нам про текст, а после второго стало понятно, о чём он.

Почему так? Слово *описание* можно ожидать в какой угодно статье, а вот *кварк* — только в ограниченном наборе статей.

Как посчитать TF-IDF

TF-IDF — term frequency * inverse document frequency

term frequency (TF) — сколько раз слово встретилось в документе

document frequency (DF) — в скольких документах встретилось слово

inverse document frequency (IDF) — $1 / DF$

$$TF-IDF = TF * \log(IDF)$$

Суть в том, что каждое слово мы награждаем за количество вхождений в наш документ и штрафуем за количество документов, в которых оно встретилось.

Ресурсы

Посмотреть / почитать

Почитать, русский:

- [про precision, recall и f-меру](#)
- [понятно про формулу Байеса](#)