# 1 Наивный Байесовский классификатор текстов

## 1.1 Примеры задач

1. **Новости**

   Есть тексты старых новостей уже размеченные (спорт, развлечения, культура, политика).

   Нужно автоматически выдать категорию очередной новости

2. **Оценки текстов**

   Пользователь Вася уже прочёл K текстов и каждому из них выставил оценку 1 или 0 (нравится или не нравится).

   Дан новый текст, определите вероятность того, что он понравится Василию

3. **Антиспам**

   Есть тексты, которые помечены пользователями как спам. И все остальные.

   Приходит очередное сообщение, нужно понять, спам ли оно?

## 1.2 Введение

- **Конкретная задача**

  Классифицировать тексты двух авторов.

- **Общая идея классификации**

  Мы видим, что определённые слова встречаются чаще у одного автора.

- **Вероятностная идея классификации**

  Мы можем насчитать частоты – где какое слово сколько раз встречается. Через них для каждого слова посчитаем вероятность, что текст, содержащий это слово, принадлежит первому (второму) автору. Из этих вероятностей построим ответ.

## 1.3 Математическая модель

- $C$ – класс; $D$ – данные (текст); $d_1, d_2, \ldots, d_n$ – слова текста
- Генерируется пара $(D, C)$ – текст и класс, к которому он относится.

  $p(C|D)$ – то, что мы хотим, вероятность того, что именно классу $C$ соответствует текст $D$

  $p(D)$ – вероятность того, что нам дадут именно текст $D$

  $p(D|C)$ – вероятность того, что если дают случайный текст класса $C$, то это именно $D$

  $p(C|D)p(D) = p(C, D) = p(D|C)p(C)$

  $p(C|D) = \frac{p(C)p(D|C)}{p(D)}$
- Наивное предположение:

  $p(D|C) = p(d_1|C)p(d_2|C)\ldots p(d_n|C)$ (независимость слов)

- Пусть в классе $C$ всего $count[C]$ текстов.

  Пусть слово $d$ встречается всего в $count[d, C]$ текстах класса $C$.

  Пусть всего у нас два класса – $C_1$, $C_2$

- Тогда $p[C_i] = \frac{count[C_i]}{count[C_1]+count[C_2]}$

  $p[d, C_i] = \frac{count[d,C_i]}{count[C_i]}$

- $p_i = p(C_i|D) = \frac{1}{p(D)} \frac{count[C_i]}{count[C_1]+count[C_2]} \prod_j \frac{count[d_j,C_i]}{count[C_i]}$

  Ответ – вероятность того, что текст принадлежит именно классу $C_1$, не $C_2$.

  $ans = \frac{p_1}{p_1+p_2} = \frac{count[C_1] \cdot \prod_j \frac{count[d_j,C_1]}{count[C_1]}}{count[C_2] \cdot \prod_j \frac{count[d_j,C_2]}{count[C_2]} + count[C_1] \cdot \prod_j \frac{count[d_j,C_1]}{count[C_1]}} = \frac{q_1}{q_1+q_2}$

  Где $q_i = count[C_i] \cdot \prod_j \frac{count[d_j,C_i]}{count[C_i]}$

## 1.4 Реализация Байесовского классификатора текстов

- На практике нам встретятся слова частоты 0. Заменим её на 0.5. `;-)`
- Слова паразиты. Нужно избавляться от слова, которые портят нам статистику. Например, слишком короткие слова: “и”, “я”. Слишком часто используемые слова: “меня”, “тебя”.
- Чтобы все тексты обрабатывались единообразно, полезно реализовать функцию `getWords(text) : string -> string[]` Например, эта функция должна пропускать все знаки препинания и приводить буквы к нижнему регистру.
- Чтобы код был короче, полезно реализовать `class Bayes`.
- Задание состоит из двух частей

  1. Обучиться на всех текстах, классифицировать все тексты
  2. Обучиться на случайной половине текстов, классифицировать оставшуюся половину