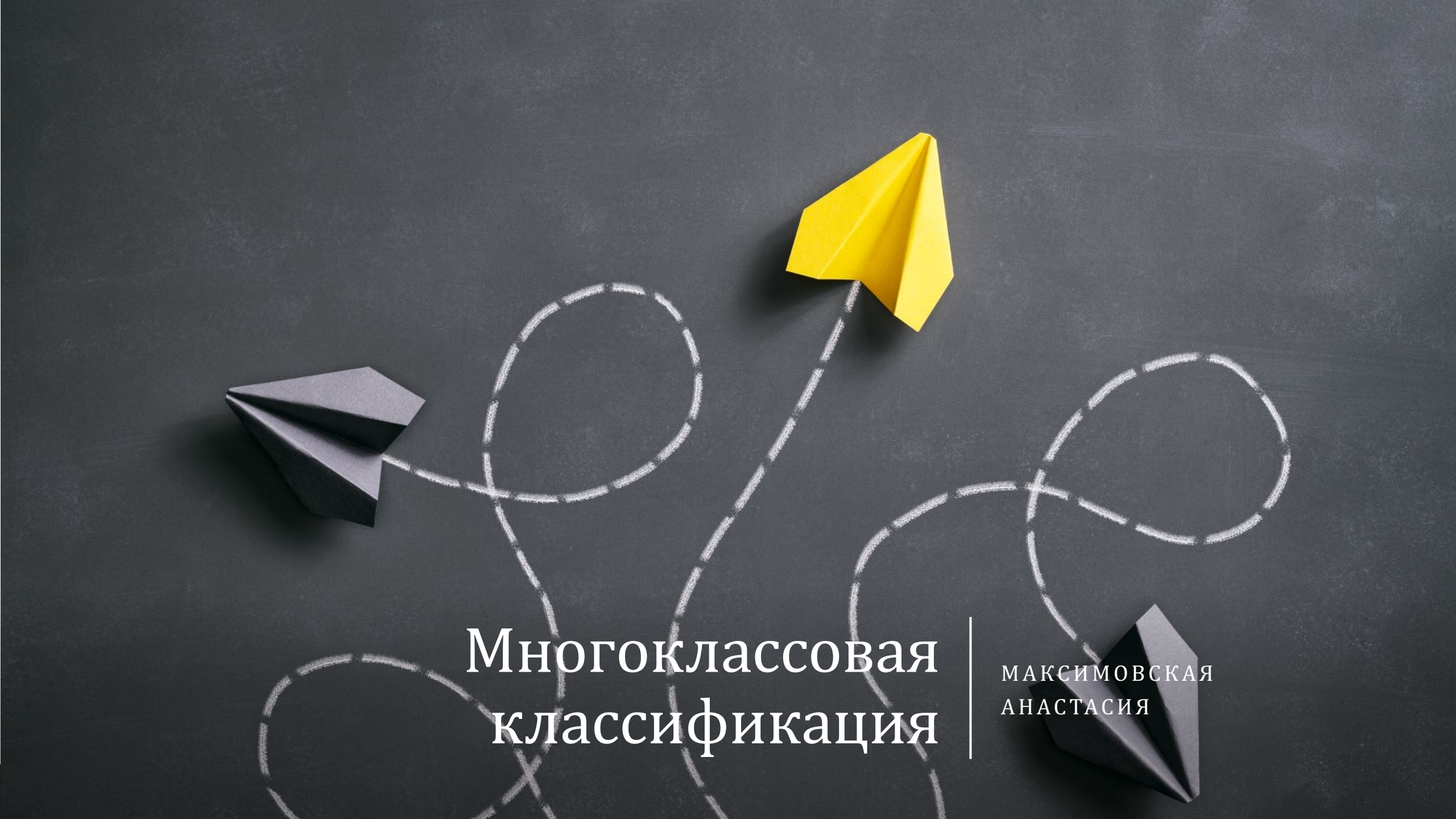


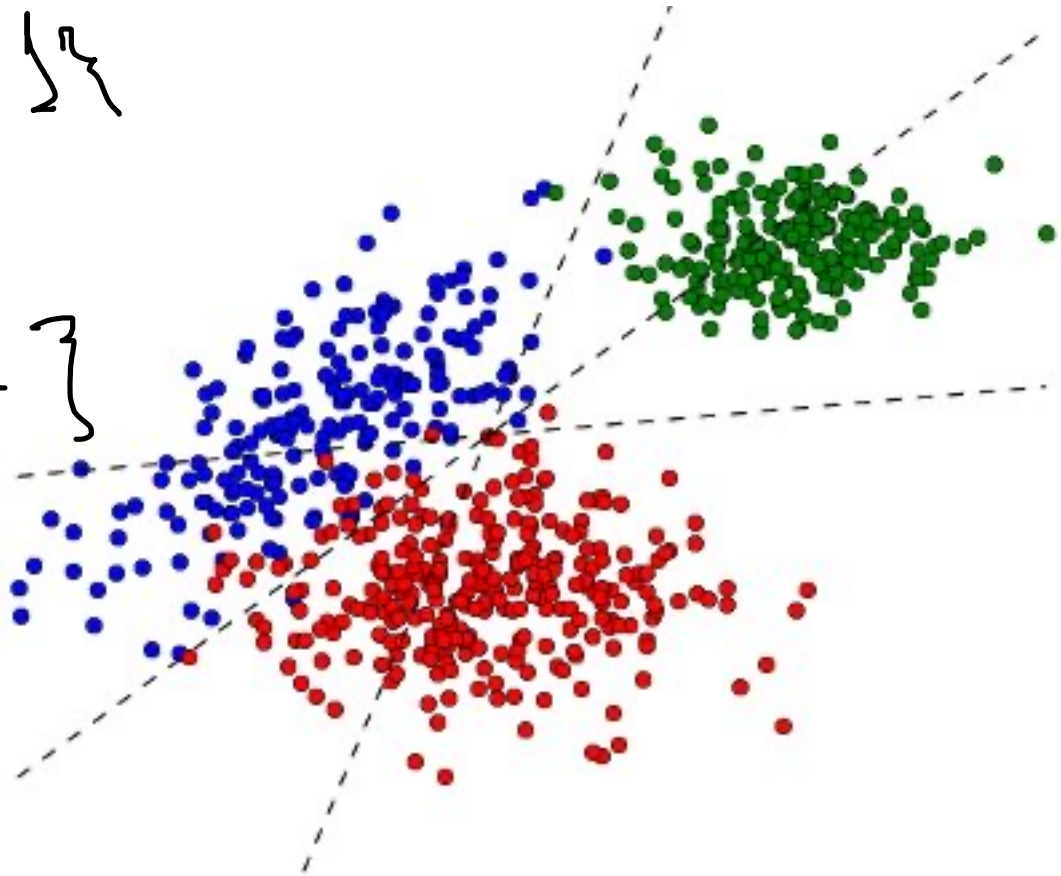
Многоклассовая классификация

МАКСИМОВСКАЯ
АНАСТАСИЯ

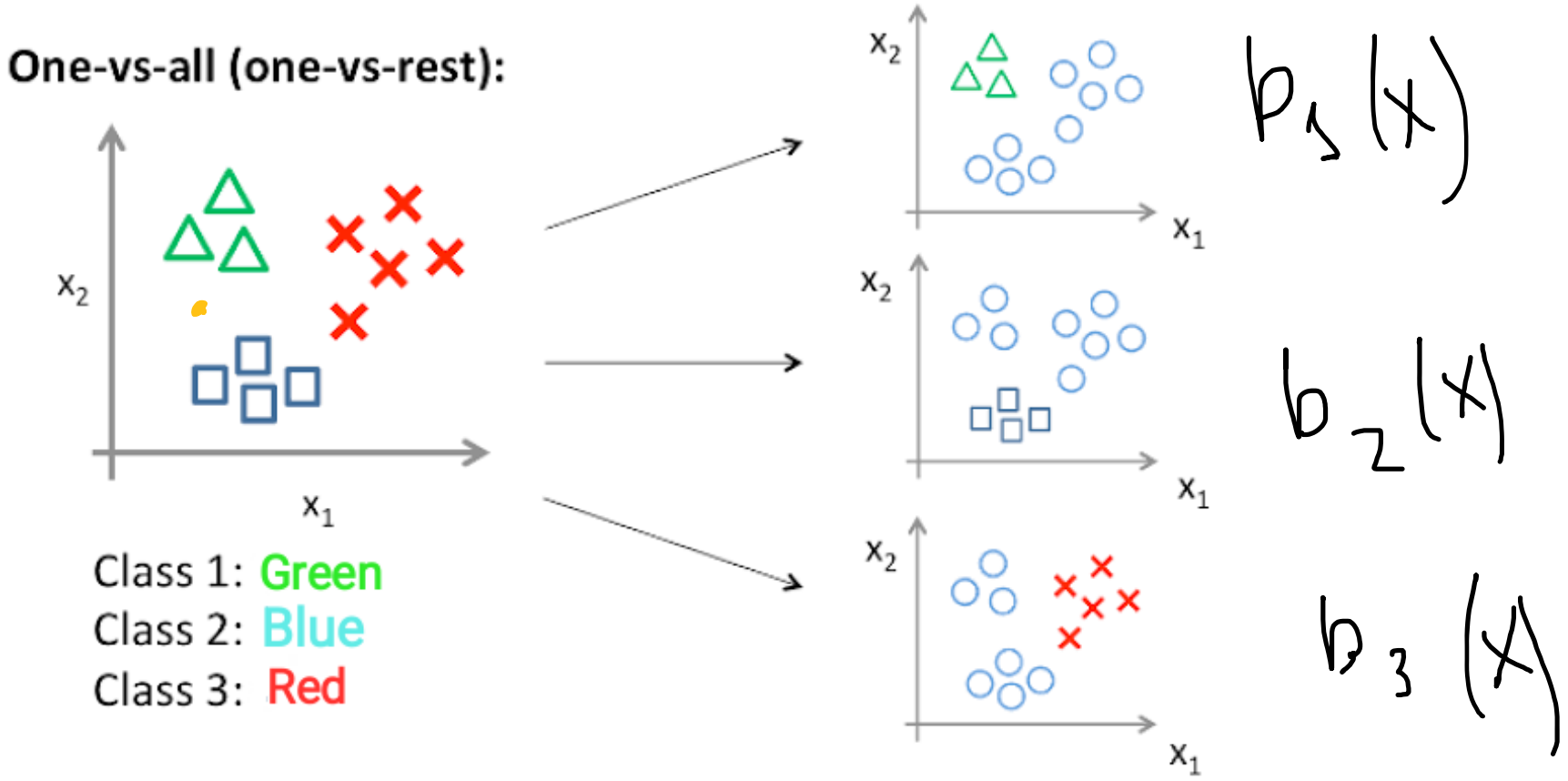
Многоклассовая классификация

$Y = \{+1, -1\}$

$Y = \{1, \ldots K\}$



Один против всех (one-versus-all)



Один против всех (one-versus-all)

- Обучаем K линейных классификаторов, выдающих оценки принадлежности к классам $1, \ldots, K$ соответственно
- Обучаем классификатор b_k по выборке, где целевая переменная равна +1, если это класс k и -1 иначе
- Итоговый классификатор:

$$a(x) = \underset{k \in \{1,\ldots,K\}}{\arg\max}\, b_k(x)$$

Один против всех (one-versus-all)

- Каждый классификатор обучается на своей выборке
- Выходы классификаторов будут иметь разные масштабы, сравнивать их некорректно
- Можно нормировать вектора весов, чтобы они выдавали ответы в одной и той же шкале
- Не всегда хорошее решение (например, для SVM)

Все против всех (all-versus-all)

- Обучим C_k^2 классификаторов $a_{ij}(x), i, j = 1, \dots, K, i \neq j$

Напоминание

$K = 3$

$C_3^2 = \frac{3!}{2!1!} = 3 = \frac{3 \cdot 2 \cdot 1}{2 \cdot 1 \cdot 1}$

$$C_n^k = \frac{n!}{k!(n-k)!}$$

x_j

$a_{12}(x) \Rightarrow 1$

$a_{13}(x) \Rightarrow 3$

$a_{23}(x) \Rightarrow 3$

$(x_j ; 3)$

Все против всех (all-versus-all)

- Обучим C_k^2 классификаторов $a_{ij}(x), i, j = 1, \dots, K, i \neq j$
- Классификатор $a_{ij}(x)$ будем обучать на подвыборке, содержащей только объекты классов i и j:

$$X_{ij} = \{(x_n, y_n) \in X \mid [y_n = i] = 1 \text{ или } [y_n = j] = 1\}$$

- Классификатор будет для любого объекта выдавать класс i или j

Все против всех (all-versus-all)

- Чтобы классифицировать новый объект, подадим его на вход каждого из построенных бинарных классификаторов
- Каждый из них проголосует за свой класс, в качестве ответа выберем тот класс, за который наберется больше всего голосов:

$$a(x) = \arg\max_{k \in \{1,\dots,K\}} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k].$$

Многоклассовая логистическая регрессия

$$b_k(x) = \langle w_1, x \rangle + w_0$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$(b_1(x), \ldots, b_K(x))$$

Многоклассовая логистическая регрессия

- Как преобразовать вектор оценок нескольких классификаторов в вероятности?

$$\mathrm{SoftMax}(z_1, \ldots, z_K) = \left(\frac{\exp(z_1)}{\sum_{k=1}^{K} \exp(z_k)}, \ldots, \frac{\exp(z_K)}{\sum_{k=1}^{K} \exp(z_k)} \right)$$

- Вероятность класса k в таком случае:

$$P(y = k \,|\, x, w) = \frac{\exp\left(\langle w_k, x \rangle + w_{0k}\right)}{\sum_{j=1}^{K} \exp\left(\langle w_j, x \rangle + w_{0j}\right)}$$

Многоклассовая логистическая регрессия

- Обучаем веса с помощью метода максимального правдоподобия:

$$\sum_{i=1}^{\ell} \log P(y = y_i \mid x_i, w) \to \max_{w_1,\ldots,w_K}$$

Метрики

- Есть 2 подхода: микро- и макро-усреднение метрик классификации
- Микро-усреднение: усредняем по всем классам, считаем итоговую

$$\text{precision}(a, X) = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FP}}}, \qquad \overline{\text{TP}} = \frac{1}{K}\sum_{k=1}^{K} \text{TP}_k.$$

- Макро-усреднение: вычисляем итоговую метрику для каждого класса, усредняем по всем классам

$$\text{precision}(a, X) = \frac{1}{K}\sum_{k=1}^{K} \text{precision}_k(a, X); \qquad \text{precision}_k(a, X) = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}.$$

Метрики: пример

		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

Метрики: пример

		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

Метрики: пример

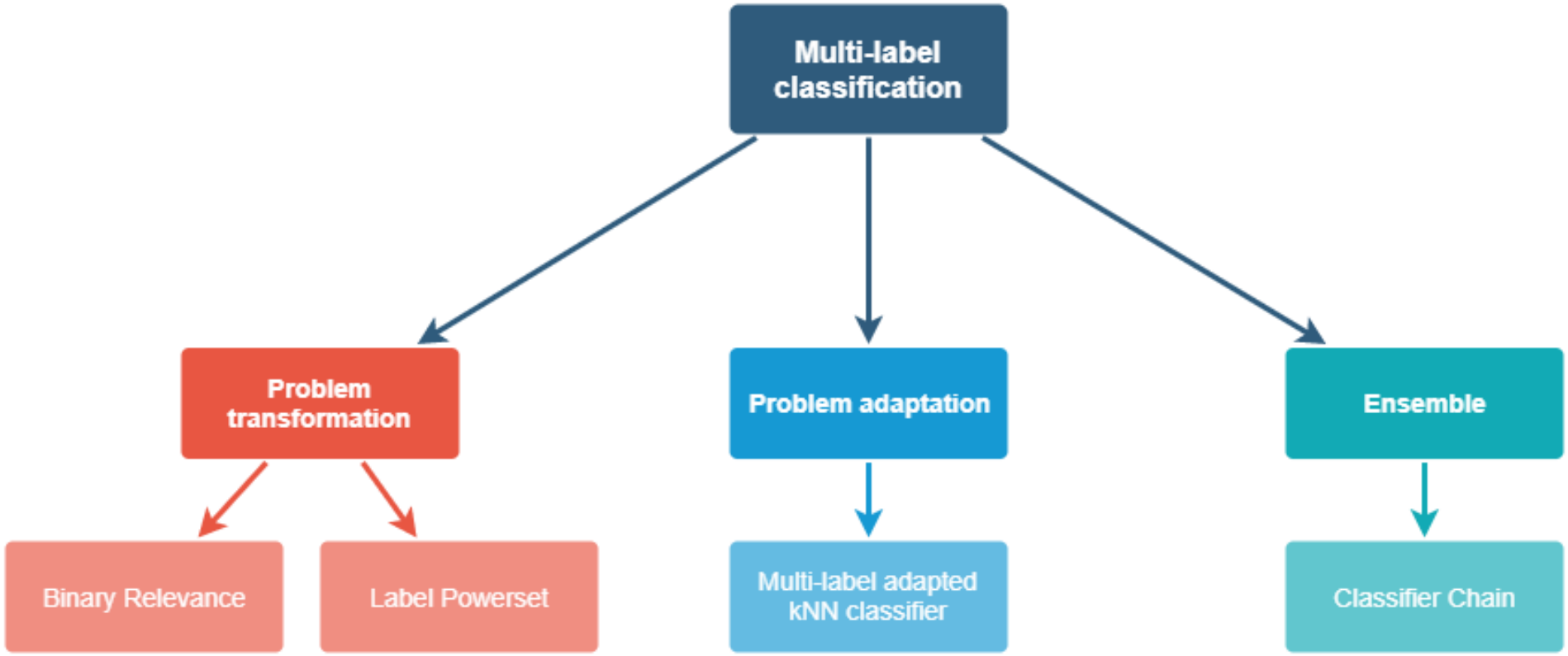
		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

Метрики: пример

		True/Actual		
		Cat ()	Fish ()	Hen ()
Predicted	Cat ()	4	6	3
	Fish ()	1	2	0
	Hen ()	1	2	6

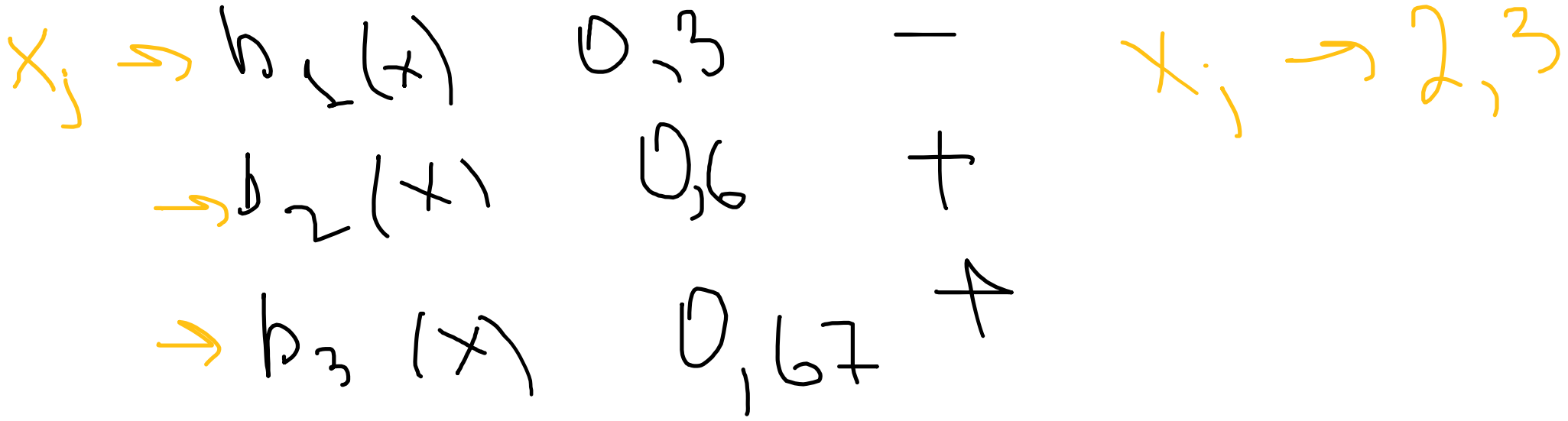
Пересекающиеся классы

Пересекающиеся классы



Binary relevance

- Предполагаем, что все классы независимы, и определяем принадлежность к каждому отдельным классификатором
- Проблема – не учитываем возможные связи между классами



Стекинг классификаторов

- Разделим выборку X на 2 части: X_1 и X_2
- На первой X_1 обучим K независимых классификаторов $b_k(x)$
- Для каждого объекта из второй выборки сформируем признаковое описание из прогнозов наших классификаторов:

$$x'_{ik} = b_k(x_i), \quad x_i \in X_2$$

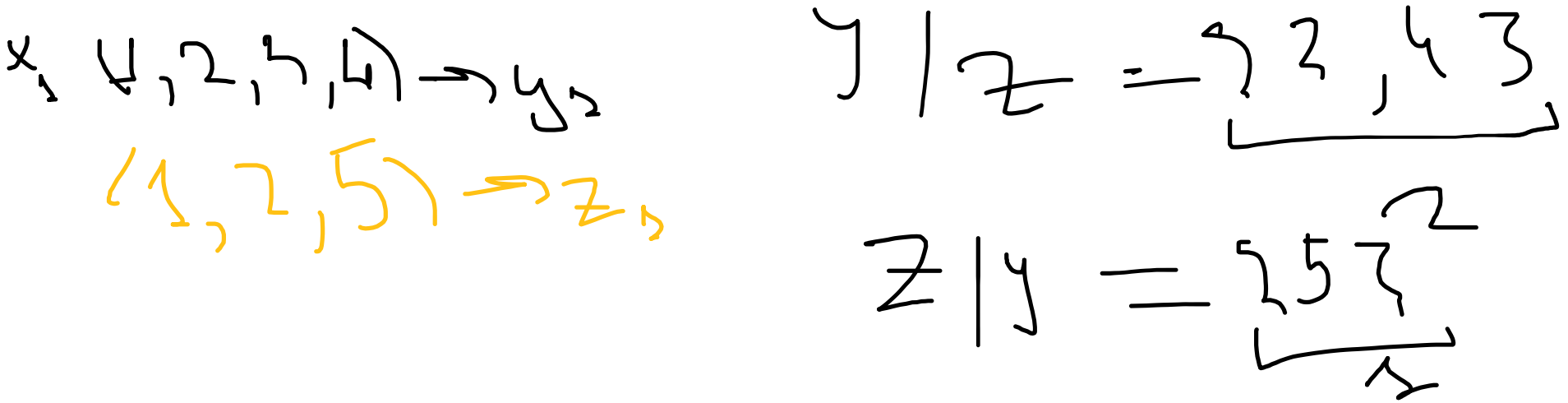
- Обучим на ***полученной*** выборке новый набор классификаторов $a_k(x)$,каждый из которых определяет принадлежность объекта к одному из классов

Стекинг классификаторов

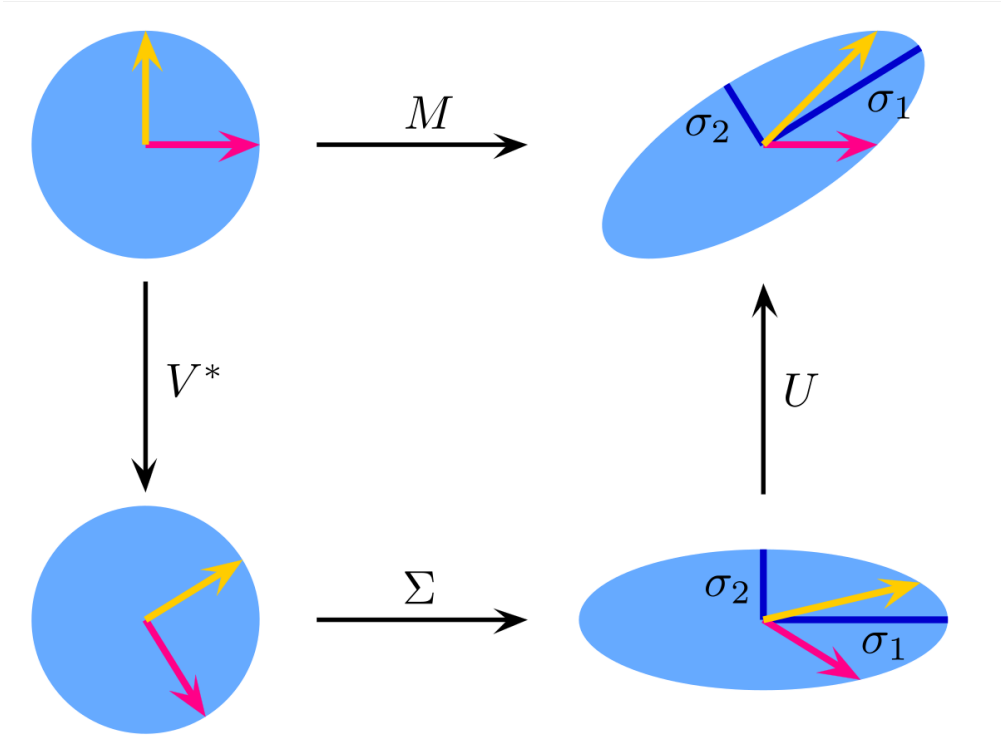
- Таким образом, новые классификаторы $a_k(x)$ опираются на прогнозы классификаторов с первого этапа $b_k(x)$
- Благодаря этому они могут обнаружить связи между классами
- Обучать $b_k(x)$ и $a_k(x)$ на одной выборке – плохая идея, т.к. может привести к переобучению

Трансформация пространства ответов

- Хотим учитывать связи между классами в рамках одной модели
- Идея: трансформировать пространство ответов так, чтобы классы стали менее зависимыми
- Используем SVD разложение для вектора ответов Y



SVD



$$M = U \cdot \Sigma \cdot V^*$$

Трансформация пространства ответов

- Обозначим через V_M матрицу, состоящую из тех M столбцов матрицы V , которые соответствуют наибольшим сингулярным числам
- Спроецируем с её помощью матрицы Y :

$$YV_M = Y' \in \mathbb{R}^{\ell \times M}$$

- Настроим на новые метки Y' независимые модели
- Получим матрицу прогнозов A' и переведем ее в исходное пространство:

$$A = A'V_M^T$$

Метрики качества для multi-label

- Z_i – множество классов, к котором можно отнести объект
- Хеммингово расстояние доля классов, факт принадлежности которым угадан неверно:

$$\text{hamming}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \setminus Z_i| + |Z_i \setminus Y_i|}{K}$$

- Хотим минимизировать

Метрики качества для multi-label

$$\text{accuracy}(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|},$$

$$\text{precision}(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{|Y_i \cap Z_i|}{|Z_i|},$$

$$\text{recall}(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{|Y_i \cap Z_i|}{|Y_i|}.$$

$Y = \{1, 2, 3, 4\}$ $Z = \{1, 2, 5\}$

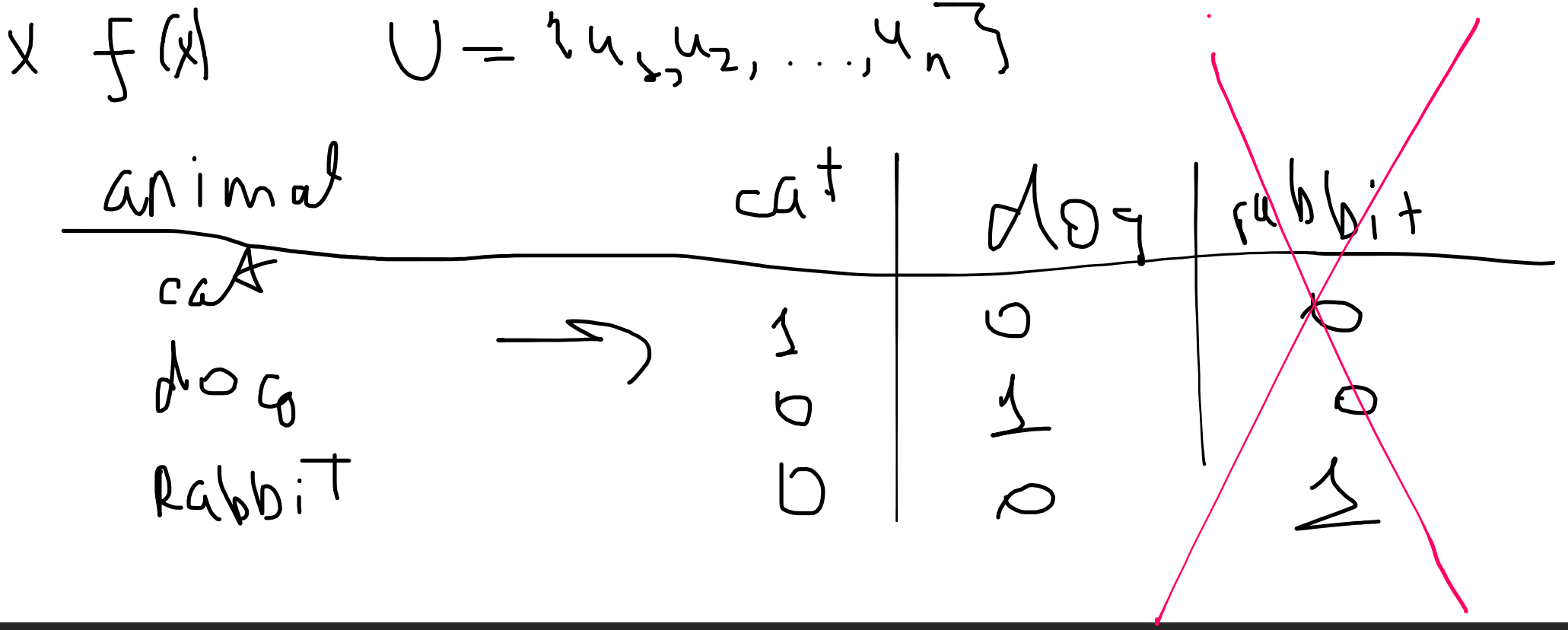
$= \frac{|\{1,2\}|}{|\{1,\dots,5\}|} = \frac{2}{5} = 0$

$= \frac{2}{3} = 0,67$

$= \frac{2}{4} = 0,5$

Категориальные признаки

- Помним: One-Hot кодирование



Категориальные признаки

- Помним: One-Hot кодирование
- Бинарное кодирование с хэшированием

Бинарное кодирование с хэшированием

- Выберем хэш-функцию $h : U \to \{1, 2, \ldots, B\}$
- После этого бинарные признаки можно индексировать значениями хэш-функции:

$$g_j(x) = [h(f(x)) = j], \quad j = 1, \ldots, B.$$

https://contrib.scikit-learn.org/category_encoders/hashing.html – *Hashing Encoder,*
https://docs.python.org/3/library/hashlib.html – *hashlib*

Бинарное кодирование с хэшированием

Color	Hash Function	Divide by	Reminder
Red	36614357519	8	3
Blue	54663777951	8	7
Green	75535549907	8	7

Feature Hashing

Reminder -->	0	1	2	3	4	5	6	7
	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8
Red	0	0	0	**1**	0	0	0	0
Blue	0	0	0	0	0	0	0	**1**
Green	0	0	0	0	0	0	0	**1**

Бинарное кодирование с хэшированием

Преимущества:

- Отпадает необходимость в хранении соответствий между значениями категориального признака и индексами бинарных признаков
- Позволяет понизить количество признаков (как правило, без существенной потери качества)

Категориальные признаки

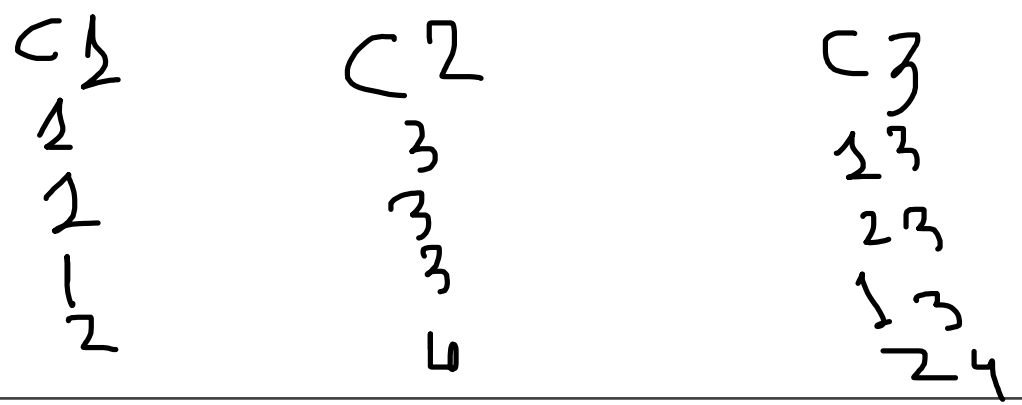
- Помним: One-Hot кодирование (и другие кодирования – см. лекции 1-2)
- Бинарное кодирование с хэшированием
- Счетчики

Счетчики

$$\text{counts}(u, X) = \sum_{(x,y)\in X} [f(x) = u],$$

$$\text{successes}_k(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], \quad k = 1, \ldots, K.$$

Счетчики



Заменим наш категориальный признак на K вещественных $g_1(x), \ldots, g_k(x)$

$$g_k(x, X) = \frac{\text{successes}_k(f(x), X) + c_k}{\text{counts}(f(x), X) + \sum_{m=1}^{K} c_m}, \quad k = 1, \ldots, K.$$