

SVM. Многоклассовая классификация

Паточенко Евгений

НИУ ВШЭ

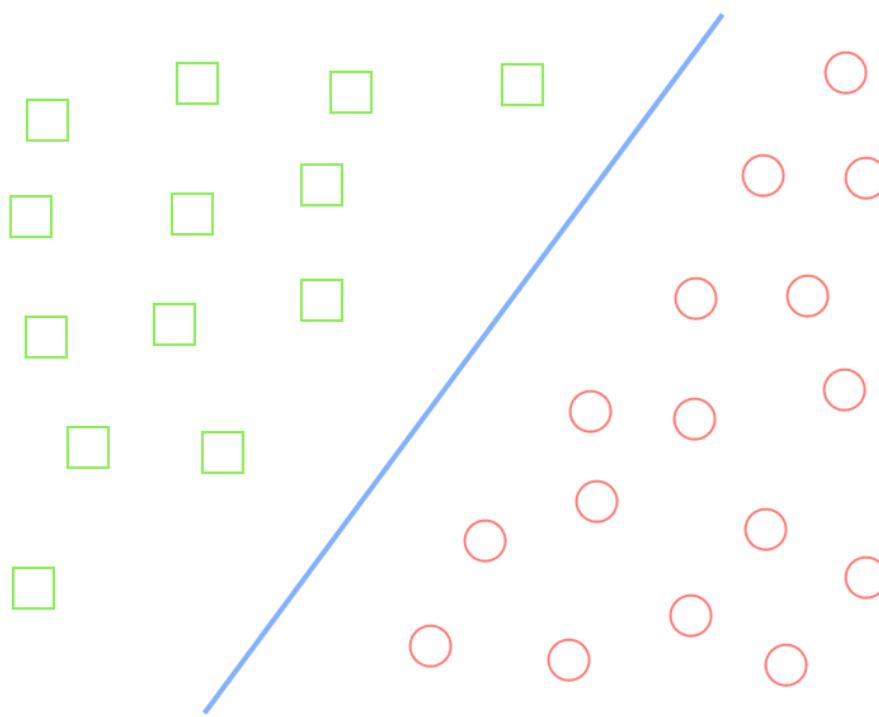
План занятия

- SVM (метод опорных векторов)
- Многоклассовая классификация
- Метрики качества многоклассовой классификации

Метод опорных векторов (SVM)

Линейно разделимая выборка

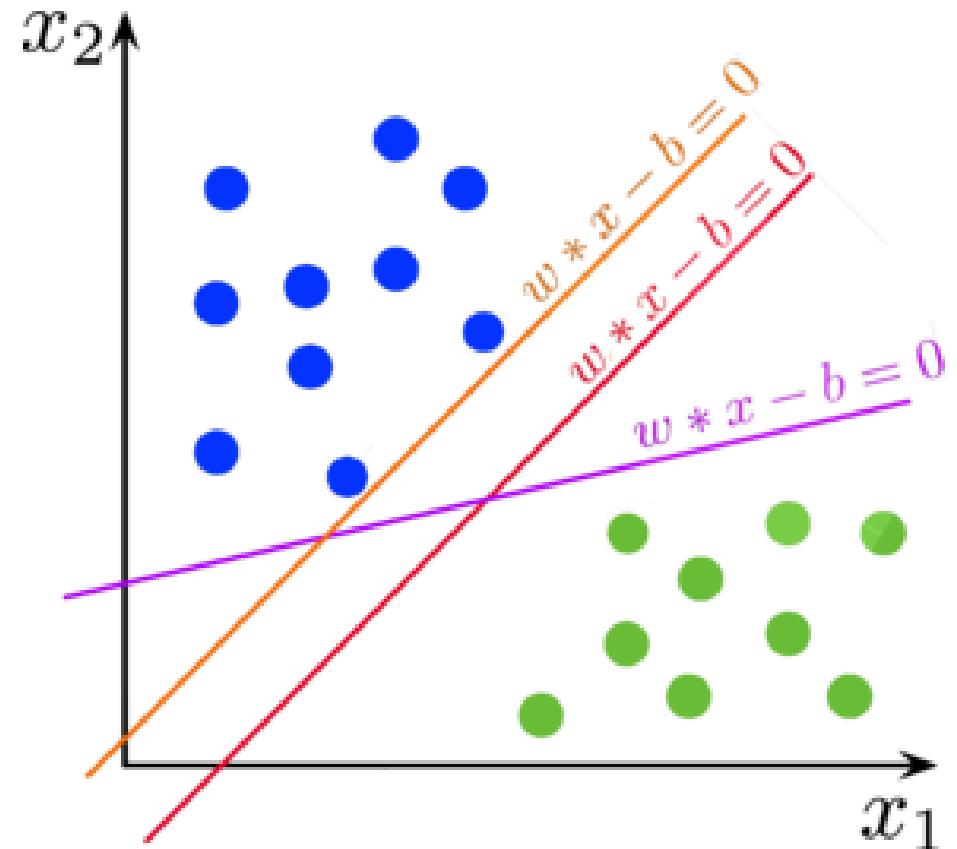
Выборка линейно разделима, если существует такой вектор параметров β , что соответствующий классификатор $f(x)$ не допускает ошибок на этой выборке



Метод опорных векторов (SVM)

Линейно разделимая выборка

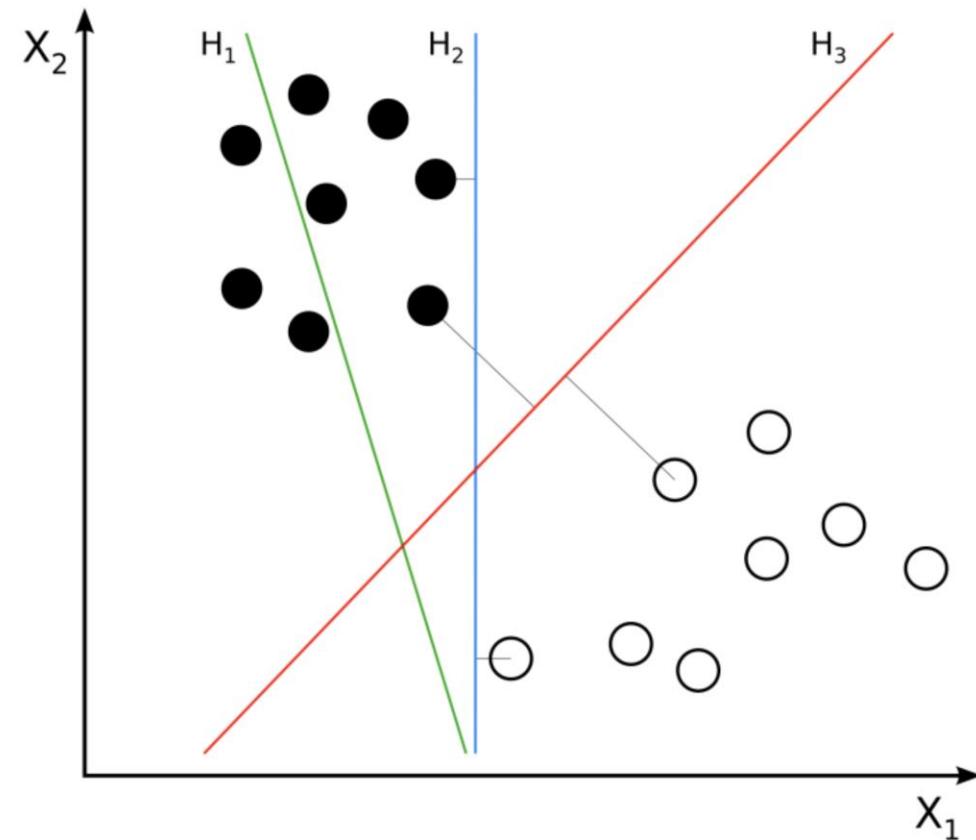
Для двух линейно разделимых классов
возможны различные варианты
построения разделяющей
гиперплоскости



Метод опорных векторов (SVM)

Критерий «хорошей» гиперплоскости

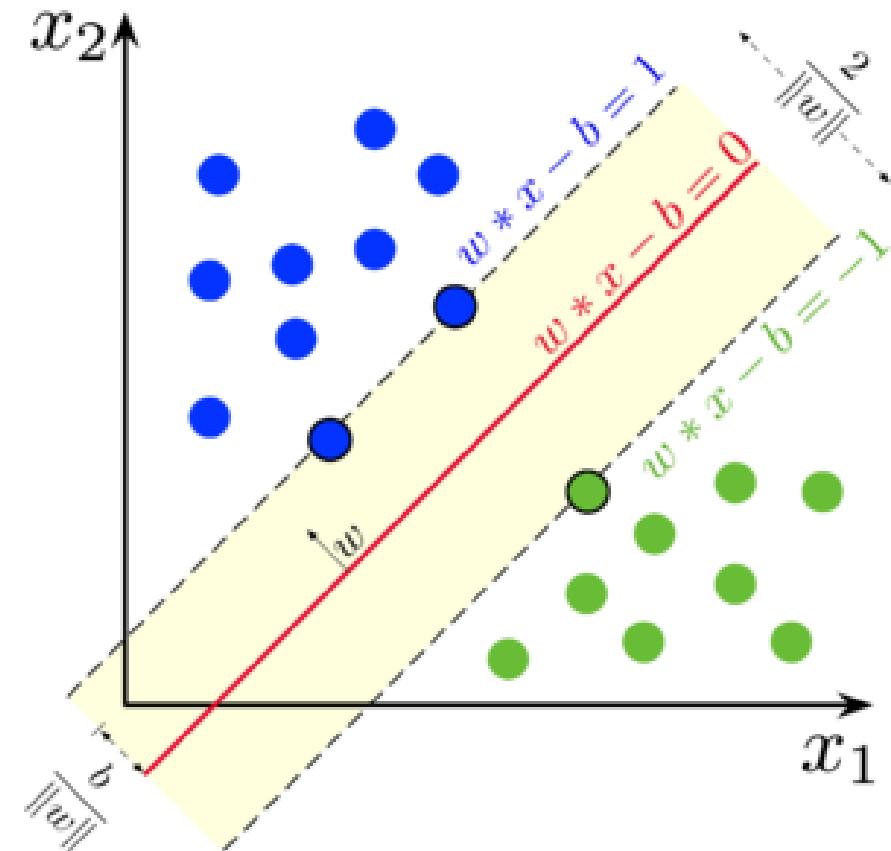
Гиперплоскость H_1 оптимальнее гиперплоскости H_2 , если расстояние от H_1 до некоторого ближайшего объекта выборки x_1 больше, чем расстояние от H_2 до своего ближайшего объекта x_2



Метод опорных векторов (SVM)

*SVM (Support Vector Machine) для
разделимого случая*

Цель SVM — максимизировать ширину
разделяющей полосы, то есть найти
наилучшую разделяющую
гиперплоскость, которая максимально
отдалена от ближайших точек каждого
класса (**опорных векторов**)



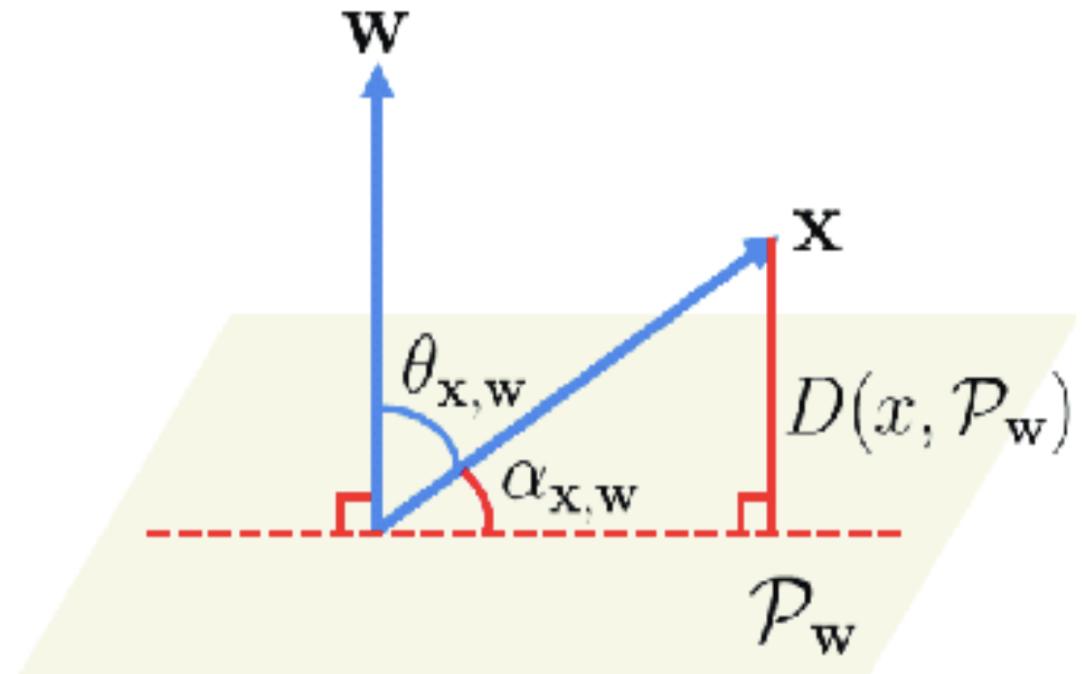
Источник: [https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_\(SVM\)](https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM))

Метод опорных векторов (SVM)

Расстояние от точки до гиперплоскости

Расстояние от точки x_0 до гиперплоскости H , заданной своим вектором нормали w и смещением b , определяется по формуле:

$$p(x_0, H) = \frac{|(w, x_0) + b|}{|w|}$$



Метод опорных векторов (SVM)

Расстояние от точки до гиперплоскости

Воспользуемся свойством функции *sign*:

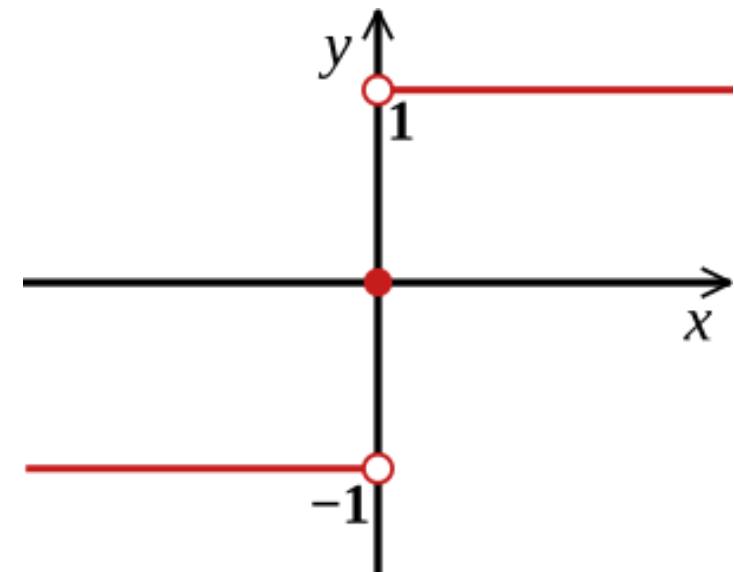
$\text{sign}(x \cdot y) = \text{sign } x \cdot \text{sign } y$, тогда для ответов $a(x)$ классификатора следует, что

$$a(x) = \text{sign}((w, x)) = \text{sign}((\hat{w}, x) + b)$$

$$= \text{sign}((k\hat{w}, x) + kb) = \text{sign}\left(k((\hat{w}, x) + b)\right)$$

$$= \text{sign}(k) \cdot \text{sign}((\hat{w}, x) + b) = \{\text{если } k > 0\}$$

$$= \text{sign}((\hat{w}, x) + b)$$



Источник: https://en.wikipedia.org/wiki/Sign_function

Метод опорных векторов (SVM)

Расстояние от точки до гиперплоскости

Воспользуемся свободой выбора коэффициента k и отнормируем веса модели так, чтобы до ближайшего объекта x^* обучающей выборки X было выполнено равенство:

$$\min_{x \in X} |(\hat{w}, x) + b| = 1$$

Тогда по формуле расстояния выходит, что расстояние до ближайшего объекта выборки:

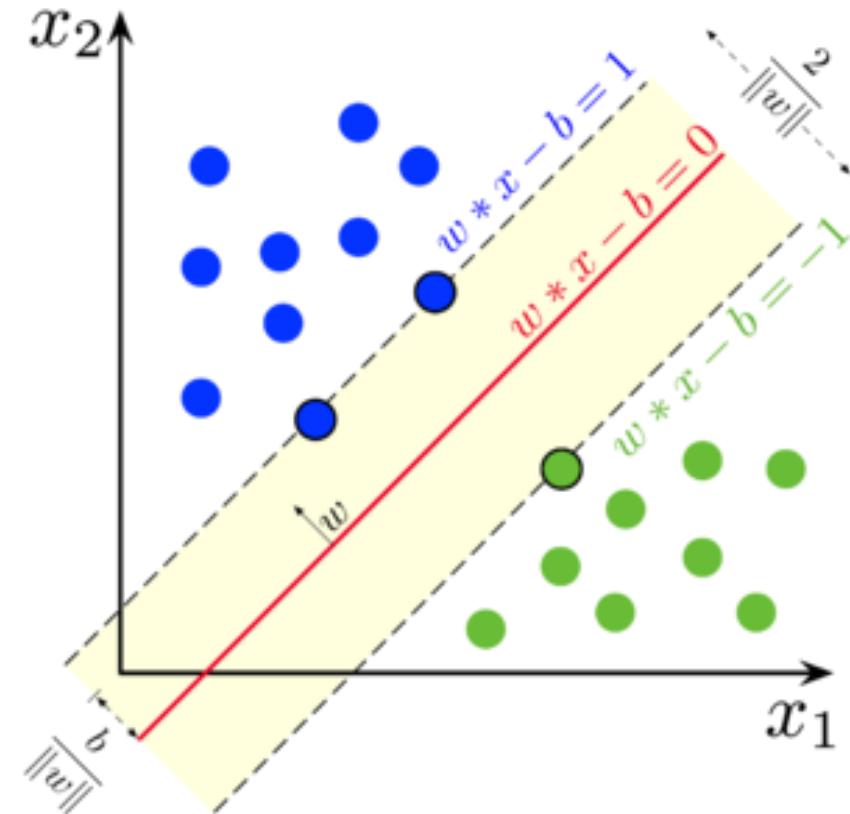
$$p(x^*, H) = \min_{x \in X} \frac{|(\hat{w}, x) + b|}{|\hat{w}|} = \frac{1}{\hat{w}} \min_{x \in X} |(\hat{w}, x) + b| = \frac{1}{|\hat{w}|}$$

Метод опорных векторов (SVM)

Расстояние от точки до гиперплоскости

Ширина зазора между положительным и отрицательным классами для линейной разделимой выборки будет удвоенной величиной: $M = \frac{2}{\|w\|}$

Ее нам и предстоит максимизировать



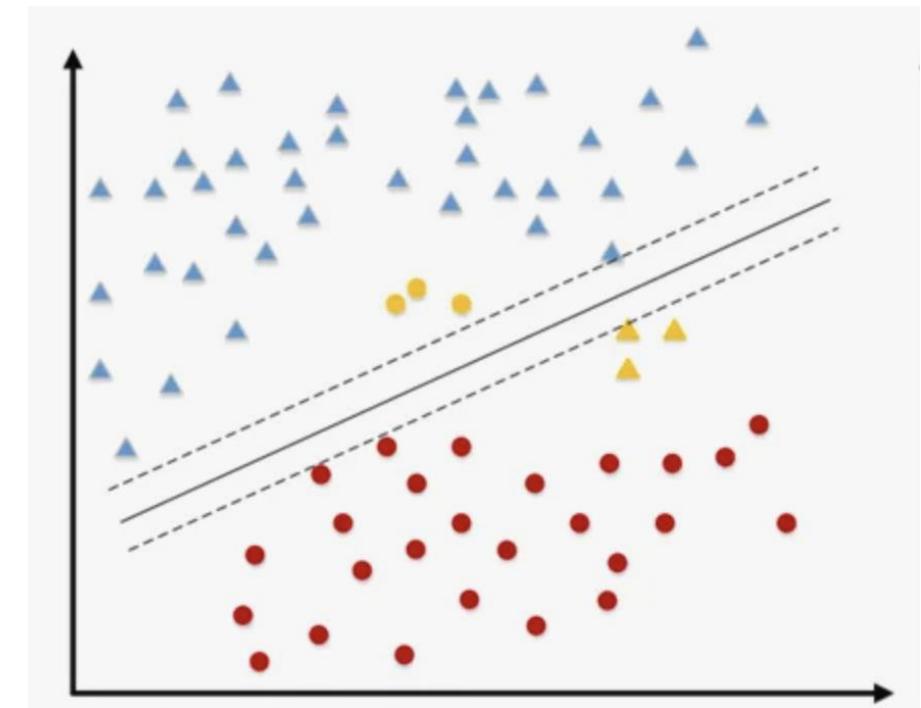
Источник: [https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_\(SVM\)](https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM))

Метод опорных векторов (SVM)

Линейно неразделимая выборка

В большинстве случаев мы имеем дело с линейно неразделимыми данными

Объекты могут попадать на другую сторону гиперплоскости или внутрь отступа



Метод опорных векторов (SVM)

Линейно неразделимая выборка

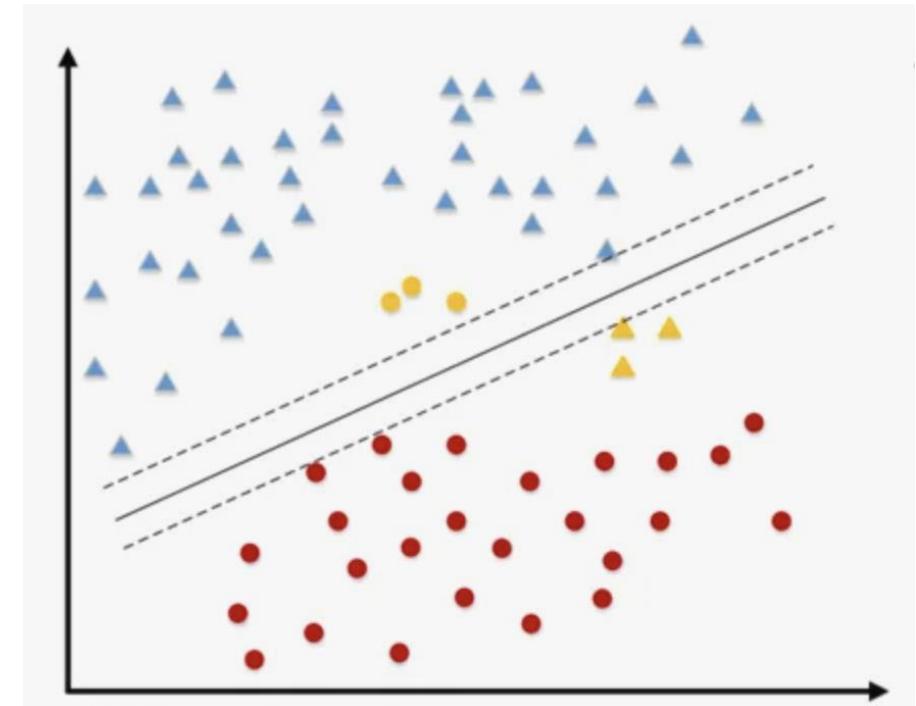
Введем штраф ξ_i такое, что $\xi_i \rightarrow 0$

$$M_i(w) \geq 1 - \xi_i \quad i = 1, 2, \dots, l \quad \xi_i \neq 0$$

Тогда:

$$\frac{\|w\|^2}{2} + C \sum_i^l \xi_i \rightarrow \min_{w, \xi}$$

$$M_i(w) \geq 1 + \xi_i$$

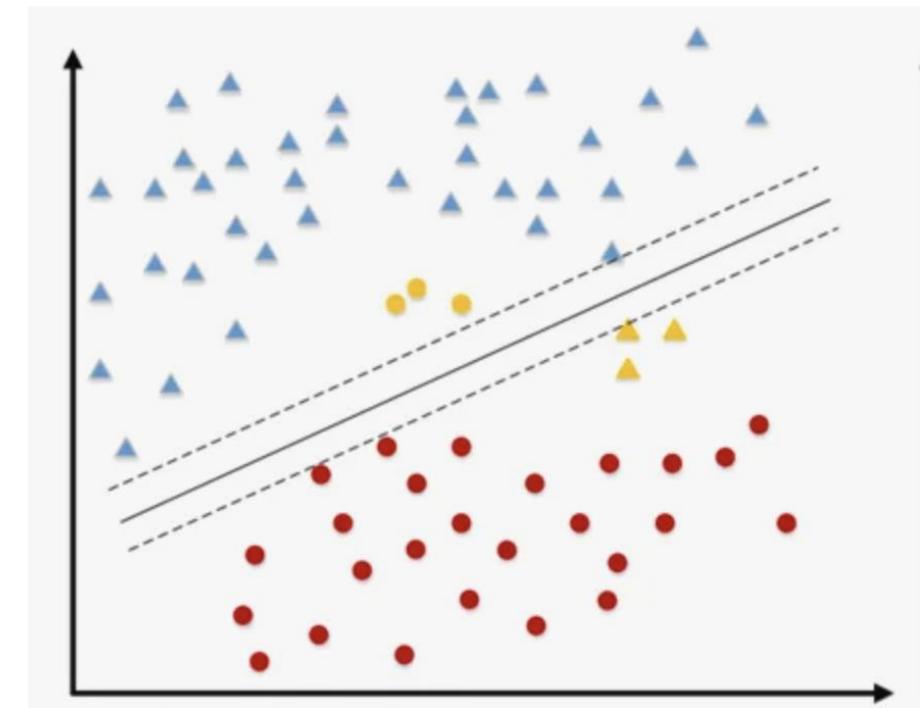


Метод опорных векторов (SVM)

Линейно неразделимая выборка

Таким образом мы получаем «мягкий» вариант SVM, в котором мы хотим, чтобы:

- алгоритм имел как можно меньшие штрафы ξ_i
- при этом имел как можно более широкий зазор



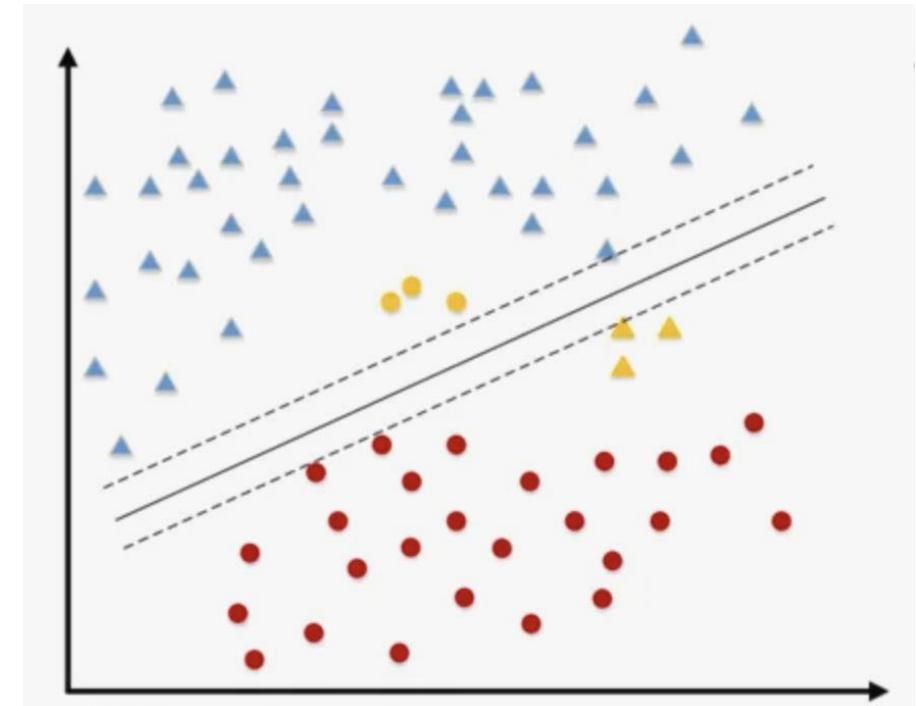
Метод опорных векторов (SVM)

Линейно неразделимая выборка

Таким образом мы получаем «мягкий» вариант SVM, в котором мы хотим, чтобы:

- алгоритм имел как можно меньшие штрафы ξ_i
- при этом имел как можно более широкий зазор

Такая задача имеет единственное решение



Метод опорных векторов (SVM)

Оптимизация

Оптимизируем функционал ошибки Q :

$$Q(f, x) = \frac{1}{N} \sum_{i=0}^N I[M_i < 0] \leq \frac{1}{N} \sum_{i=0}^N L(x_i, y_i)$$

Метод опорных векторов (SVM)

Оптимизация

Оптимизируем функционал ошибки Q :

$$Q(f, x) = \frac{1}{N} \sum_{i=0}^N I[M_i < 0] \leq \frac{1}{N} \sum_{i=0}^N L(x_i, y_i)$$

С функцией потерь $L(M) = \max(0, 1 - M) = (1 - M)$,
с L_2 -регуляризацией:

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Метод опорных векторов (SVM)

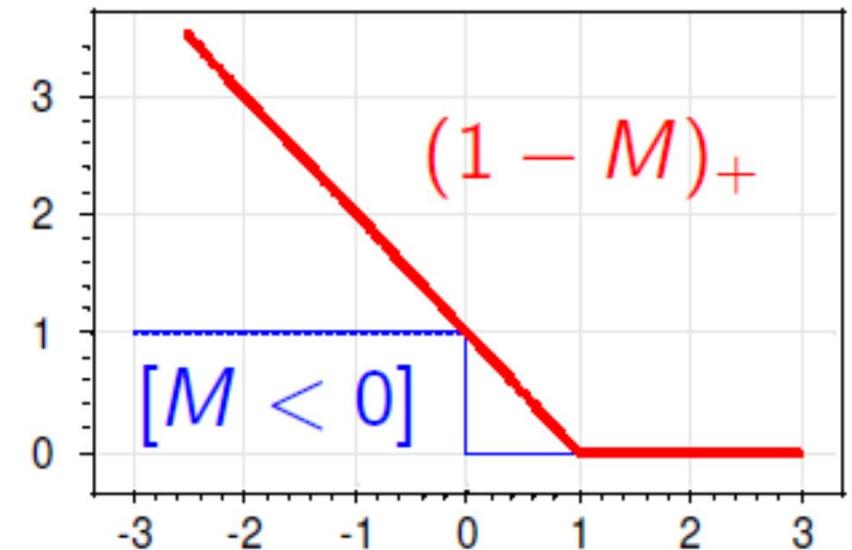
Оптимизация

Оптимизируем функционал ошибки Q :

$$Q(f, x) = \frac{1}{N} \sum_{i=0}^N I[M_i < 0] \leq \frac{1}{N} \sum_{i=0}^N L(x_i, y_i)$$

С функцией потерь $L(M) = \max(0, 1 - M) = (1 - M)$,
с L_2 -регуляризацией:

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$



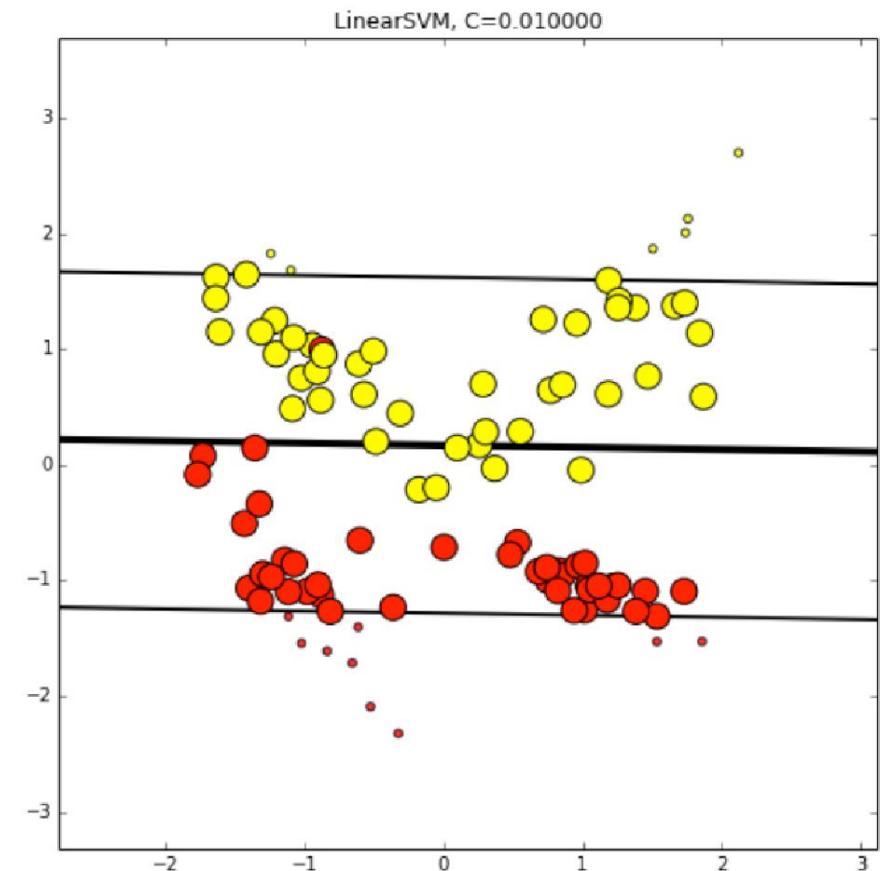
Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

Метод опорных векторов (SVM)

Значение константы C

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

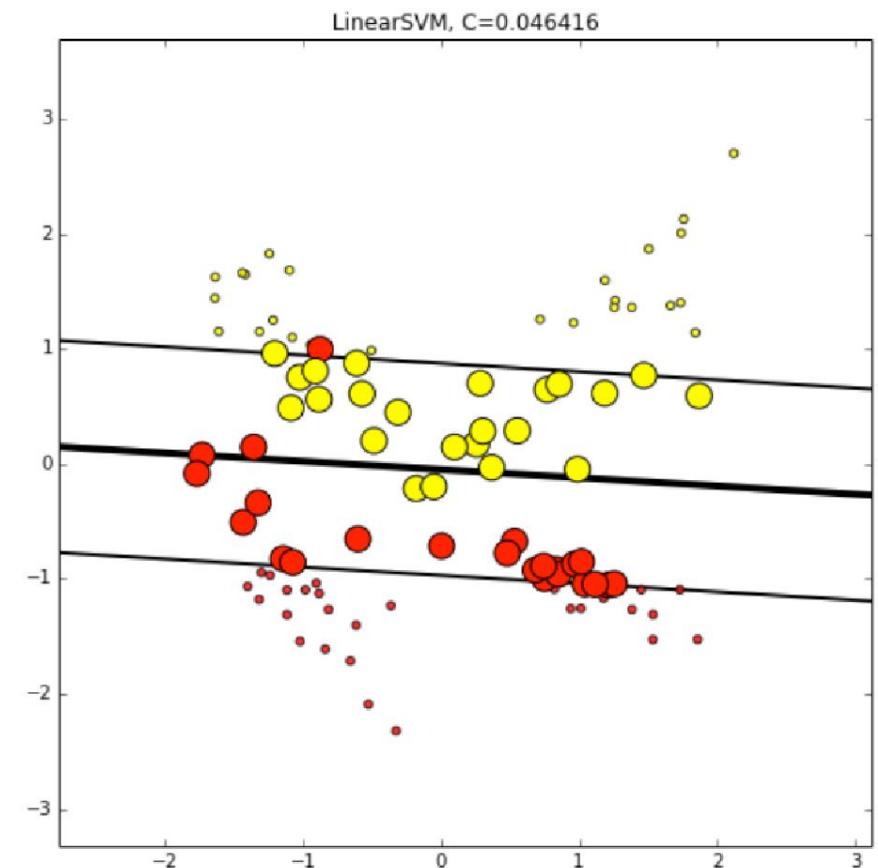


Метод опорных векторов (SVM)

Значение константы C

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

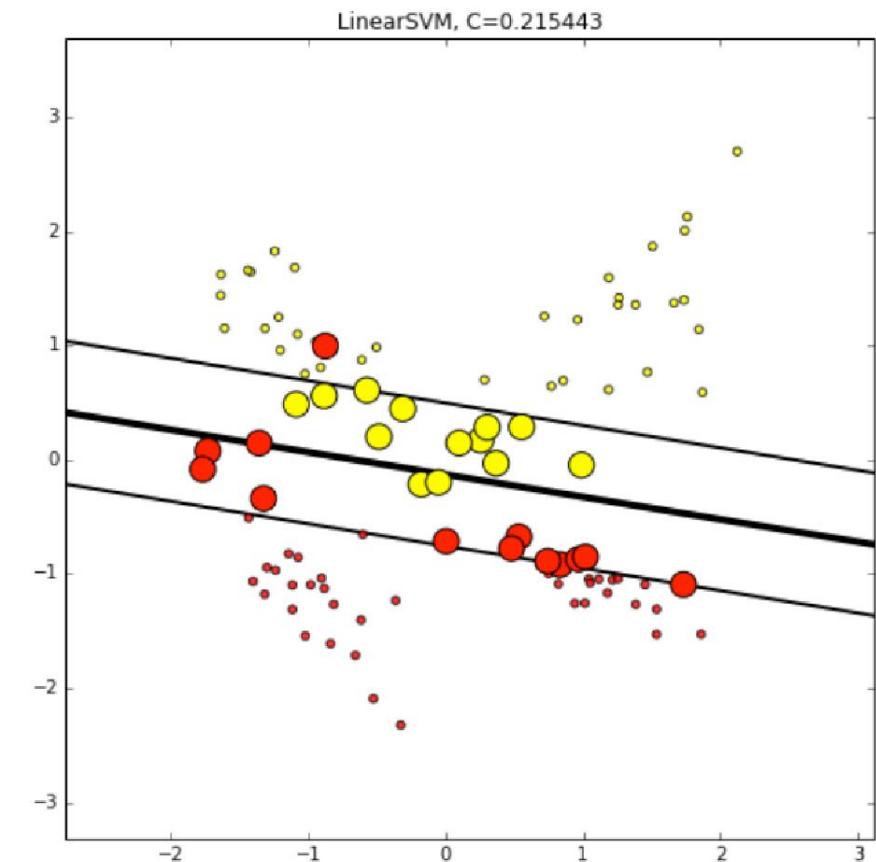


Метод опорных векторов (SVM)

Значение константы C

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

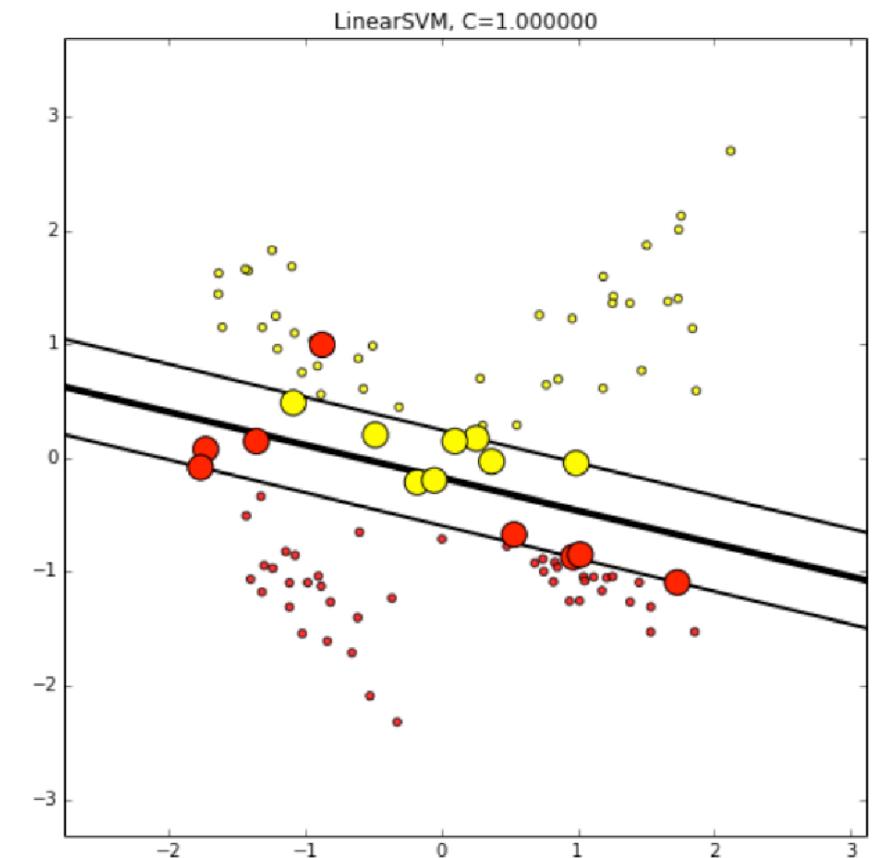


Метод опорных векторов (SVM)

Значение константы C

$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

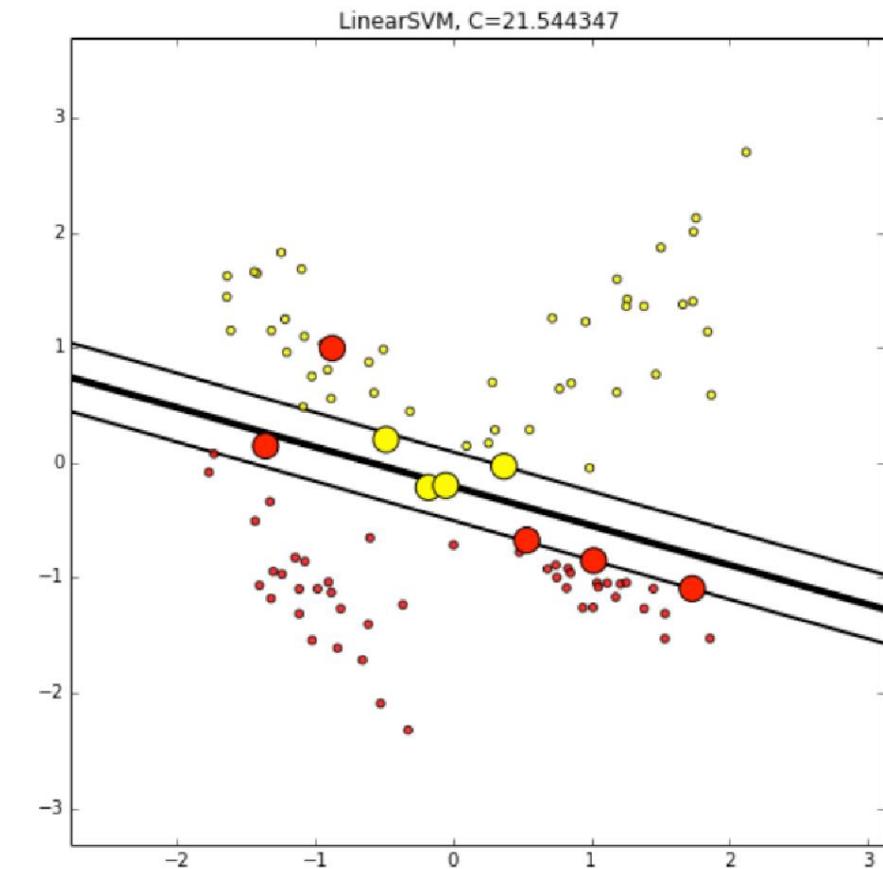


Метод опорных векторов (SVM)

Значение константы C

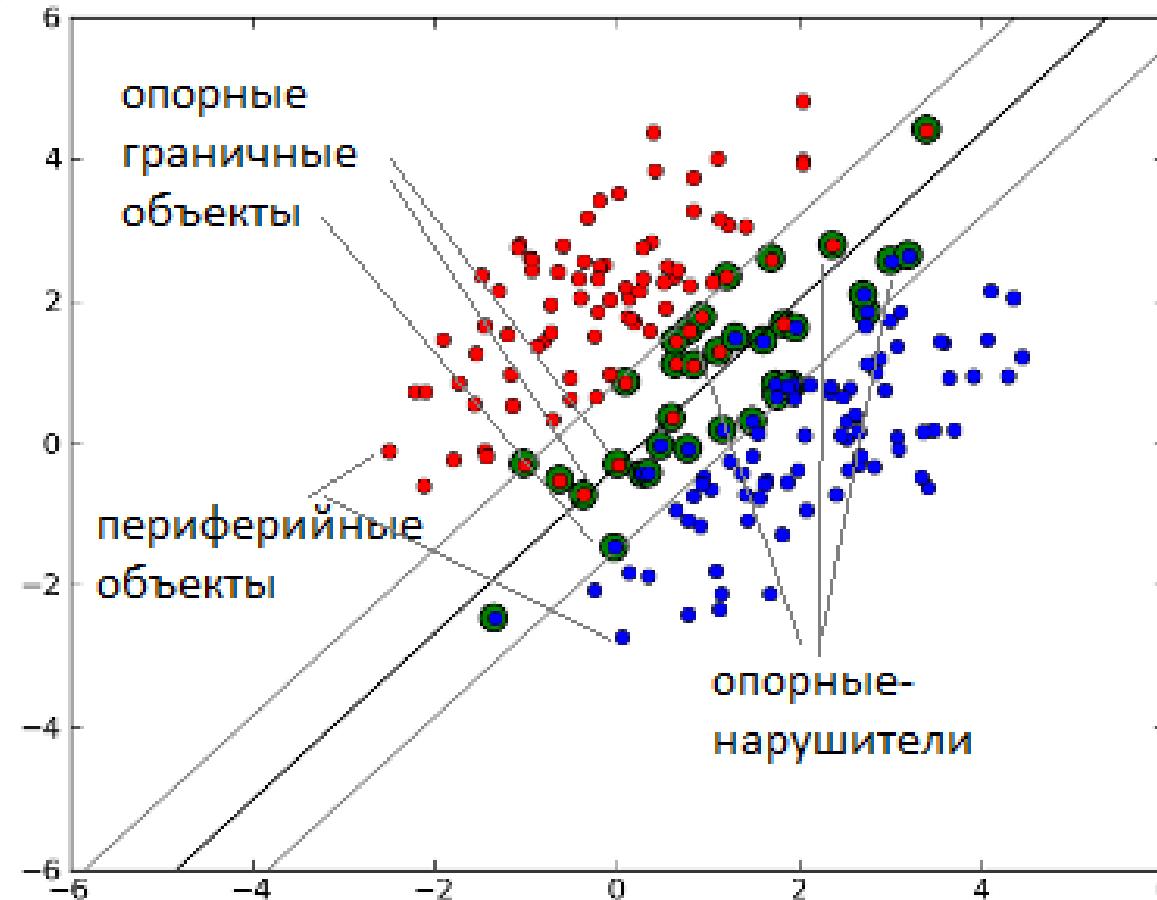
$$Q(f, x) = \sum_{i=1}^l [\max(0, 1 - y_i(w, x_i))] + \frac{1}{2C} |w|^2 \rightarrow \min_w$$

Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки



Метод опорных векторов (SVM)

Типы объектов в SVM



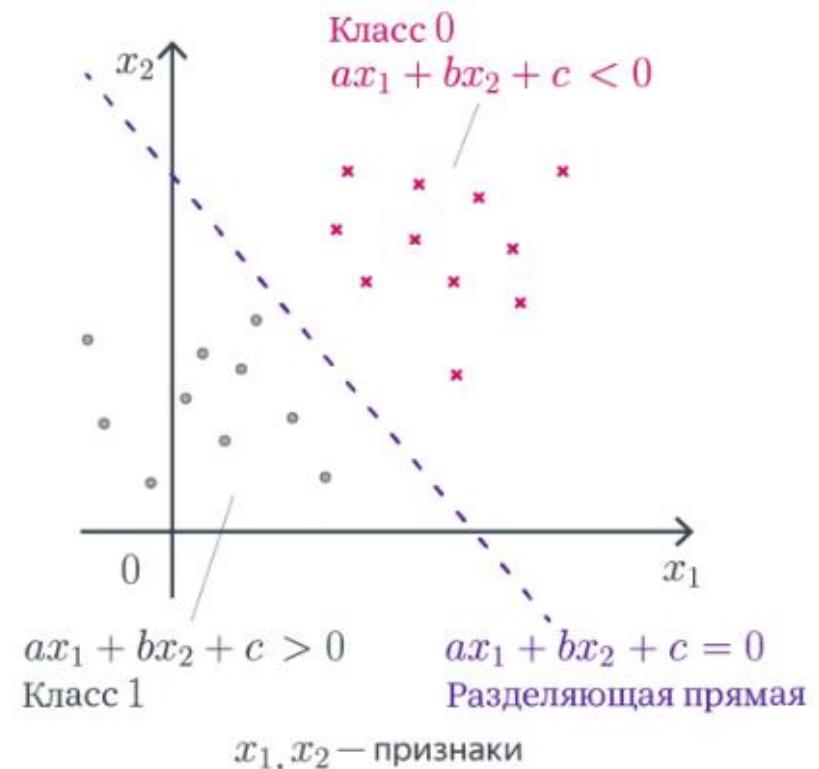
Линейные методы классификации (повтор)

Классификация

Модель машинного обучения, используемая для прогнозирования категориальной (дискретной) целевой переменной на основе одной или нескольких независимых переменных (признаков). Целевая переменная принимает конечное число классов или меток.

Может быть:

- Бинарной (классификация на два класса) $Y = \{0,1\}$
- Многоклассовой (классификация на M непересекающихся классов) $Y = \{1, \dots, M\}$
- Многоклассовой (классификация на M классов, которые могут пересекаться) $Y = \{0,1\}^M$



Источник: <https://education.yandex.ru/handbook/ml/article/linear-models>

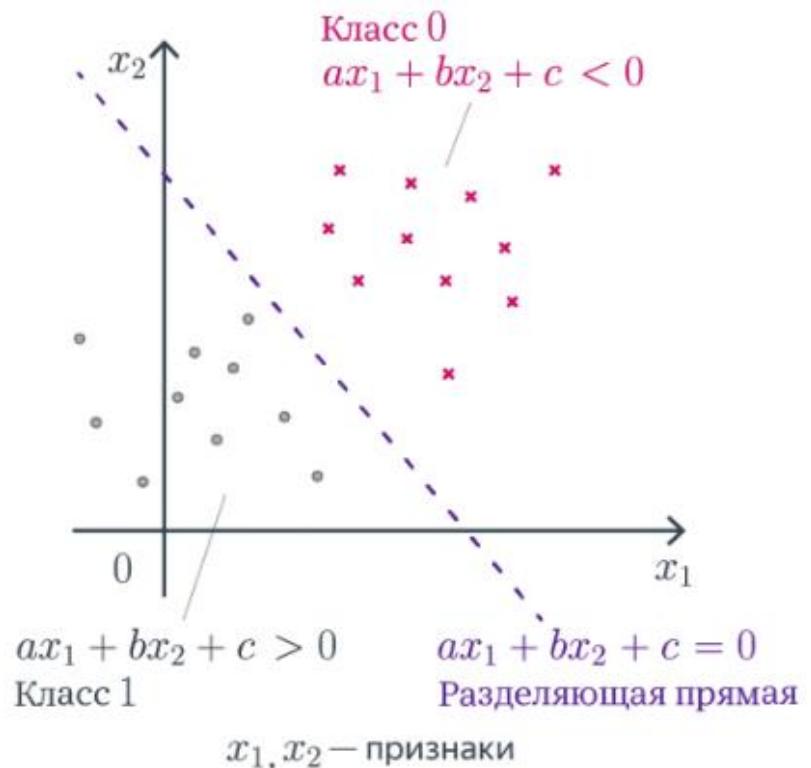
Линейные методы классификации (повтор)

Классификация

Модель машинного обучения, используемая для прогнозирования категориальной (дискретной) целевой переменной на основе одной или нескольких независимых переменных (признаков). Целевая переменная принимает конечное число классов или меток.

Может быть:

- Бинарной (классификация на два класса) $Y = \{0,1\}$
- **Многоклассовой (классификация на M непересекающихся классов)** $Y = \{1, \dots, M\}$
- **Многоклассовой (классификация на M классов, которые могут пересекаться)** $Y = \{0, 1\}^M$



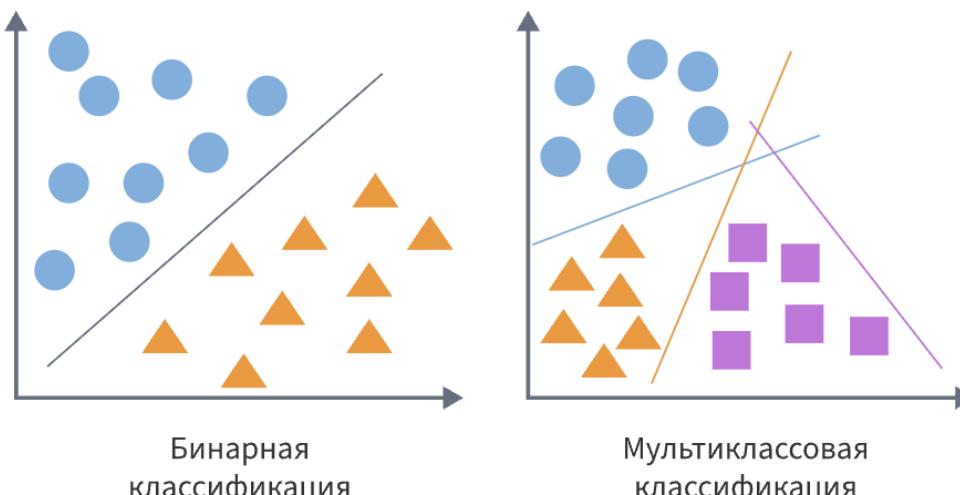
Источник: <https://education.yandex.ru/handbook/ml/article/linear-models>

Многоклассовая классификация

Постановка задачи

Многоклассовой называется тип классификации, в которой объект может относиться к одному из нескольких классов:

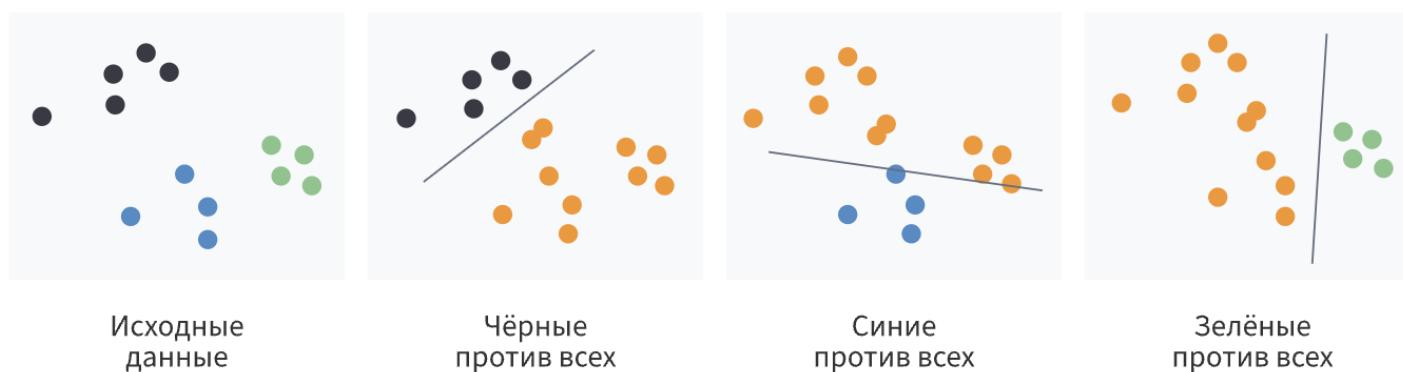
$$y_i \in \{1, \dots, K\}, \quad \text{где } K > 2$$



Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)

Обучим K бинарных классификаторов $b_1(x), \dots, b_k(x)$, каждый из которых решает задачу принадлежности объекта x к классу k_i

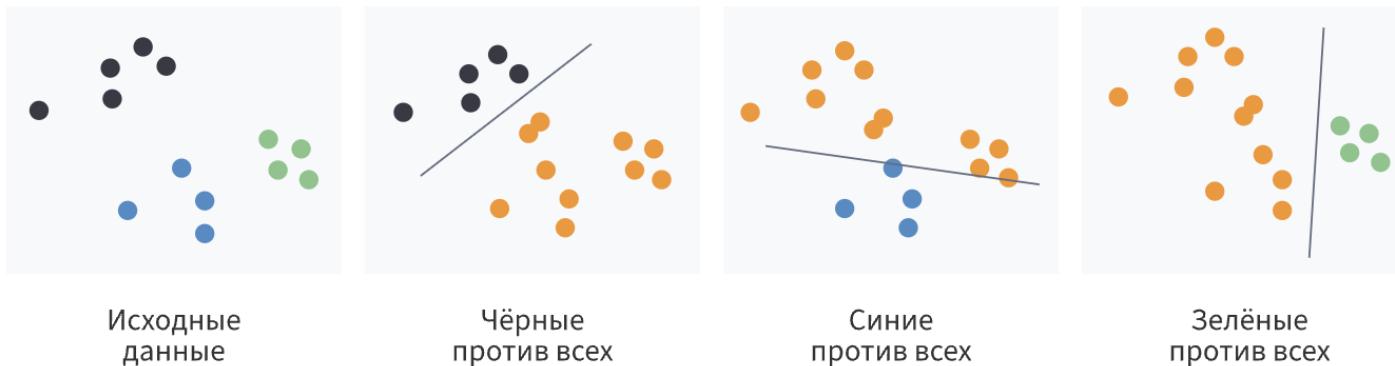


Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)

Обучим K бинарных классификаторов $b_1(x), \dots, b_k(x)$, каждый из которых решает задачу принадлежности объекта x к классу k_i

Например, линейные классификаторы будут иметь вид: $b_k(x) = sign(\beta_k \cdot x)$



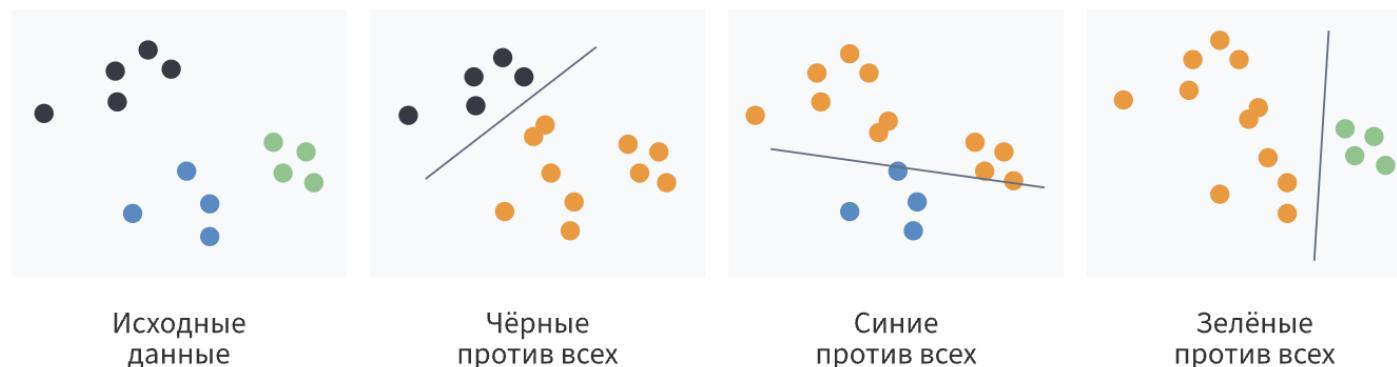
Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)

Обучим K бинарных классификаторов $b_1(x), \dots, b_k(x)$, каждый из которых решает задачу принадлежности объекта x к классу k_i

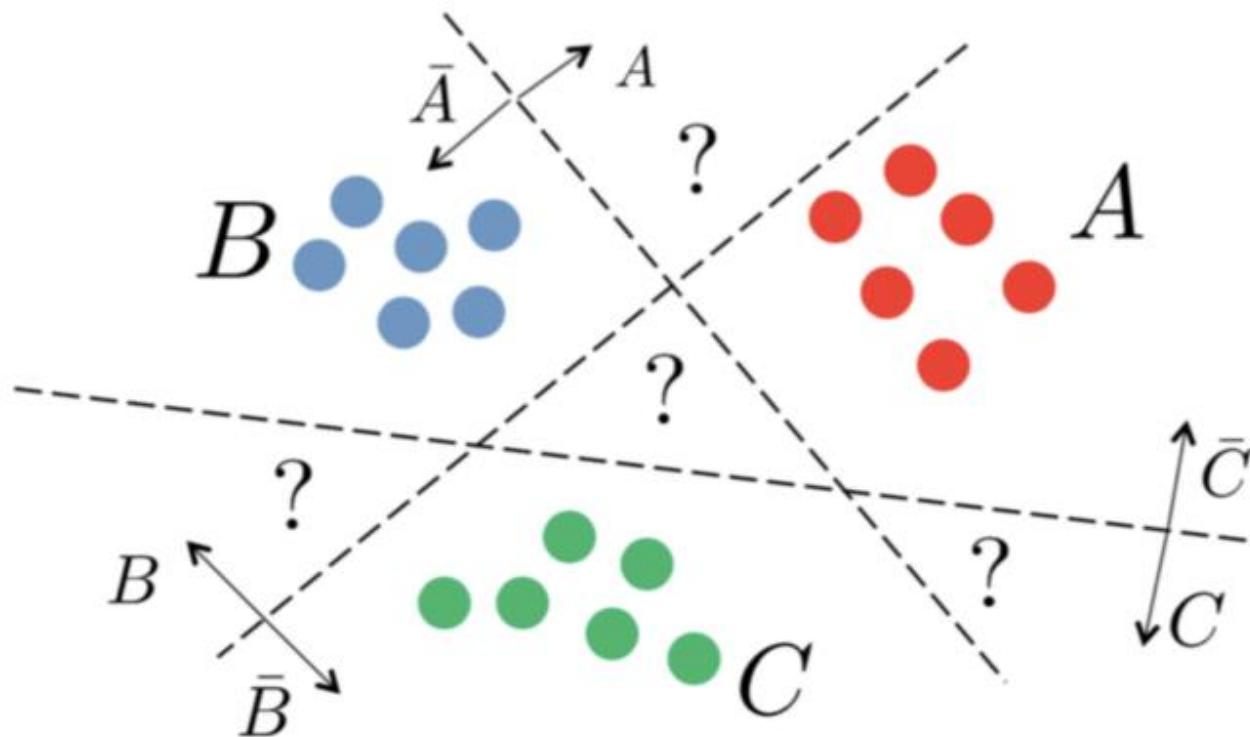
Например, линейные классификаторы будут иметь вид: $b_k(x) = \text{sign}(\beta_k \cdot x)$

Тогда итоговым предсказанием будет предсказание самого уверенного классификатора:
 $f(x) = \text{argmax}_{k \in \{1, \dots, K\}} (\beta_k \cdot x)$



Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)



Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)

Какая может быть проблема у такого подхода?

Многоклассовая классификация

Подход One-vs-All (One-vs-Rest)

Какая может быть проблема у такого подхода?

Классификаторы могут иметь различные масштабы, тогда сравнивать их будет некорректно

Многоклассовая классификация

Подход All-vs-All (One-vs-One)

Для каждой пары классов i и j обучим бинарный классификатор $f_{ij}(x)$, который будет предсказывать класс i или j

Многоклассовая классификация

Подход All-vs-All (One-vs-One)

Для каждой пары классов i и j обучим бинарный классификатор $f_{ij}(x)$, который будет предсказывать класс i или j

При K классах получим C_K^2 классификаторов. Каждый такой классификатор будем обучать только на объектах классов i и j .

Многоклассовая классификация

Подход All-vs-All (One-vs-One)

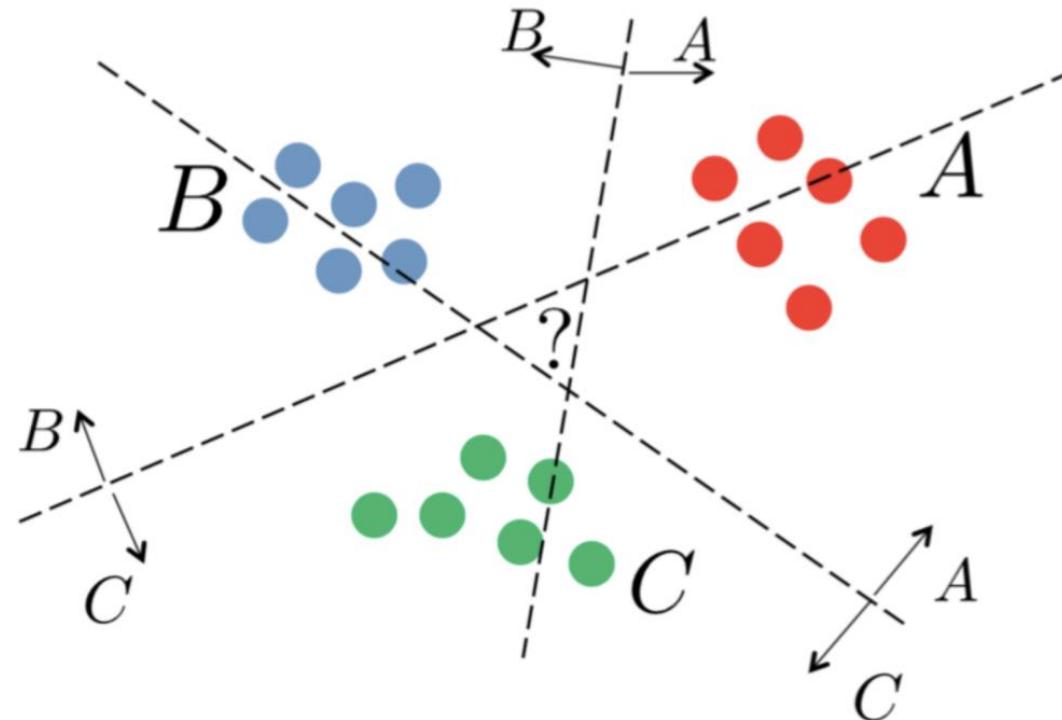
Для каждой пары классов i и j обучим бинарный классификатор $f_{ij}(x)$, который будет предсказывать класс i или j

При K классах получим C_K^2 классификаторов. Каждый такой классификатор будем обучать только на объектах классов i и j .

Итоговым предсказанием будет класс, который предсказало наибольшее число классификаторов: $f(x) = \operatorname{argmax}_{k \in \{1, \dots, K\}} \sum_{i=1}^K \sum_{i \neq j} I[f_{ij}(x) = k]$

Многоклассовая классификация

Подход All-vs-All (One-vs-One)



Многоклассовая классификация

Подход All-vs-All (One-vs-One)

Какая может быть проблема у такого подхода?

Многоклассовая классификация

Подход All-vs-All (One-vs-One)

Какая может быть проблема у такого подхода?

Нужно обучить C_K^2 классификаторов, что при больших K может быть затратно

Многоклассовая классификация

Multiclass vs multilabel-классификация

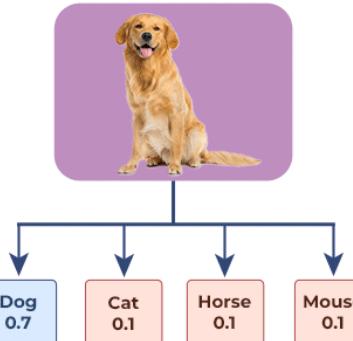
Multiclass — каждый объект может принадлежать только одному классу

Multilabel — каждый объект может принадлежать нескольким классам
(задача с пересекающимися классами)

Mutliclass Classification vs multilabel classification



Multiclass Classification

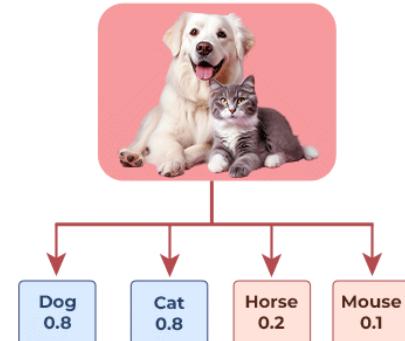


Classes

(pick one class)

- Dog
- Cat
- Horse
- Mouse

Multilabel Classification



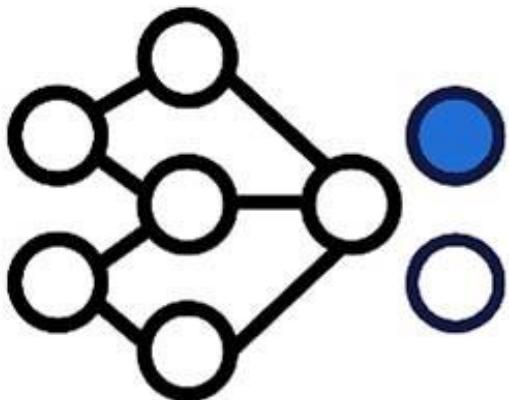
Classes

(pick all the labels present in the image)

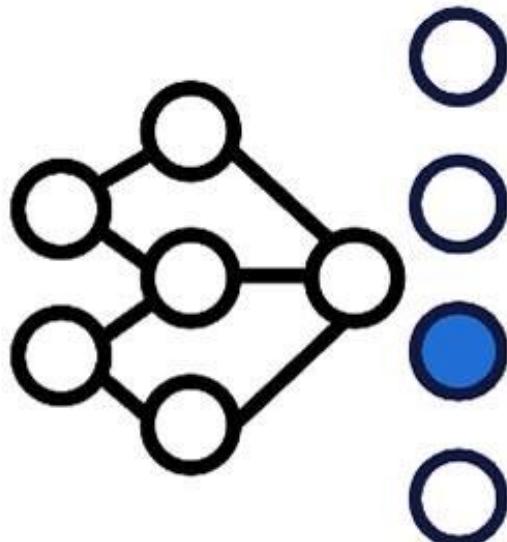
- Dog
- Cat
- Horse
- Mouse

Многоклассовая классификация

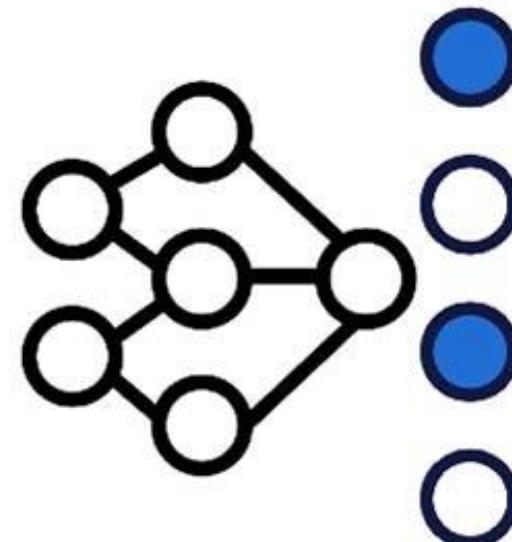
Еще раз о разнице



Binary



Multi-Class



Multi-Label

Метрики качества многоклассовой классификации

Матрица ошибок (confusion matrix)

Пример для трехклассовой классификации

		True/Actual		
		Cat (img alt="Cat icon" data-bbox="428 555 518 605")/	Fish (img alt="Fish icon" data-bbox="578 555 668 605")/	Hen (img alt="Hen icon" data-bbox="728 555 818 605")/
Predicted	Cat (img alt="Cat icon" data-bbox="238 645 328 695")/	4	6	3
	Fish (img alt="Fish icon" data-bbox="238 755 328 805")/	1	2	0
	Hen (img alt="Hen icon" data-bbox="238 855 328 905")/	1	2	6

Метрики качества многоклассовой классификации

Усреднение

Для подсчета качества работы алгоритма на всех классах применяют различные способы усреднения качеств работы на каждом из классов:

- Макро-усреднение (macro-average)
- Микро-усреднение (micro-average)
- Взвешенное усреднение (weighted-average)

Метрики качества многоклассовой классификации

Макро-усреднение

Вычисляется значение выбранной метрики для каждого бинарного классификатора.

Например:

$$\text{Macro - accuracy} = \frac{\text{accuracy}_1 + \dots + \text{accuracy}_k}{K}$$

$$\text{Macro - precision} = \frac{\text{precision}_1 + \dots + \text{precision}_k}{K}$$

Метрики качества многоклассовой классификации

Макро-усреднение

Посчитаем *macro – precision*:

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

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

Метрики качества многоклассовой классификации

Макро-усреднение

Посчитаем *macro – precision*:

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

$$\text{Precision(cat)} = \frac{4}{4 + 6 + 3} = \frac{4}{13}$$

$$\text{Precision(fish)} = \frac{2}{2 + 1 + 0} = \frac{2}{3}$$

$$\text{Precision(hen)} = \frac{6}{6 + 2 + 1} = \frac{2}{3}$$

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

Метрики качества многоклассовой классификации

Макро-усреднение

Посчитаем *macro – precision*:

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

$$\text{Precision(cat)} = \frac{4}{4 + 6 + 3} = \frac{4}{13}$$

$$\text{Precision(fish)} = \frac{2}{2 + 1 + 0} = \frac{2}{3}$$

$$\text{Precision(hen)} = \frac{6}{6 + 2 + 1} = \frac{2}{3}$$

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

$$\text{macro – precision} = \frac{\text{Precision(cat)} + \text{Precision(fish)} + \text{Precision(hen)}}{3} \approx 0,55$$

Метрики качества многоклассовой классификации

Микро-усреднение

Вычисляются значения TP, TN, FP и FN по всей матрице ошибок сразу, исходя из их определения, после чего вычисляем метрику.

Например:

$$\text{Micro} - \text{precision} = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K (TP_k + FP_k)}$$

$$\text{Micro} - \text{recall} = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K (TP_k + FN_k)}$$

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем *micro – precision*:

		True/Actual		
		Cat (img alt="Cat icon" data-bbox="295 555 325 585))	Fish (img alt="Fish icon" data-bbox="355 555 385 585))	Hen (img alt="Hen icon" data-bbox="415 555 445 585))
Predicted	Cat (img alt="Cat icon" data-bbox="105 625 135 655))	4	6	3
	Fish (img alt="Fish icon" data-bbox="105 685 135 715))	1	2	0
	Hen (img alt="Hen icon" data-bbox="105 745 135 775))	1	2	6

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем *micro – precision*:

		True/Actual		
		Cat (img alt="Cat icon" data-bbox="238 561 278 591"/>)	Fish (img alt="Fish icon" data-bbox="338 561 378 591")	Hen (img alt="Hen icon" data-bbox="438 561 478 591"/>)
Predicted	Cat (img alt="Cat icon" data-bbox="118 621 158 651"/>)	4	6	3
	Fish (img alt="Fish icon" data-bbox="118 681 158 711"/>)	1	2	0
	Hen (img alt="Hen icon" data-bbox="118 741 158 771"/>)	1	2	6

TP — количество верно угаданных

объектов положительного класса

FP — количество всех неверных

предсказаний

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем *micro – precision*:

		True/Actual		
		Cat (img alt="Cat icon" data-bbox="235 555 275 595"/>)	Fish (img alt="Fish icon" data-bbox="335 555 375 595")	Hen (img alt="Hen icon" data-bbox="435 555 475 595"/>)
Predicted	Cat (img alt="Cat icon" data-bbox="115 625 155 665"/>)	4	6	3
	Fish (img alt="Fish icon" data-bbox="115 685 155 725"/>)	1	2	0
	Hen (img alt="Hen icon" data-bbox="115 745 155 785"/>)	1	2	6

Источник: <https://shmueli.medium.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>

TP — количество верно угаданных

объектов положительного класса

FP — количество всех неверных

предсказаний

$$\text{micro} - \text{precision} = \frac{4 + 2 + 6}{6 + 3 + 1 + 0 + 1 + 2} = \frac{12}{25}$$

Метрики качества многоклассовой классификации

Взвешенное усреднение

Усредняются посчитанные для каждого класса метрики с весами, пропорциональными количеству объектов класса

Например:

$$\text{Weighted precision} = \frac{n_1}{N} \cdot \text{precision}_1 + \cdots + \frac{n_k}{N} \cdot \text{precision}_K$$

$$\text{Weighted recall} = \frac{n_1}{N} \cdot \text{recall}_1 + \cdots + \frac{n_k}{N} \cdot \text{recall}_K$$

Метрики качества многоклассовой классификации

Взвешенное усреднение

Посчитаем *weighted precision*:

		True/Actual		
		Cat (😺)	Fish (🐠)	Hen (🐓)
Predicted		Cat (😺)	4	6
		Fish (🐠)	1	2
Hen (🐓)	1	2	6	

Метрики качества многоклассовой классификации

Взвешенное усреднение

Посчитаем *weighted precision*:

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

$$\text{weighted precision} = \frac{6}{25} \cdot \text{precision(cat)} + \frac{10}{25} \cdot \text{precision(fish)} + \frac{9}{25} \cdot \text{precision(hen)} \approx 0,43$$