SVM. Многоклассовая классификация

Паточенко Евгений

НИУ ВШЭ

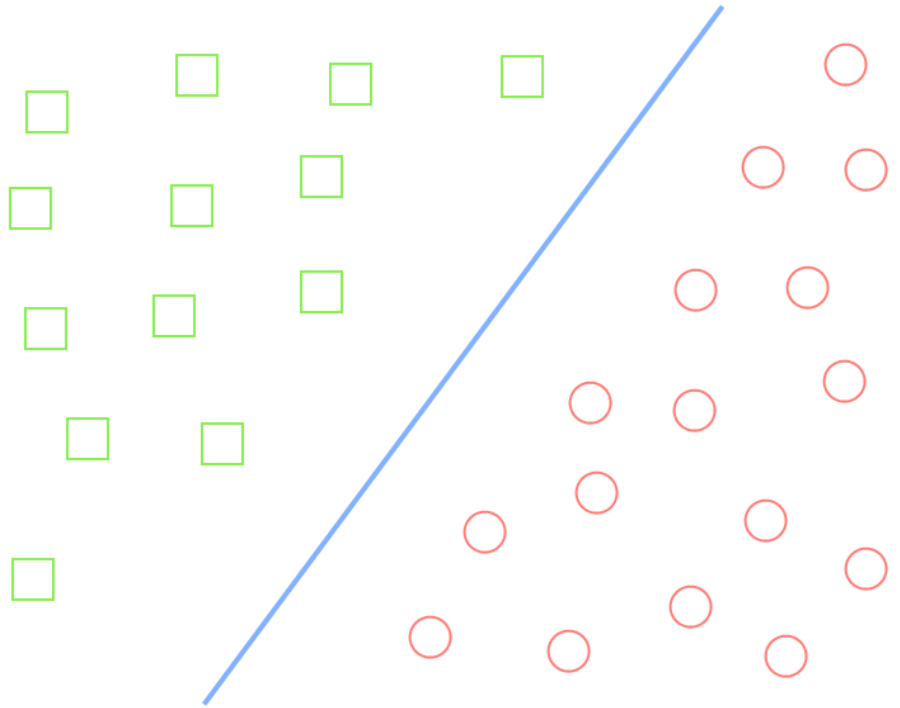
План занятия

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

Метод опорных векторов (SVM)

Линейно разделимая выборка

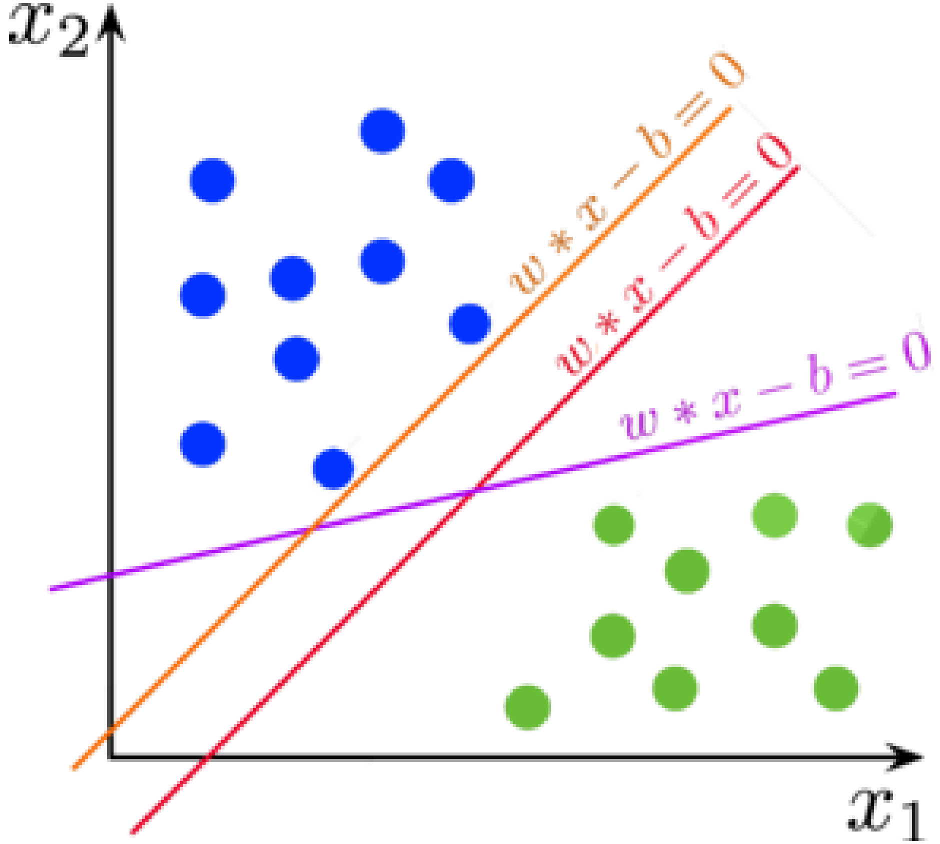
Выборка линейно разделима, если существует такой вектор параметров β, что соответствующий классификатор $f(x)$ не допускает ошибок на этой выборке



Метод опорных векторов (SVM)

Линейно разделимая выборка

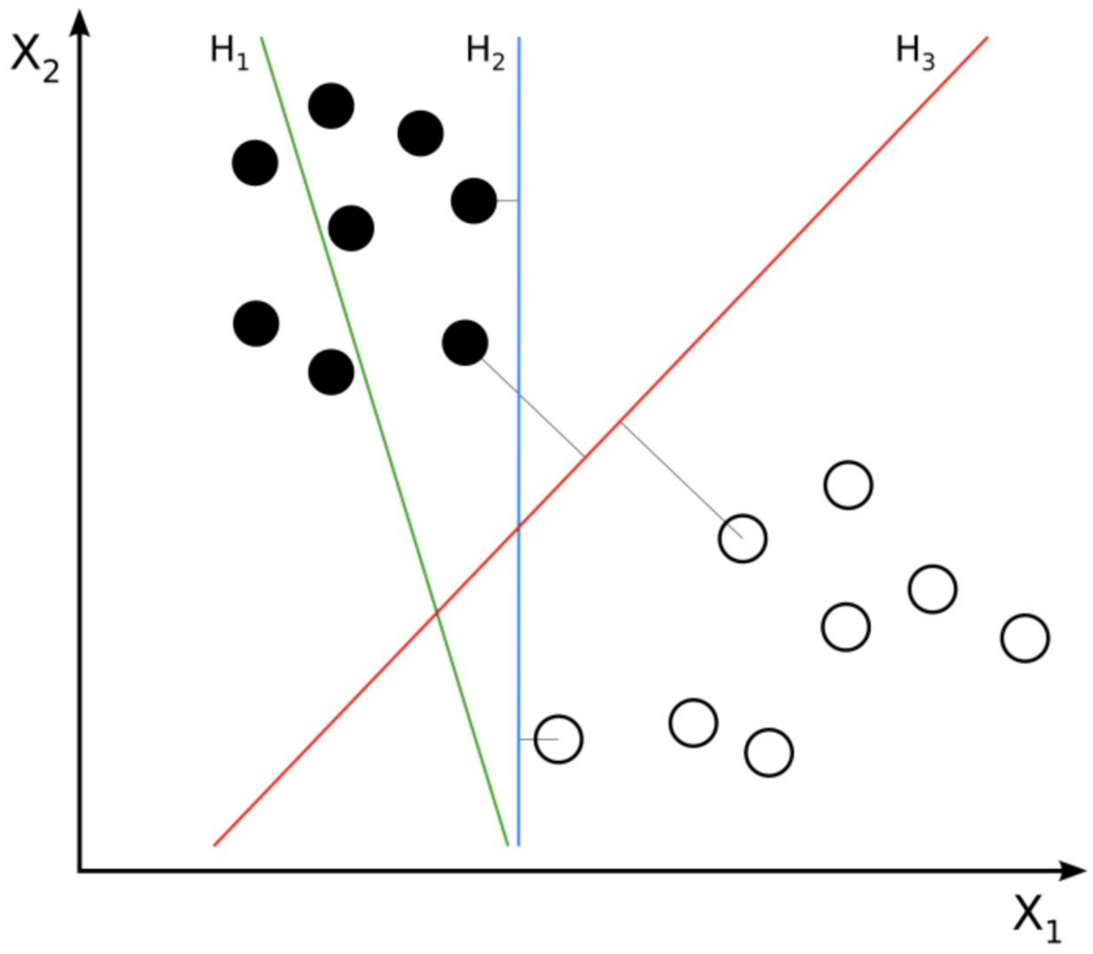
Для двух линейно разделимых классов возможны различные варианты построения разделяющей гиперплоскости



Метод опорных векторов (SVM)

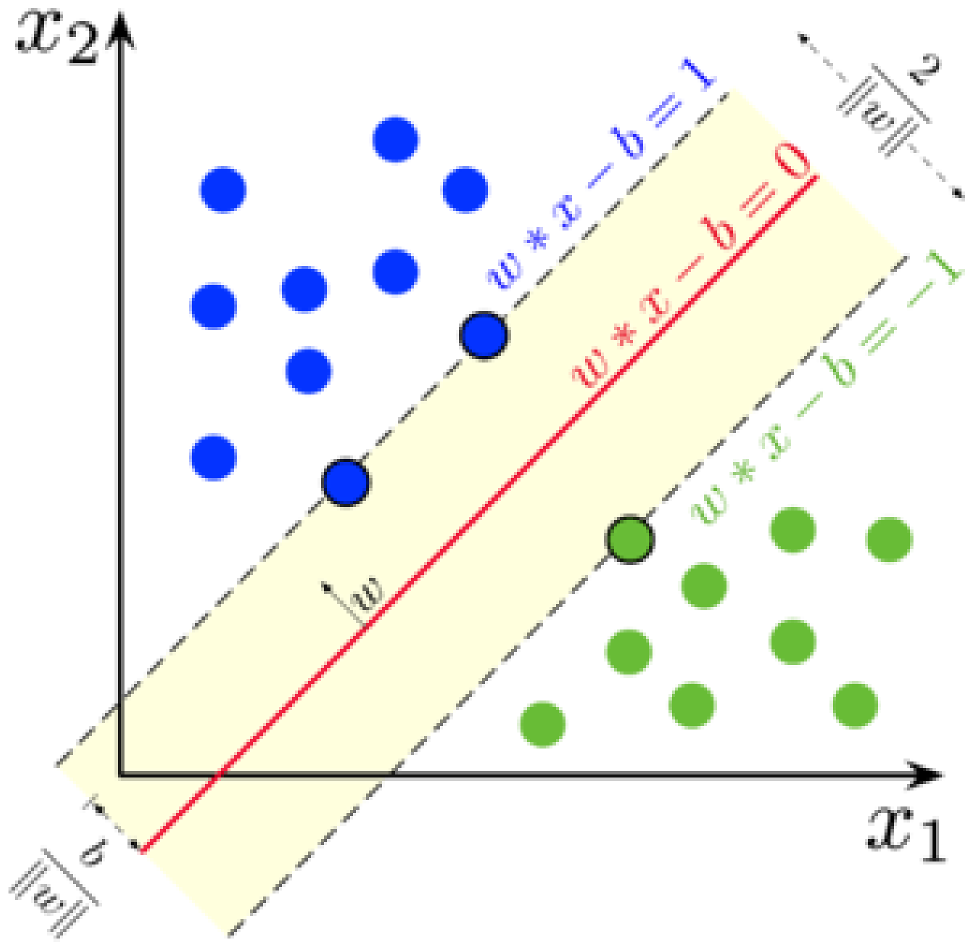
Критерий «хорошей» гиперплоскости

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



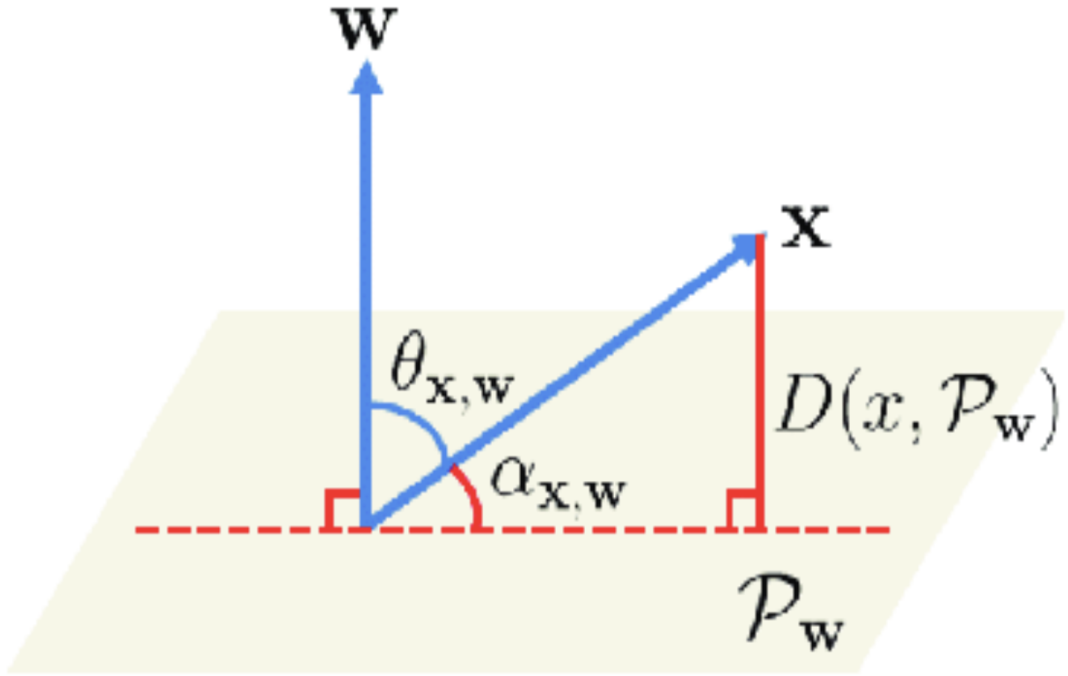
Цель SVM — максимизировать ширину разделяющей полосы, то есть найти наилучшую разделяющую гиперплоскость, которая максимально отдалена от ближайших точек каждого класса (**опорных векторов**)

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 $sign$:

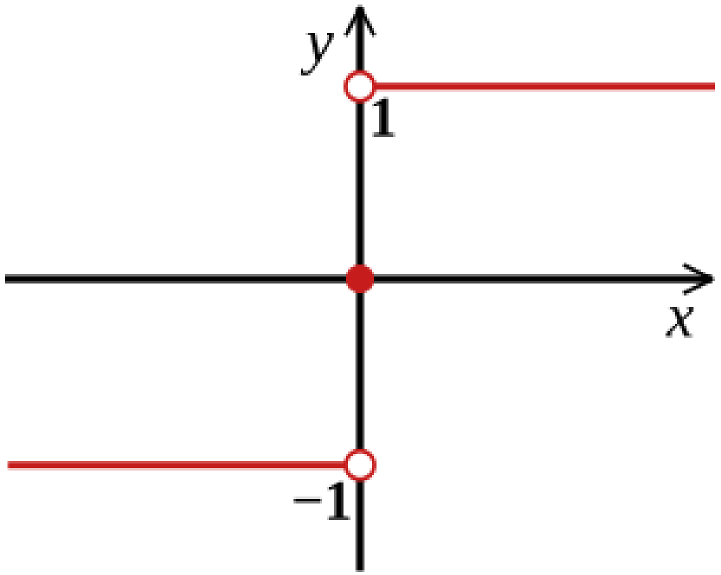
$sign(x \cdot y) = sign\, x \cdot sign\, y$, тогда для ответов $a(x)$ классификатора следует, что

$$a(x) = sign\big((w, x)\big) = sign\big((\widehat{w}, x) + b\big)$$

$$= sign\big((k\widehat{w}, x) + kb\big) = sign\Big(k\big((\widehat{w}, x) + b\big)\Big)$$

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

$$= sign\big((\widehat{w}, x) + b\big)$$



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:

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

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

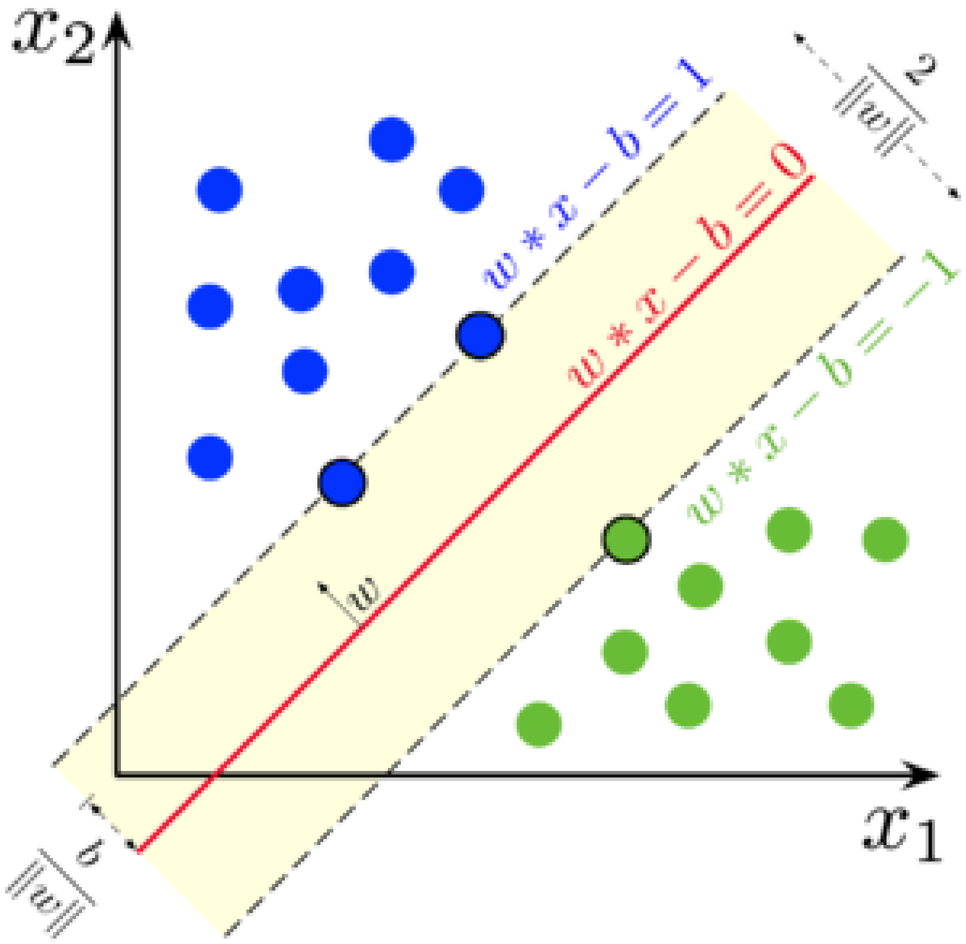
$$p(x^*,H)=min_{x\in X}\frac{|(\widehat{w},x)+b|}{|\widehat{w}|}=\frac{1}{\widehat{w}}min_{x\in X}|(\widehat{w},x)+b|=\frac{1}{\widehat{|w|}}$$

Метод опорных векторов (SVM)

Расстояние от точки до гиперплоскости

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

Ее нам и предстоит максимизировать

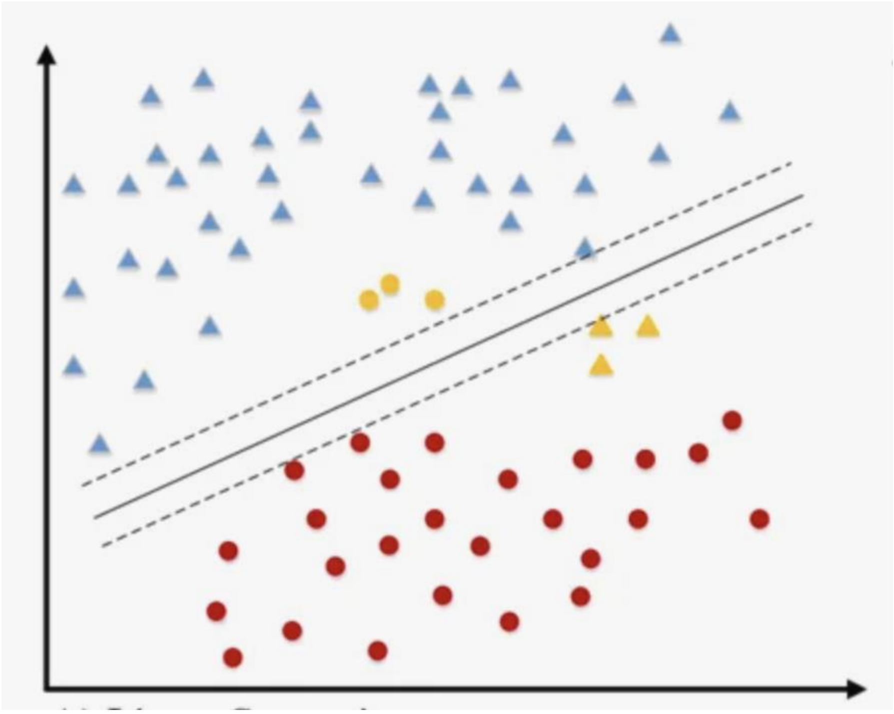


Метод опорных векторов (SVM)

Линейно неразделимая выборка

В большинстве случаев мы имеем дело с линейно неразделимыми данными

Объекты могут попадать на другую сторону гиперплоскости или внутрь отступа

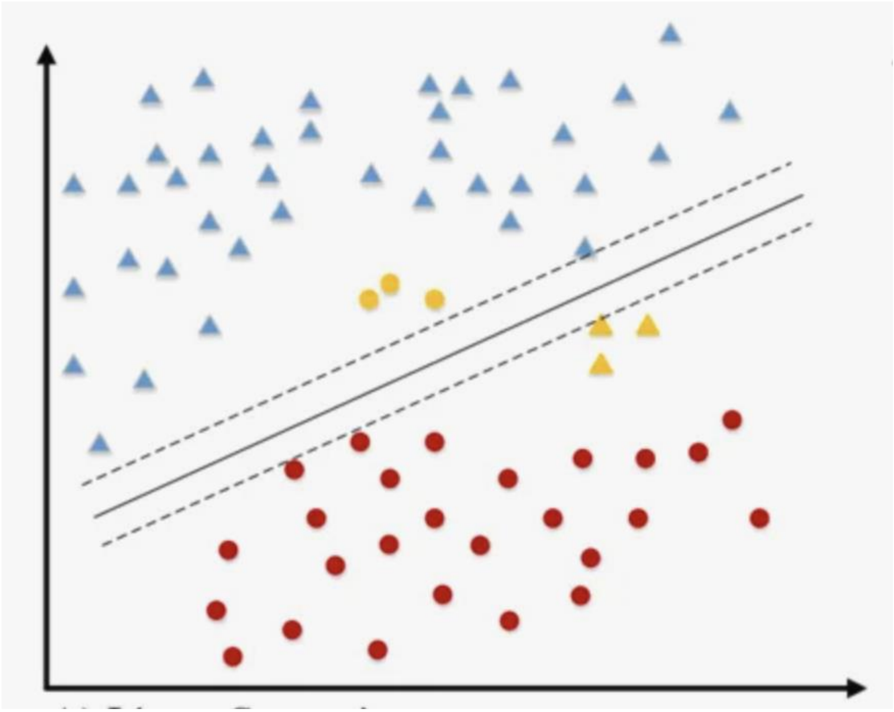


Метод опорных векторов (SVM)

Линейно неразделимая выборка

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

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



Тогда:

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

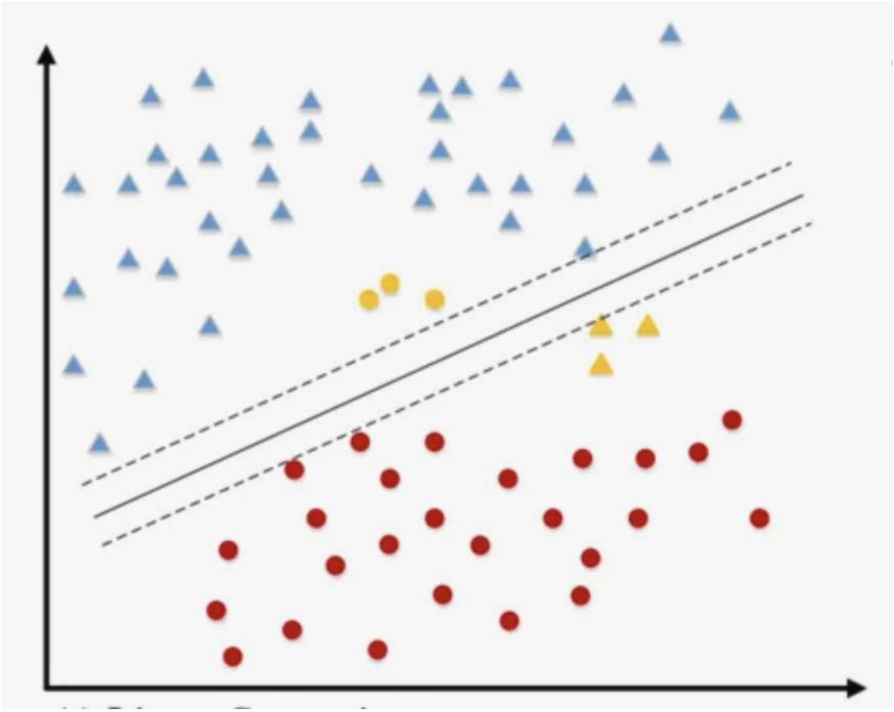
$M_i(w) \geq 1 + \xi_i$

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



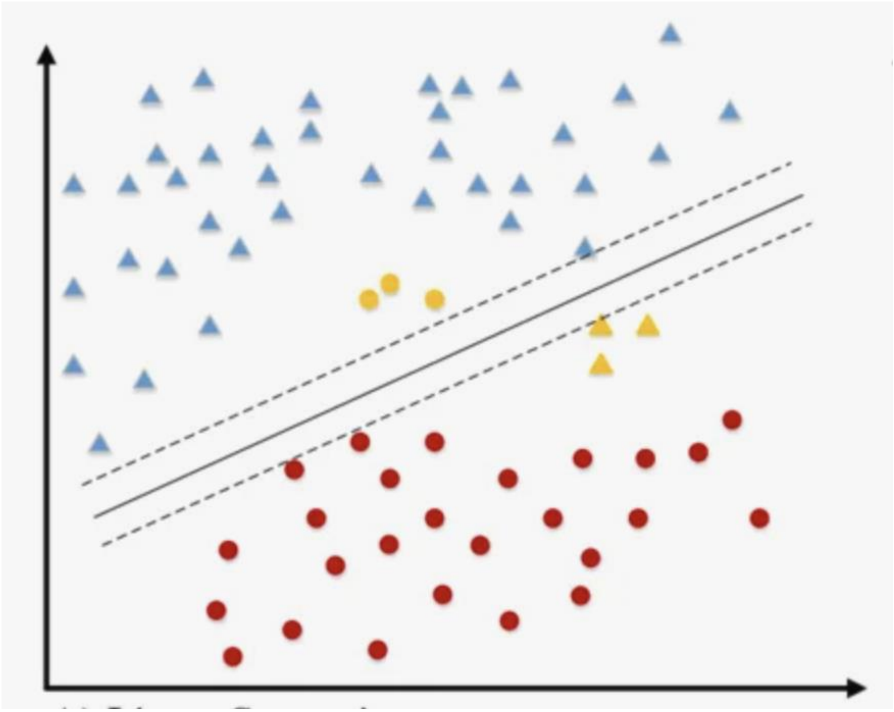
Метод опорных векторов (SVM)

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)

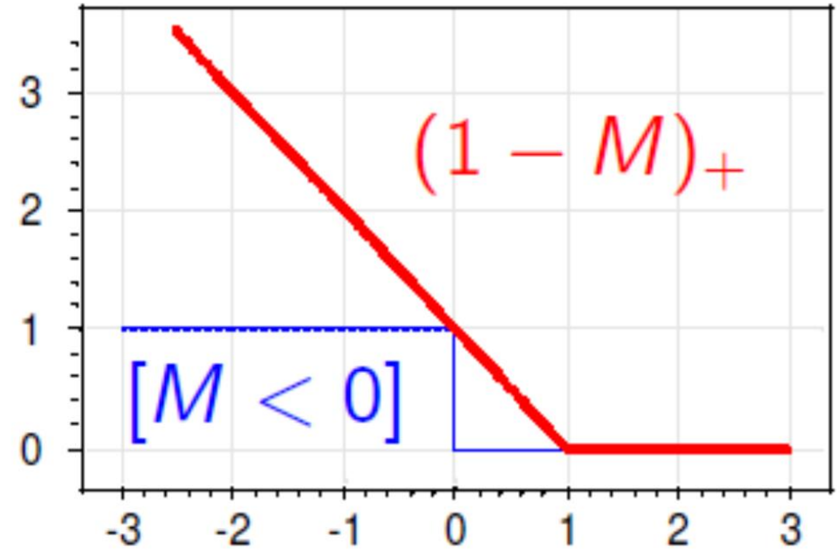
Оптимизация

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:

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



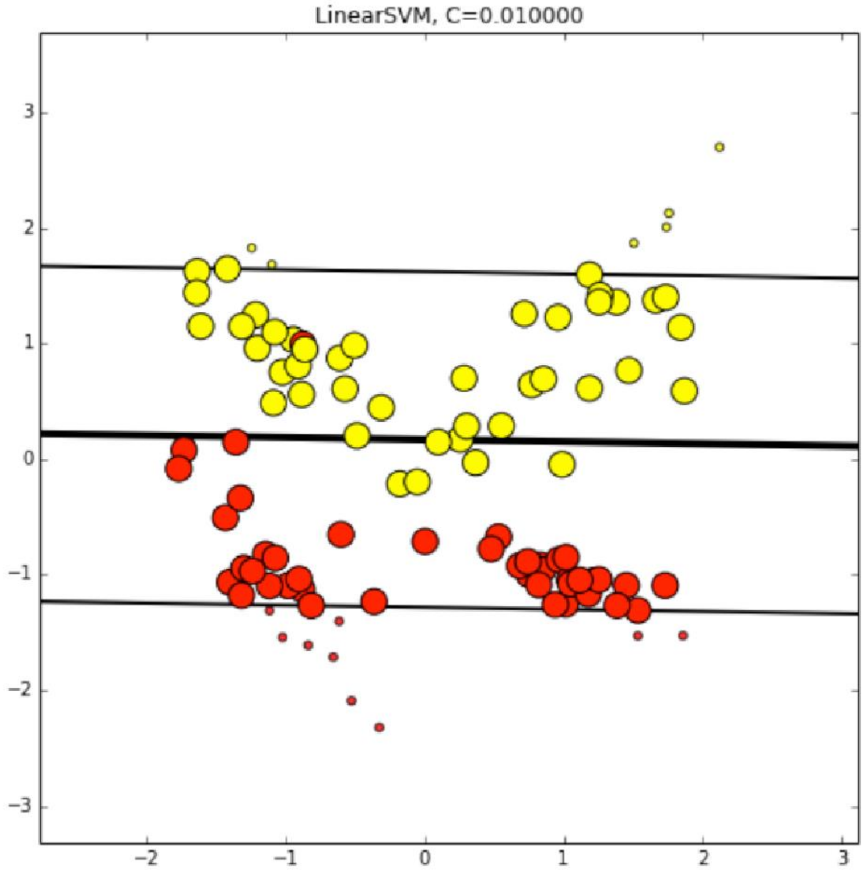
Константа C является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки

Метод опорных векторов (SVM)

Значение константы C

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

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

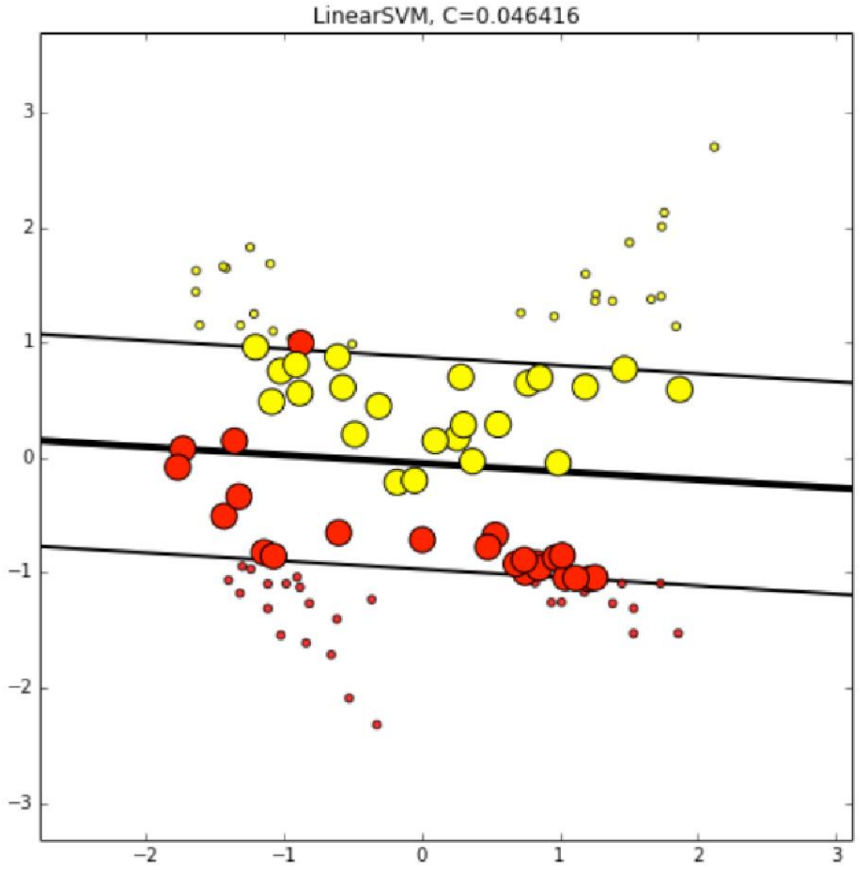


Метод опорных векторов (SVM)

Значение константы C

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

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

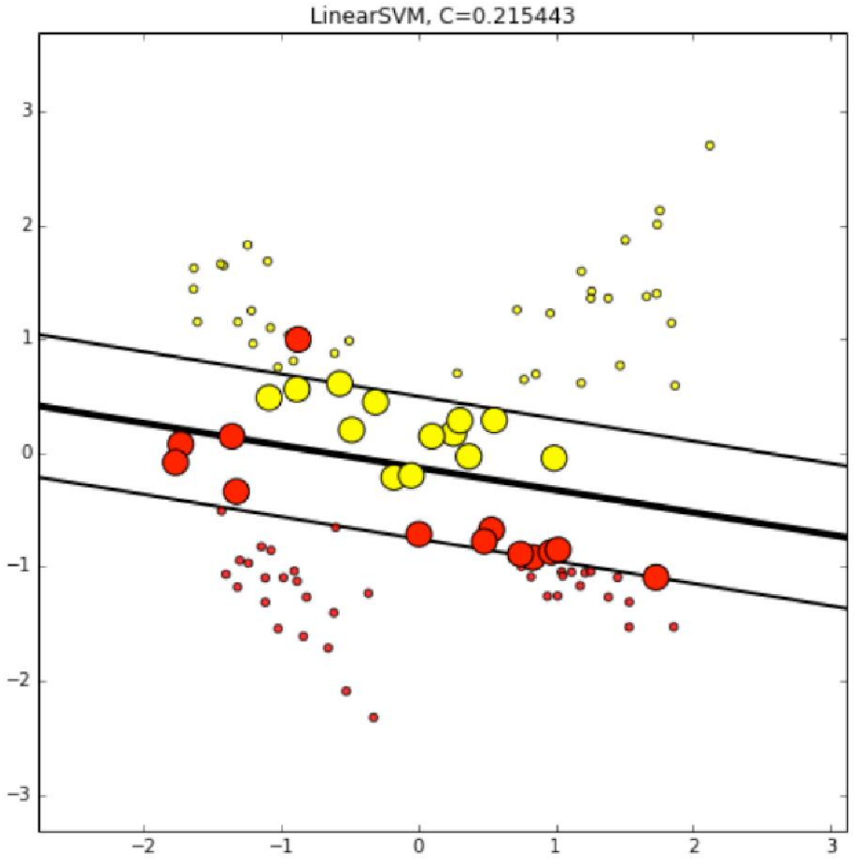


Метод опорных векторов (SVM)

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

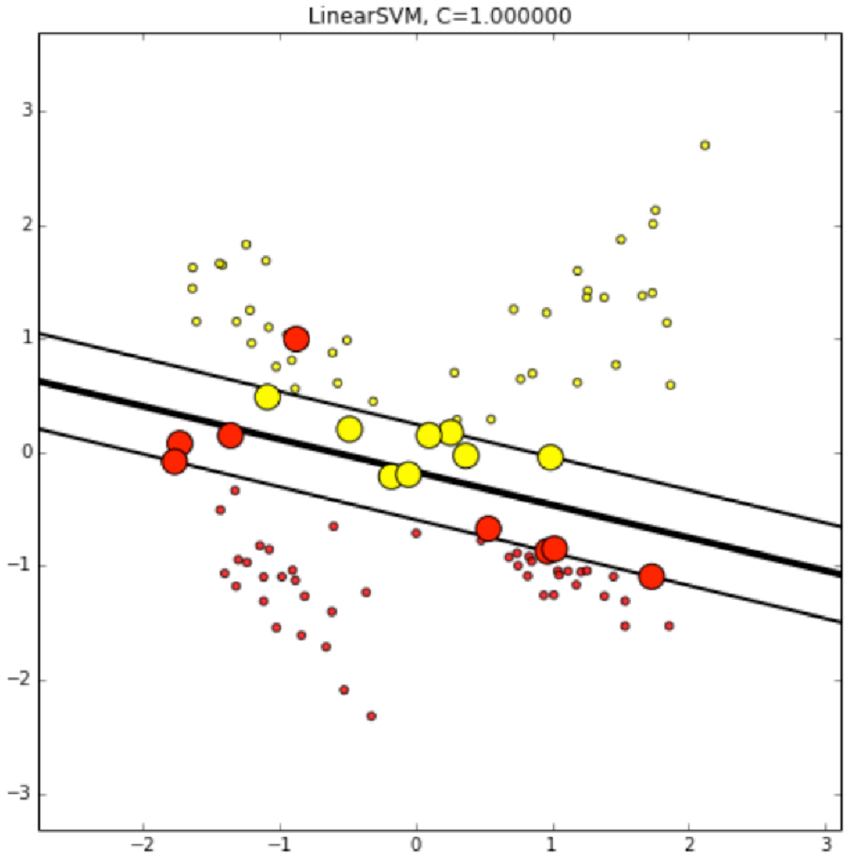


Метод опорных векторов (SVM)

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

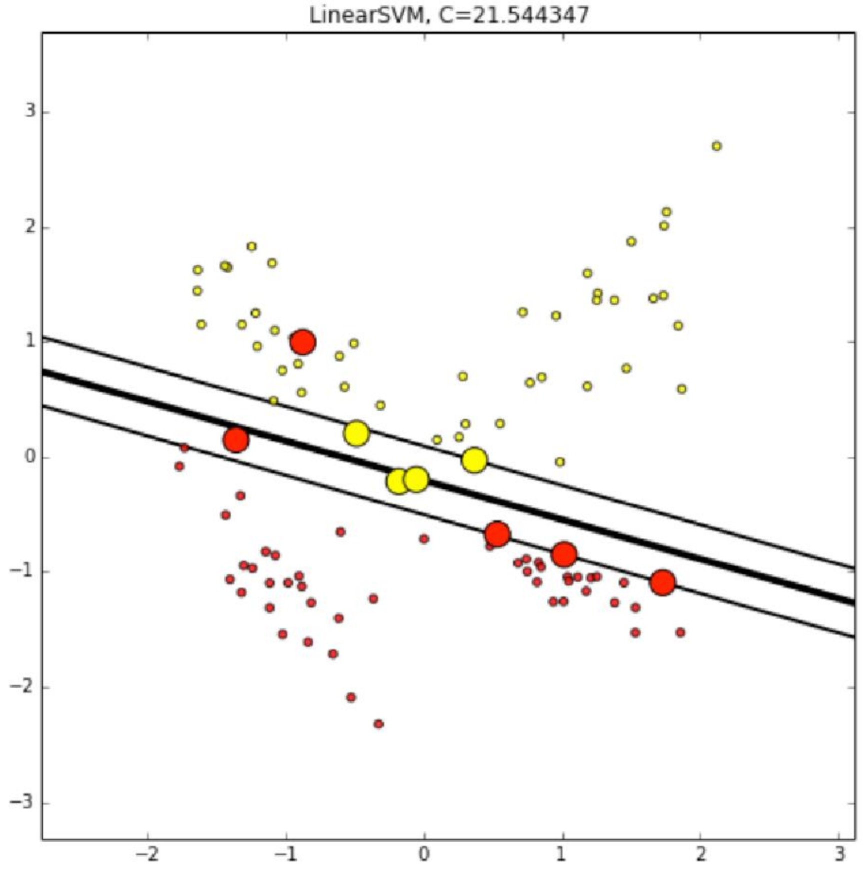


Метод опорных векторов (SVM)

Значение константы C

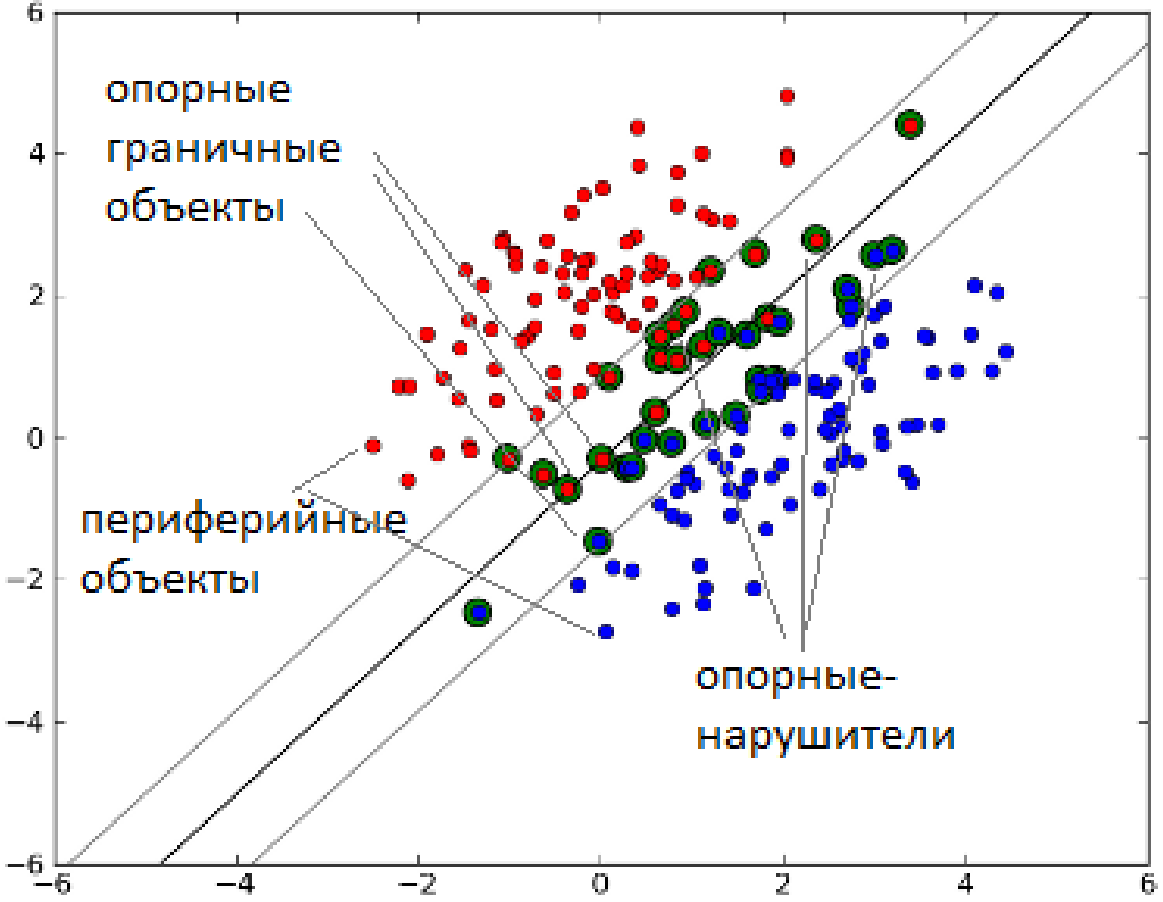
$$Q(f,x) = \sum_{i=1}^{l} [\max(0,1-y_i(w,x_i))] + \frac{1}{2C}|w|^2 \to min_w$$

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

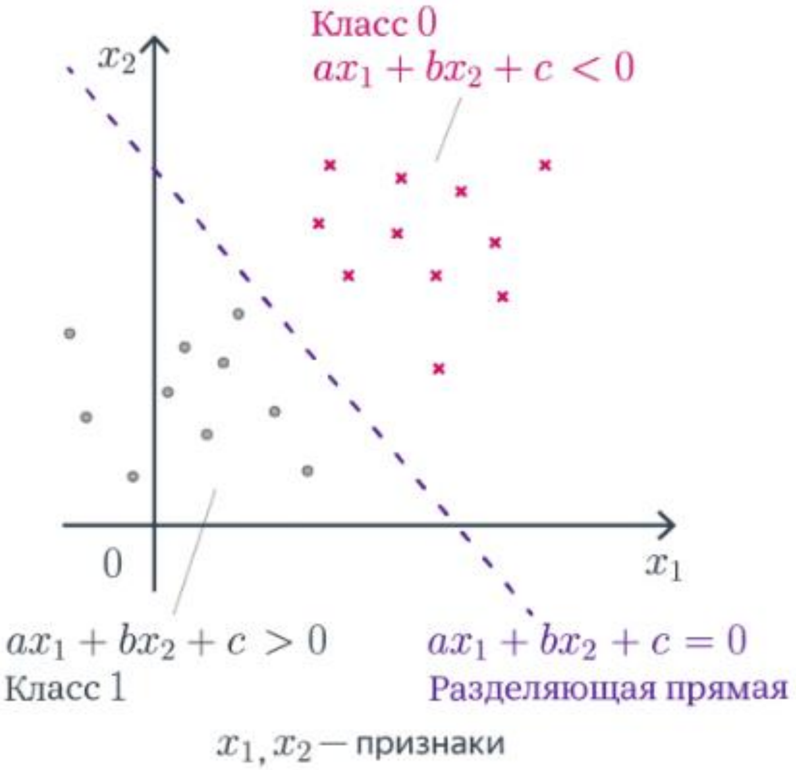
Модель машинного обучения, используемая для прогнозирования категориальной (дискретной) целевой переменной на основе одной или нескольких независимых переменных (признаков). Целевая переменная принимает конечное число классов или меток.

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) $Y = \{0,1\}^M$



Линейные методы классификации (повтор)

Классификация

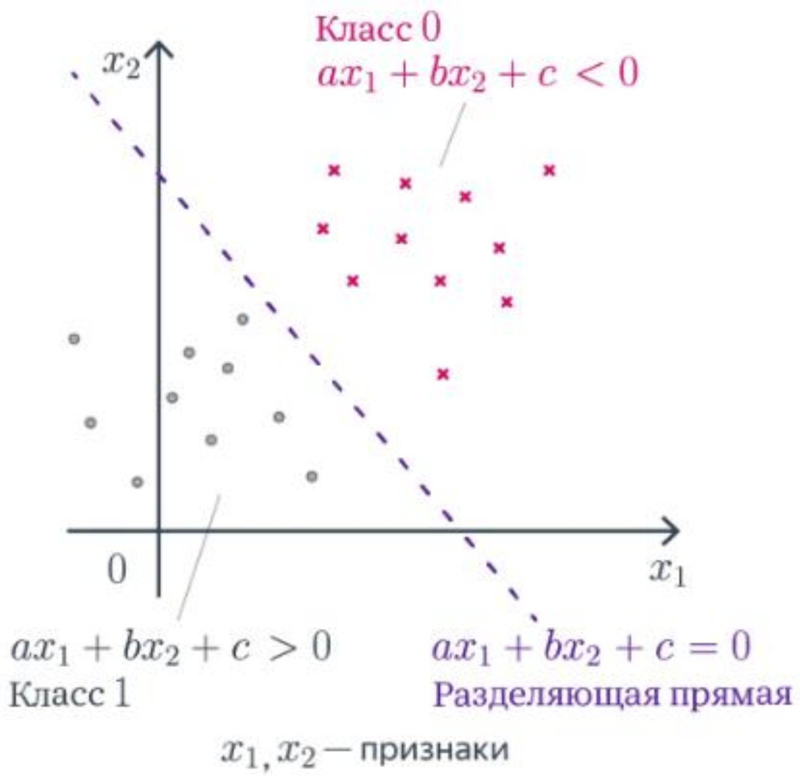
Модель машинного обучения, используемая для прогнозирования категориальной (дискретной) целевой переменной на основе одной или нескольких независимых переменных (признаков). Целевая переменная принимает конечное число классов или меток.

Может быть:

— Бинарной (классификация на два класса) $Y = \{0,1\}$

— Многоклассовой (классификация на M непересекающихся классов) $\boldsymbol{Y = \{1, \dots, M\}}$

— Многоклассовой (классификация на M классов, которые могут пересекаться) $\boldsymbol{Y = \{0, 1\}^M}$

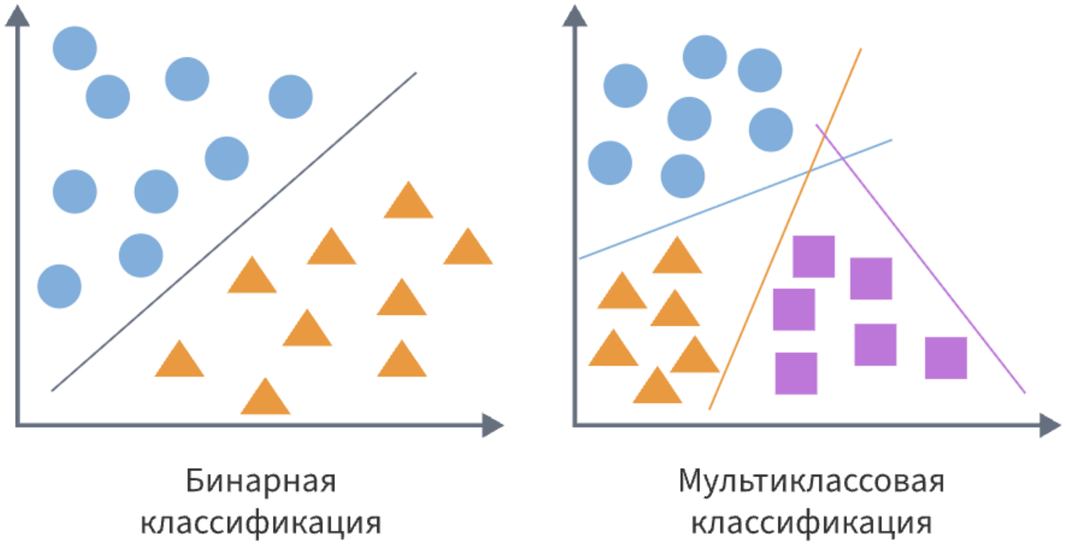


Многоклассовая классификация

Постановка задачи

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

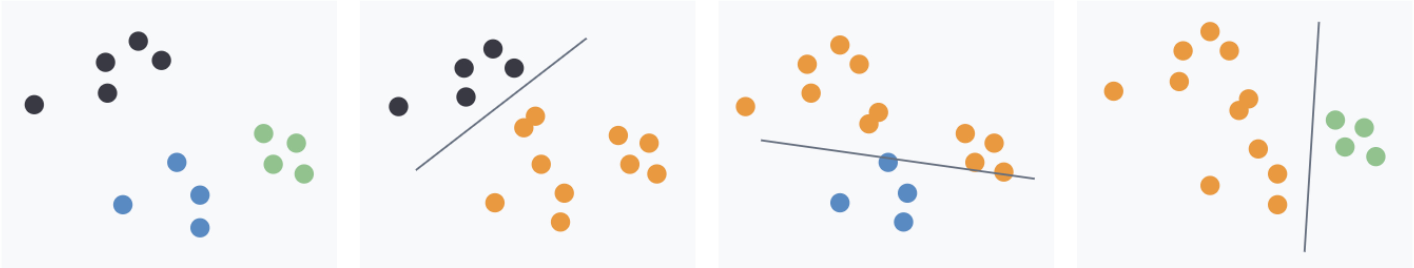
$$y_i \in \{1, \dots, K\}, \qquad \text{где } K > 2$$



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



Исходные данные

Чёрные против всех

Синие против всех

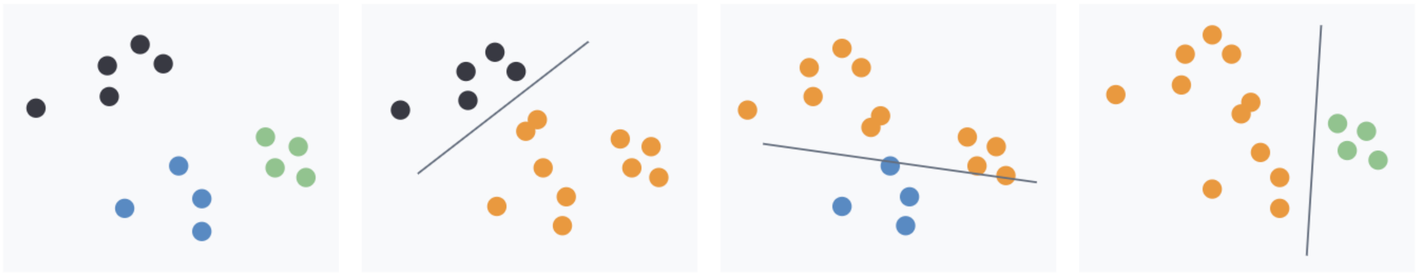
Зелёные против всех

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: $b_k(x) = sign(\beta_k \cdot x)$



Исходные данные | Чёрные против всех | Синие против всех | Зелёные против всех

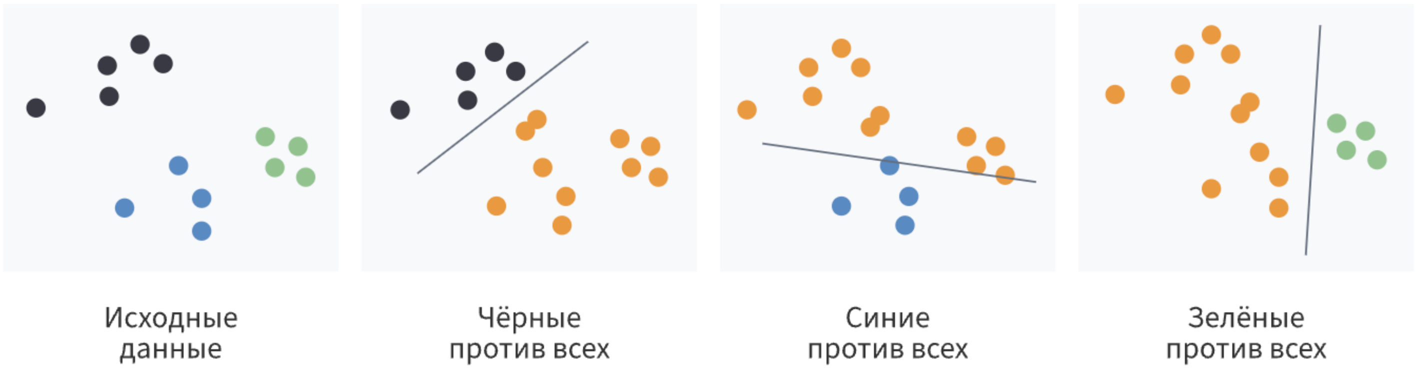
Многоклассовая классификация

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

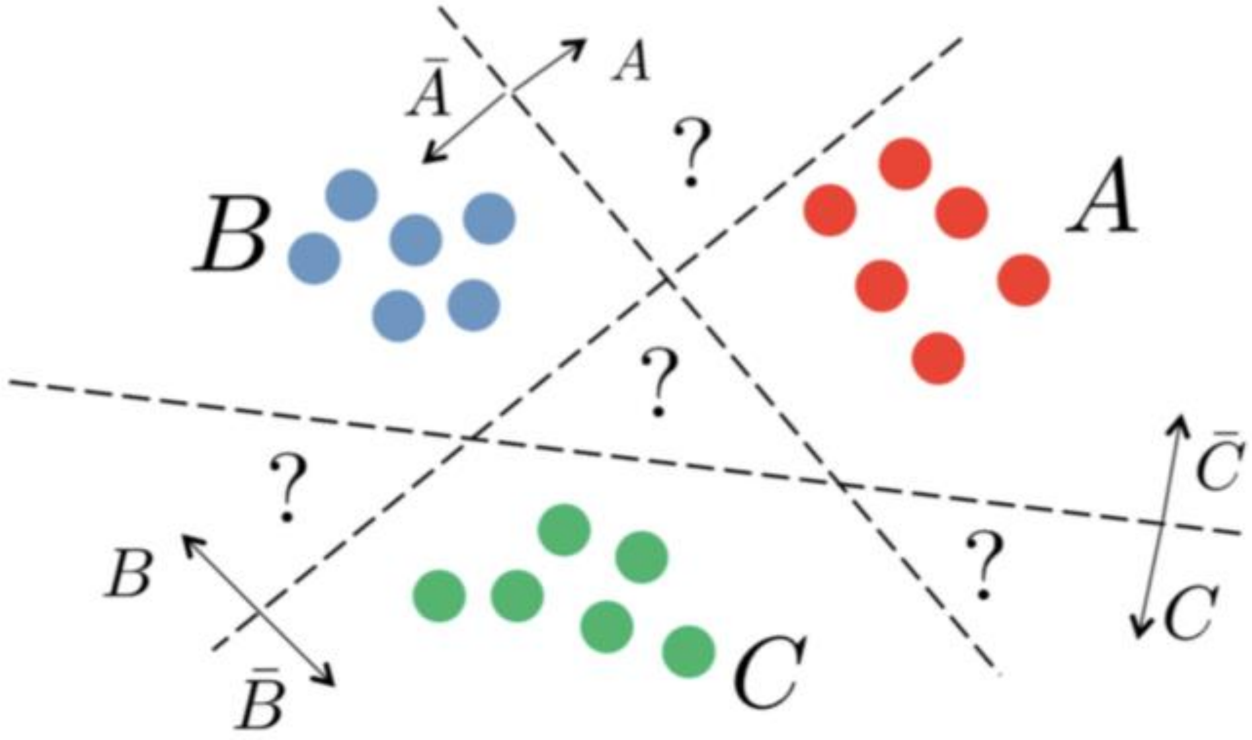
Например, линейные классификаторы будут иметь вид: $b_k(x) = sign(\beta_k \cdot x)$

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



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)

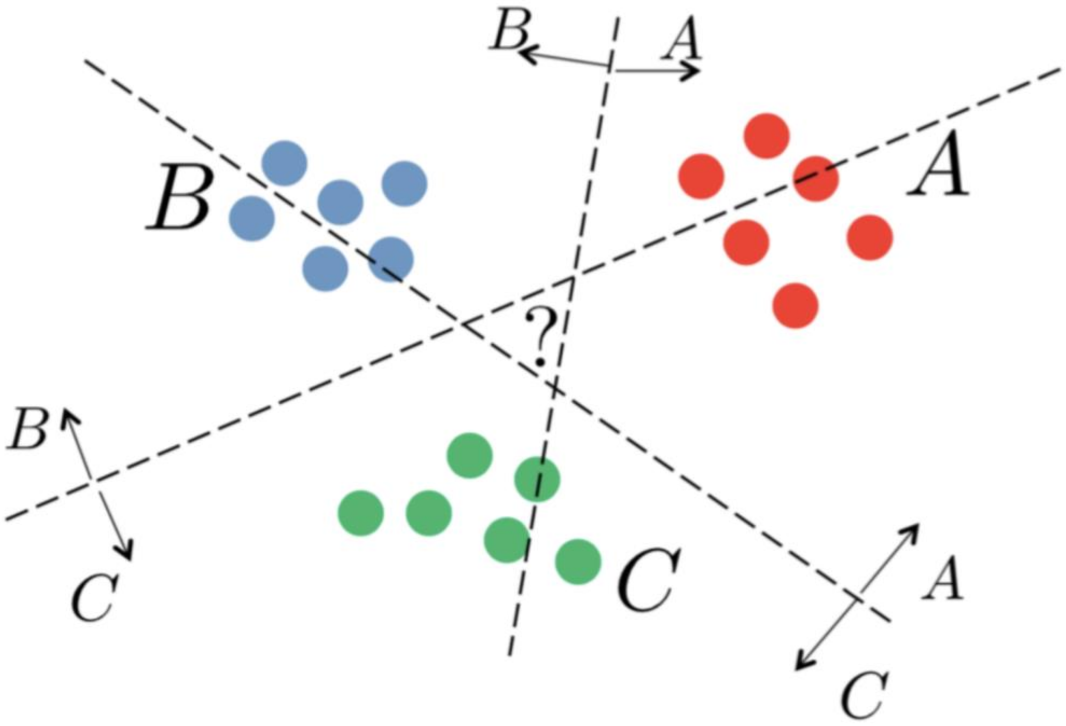
Для каждой пары классов i и j обучим бинарный классификатор $f_{ij}(x)$, который будет предсказывать класс i или j

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: $f(x) = argmax_{k\in\{1,\ldots,K\}} \sum_{i=1}^{K} \sum_{i\neq j} I\left[f_{ij}(x) = k\right]$

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?

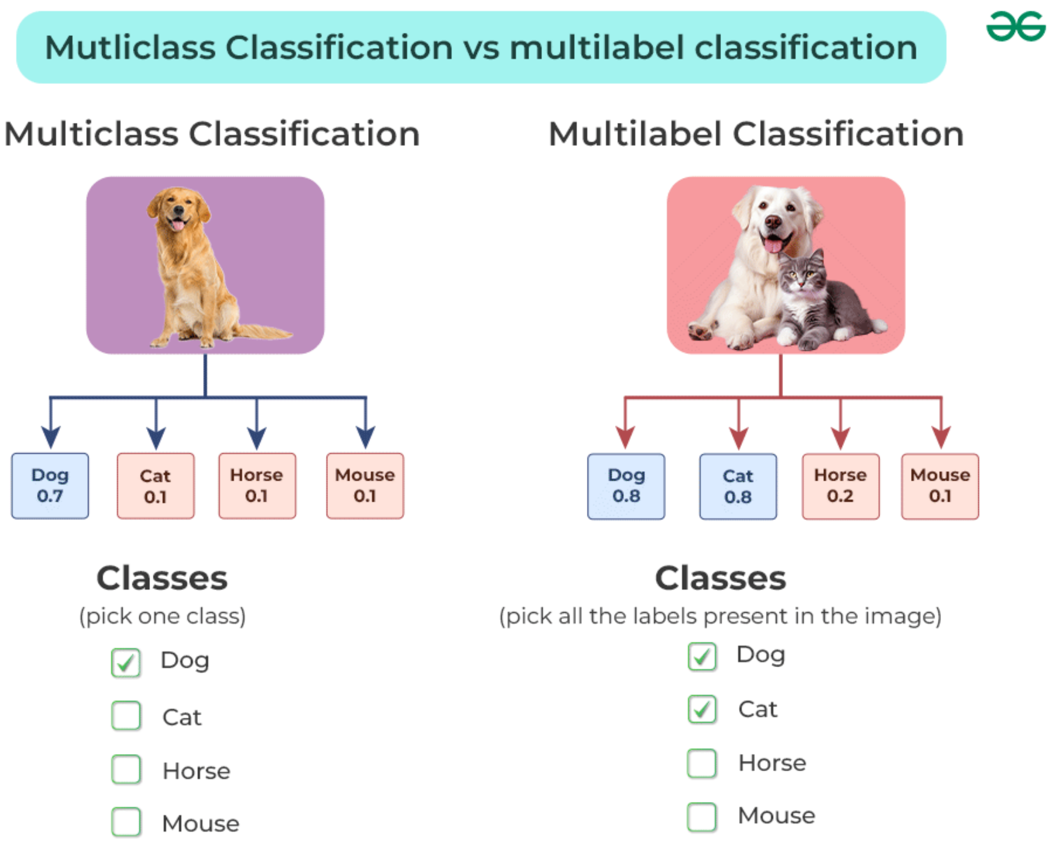
Нужно обучить C_K^2 классификаторов, что при больших K может быть затратно

Многоклассовая классификация

Multiclass vs multilabel-классификация

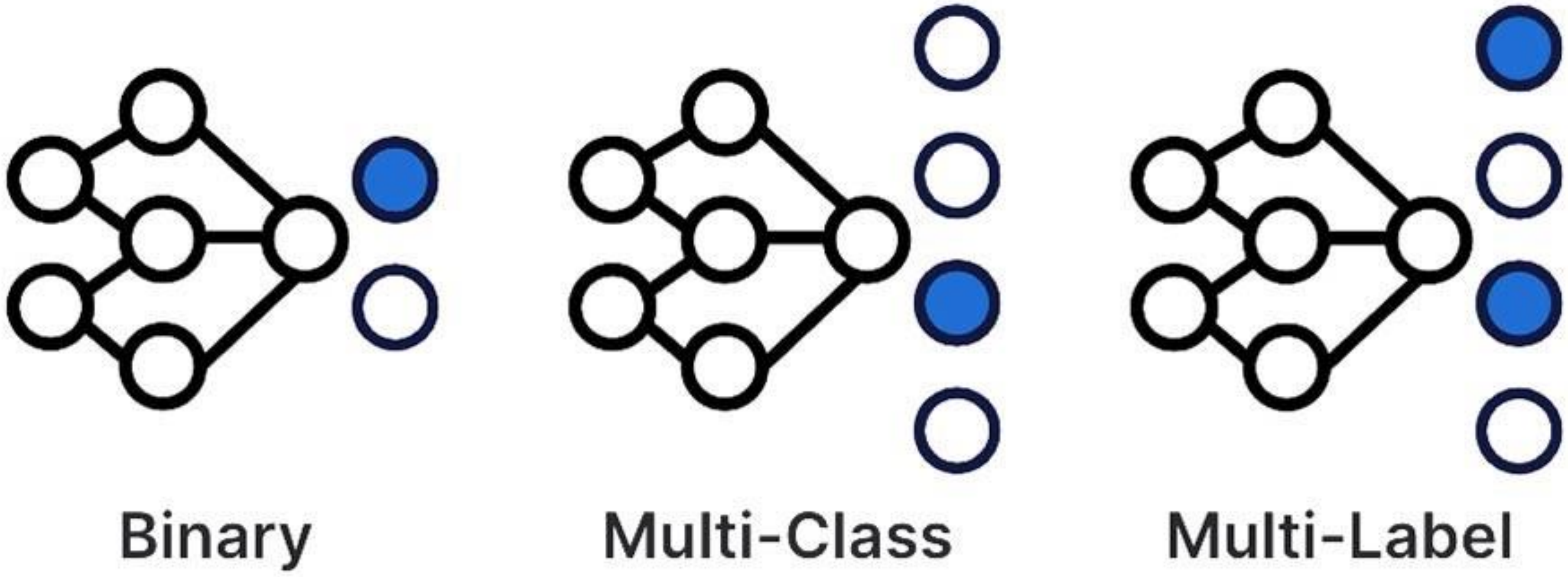
Multiclass — каждый объект может принадлежать только одному классу

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



Многоклассовая классификация

Еще раз о разнице



Метрики качества многоклассовой классификации

Матрица ошибок (confusion matrix)

Пример для трехклассовой классификации

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

$$Macro - accuracy = \frac{accuracy_1 + \cdots + accuracy_k}{K}$$

$$Macro - precision = \frac{precision_1 + \cdots + precision_k}{K}$$

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

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

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

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

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

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

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

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

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

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

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:

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

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

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем $micro-precision$:

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

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем $micro - precision$:

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

TP — количество верно угаданных объектов положительного класса

FP — количество всех неверных предсказаний

Метрики качества многоклассовой классификации

Микро-усреднение

Посчитаем $micro - precision$:

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

TP — количество верно угаданных объектов положительного класса

FP — количество всех неверных предсказаний

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

Метрики качества многоклассовой классификации

Взвешенное усреднение

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

Например:

$$Weighted\ precision = \frac{n_1}{N} \cdot precision_1 + \cdots + \frac{n_k}{N} \cdot precision_K$$

$$Weighted\ recall = \frac{n_1}{N} \cdot recall_1 + \cdots + \frac{n_k}{N} \cdot recall_K$$

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

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

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

Спасибо за внимание!