

Лекция 5. Линейные модели классификации. Часть 2

Максим Карпов



ОБУЧЕНИЕ ЛИНЕЙНОЙ РЕГРЕССИИ (НАПОМИНАНИЕ)

Обучающая выборка:

пусть x – объект (x_1, x_2, \dots, x_l - его признаки), а y – ответ на объекте (произвольное число), n – количество объектов.

Модель линейной регрессии:

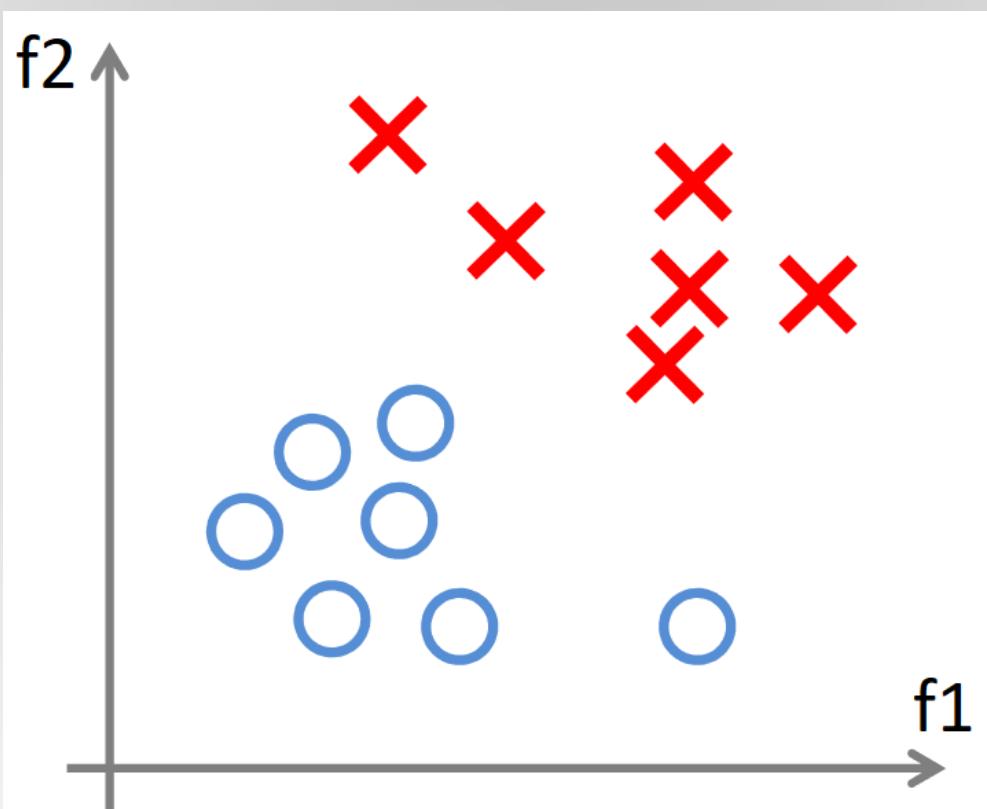
$$a(x, w) = \sum_{i=1}^l w_j x_j$$

- Метод обучения – метод наименьших квадратов
(минимизируем разность между предсказанием и правильным ответом):

$$Q(w) = \sum_{i=1}^n (a(x_i, w) - y_i)^2 \rightarrow \min_w$$

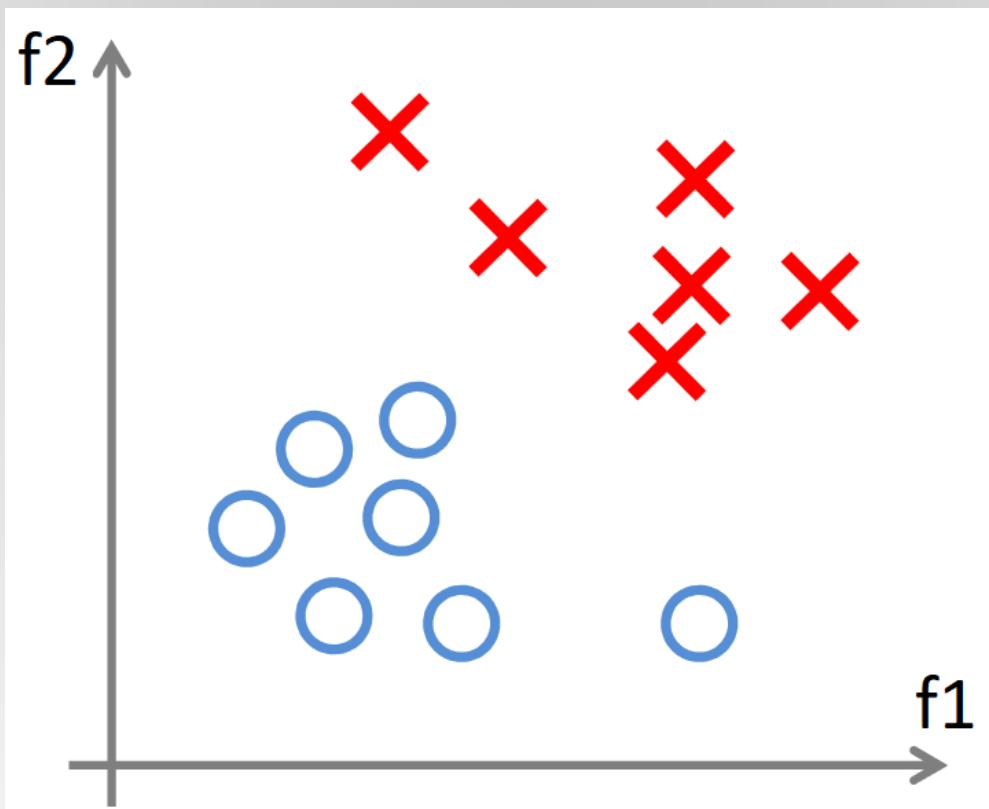
БИНАРНАЯ КЛАССИФИКАЦИЯ

y_1, y_2, \dots, y_n - ответы (+1 или -1).



БИНАРНАЯ КЛАССИФИКАЦИЯ

y_1, y_2, \dots, y_n - ответы (+1 или -1).



Как выглядит модель линейного классификатора: $a(x, w) = ?$

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \textcolor{red}{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \text{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

- если $\sum_{j=1}^l w_j x_j > 0$, то $\text{sign}(\sum_{j=1}^l w_j x_j) = +1$, то есть объект отнесён к положительному классу
- если $\sum_{j=1}^l w_j x_j < 0$, то $\text{sign}(\sum_{j=1}^l w_j x_j) = -1$, то есть объект отнесён к отрицательному классу

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \text{sign} \left(\sum_{j=1}^l w_j x_j \right)$$

- если $\sum_{j=1}^l w_j x_j > 0$, то $\text{sign}(\sum_{j=1}^l w_j x_j) = +1$, то есть объект отнесён к положительному классу
- если $\sum_{j=1}^l w_j x_j < 0$, то $\text{sign}(\sum_{j=1}^l w_j x_j) = -1$, то есть объект отнесён к отрицательному классу
- значит, $\sum_{j=1}^l w_j x_j = 0$ – уравнение разделяющей границы между классами. Это уравнение плоскости (или прямой в двумерном случае), поэтому классификатор является линейным.

БИНАРНАЯ КЛАССИФИКАЦИЯ

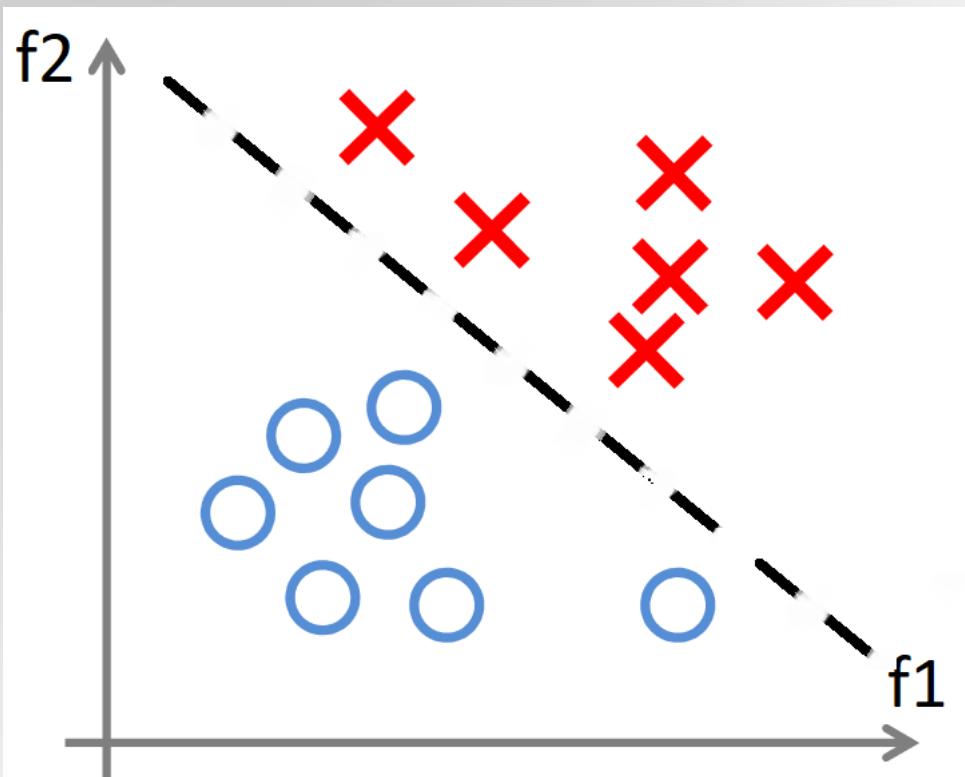
Модель линейного классификатора:

$$a(x, w) = \text{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

Уравнение

$$\sum_{j=1}^l w_j x_j = 0$$

– уравнение плоскости
(или прямой).



ОБУЧЕНИЕ КЛАССИФИКАТОРА

Как обучить линейный классификатор?

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min,$$

где $[a(x_i) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(x_i) \neq y_i$, и 0 иначе.

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min (*),$$

где $[a(x_i) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(x_i) \neq y_i$, и 0 иначе.

- Обозначим $M_i = y_i \cdot (w, x_i)$ - **отступ** на i -м объекте.

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min (*),$$

где $[a(x_i) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(x_i) \neq y_i$, и 0 иначе.

- Обозначим $M_i = y_i \cdot (w, x_i)$ - **отступ** на i -м объекте.

Утверждение. Решение задачи (*) эквивалентно решению задачи

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случай неверной классификации (предсказание не совпадает с правильным ответом):

- Если $(w, x) > 0$ (то есть объект отнесён к классу +1), а $y = -1$, то $M = y \cdot (w, x) < 0$.

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случай неверной классификации (предсказание не совпадает с правильным ответом):

- Если $(w, x) > 0$ (то есть объект отнесён к классу +1), а $y = -1$, то $M = y \cdot (w, x) < 0$.
- Аналогично, если $(w, x) < 0$, а $y = +1$, то $M = y \cdot (w, x) < 0$.

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случай неверной классификации (предсказание не совпадает с правильным ответом):

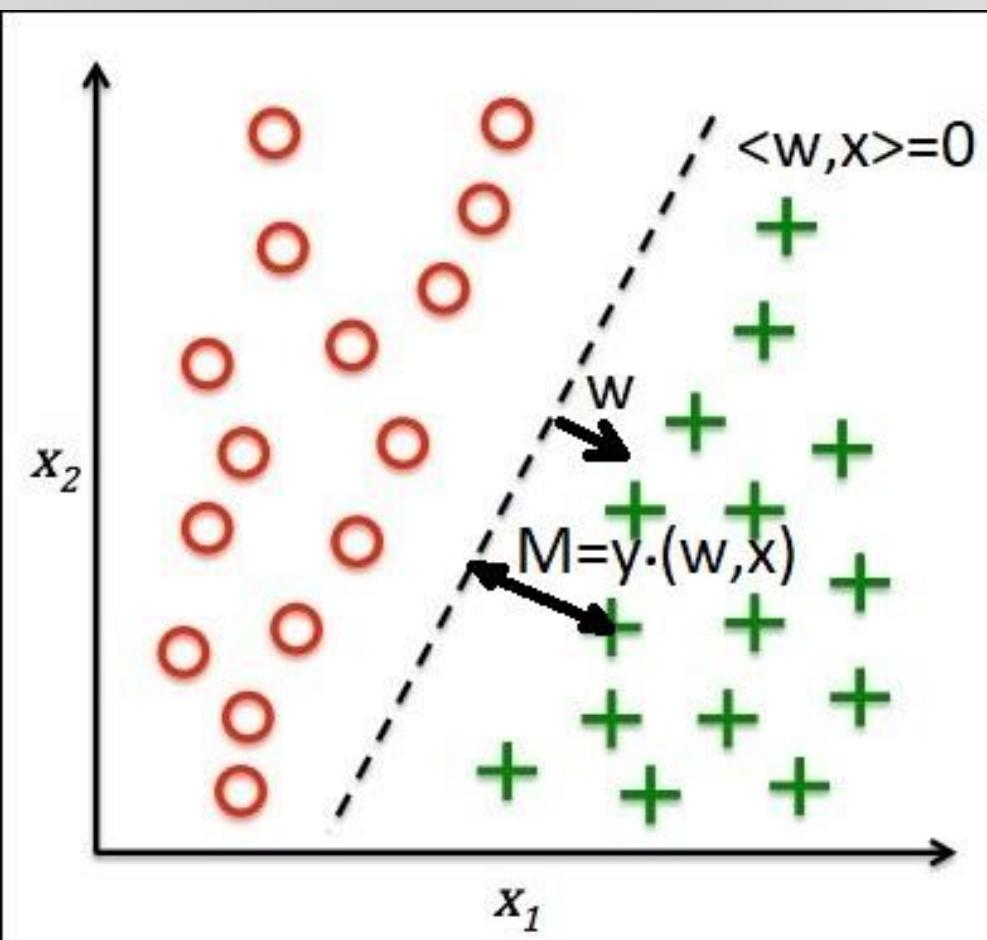
- Если $(w, x) > 0$ (то есть объект отнесён к классу +1), а $y = -1$, то $M = y \cdot (w, x) < 0$.
- Аналогично, если $(w, x) < 0$, а $y = +1$, то $M = y \cdot (w, x) < 0$.

Случай верной классификации:

- Если $(w, x) > 0$ и $y = +1$ или $(w, x) < 0$ и $y = -1$ получаем $M = y \cdot (w, x) > 0$.

ОТСТУП (MARGIN)

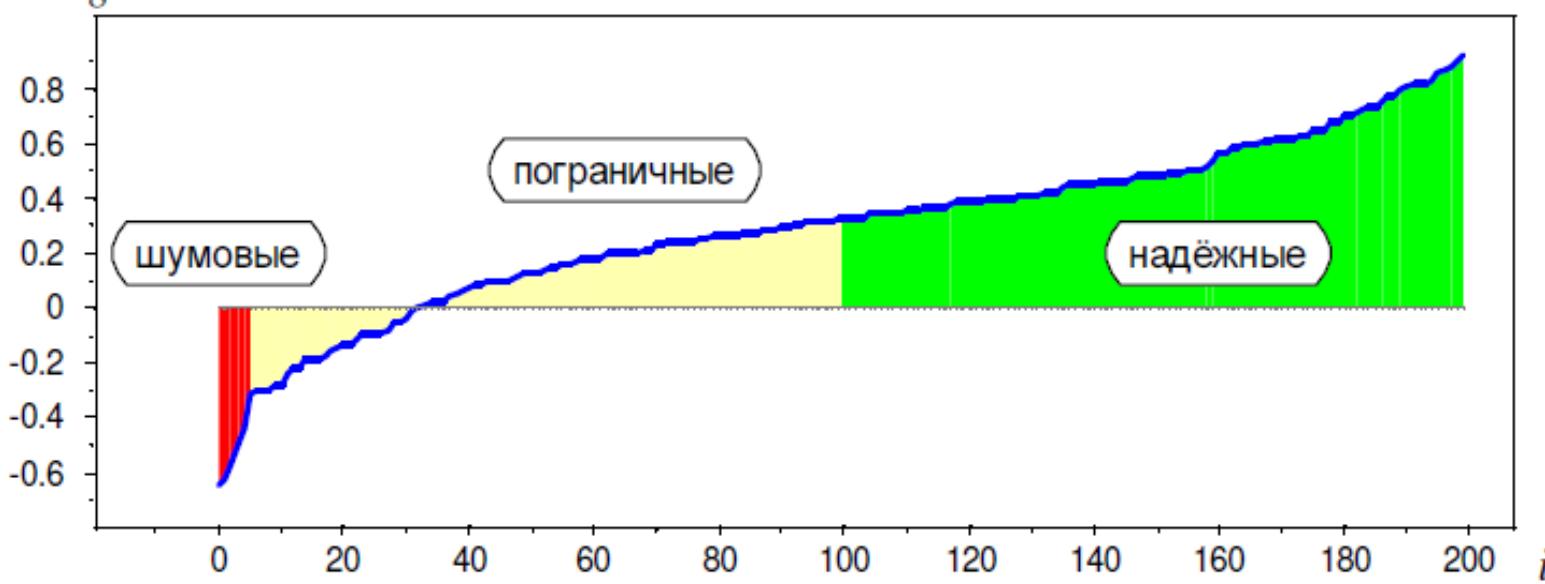
Абсолютная величина отступа M обозначает степень уверенности классификатора в ответе (чем ближе M к нулю, тем меньше уверенность в ответе)



ОТСТУП (MARGIN)

Ранжирование объектов по возрастанию отступа:

Margin



ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация *пороговой функции потерь*:

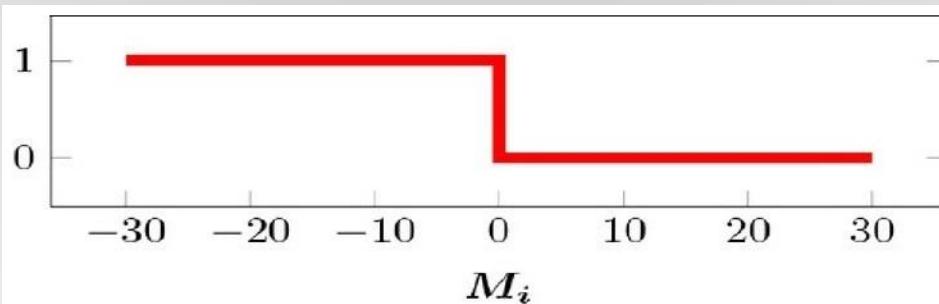
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация *пороговой функции потерь*:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- Пороговая функция потерь *разрывна*, и этот факт сильно затрудняет процесс минимизации.

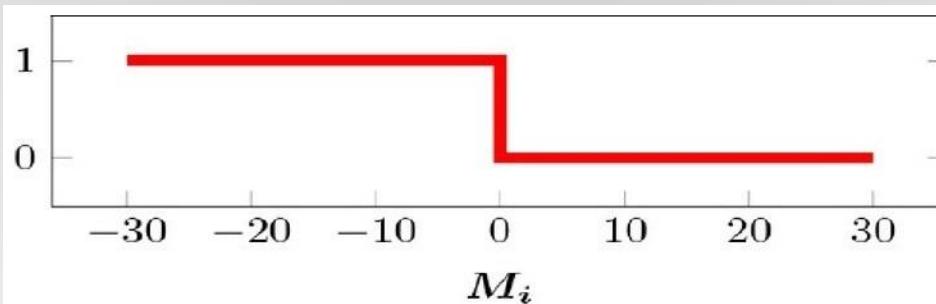


ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация *пороговой функции потерь*:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.



- Для решения этой проблемы используют *другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции*.

ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация *пороговой функции потерь*:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.
- Для решения этой проблемы используют другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции.
- Задача минимизации некоторой функции потерь называется *минимизацией эмпирического риска* (сама функция потерь – эмпирический риск).

ВЕРХНИЕ ОЦЕНКИ ЭМПИРИЧЕСКОГО РИСКА

- $L(a, y) = L(M) = [M < 0]$ – разрывная функция потерь

Оценим

$L(M) \leq \tilde{L}(M)$, где $\tilde{L}(M)$ - непрерывная или гладкая функция потерь.

- Тогда

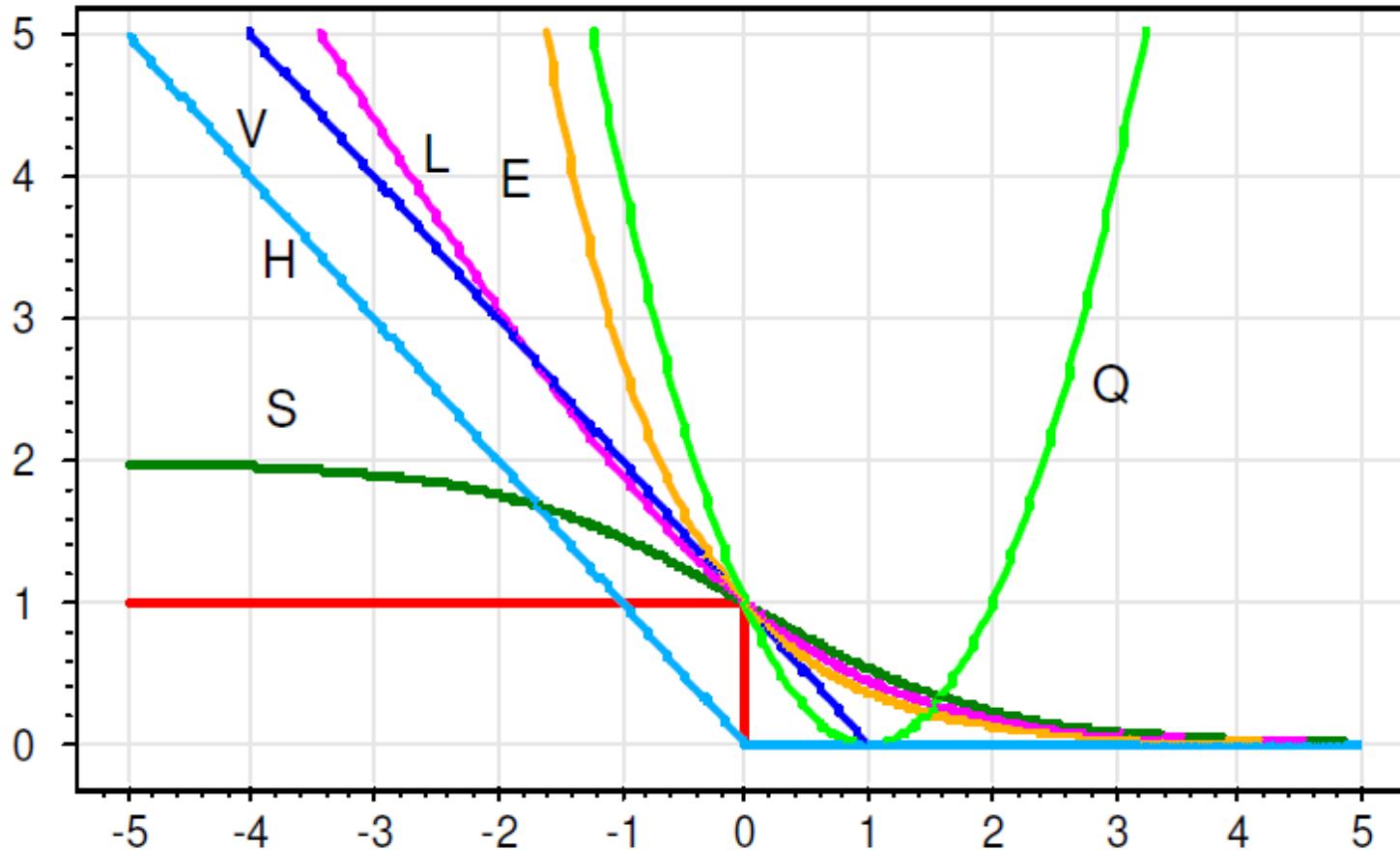
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n L(y_i \cdot (w, x_i)) \leq \frac{1}{n} \sum_{i=1}^n \tilde{L}(y_i \cdot (w, x_i)) \rightarrow \min$$

ФУНКЦИИ ПОТЕРЬ

Минимизируя различные функции потерь, получаем разные результаты. Поэтому разные функции потерь определяют различные классификаторы.

- $L(M) = \log(1 + e^{-M})$ – логистическая функция потерь
- $V(M) = (1 - M)_+ = \max(0, 1 - M)$ – кусочно-линейная функция потерь (метод опорных векторов)
- $H(M) = (-M)_+ = \max(0, -M)$ – кусочно-линейная функция потерь (персептрон)
- $E(M) = e^{-M}$ - экспоненциальная функция потерь
- $S(M) = \frac{2}{1+e^{-M}}$ - сигмоидная функция потерь
- $[M < 0]$ – пороговая функция потерь

ФУНКЦИИ ПОТЕРЬ



M

ОПТИМИЗАЦИЯ ФУНКЦИОНАЛА ПОТЕРЬ

- Нахождение минимума функции потерь Q происходит с помощью метода градиентного спуска:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \eta \cdot \nabla Q(\mathbf{w}^{(k-1)})$$

МЕТРИКИ КАЧЕСТВА

МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ

- Accuracy – доля правильных ответов:

$$\text{accuracy}(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) = y_i]$$

МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ

- Accuracy – доля правильных ответов:

$$\text{accuracy}(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) = y_i]$$

*Недостаток: при сильно несбалансированной выборке
не отражает качество работы алгоритма*

ПРИМЕР: КРЕДИТНЫЙ СКОРИНГ

- Пример: Кредитный scoring
- Модель 1: одобряет 100 кредитов
 - 80 кредитов вернули
 - 20 кредитов не вернули
- Модель 2: одобряет 50 кредитов
 - 48 кредитов вернули
 - 2 кредита не вернули
- Какая лучше?

На выборке,
где 100 вернули,
100 не вернули

МАТРИЦА ОШИБОК

Матрица ошибок (confusion matrix):

		Actual Value	
		positives	negatives
Predicted Value	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ: PRECISION, RECALL

- **Precision (точность):**

$$Precision(a, X) = \frac{TP}{TP + FP}$$

Показывает, насколько можно доверять классификатору при $a(x) = +1$.

PRECISION: ПРИМЕР

Модель $a_1(x)$:

$$\text{precision}(a_1, X) = 0.8$$

Модель $a_2(x)$:

$$\text{precision}(a_2, X) = 0.96$$

		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть	
$a(x) = 1$ Получили кредит	80	20		
	20	80		

		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть	
$a(x) = 1$ Получили кредит	48	2		
	52	98		

МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ: PRECISION, RECALL

- Precision (точность):

$$Precision(a, X) = \frac{TP}{TP + FP}$$

Показывает, насколько можно доверять классификатору при $a(x) = +1$.

- Recall (полнота):

$$Recall(a, X) = \frac{TP}{TP + FN}$$

Показывает, как много объектов положительного класса находит классификатор.

RECALL: ПРИМЕР

Модель $a_1(x)$:

$$\text{recall}(a_1, X) = 0.8$$

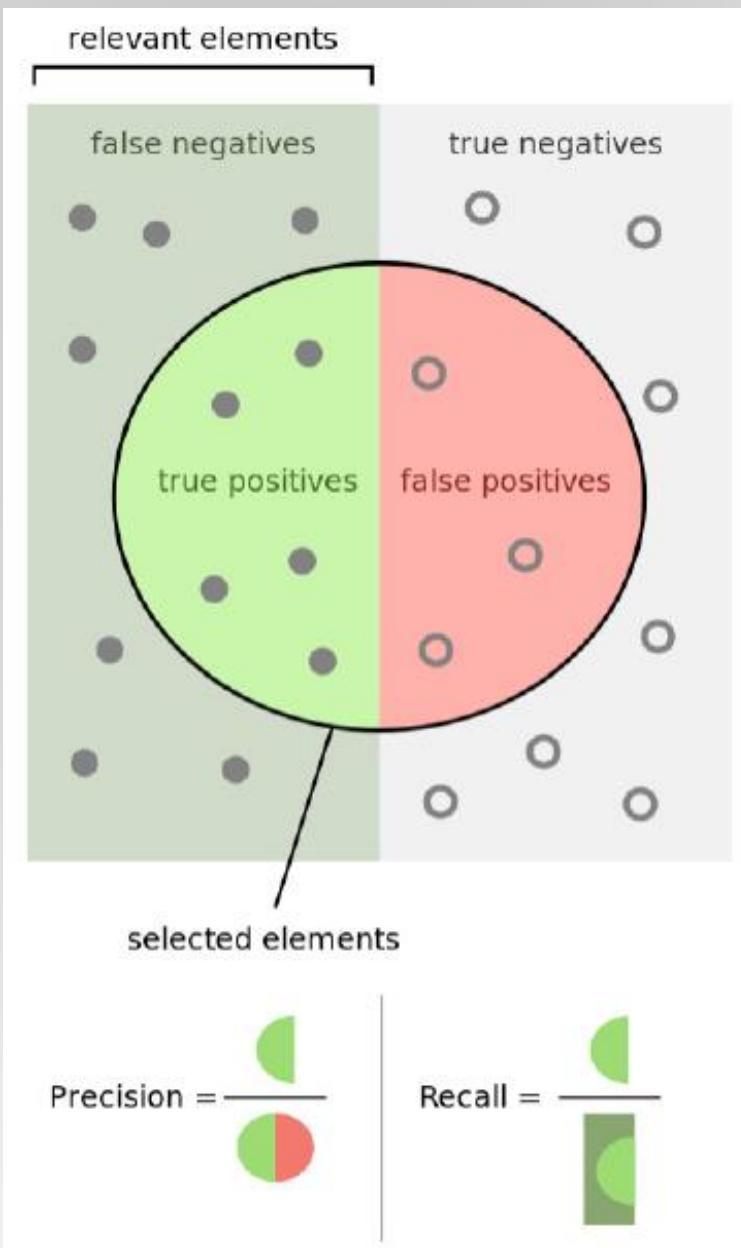
	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20
$a(x) = -1$ Не получили кредит	20	80

Модель $a_2(x)$:

$$\text{recall}(a_2, X) = 0.48$$

	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	48	2
$a(x) = -1$ Не получили кредит	52	98

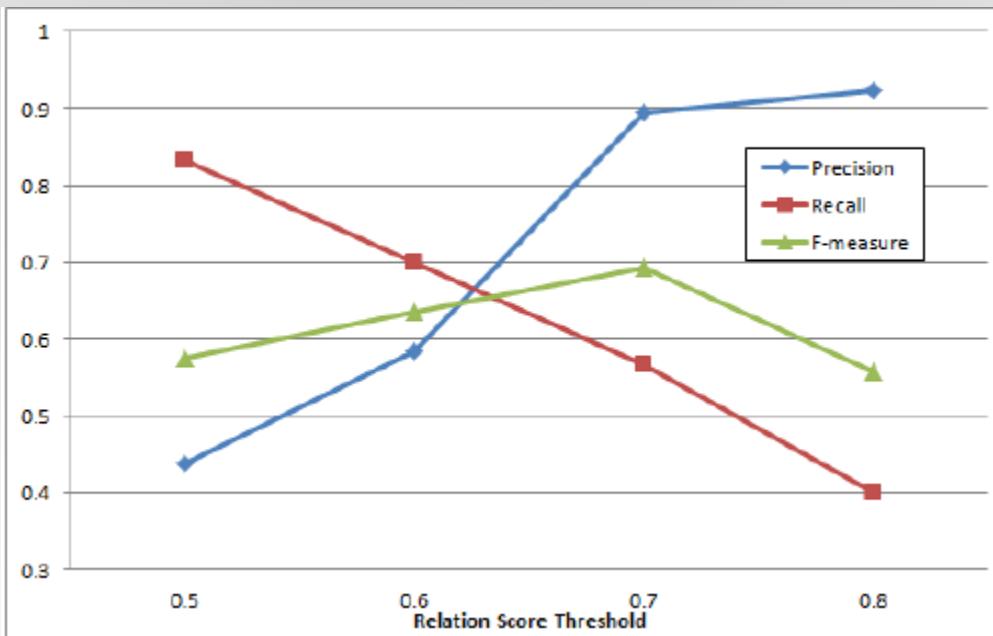
ТОЧНОСТЬ И ПОЛНОТА



F-МЕРА

F-мера – это метрика качества, учитывающая и точность, и полноту

$$F(a, X) = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но *классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.*

ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.
- Пусть $p(x)$ – вероятность того, что объект x принадлежит классу $+1$. Тогда *если $p \geq 0.5$, классификатор относит x к положительному классу, а иначе – к отрицательному.*

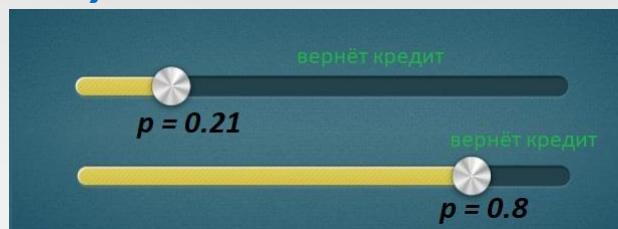


ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.
- Пусть $p(x)$ – вероятность того, что объект x принадлежит классу +1. Тогда если $p \geq 0.5$, классификатор относит x к положительному классу, а иначе – к отрицательному.



- Но *мы можем изменять порог вероятности!* Например, относить к положительному классу те объекты, для которых $p \geq 0.7$, а остальные объекты – к отрицательному классу. *Тем самым мы будем изменять точность и полноту.*



ИНТЕГРАЛЬНАЯ МЕТРИКА: ROC-AUC

Хотим измерить качество всего семейства классификаторов независимо от выбранного порога.

Для этого будем использовать метрику AUC

AUC – *Area Under ROC Curve (площадь под ROC-кривой)*

ROC-КРИВАЯ

Для каждого значения порога t вычислим:

- **False Positive Rate** (доля неверно принятых объектов отрицательного класса):

$$FPR = \frac{FP}{FP + TN} = \frac{\sum_i[y_i = -1][a(x_i) = +1]}{\sum_i[y_i = -1]}$$

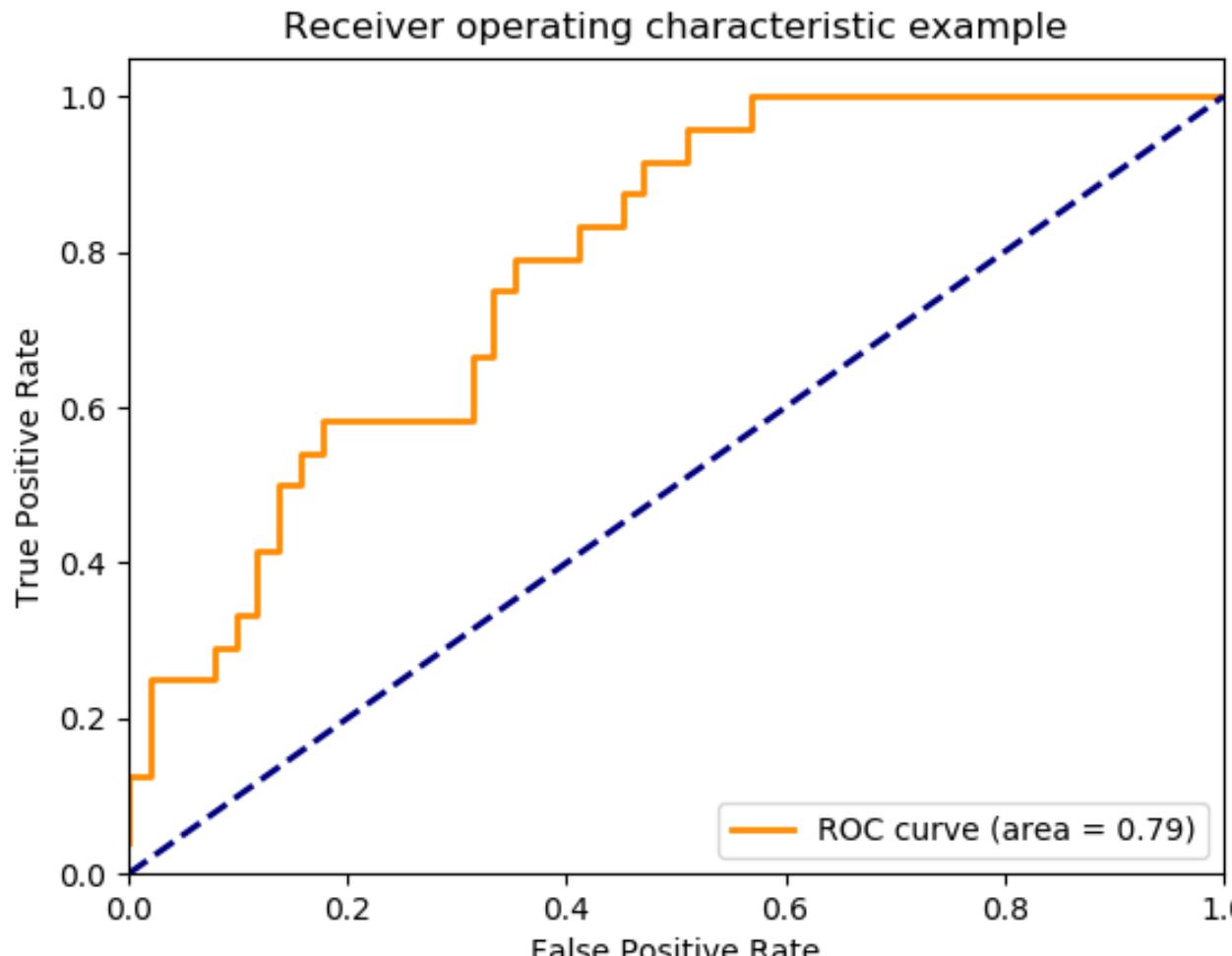
- **True Positive Rate** (доля верно принятых объектов положительного класса):

$$TPR = \frac{TP}{TP+FN} = \frac{\sum_i[y_i=+1][a(x_i)=+1]}{\sum_i[y_i=+1]}.$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

ROC-КРИВАЯ

Кривая, состоящая из точек с координатами (FPR,TPR) для всех возможных порогов – это и есть ROC-кривая.

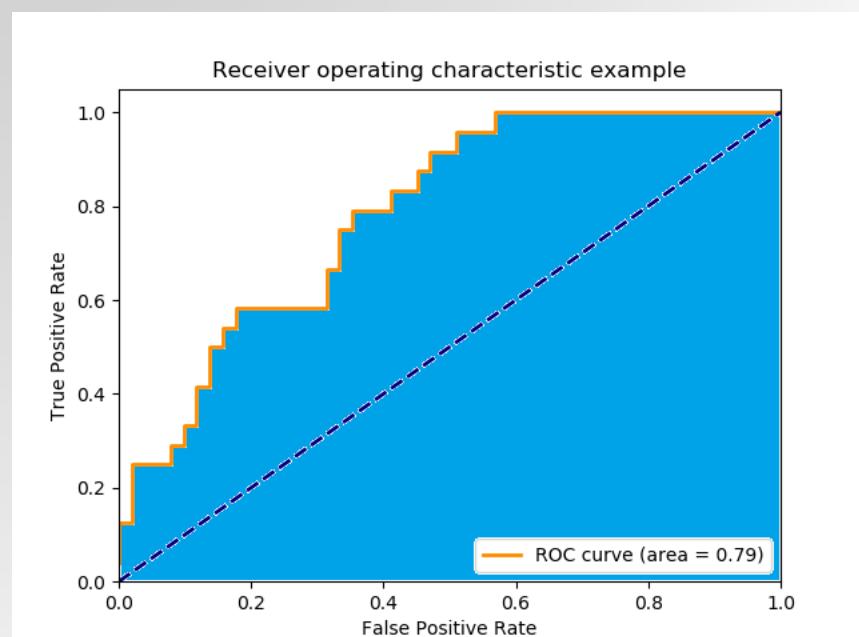


ROC-КРИВАЯ. AUC.

AUC (Area Under Curve) – площадь под ROC-кривой.

$$AUC \in [0; 1].$$

- Чему равен AUC при идеальной классификации?
- Чему равен AUC при случайной классификации?



ROC-КРИВАЯ. AUC.

AUC (Area Under Curve) – площадь под ROC-кривой.

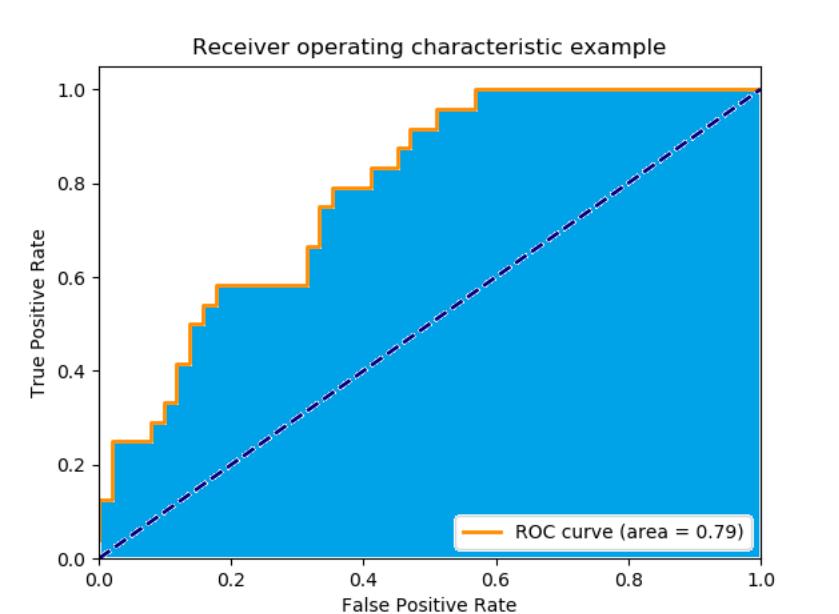
$$AUC \in [0; 1].$$

- $AUC = 1$ –

иdealная классификация

- $AUC = 0.5$ –

случайная классификация



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний: (0.7, 0.4, 0.2, 0.1, 0.05)

1 шаг: $t = 0.7$, то есть

$$a(x) = [b(x) > 0.7]$$

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний: (0.7,0.4,0.2,0.1,0.05)

1 шаг: $t = 0.7$, то есть

$$a(x) = [b(x) > 0.7]$$

$$TPR = \frac{0}{0+3} = 0, \quad FPR = \frac{0}{0+2} = 0.$$

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

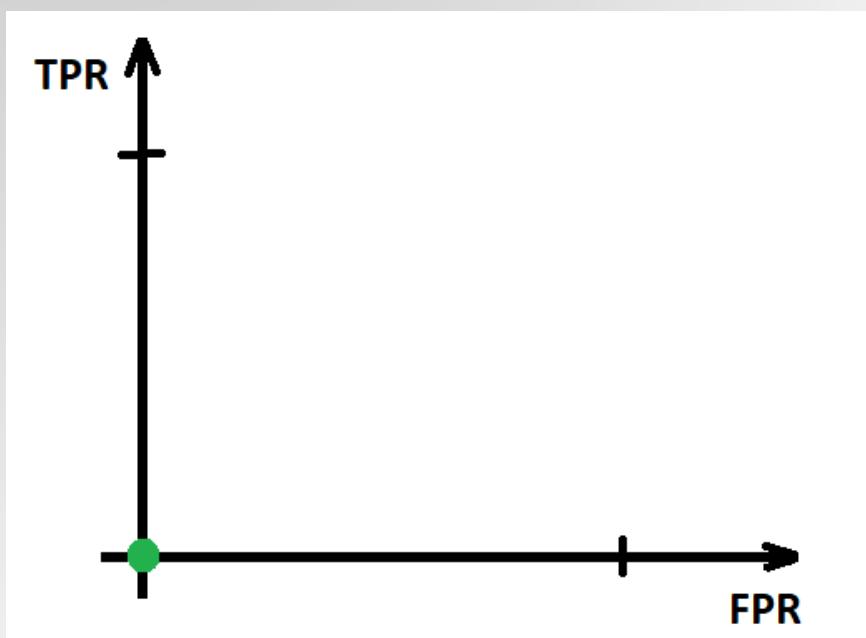
$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

1 шаг: $t = 0.7$, то есть
 $a(x) = [b(x) > 0.7]$

$$TPR = \frac{0}{0+3} = 0,$$

$$FPR = \frac{0}{0+2} = 0.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

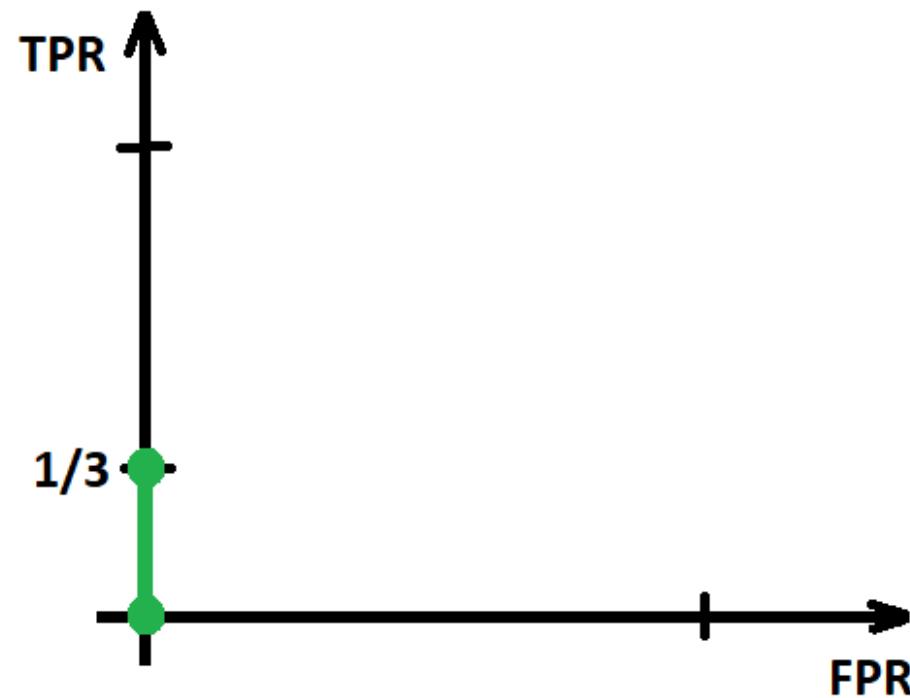
$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

**2 шаг: $t = 0.4$, то есть
 $a(x) = [b(x) > 0.4]$**

$$TPR = \frac{1}{1+2} = \frac{1}{3},$$

$$FPR = \frac{0}{0+2} = 0.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

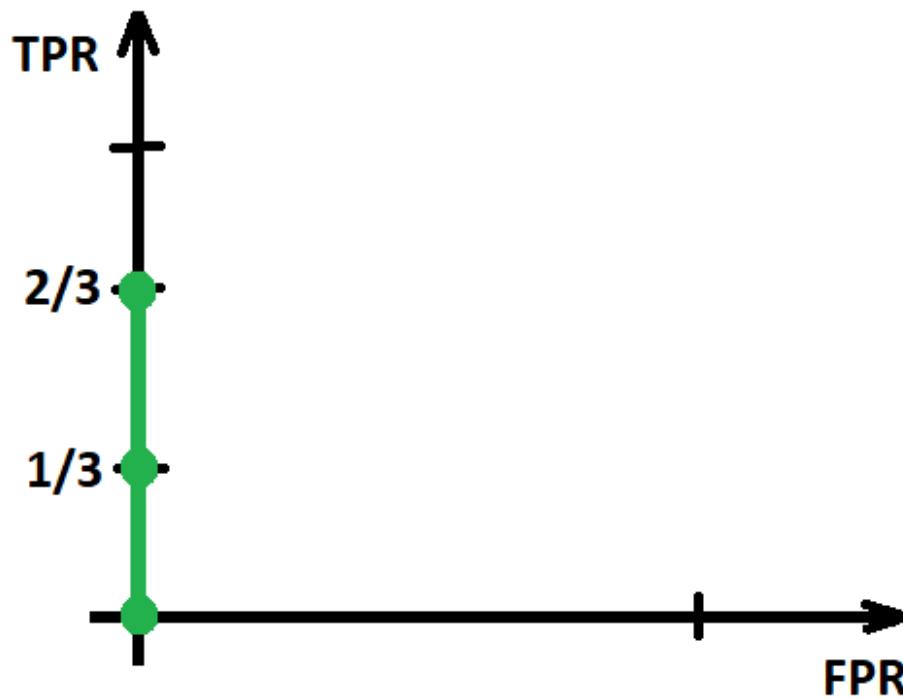
$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

3 шаг: $t = 0.2$, то есть
 $a(x) = [b(x) > 0.2]$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{0}{0+2} = 0.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:

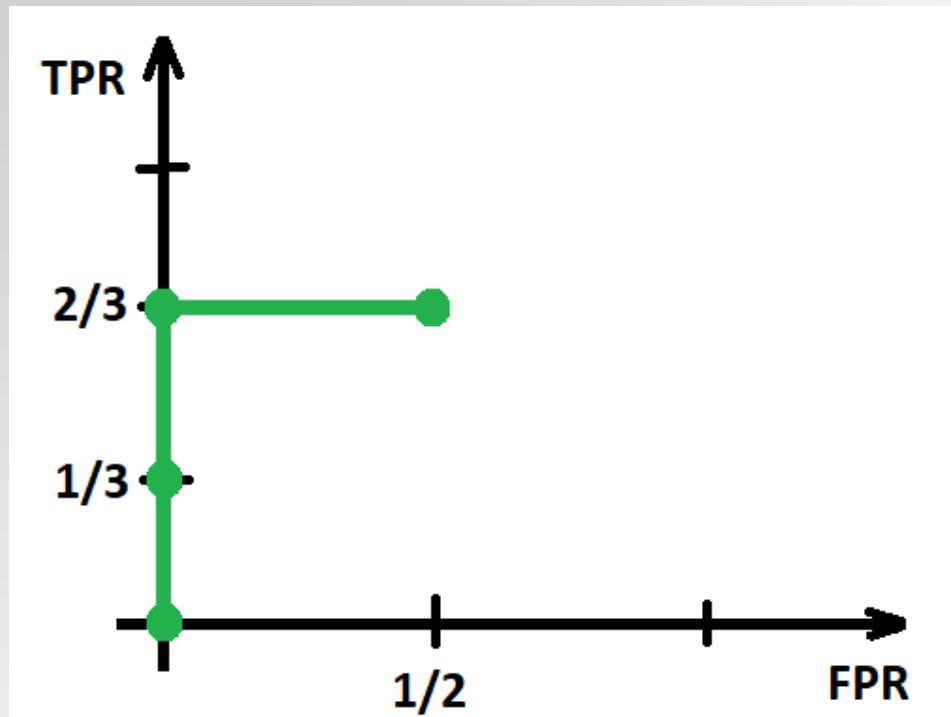
(0.7, 0.4, 0.2, 0.1, 0.05)

4 шаг: $t = 0.1$, то есть

$$a(x) = [b(x) > 0.1]$$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{1}{1+1} = \frac{1}{2}.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по

убыванию предсказаний:

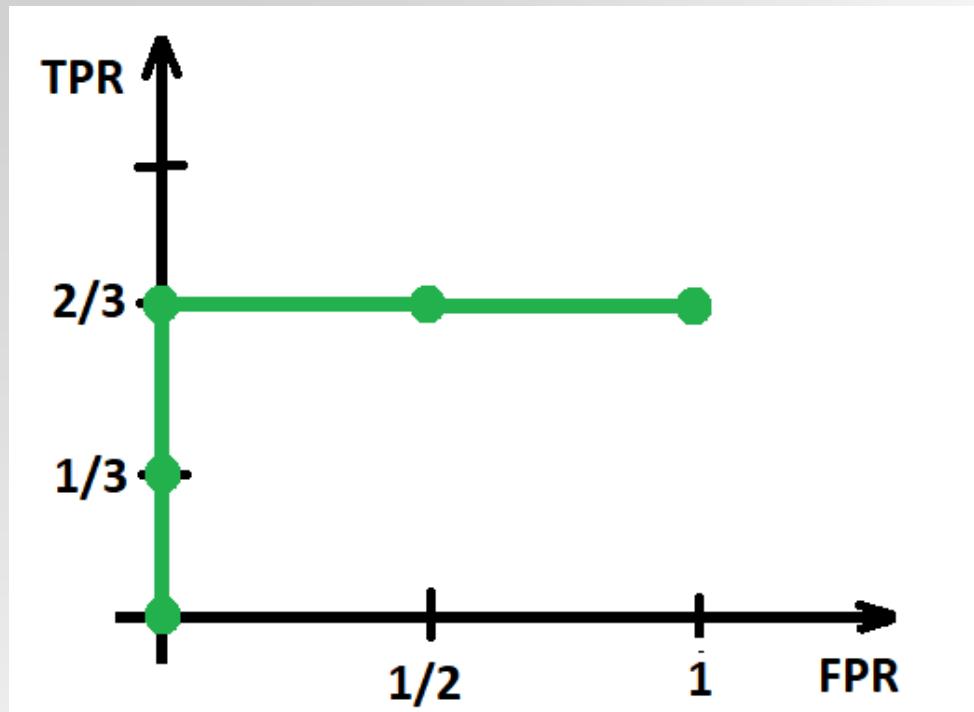
(0.7,0.4,0.2,0.1,0.05)

5 шаг: $t = 0.05$, то есть

$a(x) = [b(x) > 0.05]$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{2}{2+0} = 1.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

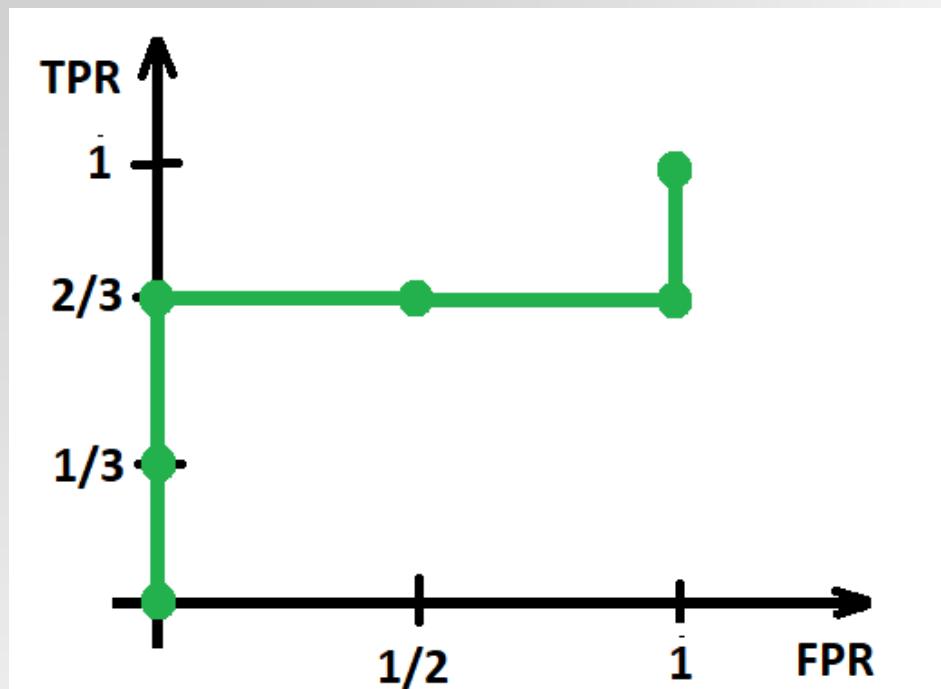
$b(x)$	0.2	0.4	0.1	0.7	0.05
y	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

5 шаг: $t = 0$, то есть
 $a(x) = [b(x) > 0]$

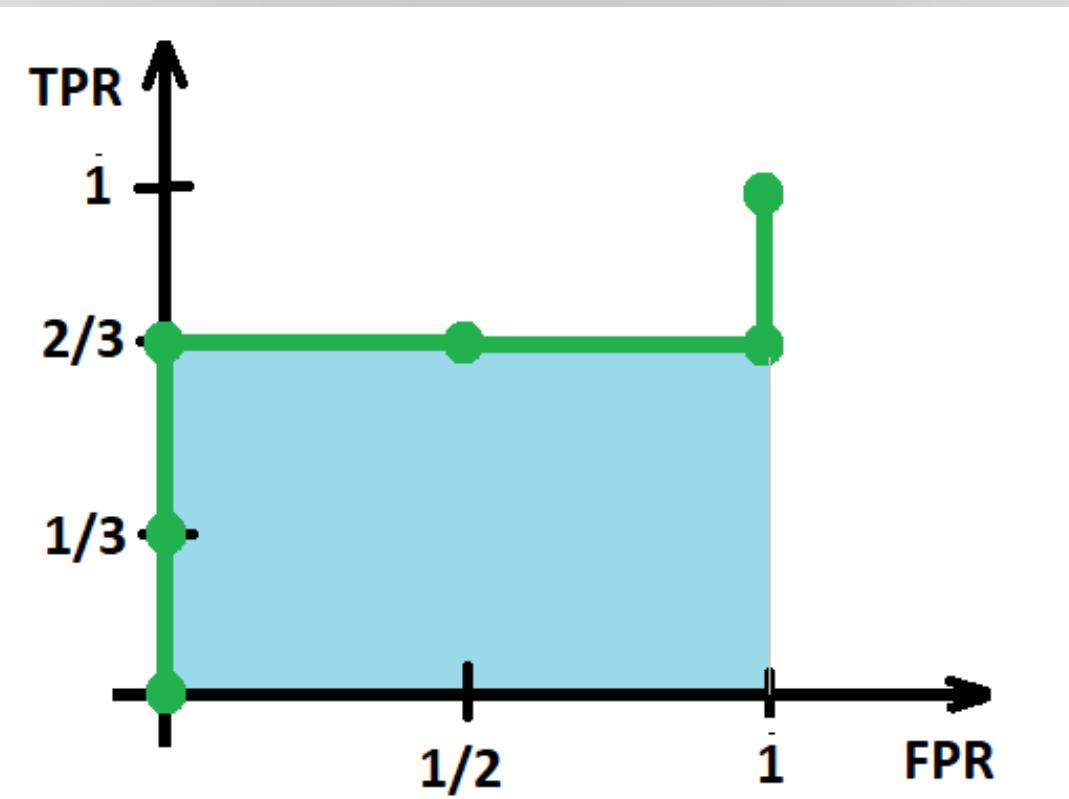
$$TPR = \frac{3}{3+0} = 1,$$

$$FPR = \frac{2}{2+0} = 1.$$



ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

$$AUC = 2/3$$

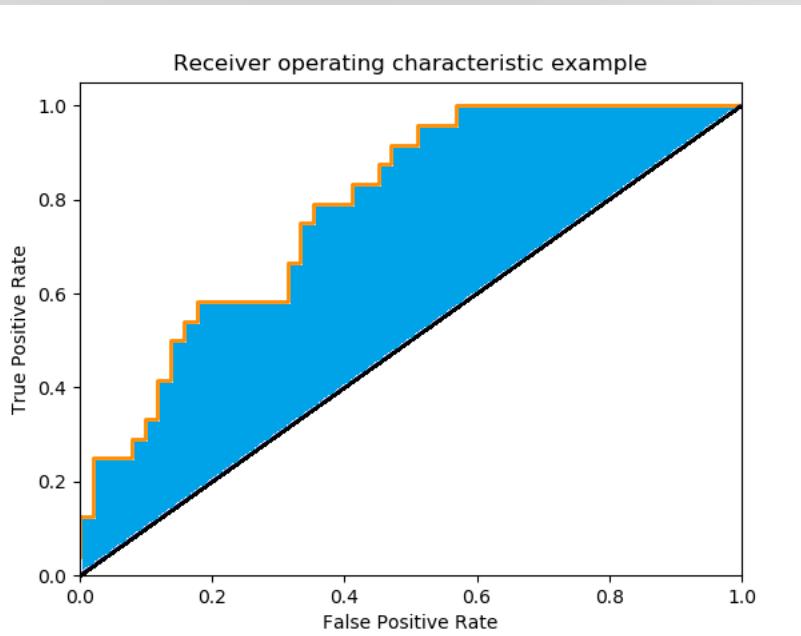


ИНДЕКС ДЖИНИ

Индекс Джини:

$$Gini = 2 \cdot AUC - 1$$

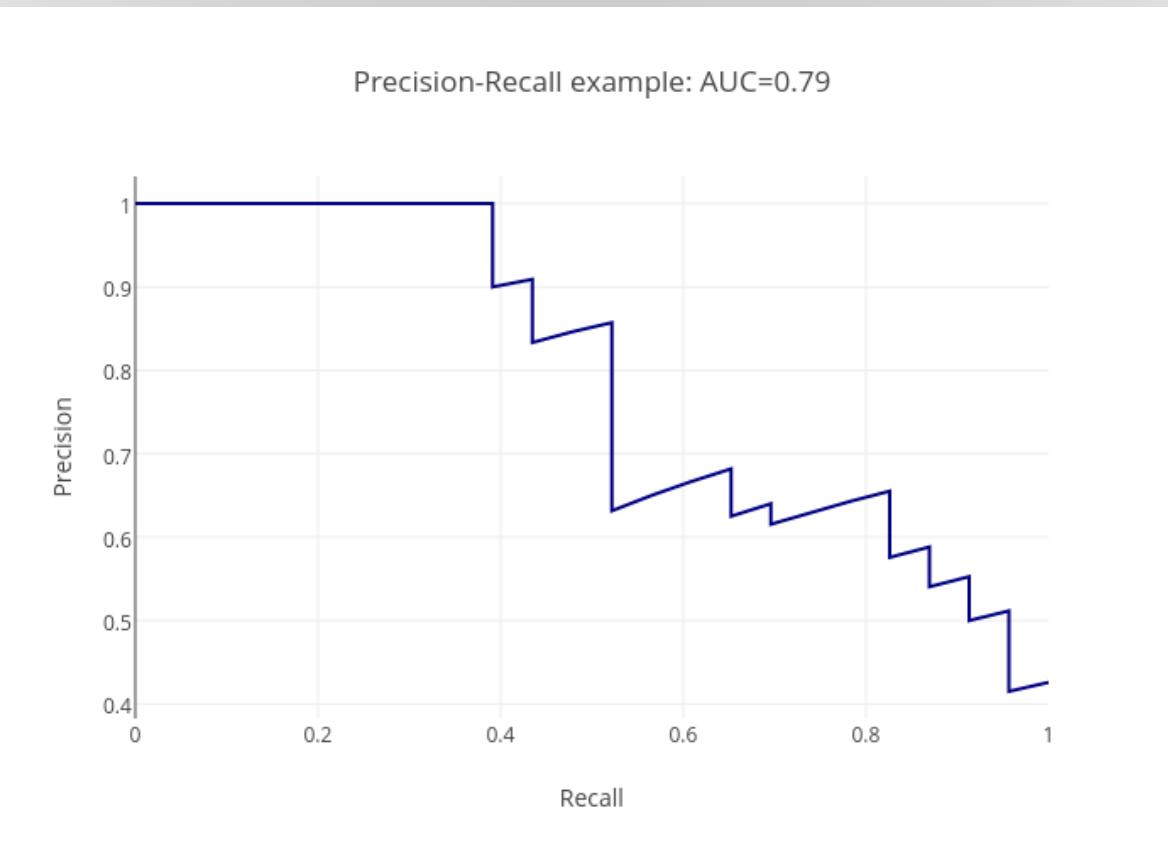
- Индекс Джини – это удвоенная площадь между главной диагональю и ROC-кривой.



PRECISION-RECALL КРИВАЯ

- В случае малой доли объектов положительного класса AUC-ROC может давать неадекватно хороший результат

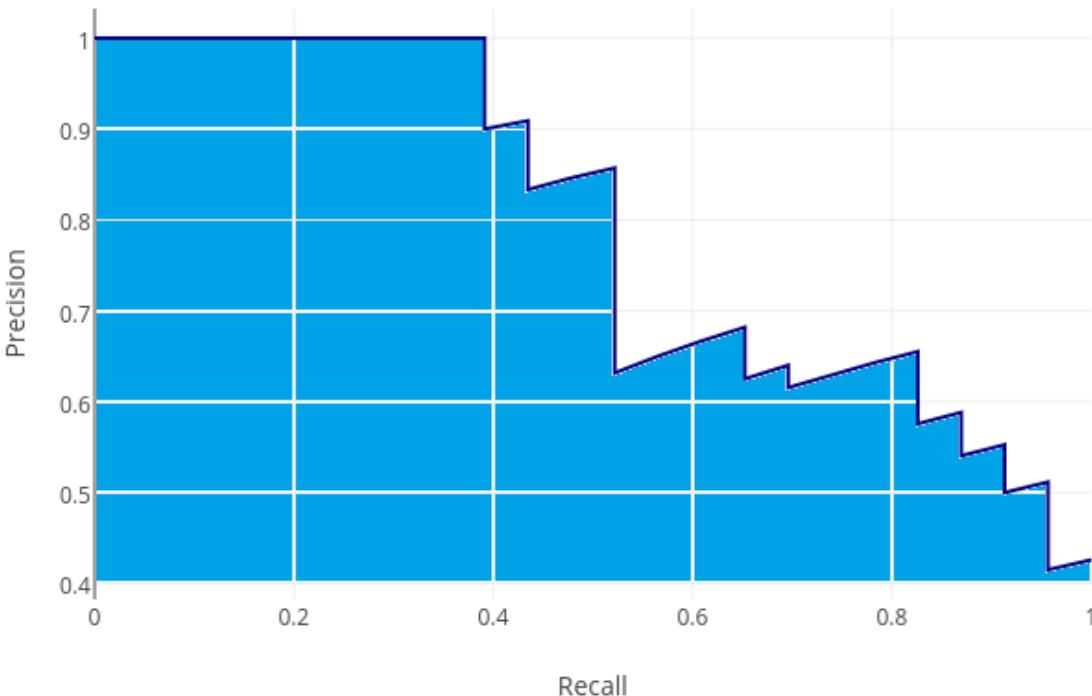
Precision-Recall кривая:



AUC-PR

AUC-PR – площадь под PR-кривой

Precision-Recall example: AUC=0.79



ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия: $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия: $\textcolor{blue}{a}(x, w) = \sigma(w^T x)$,

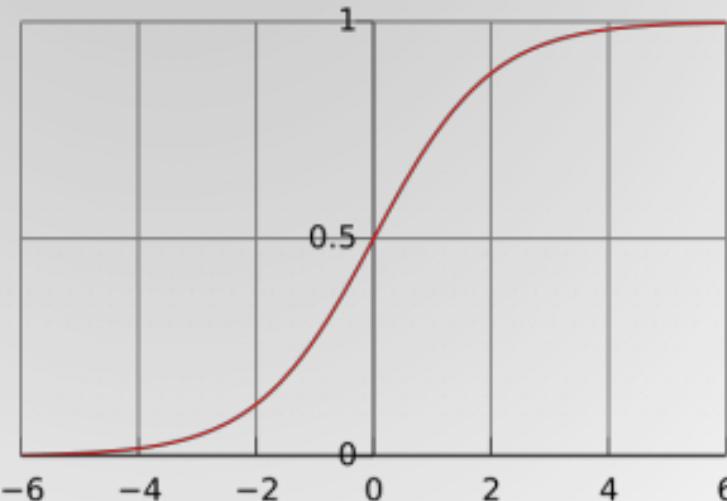
где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоида (логистическая функция)

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия: $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия: $\alpha(x, w) = \sigma(w^T x)$,

где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоида (логистическая функция),
 $\sigma(z) \in (0; 1)$.



Логистическая регрессия: $\alpha(x, w) = \frac{1}{1+e^{-w^T x}}$

ВЕРОЯТНОСТНЫЙ СМЫСЛ

Утверждение. $a(x, w)$ – вероятность того, что $y = +1$ на объекте x , т.е.

$$a(x, w) = P(y = +1|x; w)$$

Доказательство. Дальше в лекции.

РАЗДЕЛЯЮЩАЯ ГРАНИЦА

Предсказываем $y = +1$, если $a(x, w) \geq 0.5$.



$$a(x, w) = \sigma(w^T x) \geq 0.5, \text{ если } w^T x \geq 0.$$

Получаем, что

- $y = +1$ при $w^T x \geq 0$
- $y = -1$ при $w^T x < 0$,

т.е. $w^T x = 0$ – разделяющая гиперплоскость.

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Логистическая регрессия - это линейный классификатор!

ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Если взять квадратичную функцию потерь

$$L(a, y) = (a - y)^2,$$

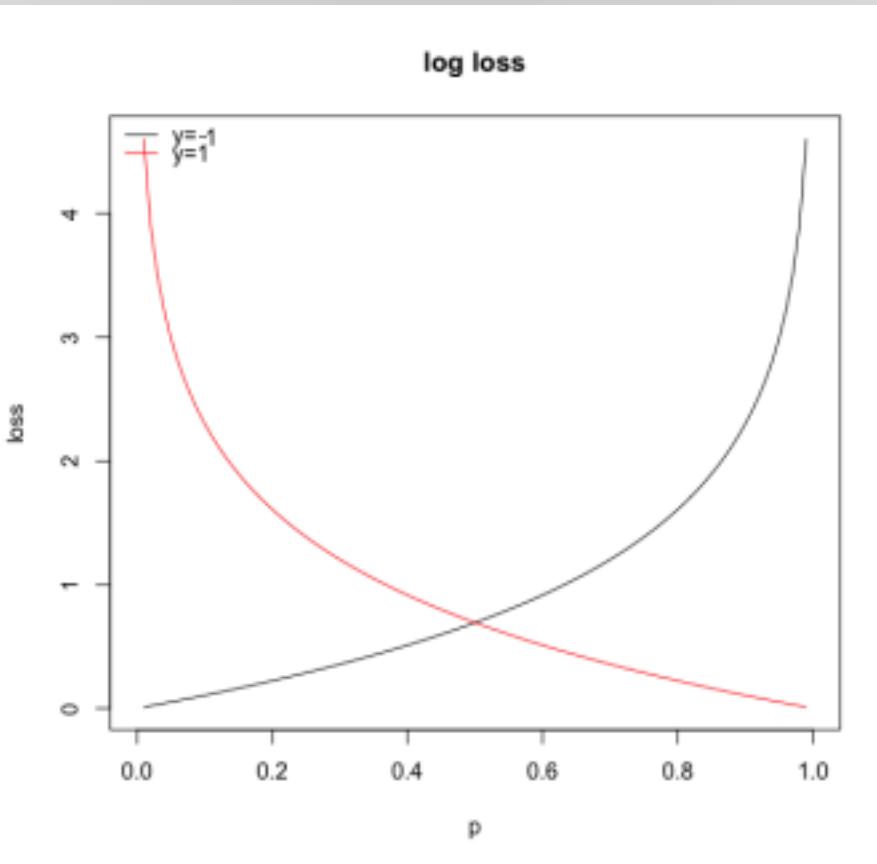
то возникнут проблемы:

- $Q(a, X) = \frac{1}{l} \sum_{i=1}^l \left(\frac{1}{1+e^{-w^T x_i}} - y_i \right)^2$ - не выпуклая функция
(можем не попасть в глобальный минимум при оптимизации)
- На совсем неправильном предсказании маленький штраф
(пусть предсказали вероятность 0% на объекте класса $y = +1$, тогда штраф всего $(1 - 0)^2 = 1$)

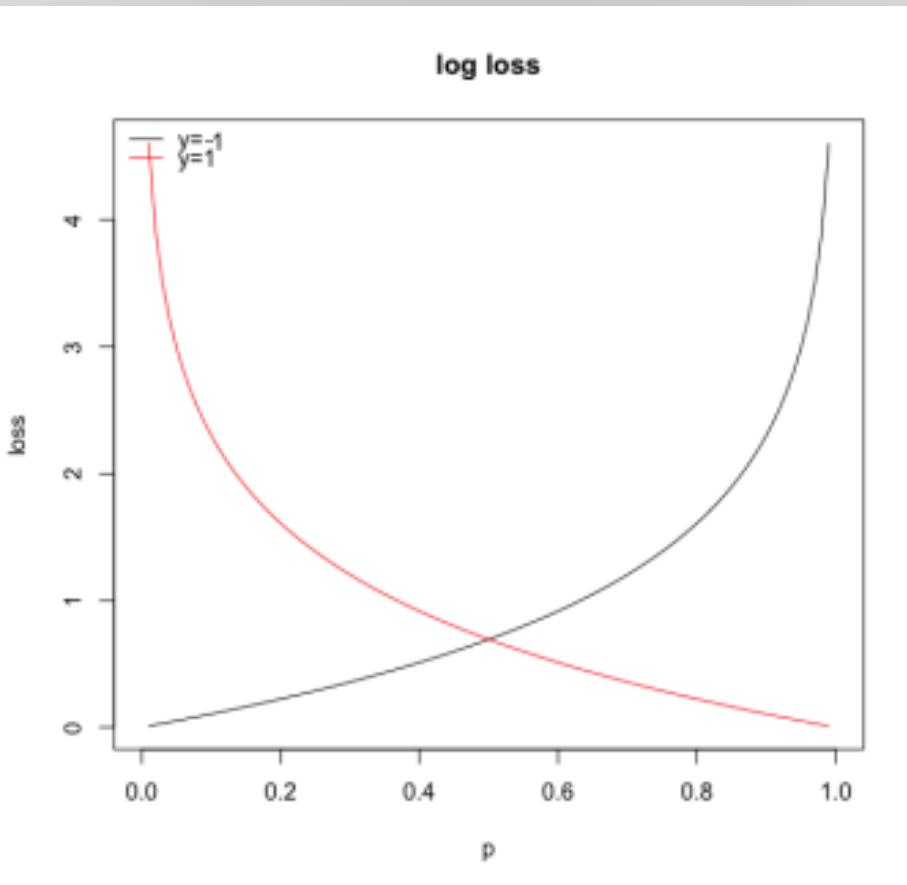
ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Возьмем логистическую функцию потерь (**log-loss**):

$$Q(w) = - \sum_{i=1}^l ([y_i = +1] \cdot \log(a(x_i, w)) + [y_i = -1] \cdot \log(1 - a(x_i, w)))$$



ЛОГИСТИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ



- если $a(x, w) = 1$ и $y = +1$, то штраф $L(a, y) = 0$
- если $a(x, w) \rightarrow 0$, а $y = +1$, то штраф $L(a, y) \rightarrow +\infty$

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ: ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

Цель: построить алгоритм $b(x)$, в каждой точке x предсказывающий $p(y = +1|x)$.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

Цель: построить алгоритм $b(x)$, в каждой точке x предсказывающий $p(y = +1|x)$.

Комментарий: пока что мы будем решать задачу в общем виде, то есть у нас нет ограничений на вид алгоритма $b(x)$ и на вид функции потерь $L(y, b)$.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при $n \rightarrow \infty$ получаем

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x]$$

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при $n \rightarrow \infty$ получаем

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x]$$

Отсюда получаем **условие на функцию потерь**:

$$\operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x] = p(y = +1|x)$$

ФУНКЦИИ ПОТЕРЬ

Подходят:

Квадратичная

$$L(y, z) = (y - z)^2$$

- Логистическая (log-loss)

$$L(y, z) = [y = +1] \cdot \log(b(x, w)) + [y = -1] \cdot \log(1 - b(x, w))$$

Не подходят:

• Модуль

$$L(y, z) = |y - z|$$

ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм $b(x)$, должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект x с классом y :

$$b(x)^{[y=+1]} \cdot (1 - b(x))^{[y=-1]}$$

ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм $b(x)$, должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект x с классом y :

$$b(x)^{[y=+1]} \cdot (1 - b(x))^{[y=-1]}$$

Правдоподобие выборки:

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]}$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это **log-loss!**

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это **log-loss!**

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

Вывод: логистическая функция потерь корректно предсказывает вероятности.

ВЫБОР АЛГОРИТМА $b(x)$

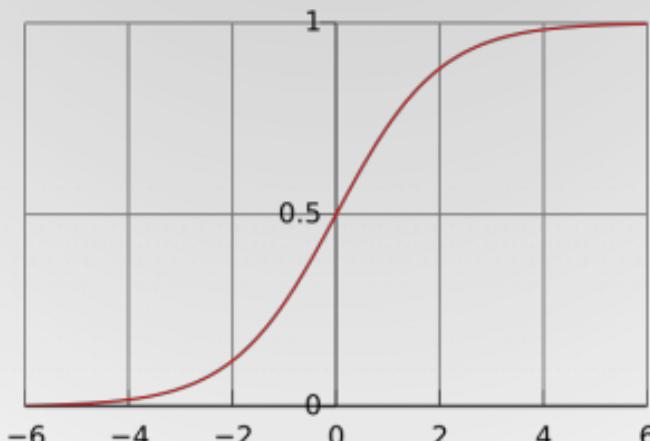
- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.

ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.
- Можно взять $b(x) = \sigma(w^T x)$, где σ – любая монотонно неубывающая функция с областью значений $[0, 1]$.

ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.
- Можно взять $b(x) = \sigma(w^T x)$, где σ – любая монотонно неубывающая функция с областью значений $[0, 1]$.
- Возьмем **сигмоиду**: $\sigma(z) = \frac{1}{1+e^{-z}}$



СМЫСЛ (w, x) В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке x предсказывает вероятность того, что x принадлежит положительному классу $p(y = +1|x)$.
- То есть $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$. Отсюда можно выразить $(w, x) = w^T x$:

$$(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}$$

СМЫСЛ (w, x) В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке x предсказывает вероятность того, что x принадлежит положительному классу $p(y = +1|x)$.
- То есть $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$. Отсюда можно выразить $(w, x) = w^T x$:

$$(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}$$

- Величина $\log \frac{p(y=+1|x)}{p(y=-1|x)}$ называется **логарифм отношения шансов (log odds)**. Из формулы видно, что величина может принимать любое значение.

ЛОГАРИФМИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ

Утверждение. Логарифмическая функция потерь может быть записана в виде

$$L(b, X) = \sum_{i=1}^l \log(1 + e^{-y_i(w, x)})$$

Идея доказательства:

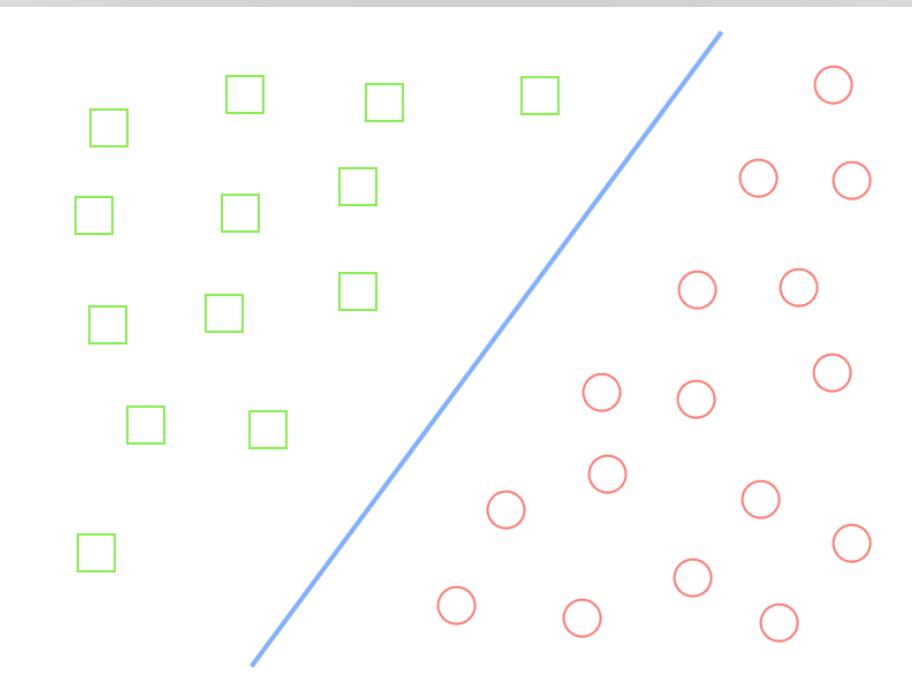
Подставляем явный вид сигмоиды в логарифмическую функцию потерь:

$$-\sum_{i=1}^l ([y_i = +1] \log \sigma(w^T x_i) + [y_i = -1] \log(1 - \sigma(w^T x_i))) \rightarrow \min_w$$

МЕТОД ОПОРНЫХ ВЕКТОРОВ

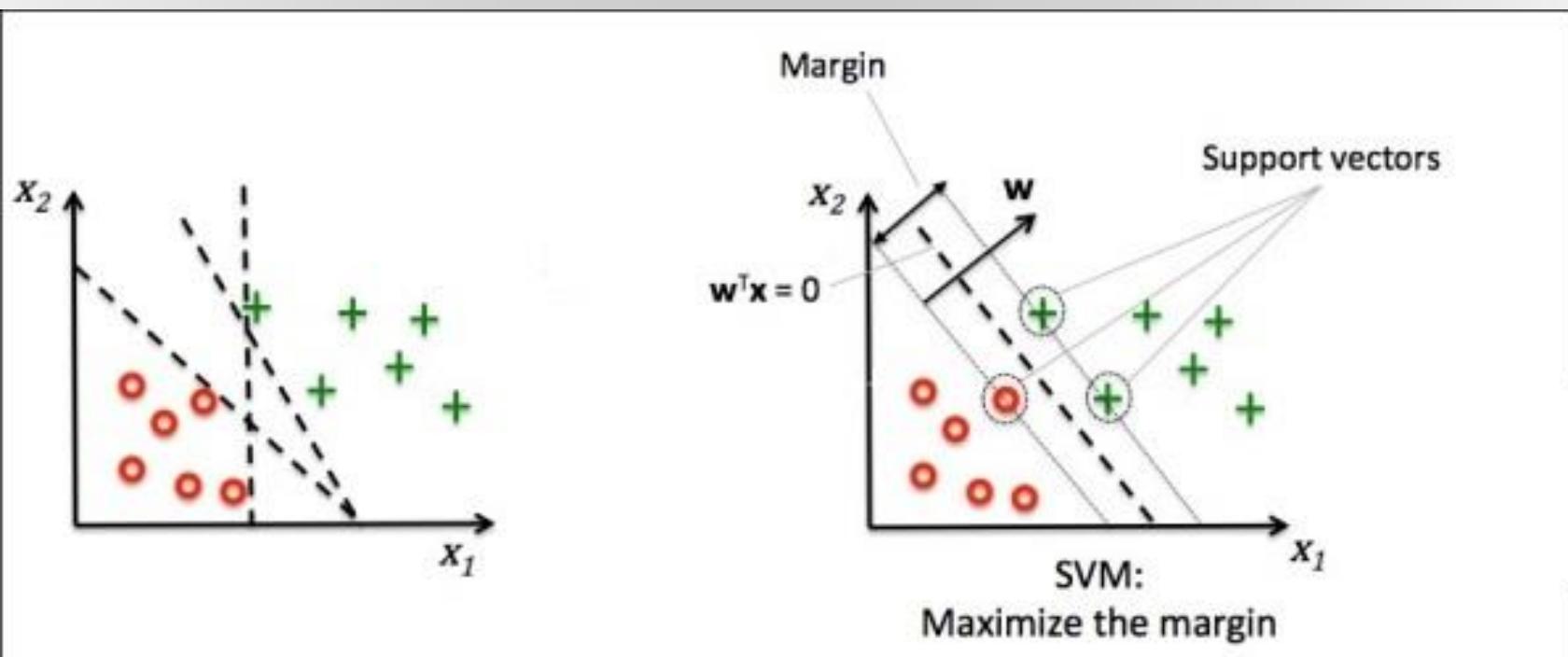
ЛИНЕЙНО РАЗДЕЛИМАЯ ВЫБОРКА

Выборка *линейно разделима*, если существует такой вектор параметров w^* , что соответствующий классификатор $a(x)$ не допускает ошибок на этой выборке.



МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

Цель метода опорных векторов (Support Vector Machine) –
максимизировать ширину разделяющей полосы.



МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ



- $a(x) = \text{sign}((w, x) + w_0)$
- Нормируем параметры w и w_0 так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$



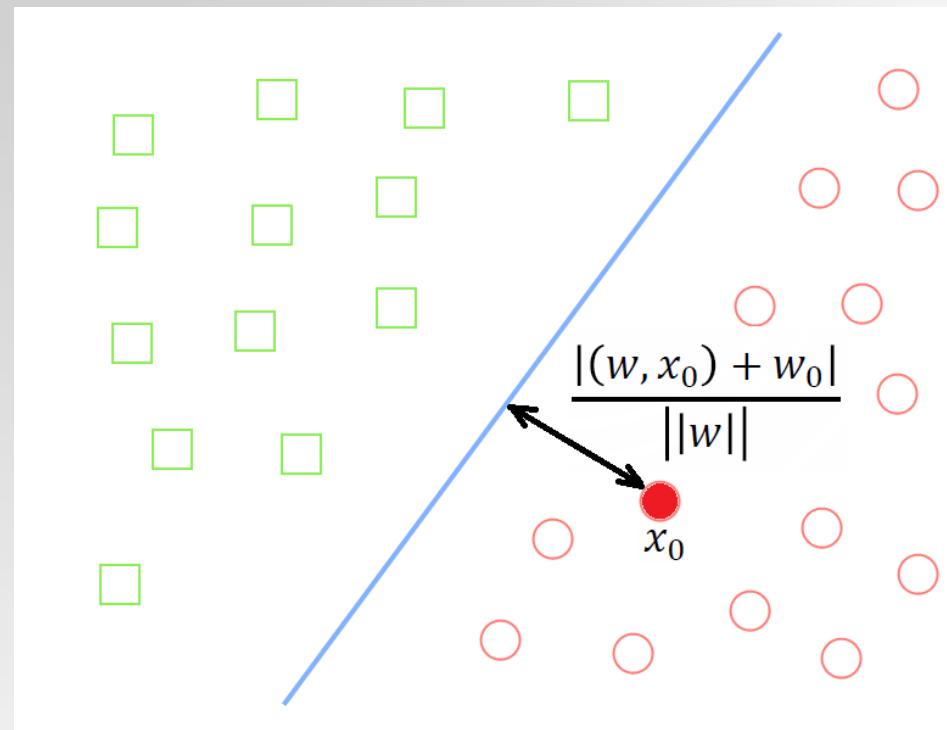
МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

- $a(x) = \text{sign}((w, x) + w_0)$
- Нормируем параметры w и w_0 так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$

Расстояние от точки x_0 до разделяющей гиперплоскости, задаваемой классификатором:

$$\rho(x_0, a) = \frac{|(w, x_0) + w_0|}{\|w\|}$$



МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

- Нормируем параметры w и w_0 так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$

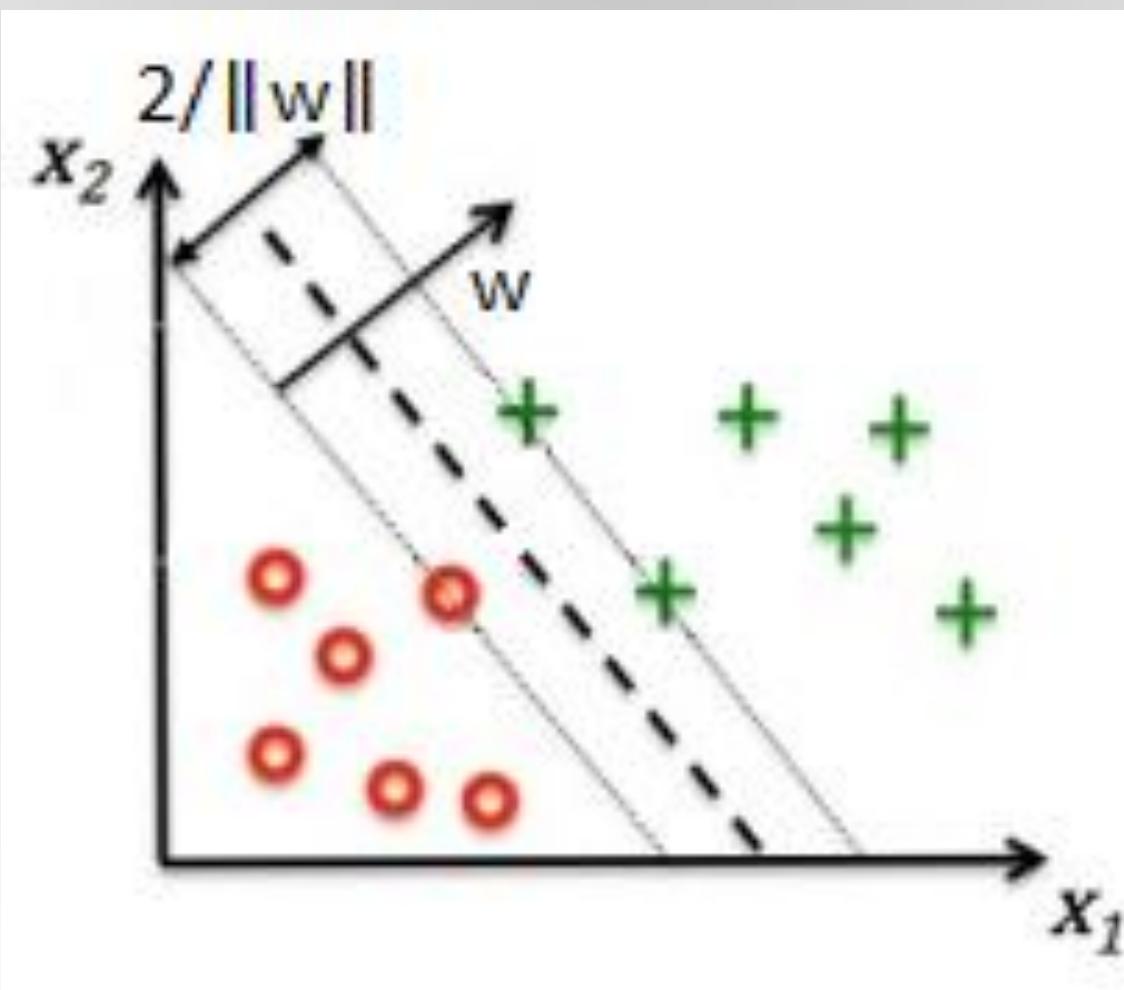
Тогда расстояние от точки x_0 до разделяющей гиперплоскости, задаваемой классификатором:

$$\rho(x_0, a) = \frac{|(w, x_0) + w_0|}{\|w\|}$$

- Расстояние до ближайшего объекта $x \in X$:

$$\min_{x \in X} \frac{|(w, x) + w_0|}{\|w\|} = \frac{1}{\|w\|} \min_{x \in X} |(w, x) + w_0| = \frac{1}{\|w\|}$$

РАЗДЕЛЯЮЩАЯ ПОЛОСА



ОПТИМИЗАЦИОННАЯ ЗАДАЧА SVM ДЛЯ РАЗДЕЛИМОЙ ВЫБОРКИ

$$\begin{cases} \frac{1}{2} \|w\|^2 \rightarrow \min_w \\ y_i((w, x_i) + w_0) \geq 1, i = 1, \dots, l \end{cases}$$

Утверждение. Данная оптимизационная задача имеет единственное решение.

ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

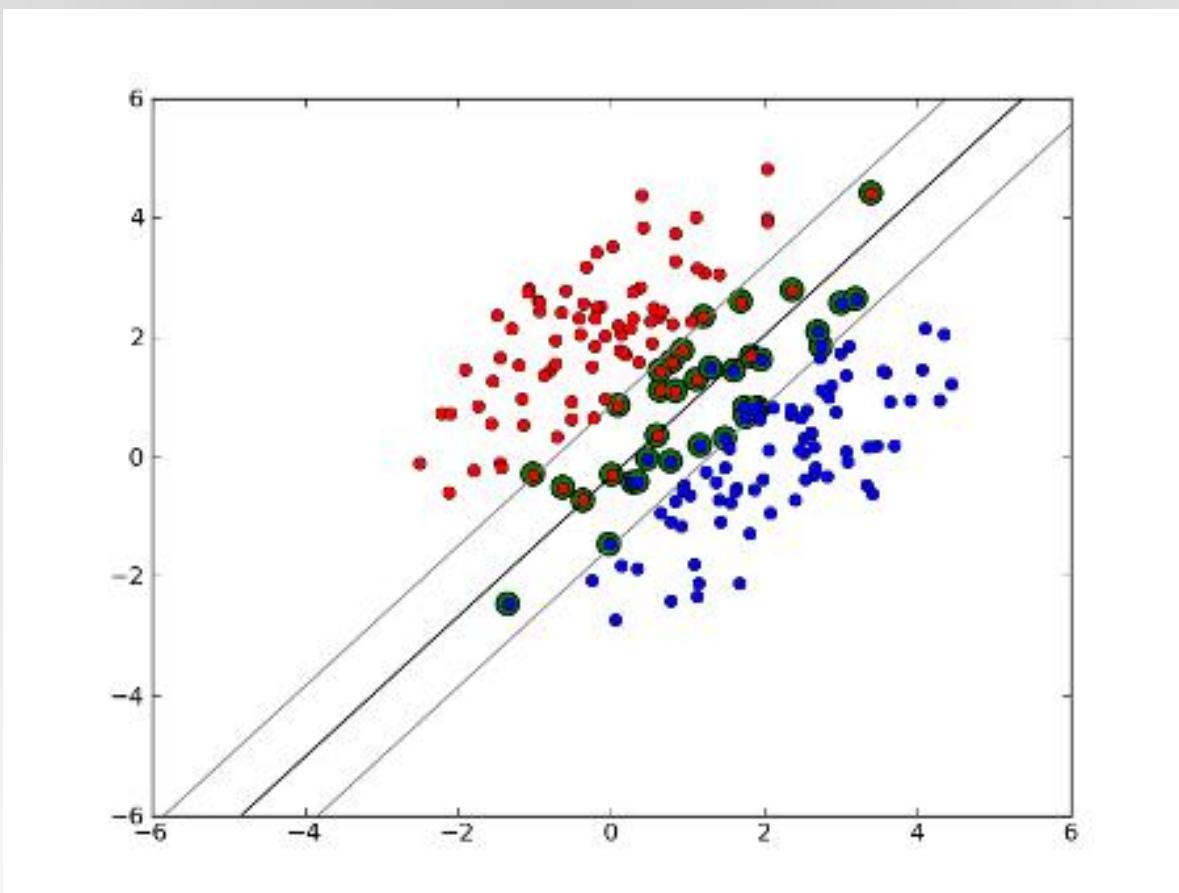
- Существует хотя бы один объект $x \in X$, что

$$y_i((w, x_i) + w_0) < 1$$

ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

- Существует хотя бы один объект $x \in X$, что

$$y_i((w, x_i) + w_0) < 1$$



ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

- Существует хотя бы один объект $x \in X$, что

$$y_i((w, x_i) + w_0) < 1$$

Смягчим ограничения, введя штрафы $\xi_i \geq 0$:

$$y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l$$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Хотим:

- Минимизировать штрафы $\sum_{i=1}^l \xi_i$
- Максимизировать отступ $\frac{1}{||w||}$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Хотим:

- Минимизировать штрафы $\sum_{i=1}^l \xi_i$
- Максимизировать отступ $\frac{1}{\|w\|}$

Задача оптимизации:

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l \\ \xi_i \geq 0, i = 1, \dots, l \end{cases}$$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Утверждение. Задача

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l \\ \xi_i \geq 0, i = 1, \dots, l \end{cases}$$

Является выпуклой и имеет единственное решение.

СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) = 1 - M_i \\ \xi_i \geq 0 \end{cases}$$

СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) \\ \xi_i \geq 0 \end{cases} \Rightarrow \xi_i = \max(0, 1 - y_i((w, x_i) + w_0))$$

СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} & (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l & (2) \\ \xi_i \geq 0, i = 1, \dots, l & (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) \\ \xi_i \geq 0 \end{cases} \Rightarrow \xi_i = \max(0, 1 - y_i((w, x_i) + w_0))$$

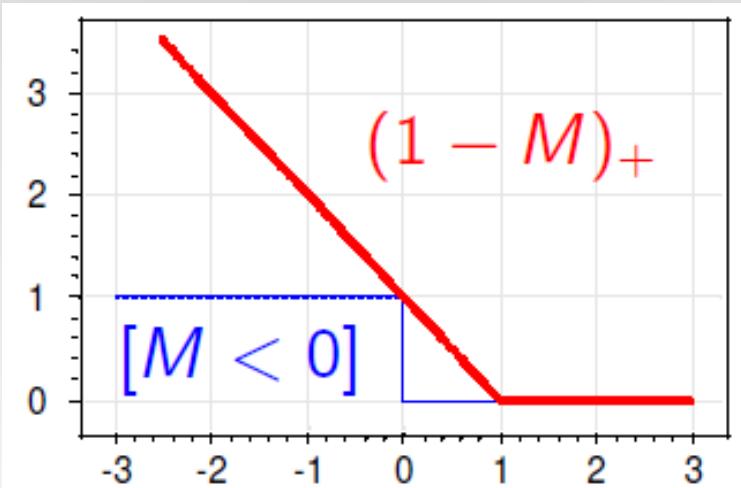
Получаем безусловную задачу оптимизации:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \max(0, 1 - y_i((w, x_i) + w_0)) \rightarrow \min_{w, w_0}$$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: ЗАДАЧА ОПТИМИЗАЦИИ

- На задачу оптимизации SVM можно смотреть, как на оптимизацию функции потерь $L(M) = \max(0, 1 - M) = (1 - M)_+$ с регуляризацией:

$$Q(a, X) = \sum_{i=1}^l (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}$$

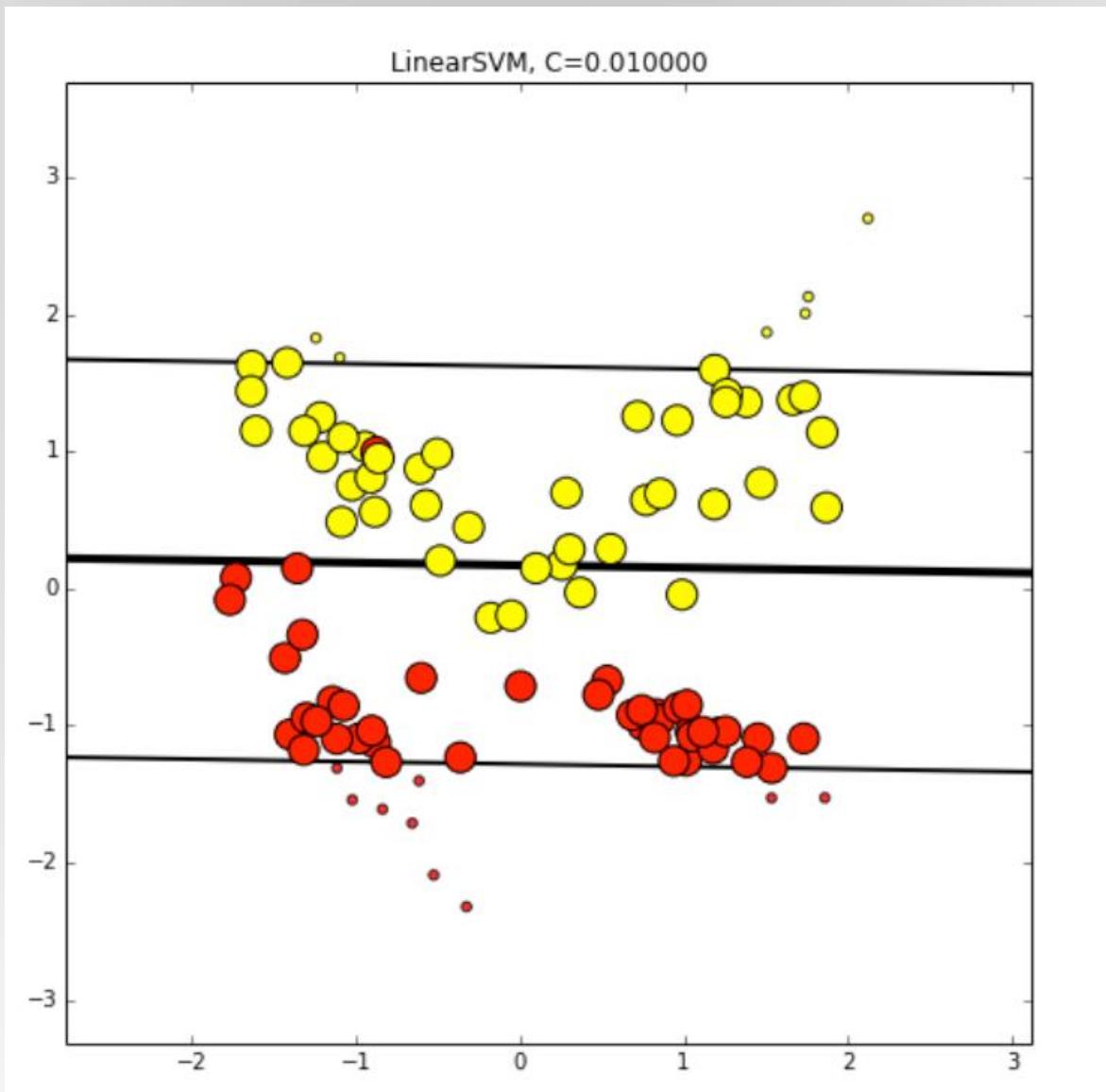


ЗНАЧЕНИЕ КОНСТАНТЫ С

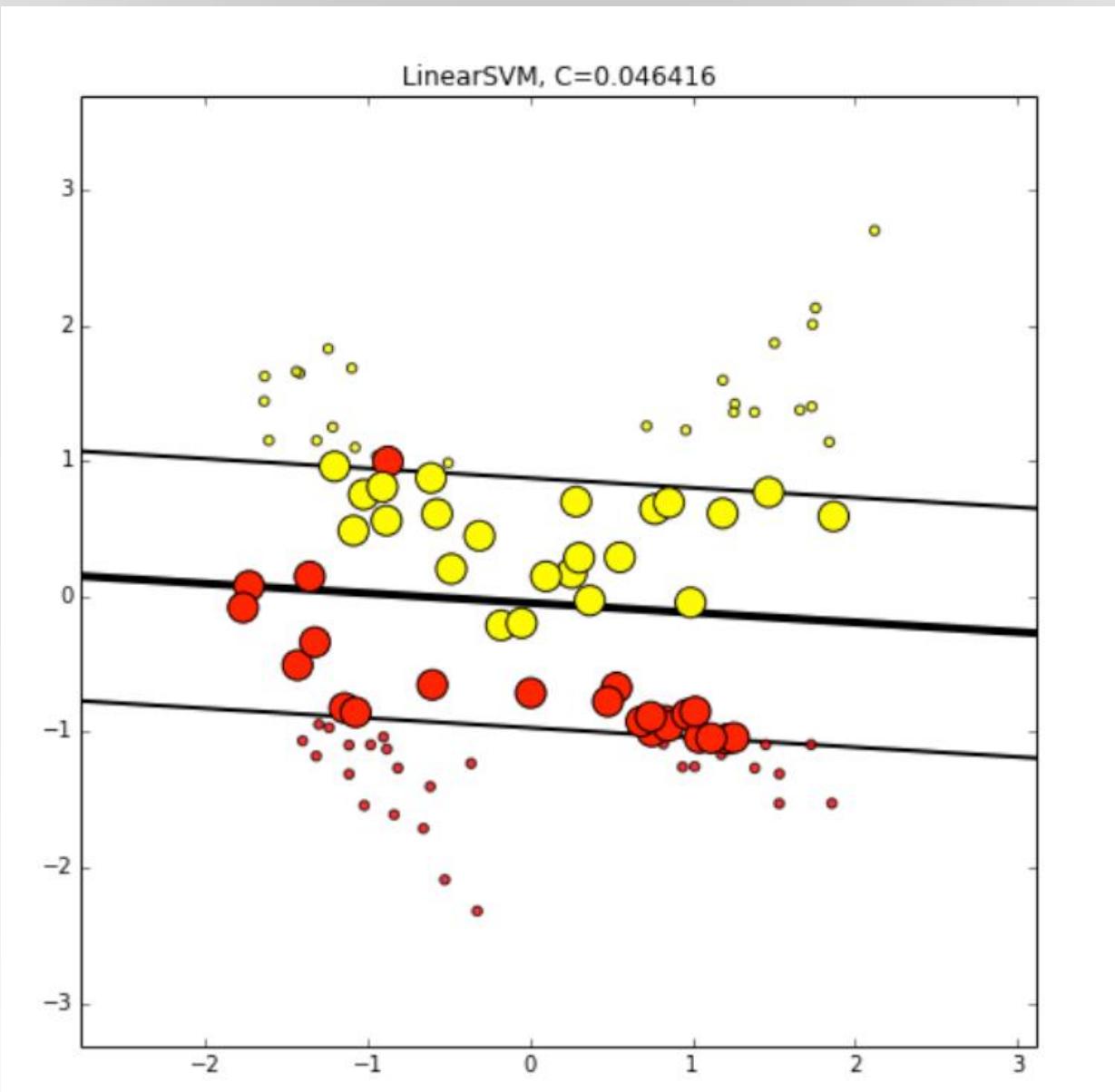
$$\left\{ \begin{array}{l} \frac{1}{2} \|w\|^2 + \textcolor{red}{C} \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{array} \right.$$

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

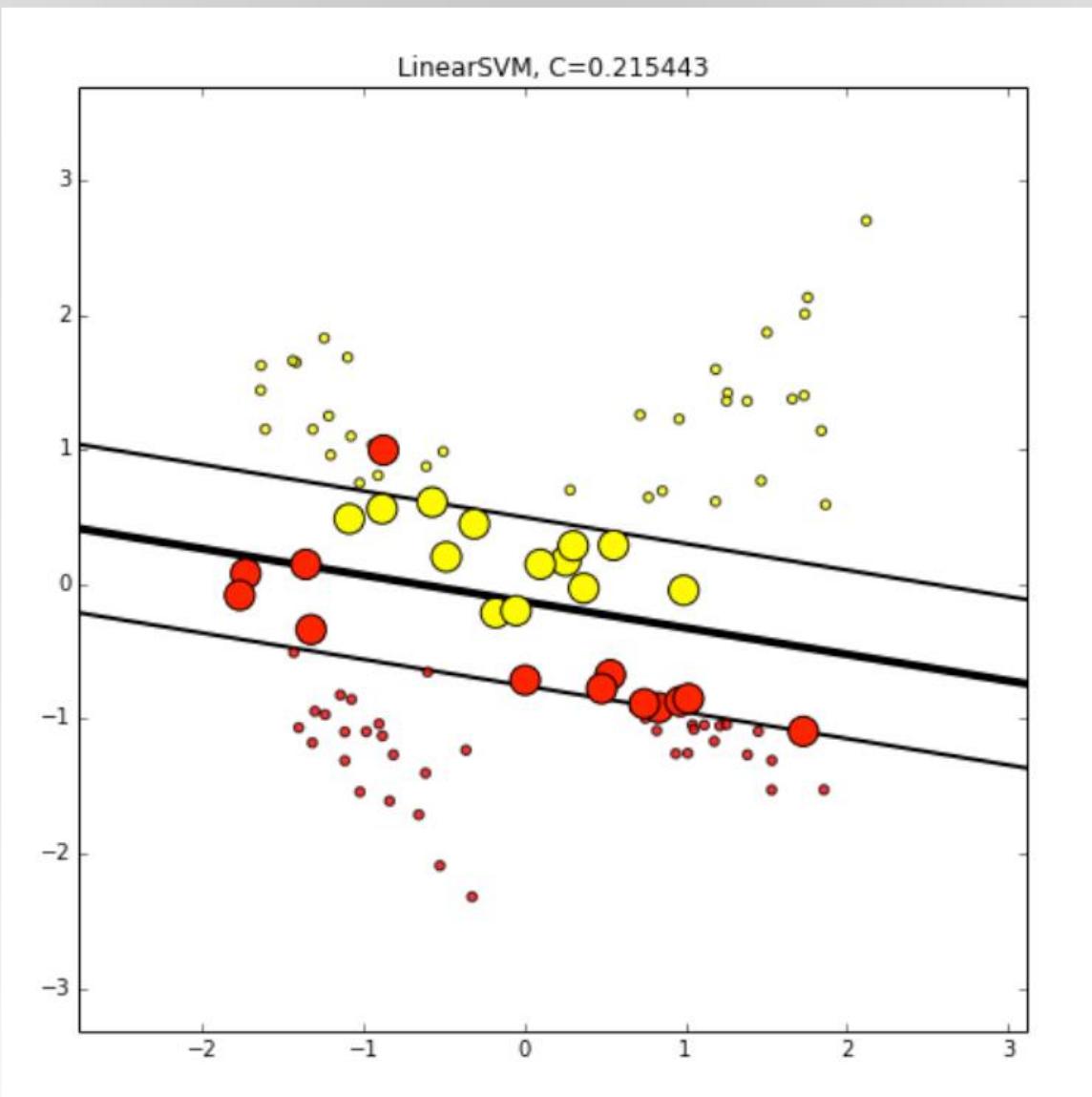
ЗНАЧЕНИЕ КОНСТАНТЫ С



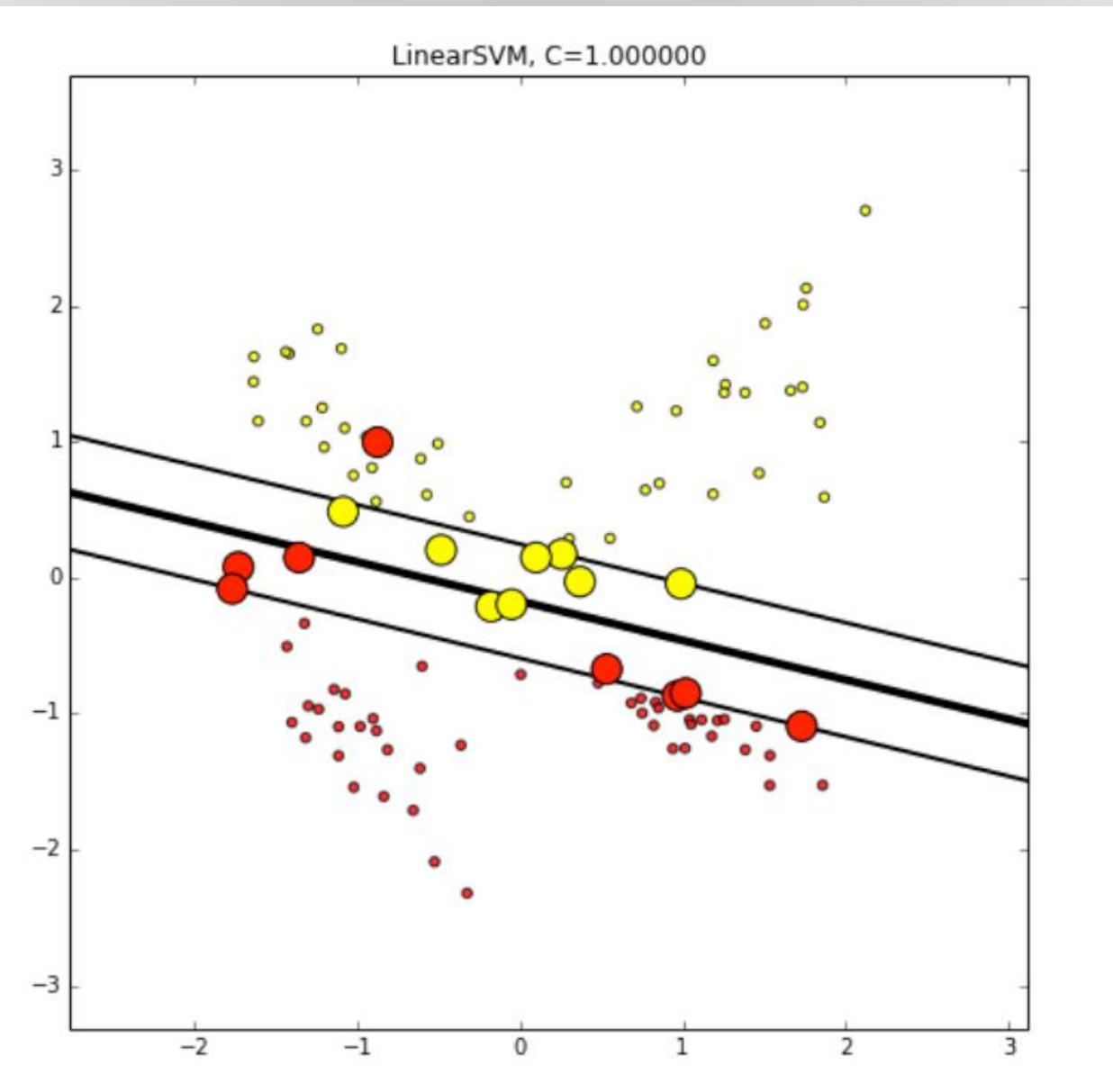
ЗНАЧЕНИЕ КОНСТАНТЫ С



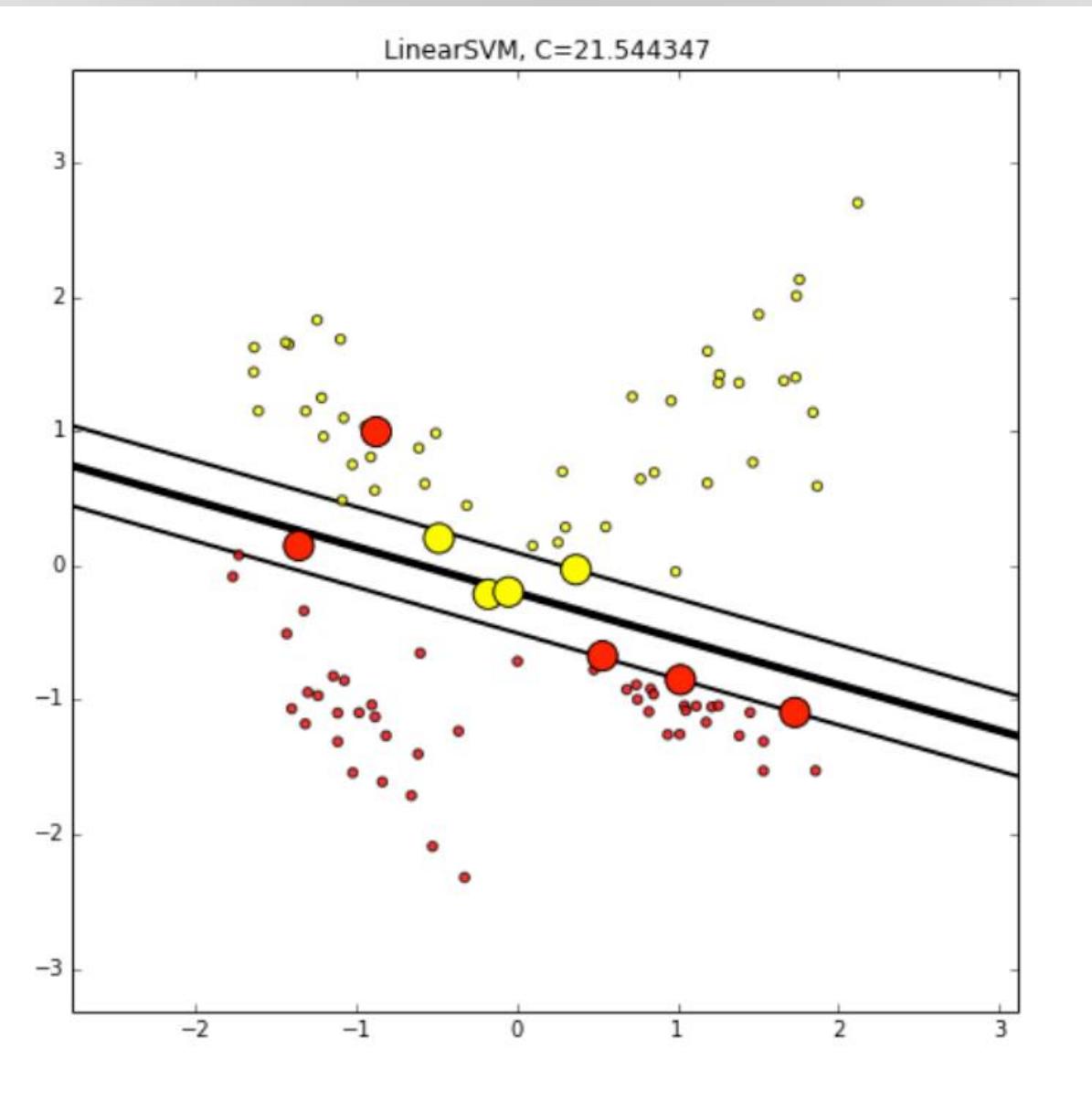
ЗНАЧЕНИЕ КОНСТАНТЫ С



ЗНАЧЕНИЕ КОНСТАНТЫ С



ЗНАЧЕНИЕ КОНСТАНТЫ С



ТИПЫ ОБЪЕКТОВ В SVM

