



# Занятие 4. Бинарная классификация

Колмагоров Евгений  
[ml.hse.dpo@yandex.ru](mailto:ml.hse.dpo@yandex.ru)

28 октября 2024

# План лекции

1. Постановка задачи классификации
2. Бинарная классификация
3. Линейные методы классификации
4. Функционал ошибки
5. Метрики качества



# Вспомним предыдущий материал

Прежде чем начать попробуем ответить на следующие, вопросы:

- В чём заключается задача классификации?
- Какие виды бывают?
- В чём ключевое отличие задачи классификации от регрессии?

# Напоминание. Классификация

Задача *классификации* - определить к какому из ограниченного набора классов принадлежит рассматриваемый объект

**iris setosa**



petal

sepal

**iris versicolor**



petal

sepal

**iris virginica**



petal

sepal

# Напоминание. Виды задач классификации

В случае если множество допустимых значений  $y \subset \mathbb{N}$  - это натуральные числа, то решается задача классификации:

- Если  $y = \{0, 1\}$  - бинарная классификация
- Если  $y = \{1, \dots, M\}$  - многоклассовая классификация на  $M$  непересекающихся классов
- Если  $y = \{0, 1\}^M$  - многоклассовая классификация с  $M$  пересекающимися классами. Например, объект с  $Y = \{0, 1, 1, 0, 1\}$ , относится ко 2, 3 и 5 классу

# Построение решения для задачи классификации

При построении решения для задачи классификации необходимо ответить на те же самые вопросы, которые возникают при решении задачи регрессии (да и вообще любой задачи МО):

- Как будет выглядеть алгоритм?
- Какой будет функционал ошибки?
- Какие использовать метрики качества для его оценки?



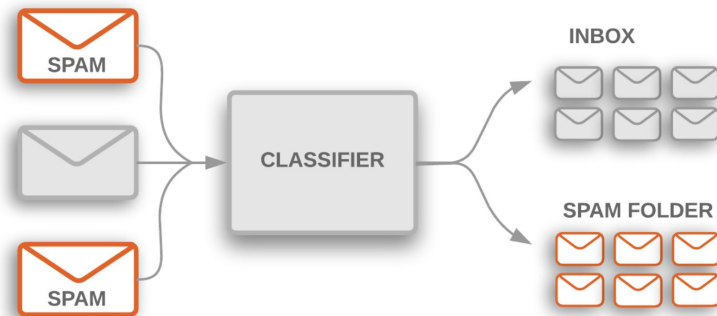
# Бинарная классификация

Начнём знакомство с простейшего вида классификации - бинарной, где метки классов  $Y$  состоит всего из двух видов:

- *положительных (+)*
- *отрицательных (-)*

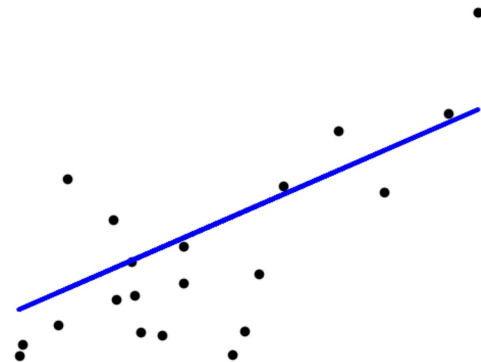
Несмотря на свою простоту, множество задач сводится к данному виду классификации:

- Определение болезни?
- Содержание неприемлемого контента?
- Допуск к секретным данным?
- И ещё множество других задач.....





Можно ли с помощью линейной модели  
построить алгоритм бинарной  
классификации?





# Правильный ответ

Да, с помощью модели линейной регрессии можно решать задачу бинарной классификации если *данные могут быть линейно разделимы* и в алгоритм добавить следующие изменения

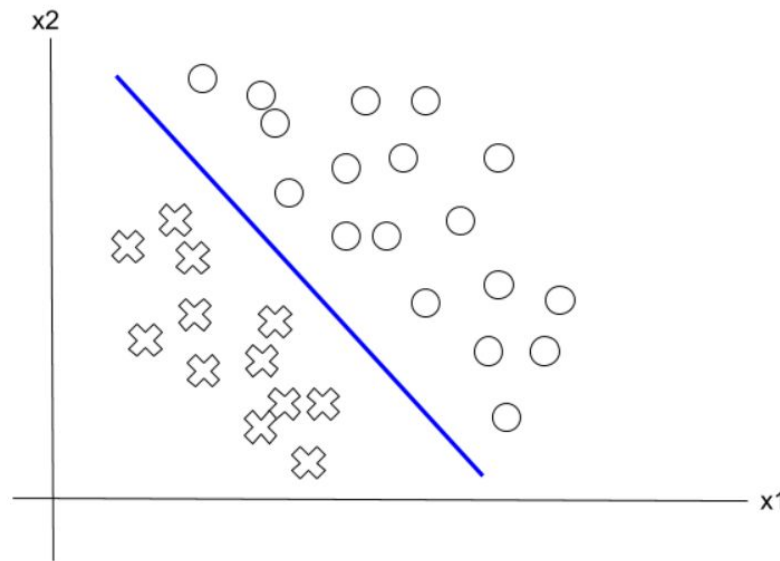
1. Изменить функцию ответа модели  $a(x, w)$  на бинарную  $+1/-1$
2. При обучении использовать другой функционал ошибки  $Q(a, X)$



# Линейная разделяемость данных

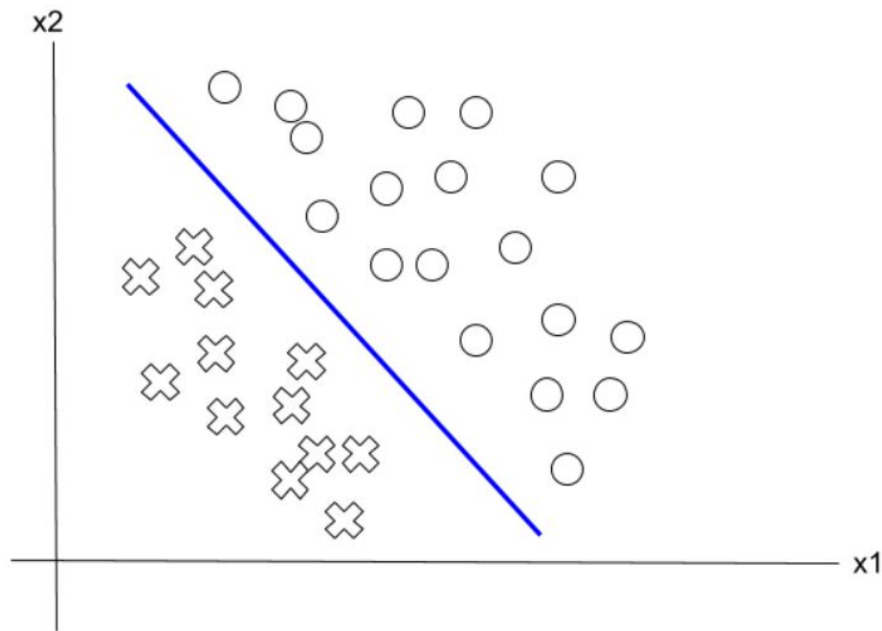
**Определение:** Данные линейно отделимы друг от друга, если можно построить гиперплоскость  $A$  в пространстве признаков  $X$  так, что первый класс лежит по одну сторону от плоскости, а второй класс по другую

**Замечание:** В реальной жизни редко, когда можно встретить данные, которые могут быть идеально поделены линейной гиперплоскостью



# Бинаризация ответов

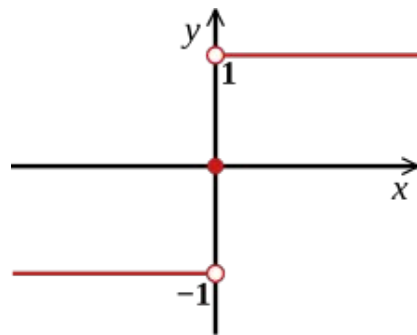
Как можно в линейной регрессии перейти к бинарному ответу +1 и -1?



# Функция Sign

Вспомним как выглядит функция  $sign(x)$ :

$$sign(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$



Теперь добавим к ответам линейной регрессии функцию sign:

$$a(x, w) = sign(\sum_{i=0}^d w_i x_i)$$

# Бинарная классификация

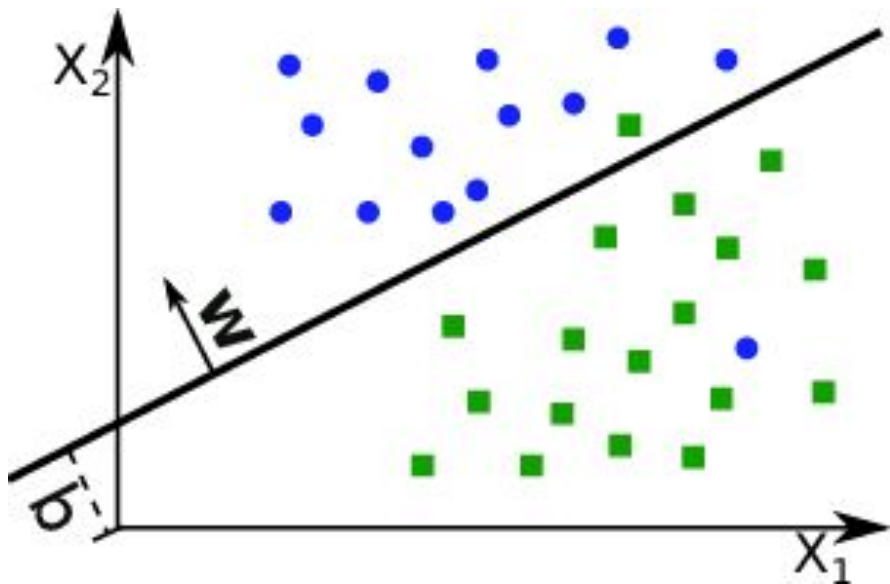
Модель линейного классификатора:

$$a(x, w) = \textit{sign}(\sum_{i=0}^d w_i x_i)$$

- Если  $\sum_{i=0}^d w_i x_i > 0$ , то  $\textit{sign}(\sum_{i=0}^d w_i x_i) = +1$ , то объект отнесён к положительному классу;
- Если  $\sum_{i=0}^d w_i x_i < 0$ , то  $\textit{sign}(\sum_{i=0}^d w_i x_i) = -1$ , то объект отнесён к отрицательному классу;
- Если  $\sum_{i=0}^d w_i x_i = 0$ , то объект лежит на самой разделяющей гиперплоскости.  
Так как сумма задаёт уравнение гиперплоскости в  $d+1$  мерном пространстве

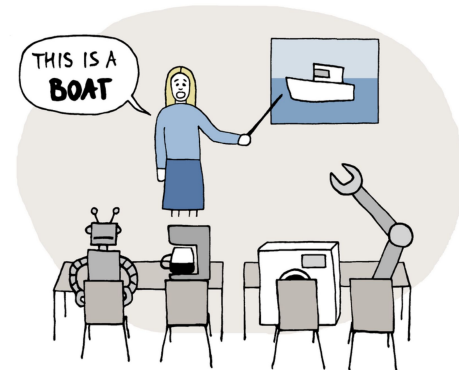
# Геометрическая интерпретация

Так как веса  $\mathbf{w}=(w_0, w_1, \dots, w_d)$  - задаёт гиперплоскость вместе со смещением  $w_0(b)$ , то вектор  $\mathbf{w}^n=(w_1, \dots, w_d)$  задаёт вектор нормали к разделяющей поверхности. Те объекты, которые лежат в стороне  $\mathbf{w}^n$ , имеют положительное значение суммы  $\sum_{i=0}^d w_i x_i$ , а те которые лежат в противоположной стороне имеют отрицательное значение





Как обучать линейную классификацию?



# Основное свойство функционала ошибки

Как и любому алгоритму машинного обучения для поиска оптимальных весов  $w^*$ , необходим функционал ошибки  $Q(a, X)$ , который удовлетворял бы следующему свойству - **функционал  $Q$  тем меньше, чем лучше качество решения задачи на обучающем множестве  $X$**

$$w^* = \operatorname{argmin}_w Q(a, X)$$



# Ошибка на одном объекте

Введём ошибку классификации на одном объекте:

$$err_i = I[a(x_i, w) \neq y_i]$$

$I$  - индикаторная функция того, что ответ алгоритма отличается от целевой переменной  $y$



# Ошибка на всём обучающем множестве

Тогда средняя ошибка на всём тренировочном множестве  $X$ :

$$Q(a, X) = \frac{1}{N} \sum_{i=0}^d err_i = \frac{1}{N} \sum_{i=0}^N I[a(x_i, w) \neq y_i] \rightarrow \min_w$$

*Вопрос: можно ли напрямую проводить минимизацию функционала  $Q$ ?*

# Ошибка на всём обучающем множестве

Тогда средняя ошибка на всём тренировочном множестве  $X$ :

$$Q(a, X) = \frac{1}{N} \sum_{i=0}^d err_i = \frac{1}{N} \sum_{i=0}^N I[a(x_i, w) \neq y_i] \rightarrow \min_w$$

*Вопрос: можно ли напрямую проводить минимизацию функционала  $Q$ ?*

*Ответ: Проводить минимизацию не дифференцируемого функционала  $Q$  сложная задача и стандартными методами она не решается.*

*Решение: Попробуем придумать другой функционал  $Q^*$ , который был бы гладкой аппроксимацией  $Q$ .*

# Функция отступа

Введём функцию отступа (margin) для объекта  $x_i$  с меткой  $y_i$ :

$$M_i = y_i \cdot a(x_i, w) = y_i \cdot (w, x_i)$$

Утверждение: Решение задачи:

$$Q(a, X) = \frac{1}{N} \sum_i^N I[a(x_i, w) \neq y_i] \rightarrow \min_w$$

эквивалентно задаче:

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

# Доказательство эквивалентности

Функция отступа принимает отрицательные значения только тогда, когда метка  $y_i$  на объекте  $x_i$  не совпадает с ответом алгоритма.

$$I[a(x_i) \neq y_i] = I[M_i < 0] = I[y_i \cdot a(x_i, w) < 0]$$

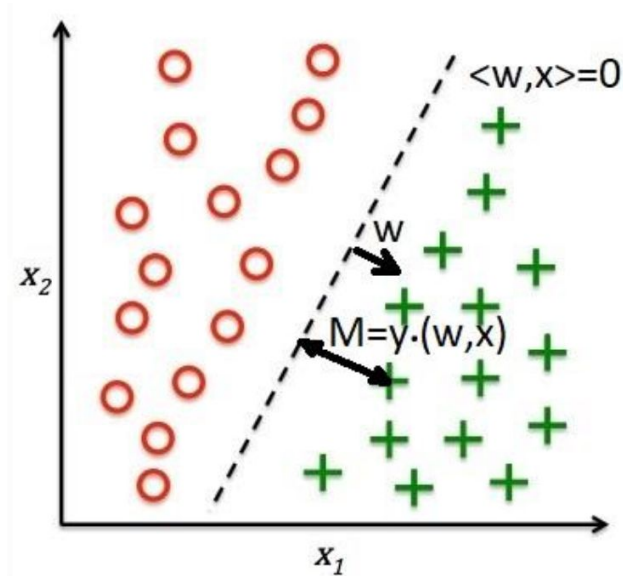
Рассмотрим всевозможные варианты ответа алгоритма и целевой переменной  $y$

- $y_i = 1, a(x_i) = 1: I[M_i < 0] = 0$
- $y_i = -1, a(x_i) = 1: I[M_i < 0] = 1$
- $y_i = 1, a(x_i) = -1: I[M_i < 0] = 1$
- $y_i = -1, a(x_i) = -1: I[M_i < 0] = 0$

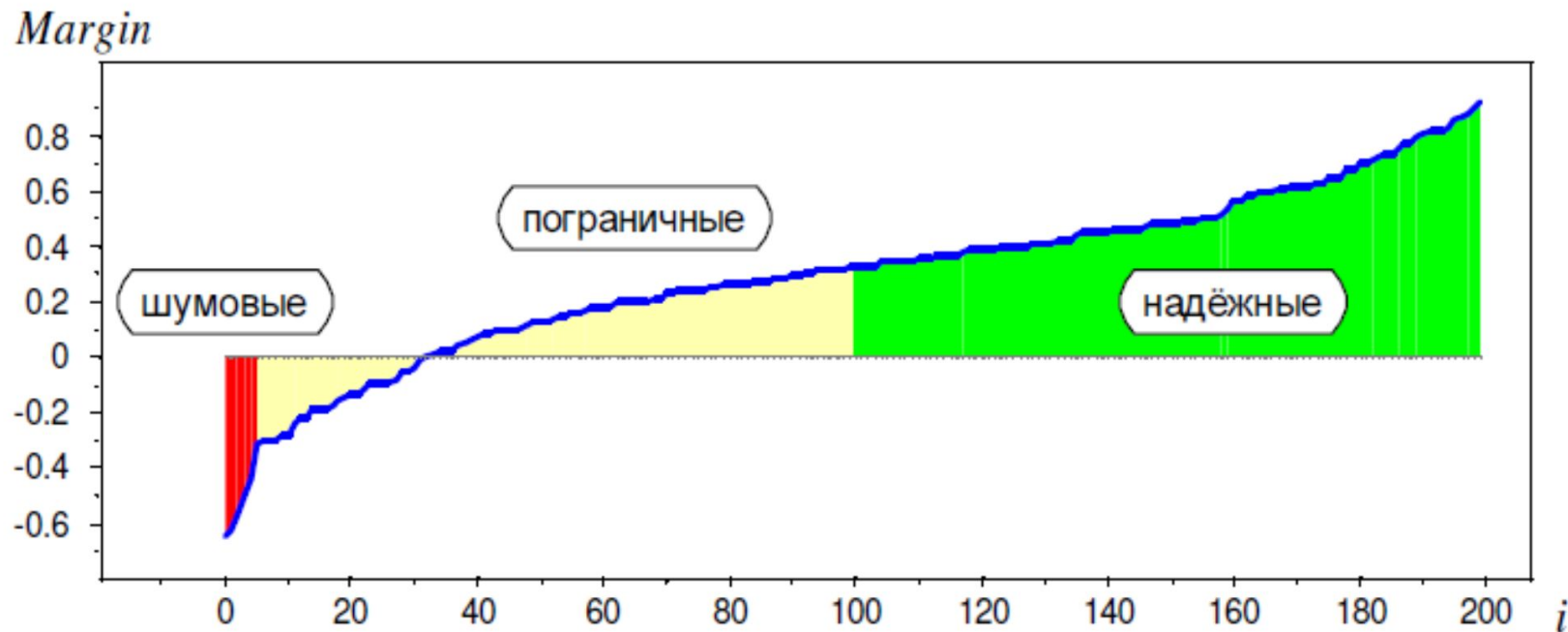
Таким образом знак отступа  $M_i$  говорит о корректности классификации на  $i$ -ом объекте

# Абсолютное значение отступа

Абсолютная величина отступа  $M_i$  обозначает степень уверенности классификатора  $a(x_i, w)$  в своём ответе. Чем ближе  $M_i$  к нулю, тем меньше уверенности в ответе.



# Ранжирование объектов по возрастанию отступа



# Дифференцируемость функционала

*Вопрос: Дифференцируема ли функция отступа  $M_i$ ? Дифференцируем ли функционал  $Q$  с порогом  $M$ ?*

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

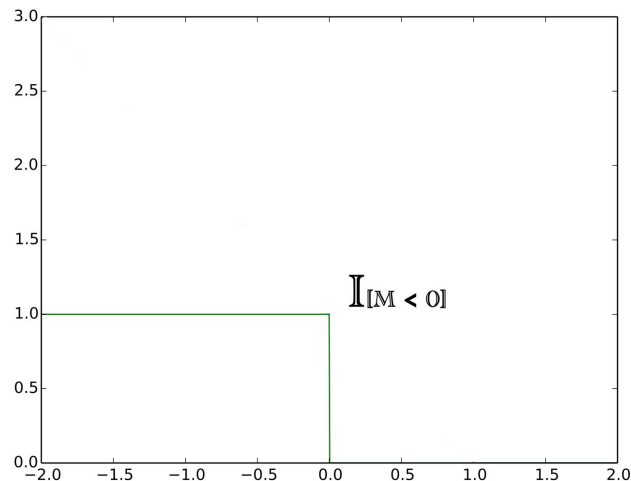


# Дифференцируемость функционала

*Вопрос: Дифференцируема ли функция отступа  $M_i$ ? Дифференцируем ли функционал  $Q$  с порогом  $M$ ?*

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

*Ответ: Функция  $I[M_i < 0]$  не является дифференцируемой в нуле, следовательно и сумма индикаторов тоже не дифференцируемая функция*



# Верхняя оценка функции потерь

Для оптимизации функционала  $Q(a, X)$  будем использовать верхние оценки для индикаторной функции  $I[M_i < 0]$ .

Определение: Функция  $L(x_i)$  является верхней оценкой функции  $F(x_i)$ , если для любого объекта  $x_i$  выполнено неравенство:  $F(x_i) \leq L(x_i)$

В качестве верхней оценки будем использовать **дифференцируемую** функцию.

*Вопрос: Почему оптимизация функционала будет приводить к оптимизации исходного функционала  $Q$ ?*

$$\hat{Q}(a, X) = \frac{1}{N} \sum_{i=0}^N L(x_i, y_i) \rightarrow \min_w$$

# Оптимизация верхней оценки

Поскольку между каждым элементом суммы выполняется нестрогое неравенство  $I[M_i < 0] \leq L(x_i, y_i)$ , то выполняется неравенство между функционалами

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

Поэтому минимизация  $\hat{Q}(a, X) \rightarrow \min_w$ , будет приводить к минимизации  $Q(a, X)$

# Примеры верхних оценок

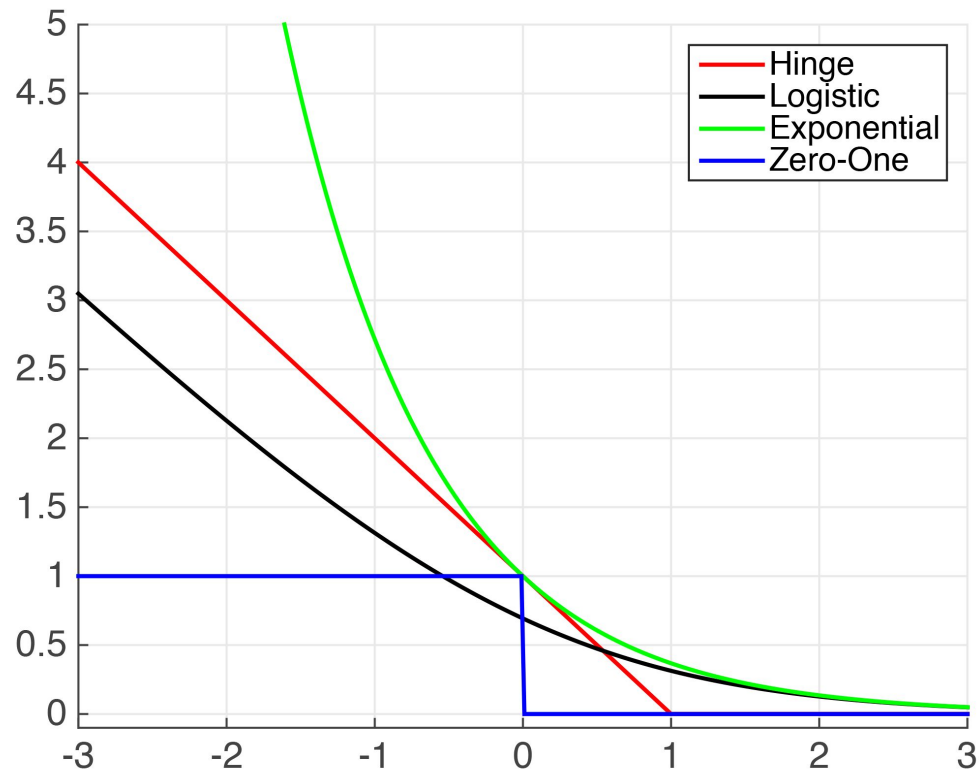
Примеры верхних оценок:

- $L(M) = \log(1 + e^{-M})$  – логистическая функция потерь (logistic)
- $V(M) = (1 - M)_+ = \max(0, 1 - M)$  – кусочно-линейная функция со сдвигом (hinge)
- $H(M) = (-M)_+ = \max(0, -M)$  – кусочно-линейная функция потерь
- $E(M) = e^{-M}$  – экспоненциальная функция потерь (exponential)
- $I[M < 0]$  – пороговая функция потерь (zero-one)
- $S(M) = 2/(1 + e^{-M})$  – сигмоидная функция

---

Напоминание: отступ  $M = y_i(w, x_i)$

# Гладкие аппроксимации



# Гладкие аппроксимации

*Каждая из введённых аппроксимаций при оптимизации своего функционала ошибки будет давать различные результаты. Поэтому разные функции потерь определяют различные классификаторы.*

Сегодня будет рассматриваться логистическая регрессия, в которой в качестве верхней оценки используется **логистическая** функция потерь:

$$L(x_i, y_i) = \log(1 + e^{-y_i \cdot (w, x_i)})$$

*Вопрос: Как будем оптимизировать?*

$$\hat{Q}(a, X) = \frac{1}{N} \sum_{i=0}^N L(x_i, y_i) = \frac{1}{N} \sum_{i=0}^N \log(1 + e^{-y_i \cdot (w, x_i)}) \rightarrow \min_w$$

# Градиентный спуск

После того как есть гладкий функционал  $Q(a, X)$  алгоритм обучения в задаче классификации не отличается от обучения задачи регрессии.

Используем оптимизацию гладких функционалов градиентным спуском:

$$w^k = w^{k-1} - \eta \cdot \nabla_w \hat{Q}(a, X)$$

Метрики качества?



# Как измерять качество обученного алгоритма?

Самая простая метрика — доля правильных ответов модели (Accuracy):

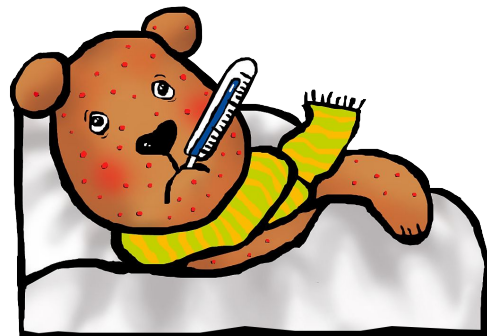
$$Accuracy(a, X) = \frac{1}{N} \sum_{i=0}^N I[a(x_i) = y_i]$$

*Вопрос: Чем плоха и хороша данная метрика?*

# Пример. Классификация больных

Предположим, что некоторое заболевание встречается 3 раза на 1000 человек. Задача классификатора состоит в том, чтобы по анализам найти тех, кто болен, а кто нет

*Вопрос:* Чему будет равно Ассигасу константного классификатора, который говорит всем, что все здоровы?

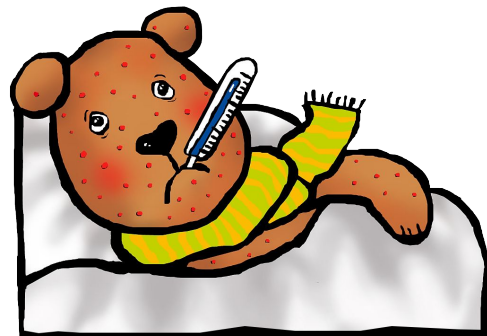


# Пример. Классификация больных

Предположим, что некоторое заболевание встречается 3 раза на 1000 человек. Задача классификатора состоит в том, чтобы по анализам найти тех, кто болен, а кто нет

*Вопрос:* Чему будет равно Ассигасу константного классификатора, который говорит всем, что все здоровы?

*Ответ:* Качество работы такого классификатора будет 0.997, что в целом очень хорошо и близко к 1.  
Но все ли классы для нас равнозначны?



# Матрица ошибок

Так как в бинарной классификации множества ответов это две метки +/- и множество предсказаний те же самые две метки +/-, то множество ответов состоит из 4 возможных вариантов.

		Actual Value	
		Positive	Negative
Predicated Value	Positive	TP	FP
	Negative	FN	TN

**Confusion Matrix**

# False Positive & False Negative

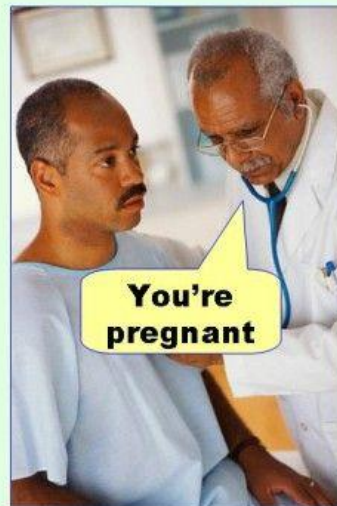
Из 4-х возможных вариантов пар предсказание-ответ 2 являются ошибочными.

В статистике такие ошибки называют **ошибками первого и второго рода**.

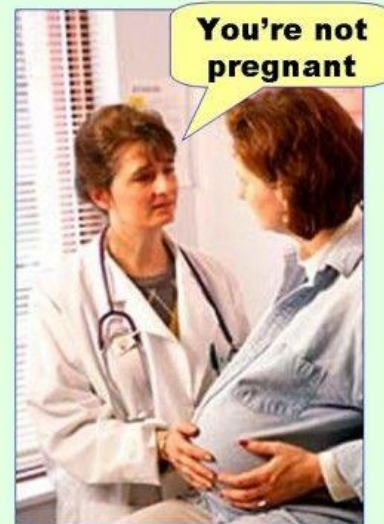
В зависимости от решаемой задачи необходимо понимать какая из ошибок более критична.

*Вопрос:* Какая из ошибок более критична для предыдущего примера?

**Type I error**  
(false positive)



**Type II error**  
(false negative)



# Сравнение моделей по матрице ошибок

Предположим, что есть 2 модели кредитного скоринга со следующими матрицами ошибок. Какая из моделей решает задачу лучше?

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

Модель 1

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

Модель 2

# Метрики качества классификации: Precision & Recall

Precision (точность):

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

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

# Сравнение по точности

Модель 1:

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

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

Модель 1

Модель 2:

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

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

Модель 2



# Метрики качества классификации: Precision & Recall

Precision (точность):

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

Показывает, то 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

# Сравнение по полноте

Модель 1:

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

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

Модель 1

Модель 2:

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

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


Модель 2

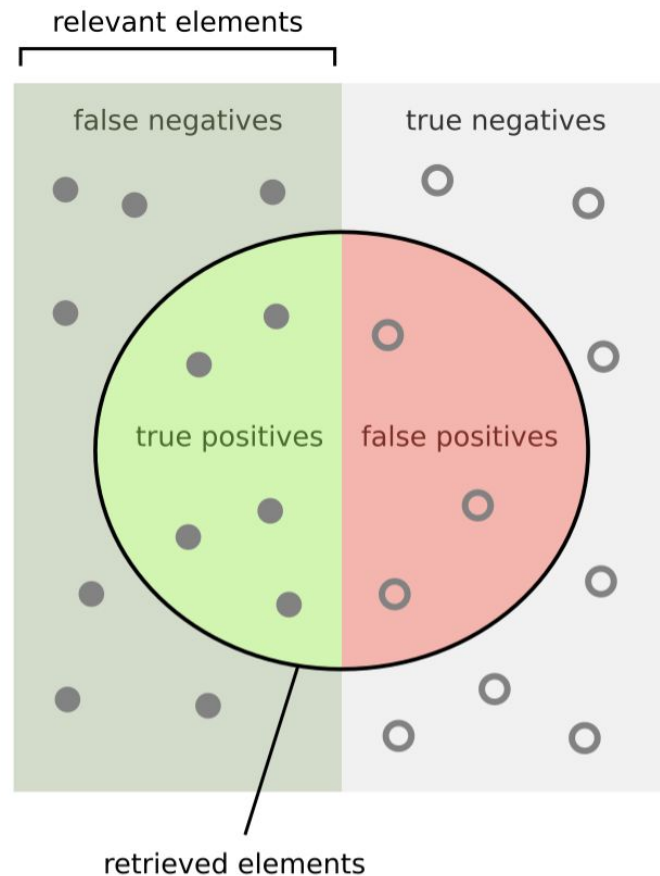
# Точность и полнота

How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$


How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$




# Ошибки классификации $F_1$ -мера

У precision и recall есть один существенный недостаток - это парные многомерные метрики, качество которых нужно сравнивать в совокупности.

Поэтому возникает вопрос, что лучше иметь в модели:

- 0.8 precision и 0.6 recall
- Или 0.6 precision и 0.8 recall

Для того, чтобы свести качество решения задачи в одно число удобное для сравнения применяют различные способы усреднения

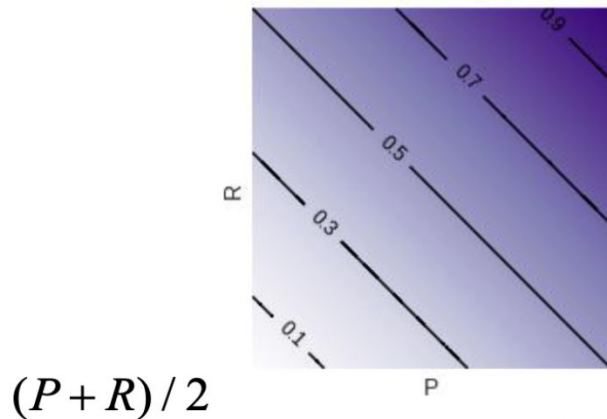
# Ошибки классификации $F_1$ -мера

$F_1$  - мера как среднее гармоническое точности и полноты:

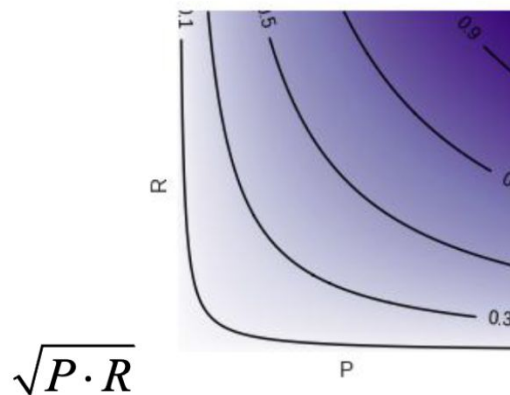
$$F_1(a, X) = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

# Ошибки классификации $F_1$ -мера

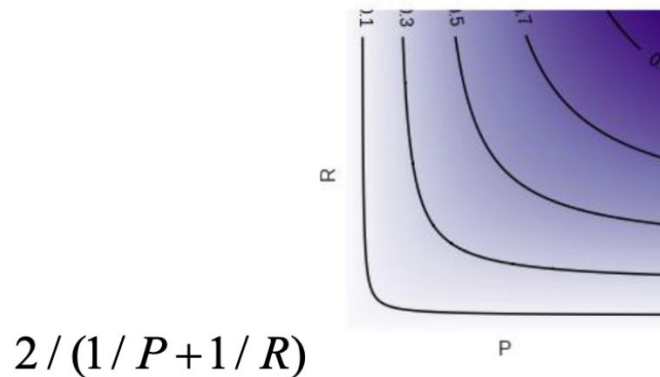
Почему используется F-мера



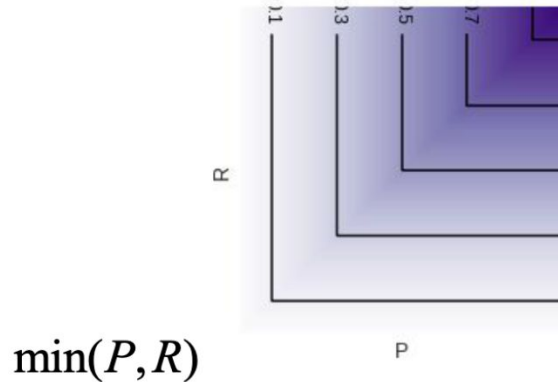
$$(P+R)/2$$



$$\sqrt{P \cdot R}$$



$$2 / (1/P + 1/R)$$



$$\min(P, R)$$

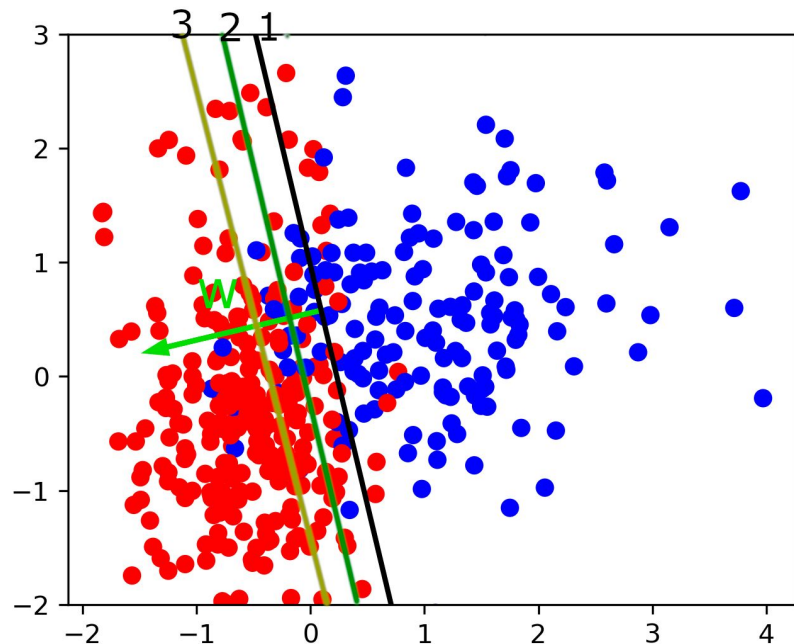
# Настройка точности и полноты

В зависимости от вида задачи можно варьировать точность и полноту.

Вспомним, что отступ  $M_i$  ещё несёт смысл уверенности модели

Можно увеличить уверенность модели, добавив порог  $\delta$ :

$$a_\delta(x, w) = \text{sign}((\sum_{i=0}^d w_i x_i) - \delta)$$



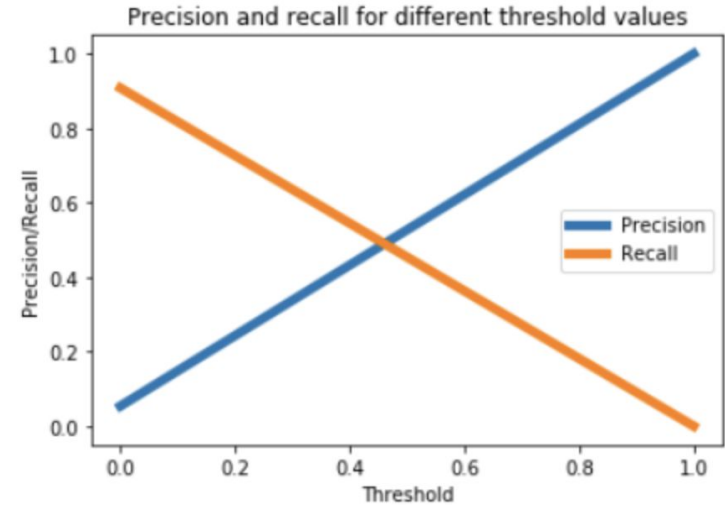
*Разделяющие гиперплоскости при различных значениях порога*

# Настройка точности и полноты

При увеличении порога  $\delta$  происходит увеличение точности за счёт уменьшения полноты.

Так как **каждое значение порога** определяет **свой классификатор**, то у каждого из них будет своё качество работы

*Вопрос: Какая из моделей лучше  $a_1$  с порогом  $\delta_1$  или  $a_2$  с порогом  $\delta_2$*



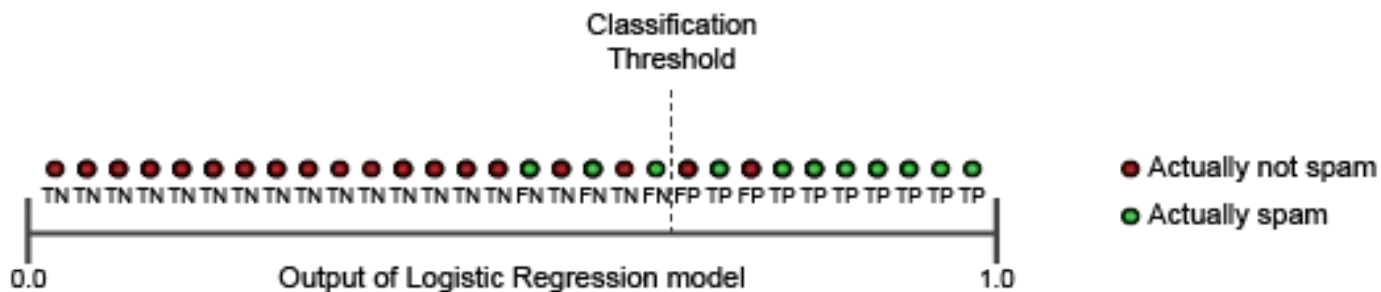
*Пример зависимости точности и полноты от значения порога*



# Измерение качества работы на все возможные

Будем измерять качество всего семейства классификаторов независимо от выбранного порога  $\delta$ .

*Как делать:* будем варьировать порог от  $(-\infty, +\infty)$  на валидации и смотреть на значения нормированные значения TP и FP.



# True Positive Rate & False Positive Rate

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

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

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

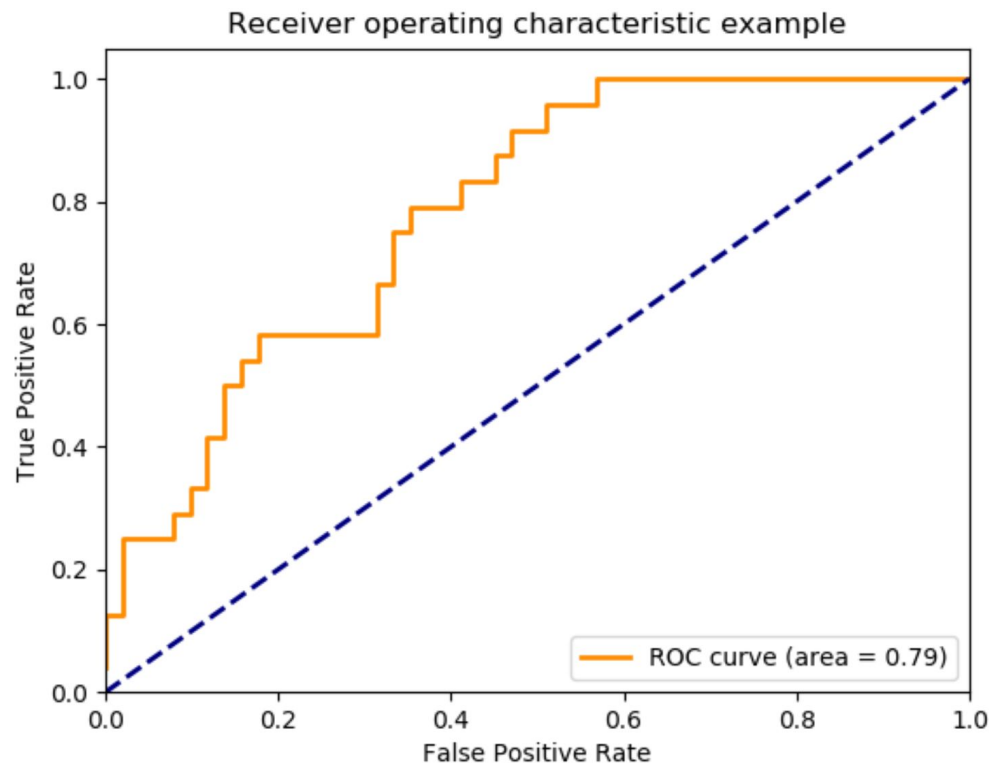
$$FPR = \frac{FP}{FP+TN}$$

		Actual Value	
		Positive	Negative
Predicated Value	Positive	TP	FP
	Negative	FN	TN

**Confusion Matrix**

# ROC - кривая

ROC – кривая, состоящая из точек с координатами (FPR, TPR) для всевозможных порогов



# Как построить

- 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:**  $\delta = 0.7$ , то есть  $a(x) = I[b(x) > 0.7]$

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

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

# Как построить

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

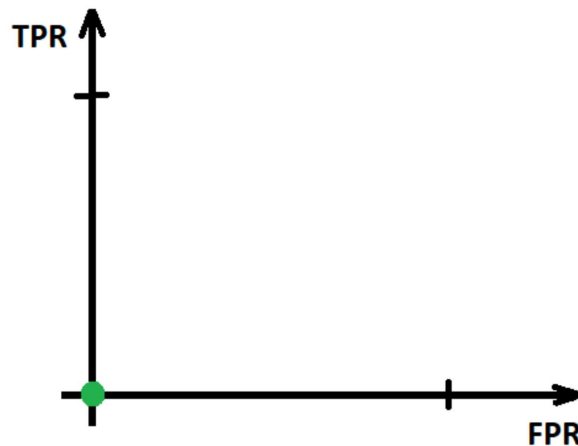
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

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

**1 шаг:**  $\delta = 0.7$ , то есть  $a(x) = I[b(x) > 0.7]$

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

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



# Как построить

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

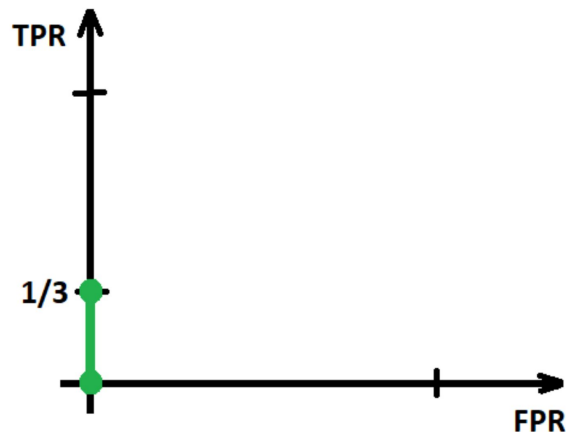
$b(x)$	0.2	0.4	0.1	<b>0.7</b>	0.05
$y$	-1	+1	-1	<b>+1</b>	+1

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

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

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

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



# Как построить

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

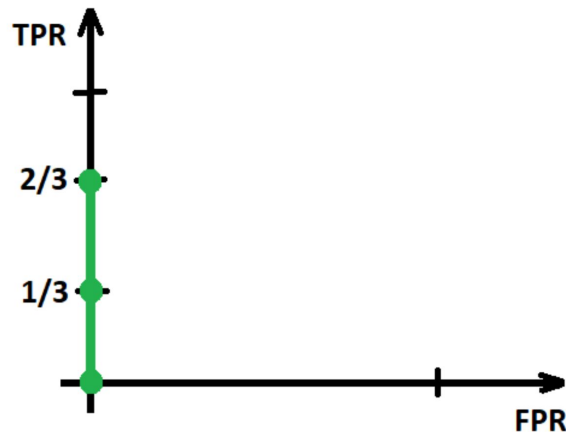
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

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

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

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

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



# Как построить

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

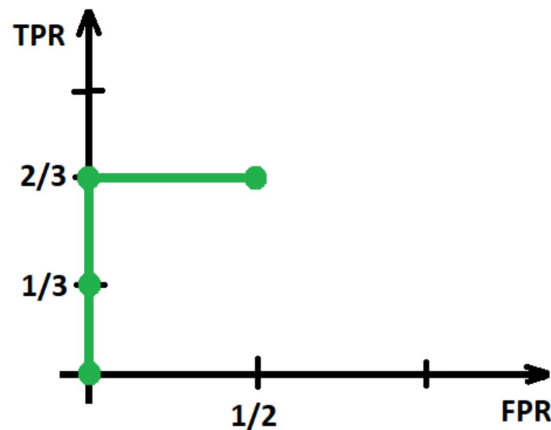
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

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

4 шаг:  $\delta = 0.1$ , то есть  $a(x) = I[b(x) > 0.1]$

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

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





# Как построить

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

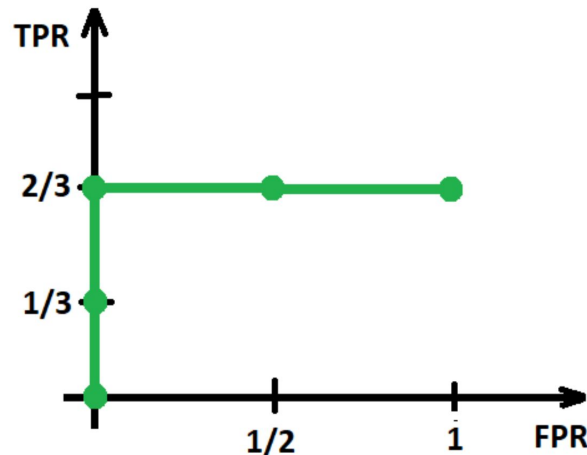
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

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

**5 шаг:**  $\delta = 0.05$ , то есть  $a(x) = I[b(x) > 0.05]$

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

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



# Как построить

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

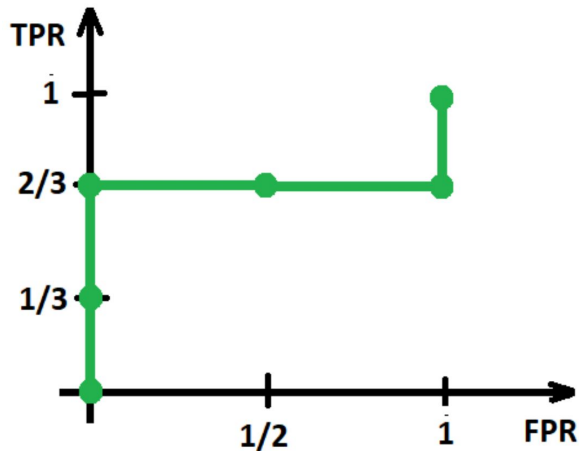
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

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

6 шаг:  $\delta = 0$ , то есть  $a(x) = I[b(x) > 0]$

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

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



# ROC AUC

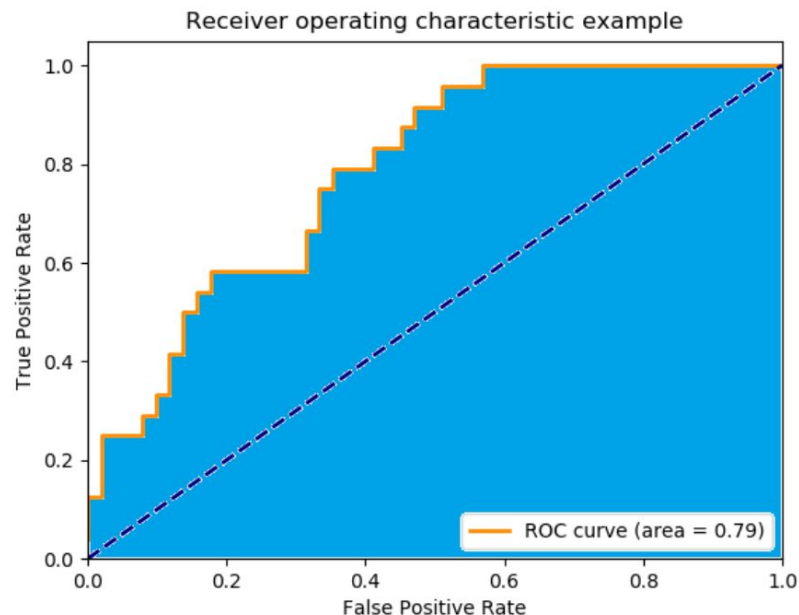
Как правило при классификации первичен вопрос не того, какую именно форму имеет кривая ROC, а какую площадь она имеет под собой.

AUC (area under curve) - площадь под ROC кривой.

$$AUC \in [0, 1]$$

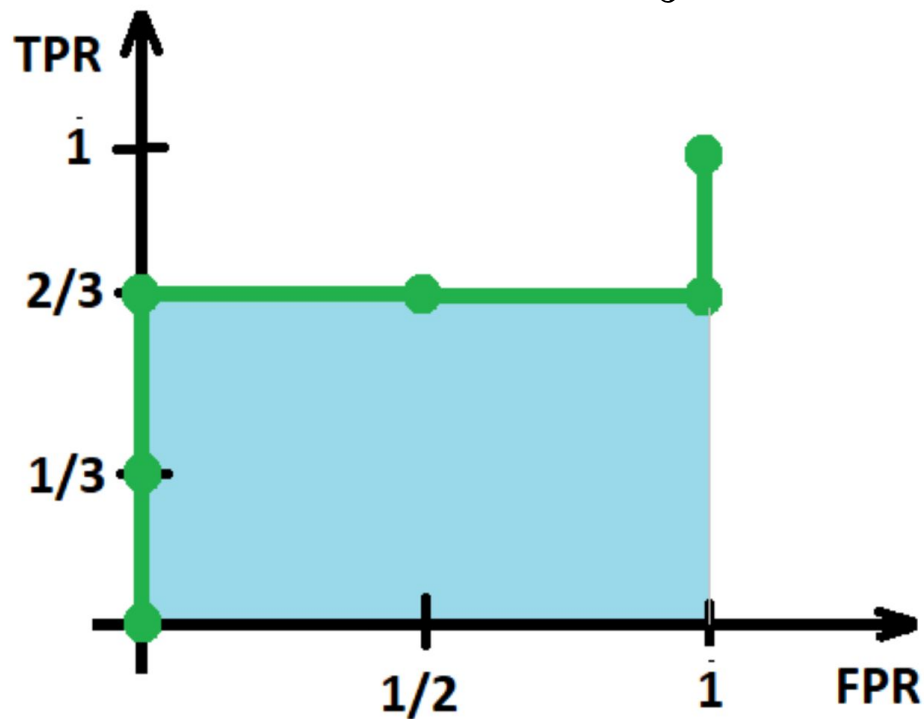
*Чему равен ROC AUC константного классификатора?*

*Чему равен ROC AUC идеального классификатора?*



## ROC-AUC из примера

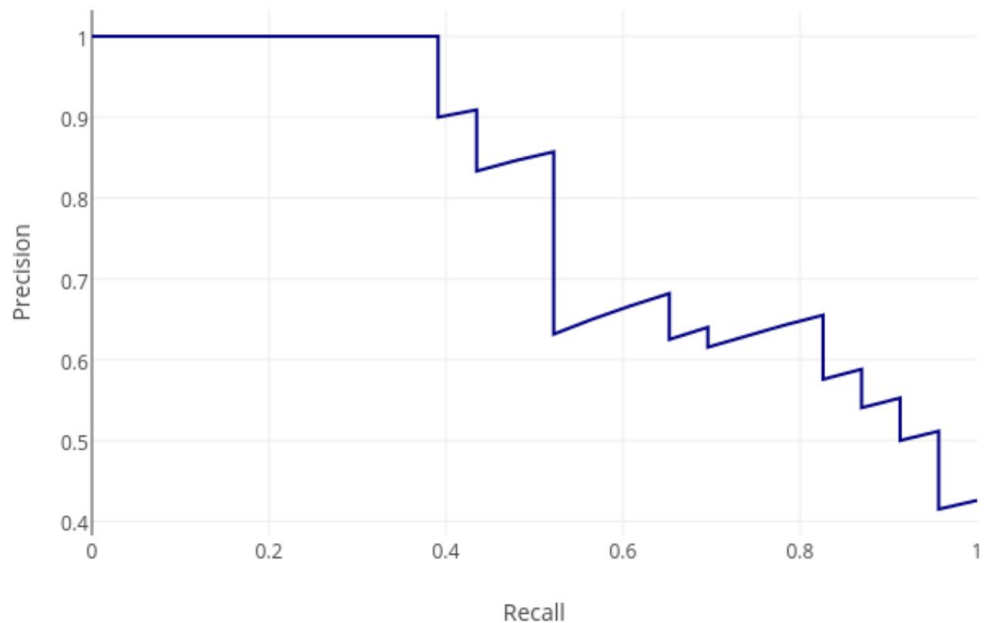
$$ROC\ AUC = \frac{2}{3}$$



# Precision-Recall кривая

Аналогично ROC AUC  
можно построить  
кривую в координатах  
Precision/Recall.

Precision-Recall example: AUC=0.79



# AUC PR

Также можно считать  
площадь под PR кривой

Precision-Recall example: AUC=0.79

