

# Машинное обучение

Лекция 5

Метрики качества. Переобучение.

Многоклассовая классификация.

Михаил Гущин

[mhushchyn@hse.ru](mailto:mhushchyn@hse.ru)

НИУ ВШЭ, 2024



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# На прошлой лекции

- ▶ Модель логистической регрессии:

$$\hat{y} = \sigma(Xw)$$

- ▶ Функция потерь log-loss:

$$L = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

- ▶ Мы хотим минимизировать  $L$ :

$$L \rightarrow \min_w$$

- ▶ Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)})$$

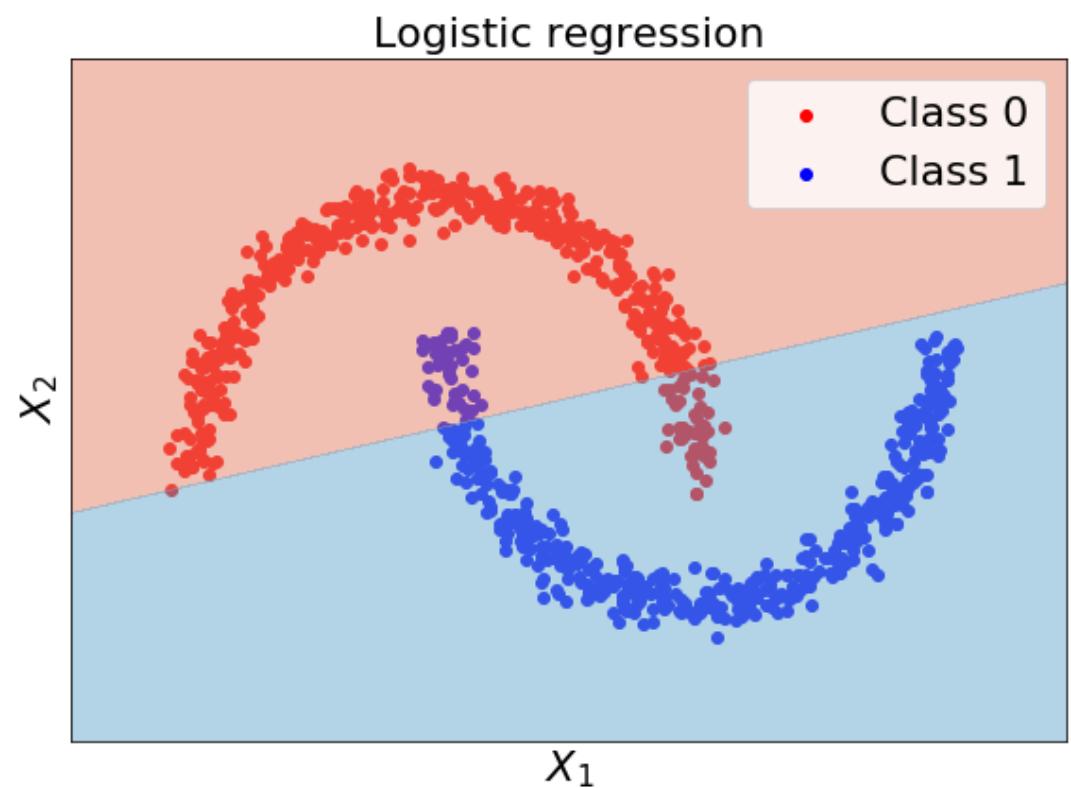
# Метрики качества для классификации



# Задача

Рассмотрим задачу бинарной классификации для некоторого набора данных.

Цель – **оценить качество классификатора**, определить как хорошо он разделяет объекты разных классов.



# Матрица ошибок (confusion matrix)

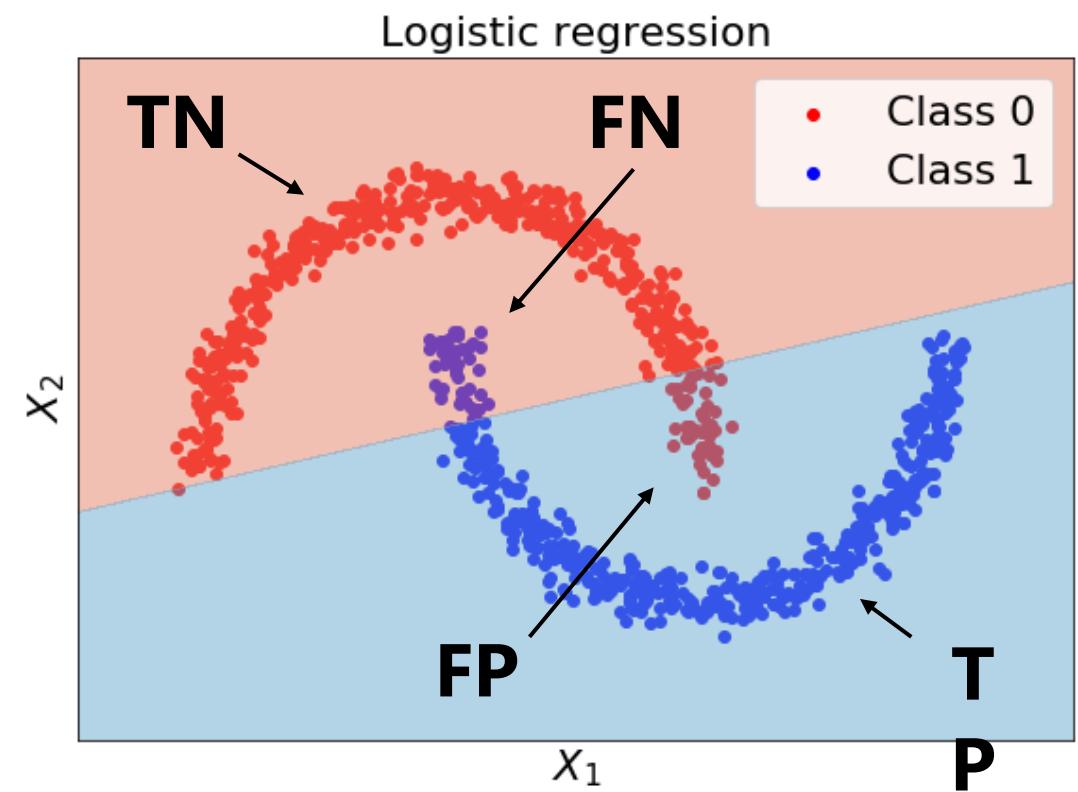
- ▶ **TP** (True Positive) – правильно предсказанные **1**
- ▶ **FP** (False Positive) – предсказанные как **1**, но правильно **0** (ошибка 1го рода)
- ▶ **TN** (True Negative) – правильно предсказанные **0**
- ▶ **FN** (False Negative) – предсказанные как **0**, но правильно **1** (ошибка 2го рода)

		PREDICTIVE VALUES	
		POSITIVE (1)	NEGATIVE (0)
ACTUAL VALUES	POSITIVE (1)	TP	FN
	NEGATIVE (0)	FP	TN



# Матрица ошибок (confusion matrix)

- ▶ **TP** (True Positive) – правильно предсказанные **1**
- ▶ **FP** (False Positive) – предсказанные как **1**, но правильно **0** (ошибка 1го рода)
- ▶ **TN** (True Negative) – правильно предсказанные **0**
- ▶ **FN** (False Negative) – предсказанные как **0**, но правильно **1** (ошибка 2го рода)



# Матрица ошибок (confusion matrix)

- ▶ Все **1** (*Pos*):

$$Pos = TP + FN$$

- ▶ Все **0** (*Neg*):

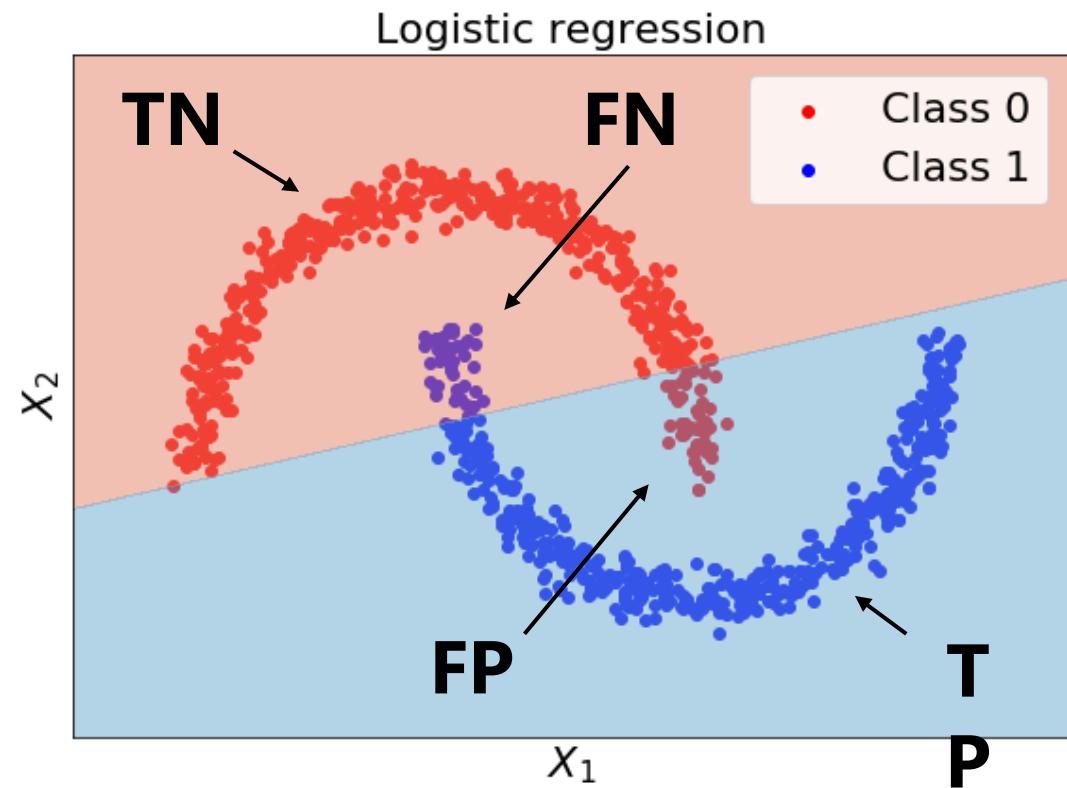
$$Neg = TN + FP$$

- ▶ Все прогнозы **1** (*PosPred*):

$$PosPred = TP + FP$$

- ▶ Все прогнозы **0** (*NegPred*):

$$NegPred = TN + FN$$



# Доля правильных ответов (accuracy)

- ▶ Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} = \frac{TP + TN}{Pos + Neg}$$

- ▶ Error rate:

$$\text{Error rate} = 1 - \text{Accuracy}$$

- ▶ Измеряет долю верных прогнозов во всех классах

# Точность (precision)

- ▶ Precision:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{PosPred}$$

- ▶ Показывает какая доля прогнозов **1** правильная

**Пример:** предсказали 100 объектов класса 1, но только 90 прогнозов верны.  
Тогда точность = 0.9.

# Полнота (recall)

- ▶ Recall:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{Pos}$$

- ▶ Показывает какую долю настоящих 1 классификатор предсказал правильно.

**Пример:** в данных 50 объектов класса 1, классификатор правильно предсказал 40 этих объектов. Тогда полнота = 0.8.

# F-мера

- ▶  $F_1$ -score:

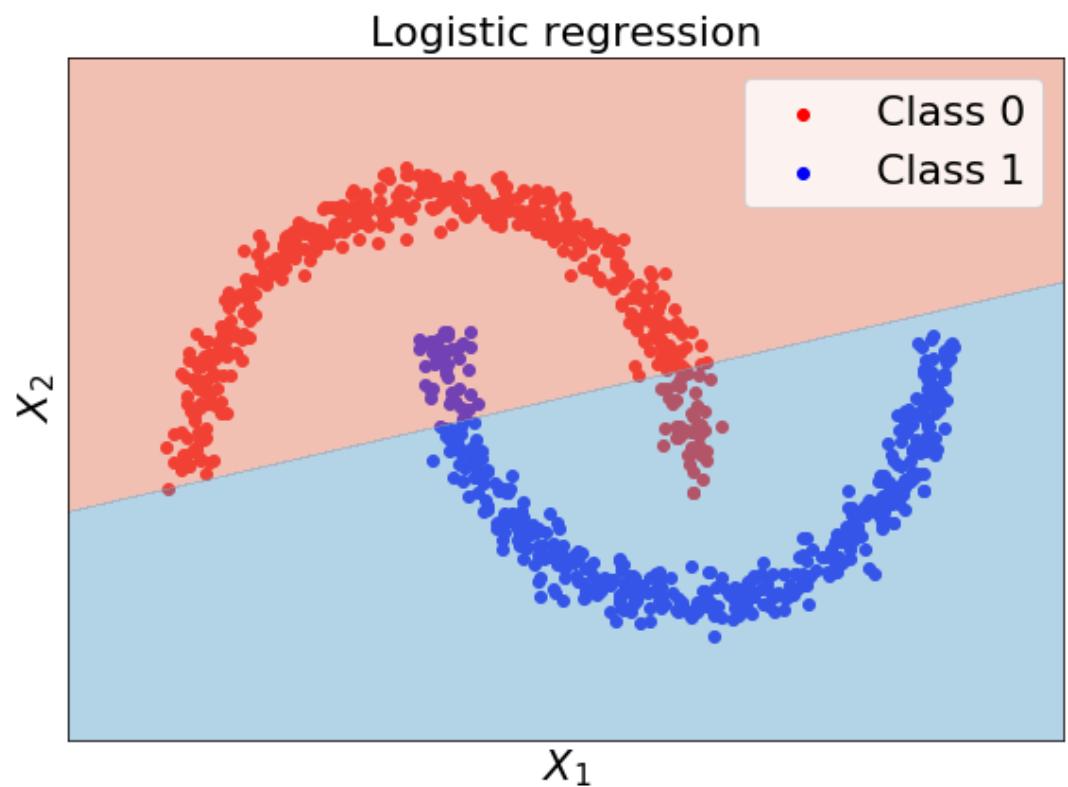
$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- ▶ Показывает среднее геометрическое точности и полноты

# Пример

Metric	Value
Accuracy	0.89
Precision	0.89
Recall	0.89
$F_1$	0.89

- ▶ В этом простом симметричном примере все метрики равны
- ▶ Далее увидим другие примеры



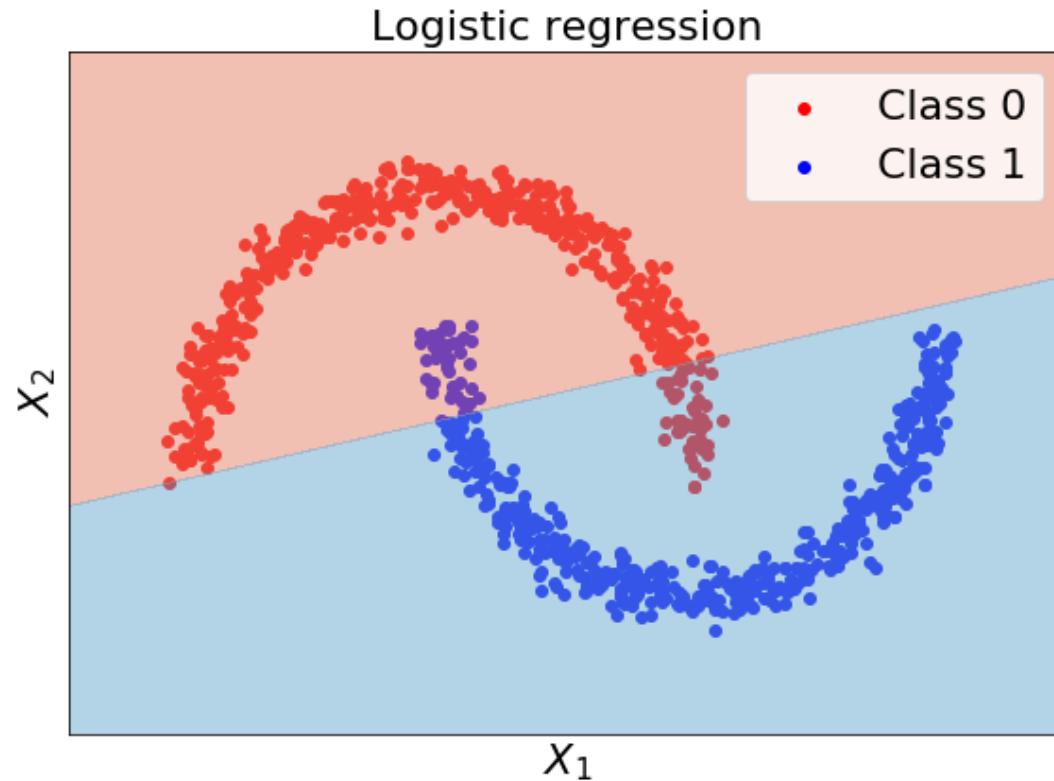
An aerial photograph of a winding road through a dense forest. The road curves back and forth, creating a series of S-shaped bends. The surrounding trees are a mix of dark evergreens and lighter deciduous species, with patches of sunlight filtering through the canopy.

ROC кривая

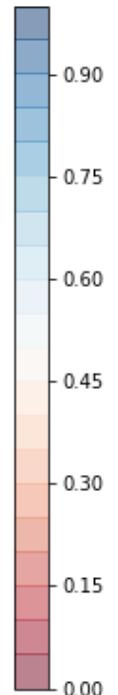
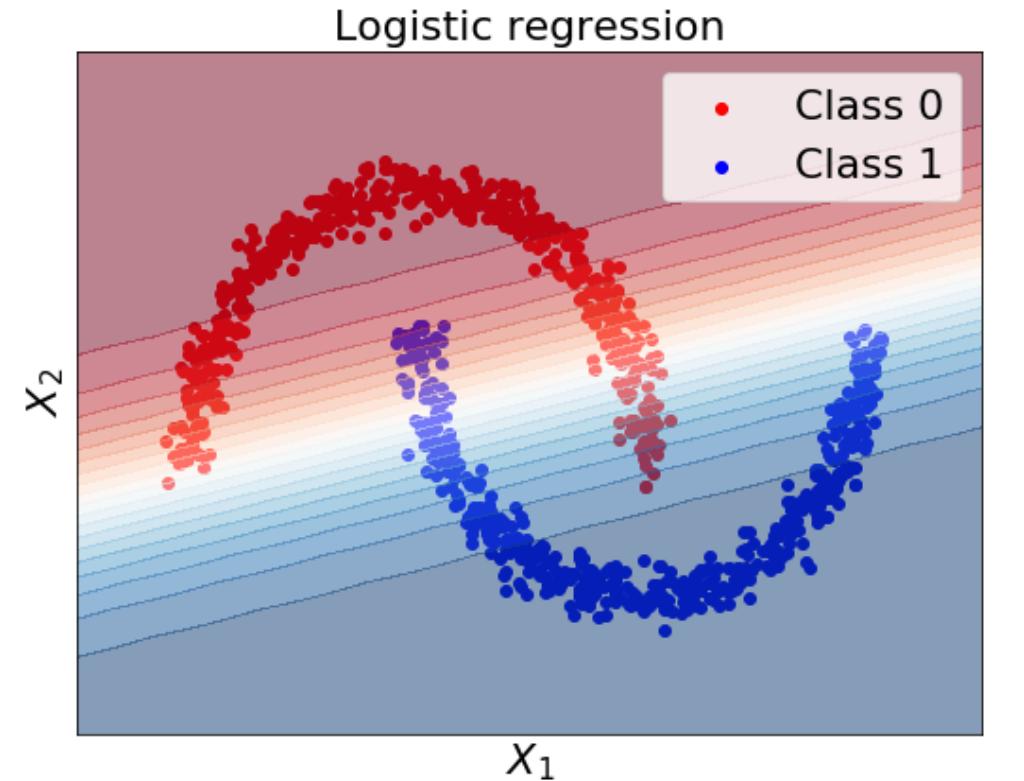
# Метка класса vs вероятность класса

Прогноз **1** если  $p \geq 0.5$

Прогноз **0** если  $p < 0.5$



Вероятность класса **1**  $p$ :



# ROC кривая

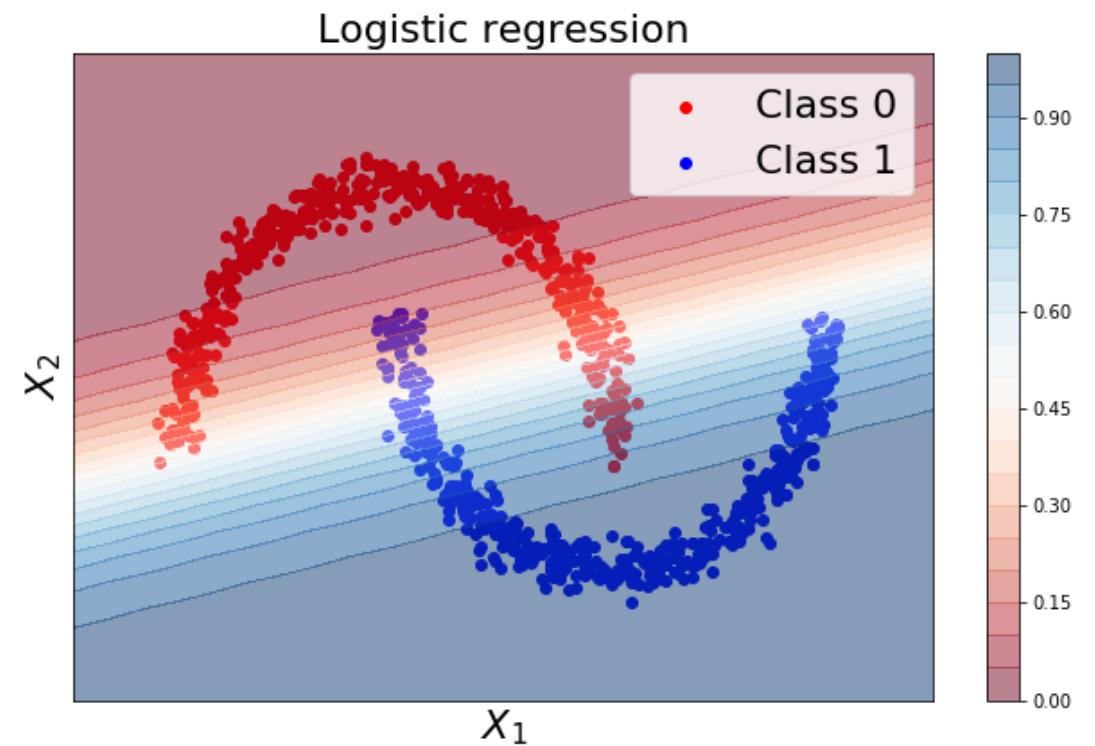
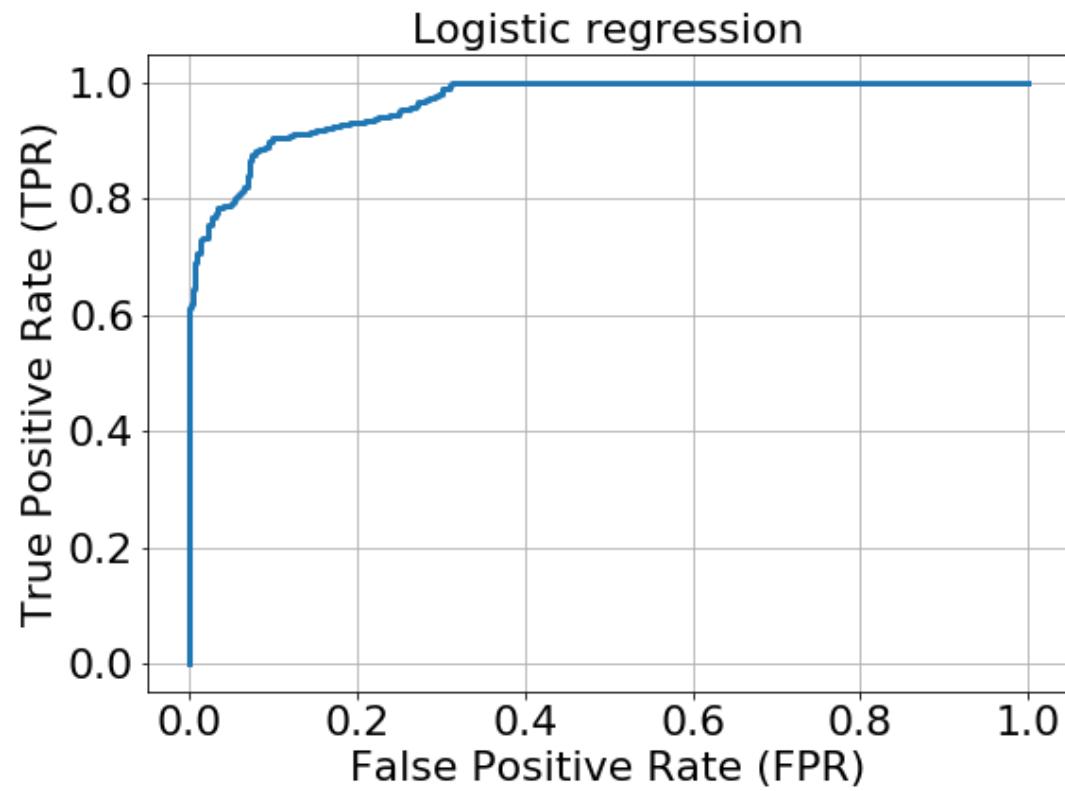
- ▶ ROC (Receiver operating characteristic) кривая – зависимость  $\textcolor{red}{TPR}(\mu)$  от  $\textcolor{red}{FPR}(\mu)$  для разных пороговых значений  $\mu$  вероятности  $p$
- ▶  $TPR(\mu)$  (True Positive Rate):

$$TPR(\mu) = \frac{1}{Pos} \sum_{i \in Pos} I[p_i \geq \mu] = \frac{TP(\mu)}{Pos}$$

- ▶  $FPR(\mu)$  (False Positive Rate):

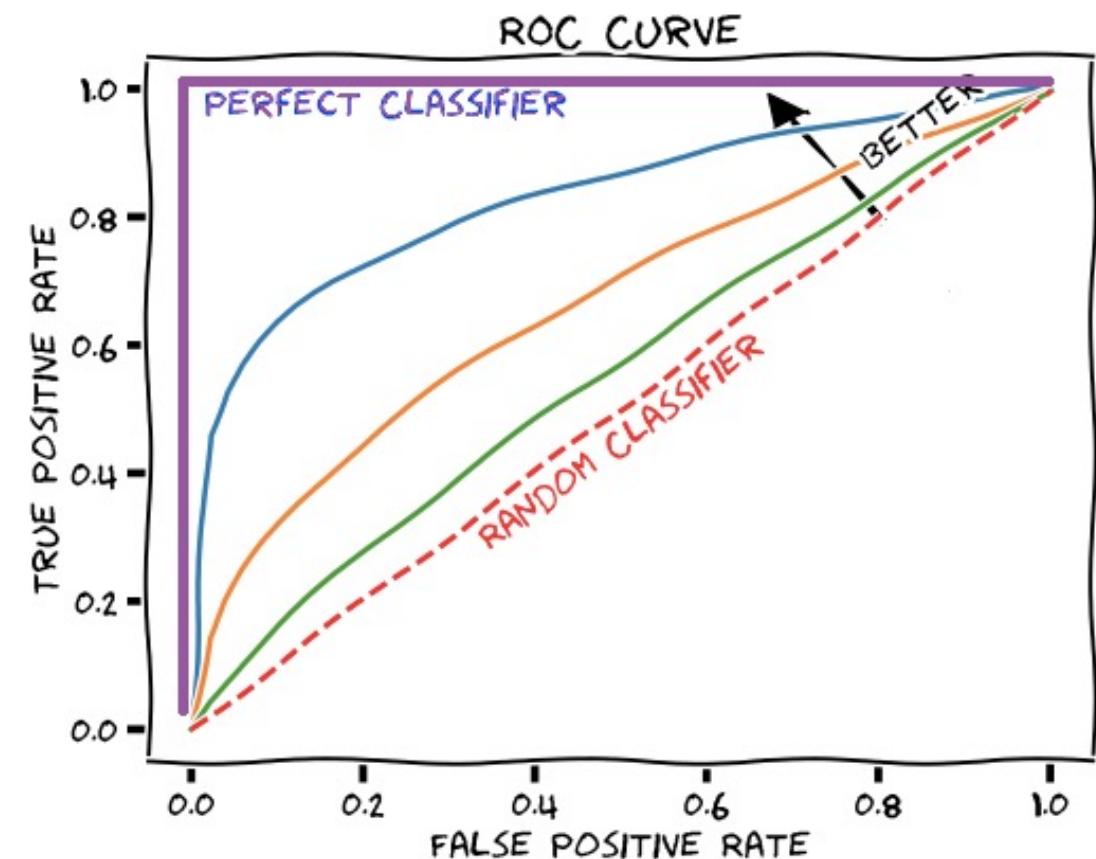
$$FPR(\mu) = \frac{1}{Neg} \sum_{i \in Neg} I[p_i \geq \mu] = \frac{FP(\mu)}{Neg}$$

# ROC кривая



# ROC AUC

- ▶ Можно сравнивать классификаторы с помощью площади под ROC кривой (ROC AUC)
- ▶ ROC AUC  $\in [0, 1]$
- ▶ ROC AUC = 0.5 – случайные прогнозы
- ▶ ROC AUC = 1 – идеальный классификатор
- ▶ ROC AUC = 0 – тоже идеальный классификатор, но с противоположными ответами ☺



Img: <https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/>

# Индекс Джини

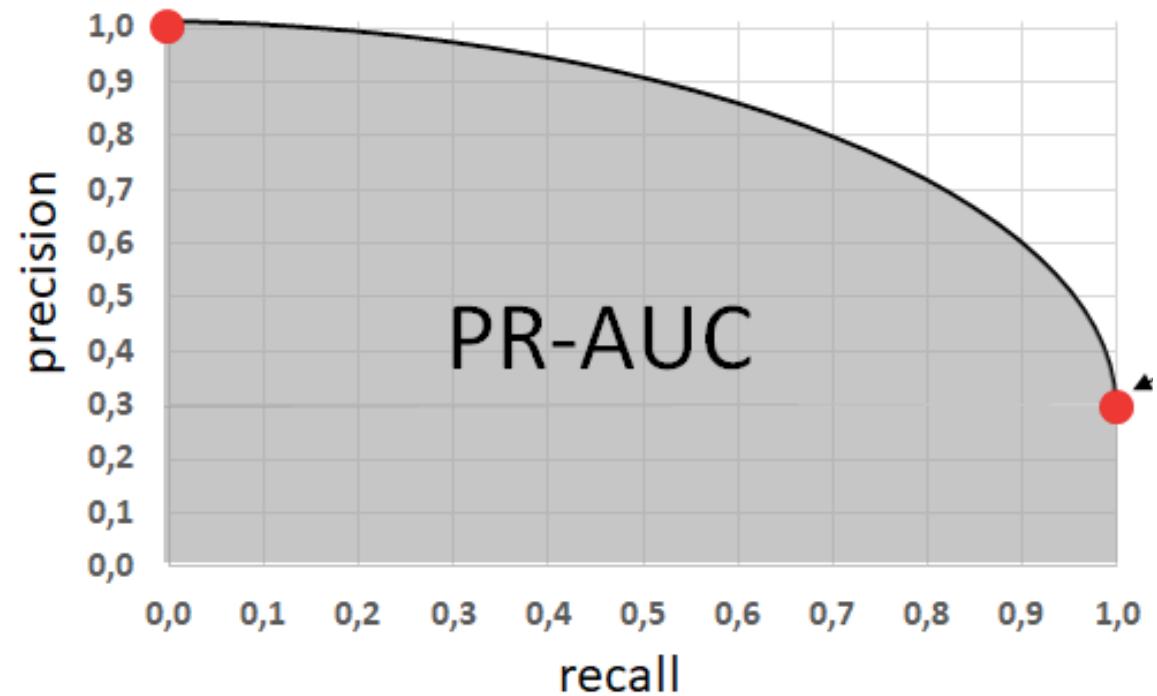
- ▶ Gini:

$$Gini = 2(ROC AUC) - 1$$

- ▶ Измеряется в диапазоне от 0 до 1

# Precision-Recall кривая

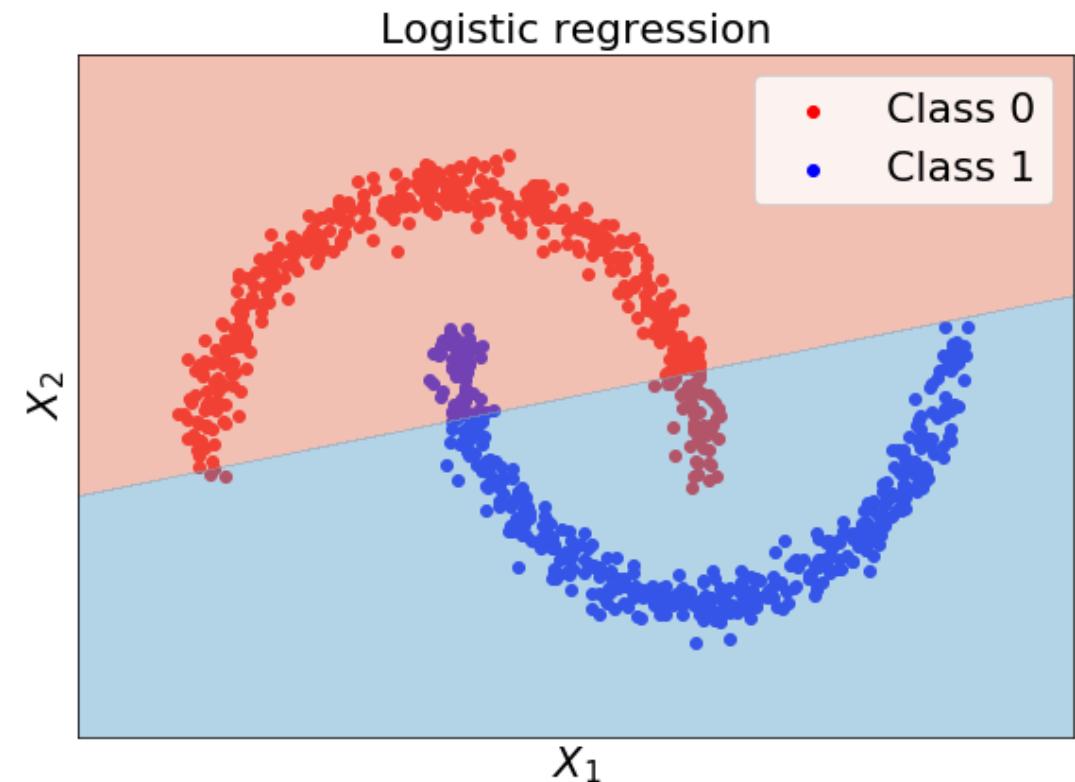
- ▶ По аналогии с ROC кривой, можно построить Precision-Recall (PR) кривую
- ▶ PR – зависимость **Precision( $\mu$ ) от Recall( $\mu$ )** для разных пороговых значений  $\mu$  вероятности  $p$



# Демонстрация

Metric	1:1	1:10	10:1
Accuracy	0.89		
Precision	0.89		
Recall	0.89		
$F_1$	0.89		
ROC AUC	0.97		

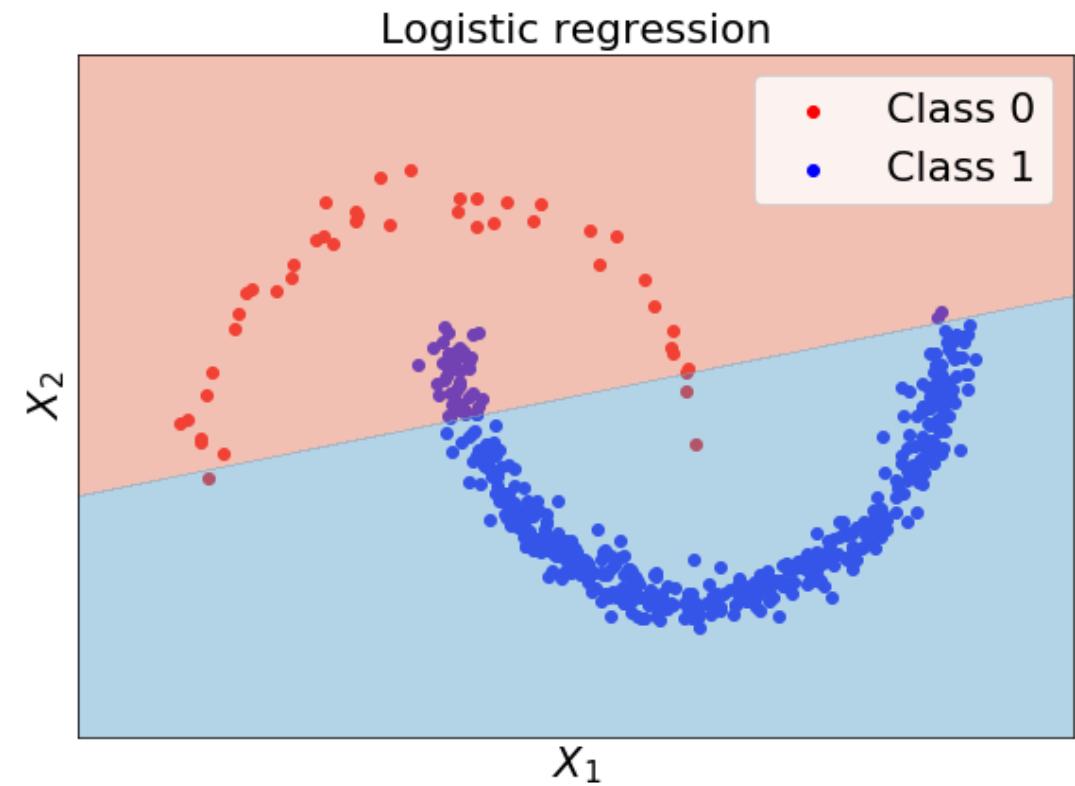
- ▶ Обучили модель на сбалансированной выборке
- ▶ **Фиксируем модель** и будем менять баланс классов



# Демонстрация

Metric	1:1	1:10	10:1
Accuracy	0.89	0.89	
Precision	0.89	0.99	
Recall	0.89	0.89	
$F_1$	0.89	0.94	
ROC AUC	0.97	0.97	

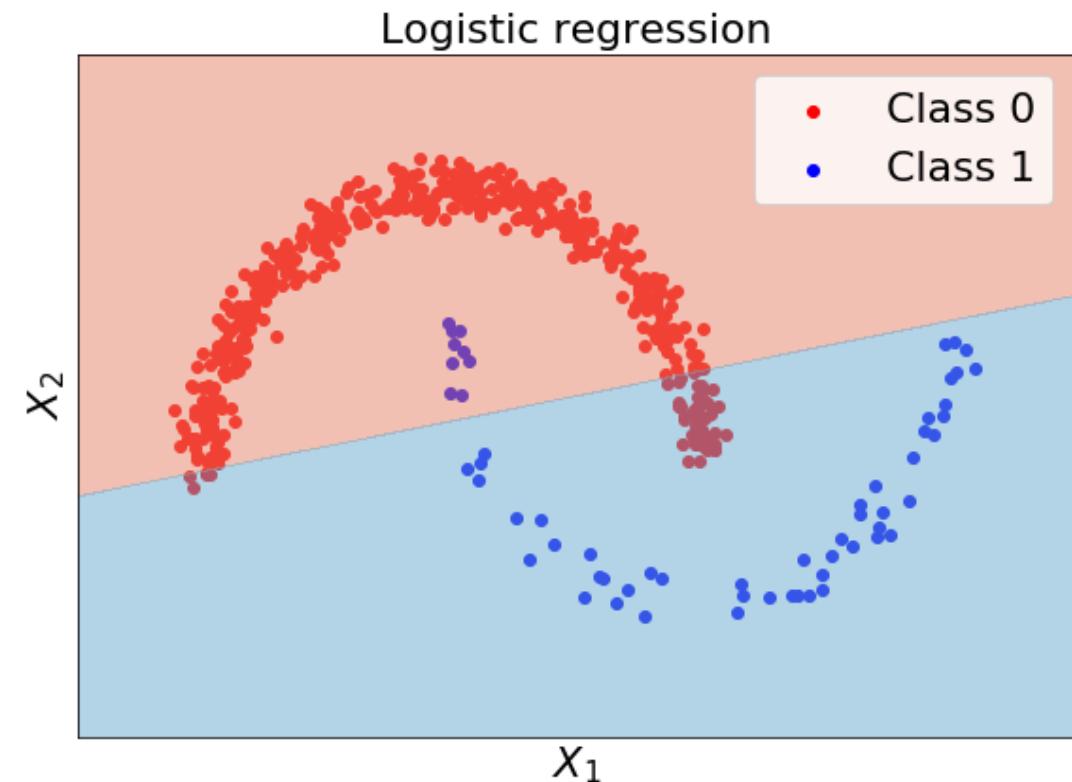
- ▶ Значения некоторых метрик меняются при смене баланса классов



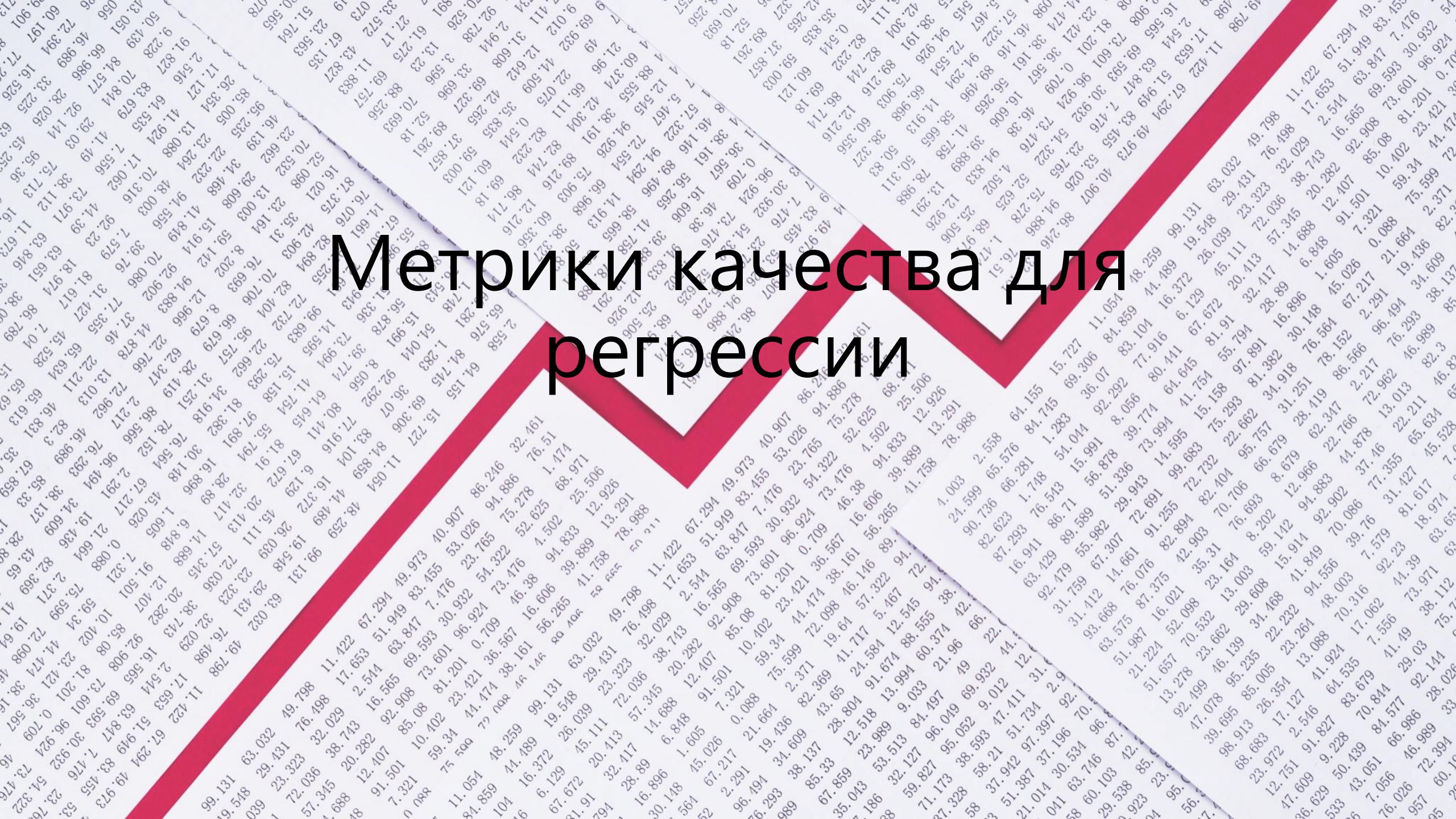
# Демонстрация

Metric	1:1	1:10	10:1
Accuracy	0.89	0.89	0.89
Precision	0.89	0.99	0.47
Recall	0.89	0.89	0.89
$F_1$	0.89	0.94	0.61
ROC AUC	0.97	0.97	0.97

- ▶ **Recall** и **ROC AUC** устойчивы к дисбалансу классов
- ▶ Для **Accuracy** это не выполняется в общем случае



# Метрики качества для регрессии

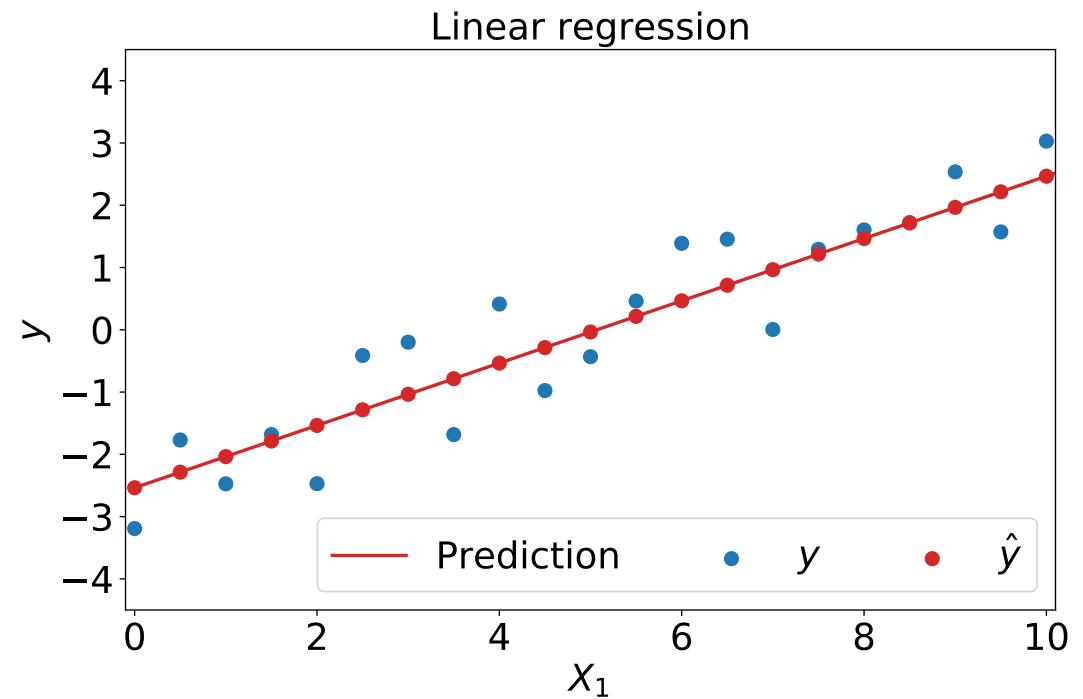


# Задача

Пусть даны  $X, y$  и линейная модель:

$$\hat{y} = Xw$$

**Цель** – измерить **качество модели**,  
определить насколько близки  
прогнозы  $\hat{y}$  к реальным значениям  $y$ .



# Популярные метрики качества

- ▶ Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}$$

- ▶ Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

- ▶ Трудно определить хорошую модель:  $RMSE = 1$  выражает разное качество моделей при  $\bar{y} = 100$  and  $\bar{y} = 1$

# Другие метрики качества #1

- ▶ Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

- ▶ Измеряет относительную ошибку модели
- ▶ Легко интерпретировать
- ▶ Чувствительна к масштабу у

# Другие метрики качества #2

- ▶ Relative Squared Error (RSE):

$$RSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}}$$

- ▶ Relative Absolute Error (RAE):

$$RAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{\sum_{i=1}^N |y_i - \bar{y}|}$$

- ▶ Робастны (мене чувствительны) к масштабу  $y$

# Other quality metrics #3

- ▶ Root Mean Squared Logarithmic Error (RMSLE):

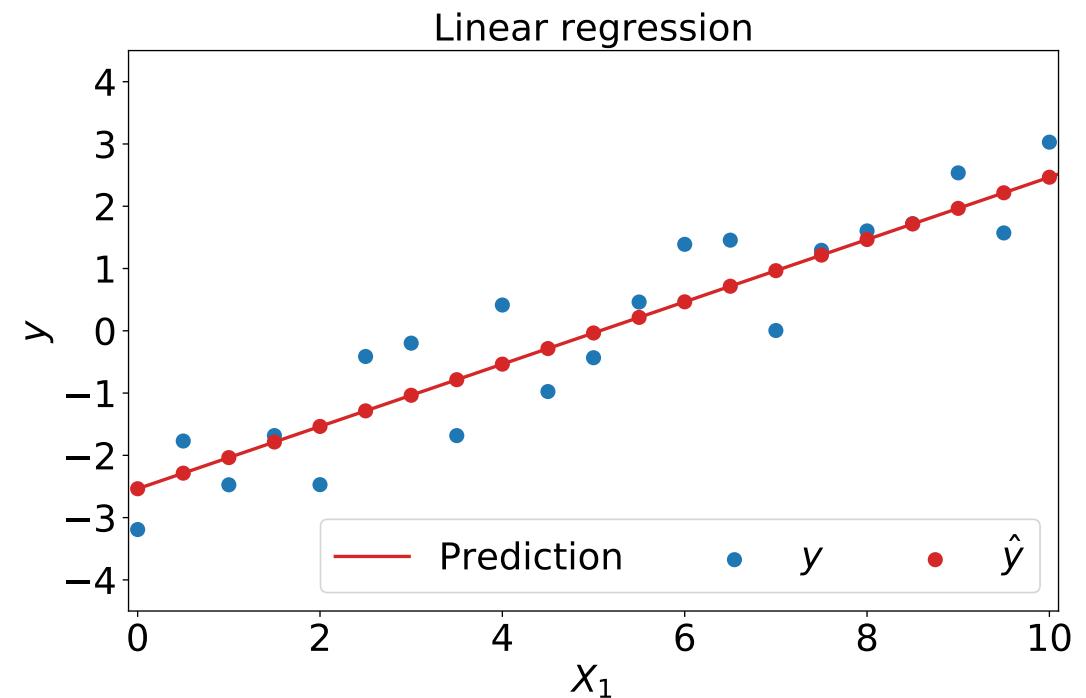
$$RMSLE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$$

- ▶ Отличный выбор, когда  $y_i$  меняется на несколько порядков:  $y_i \in [0, 10^6]$

# Пример

Metric	No outliers
RMSE	0.67
MAE	0.59
MAPE, %	1035
RSE	0.39
RAE	0.40

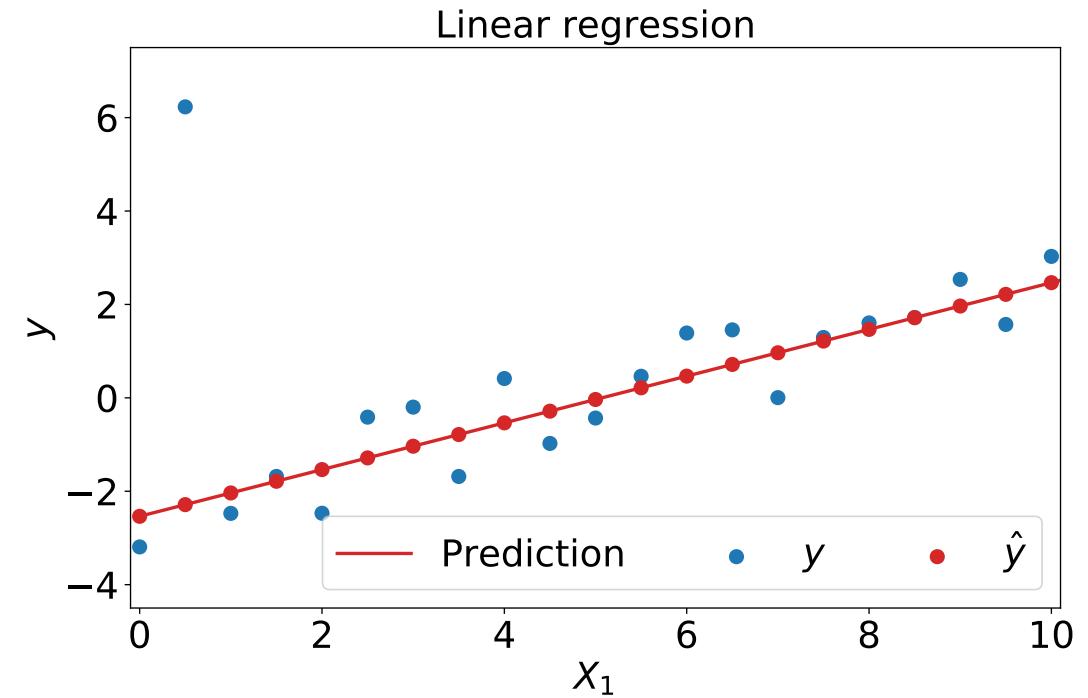
МАРЕ ведет себя плохо, потому что  $y$  и  $y_i$  близки к 0



# Demonstration

Metric	No outliers	With outlier
RMSE	0.67	1.93
MAE	0.59	0.96
MAPE, %	1035	1040
RSE	0.39	0.92
RAE	0.40	0.58

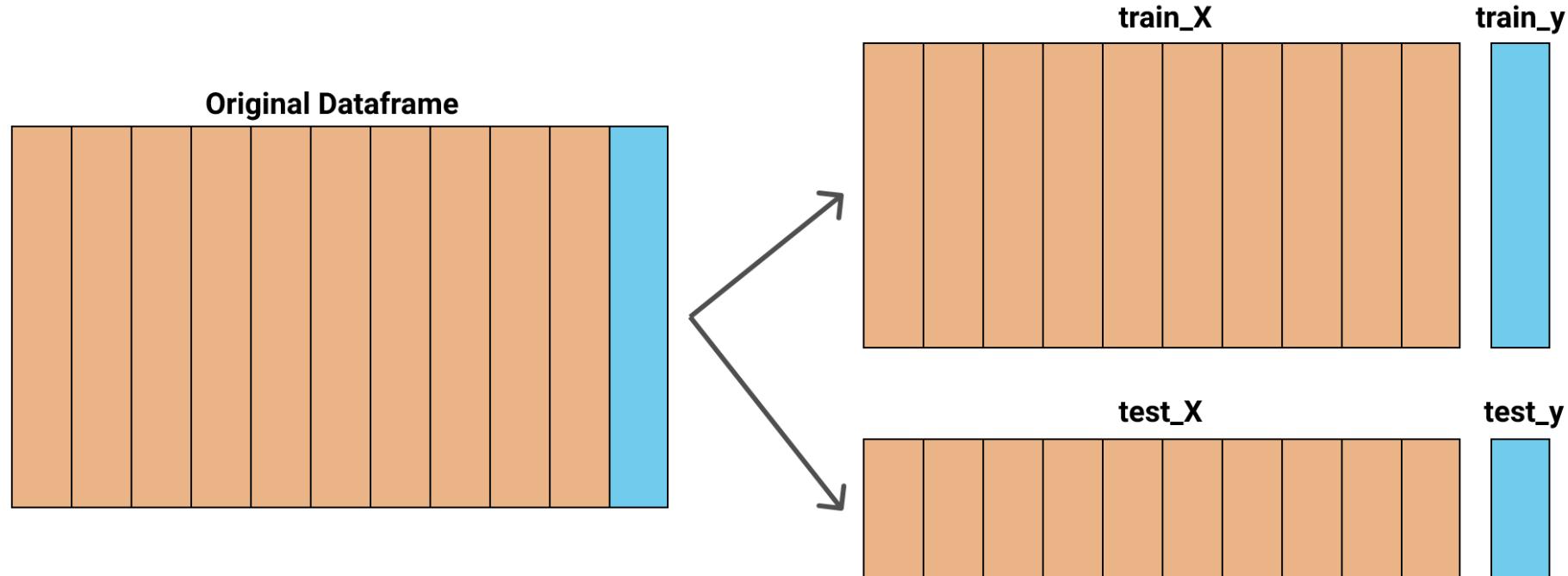
- ▶ Выбросы могут сместить метрики
- ▶ MAE и RAE более робастны



Переобучение

# Обучение и тест

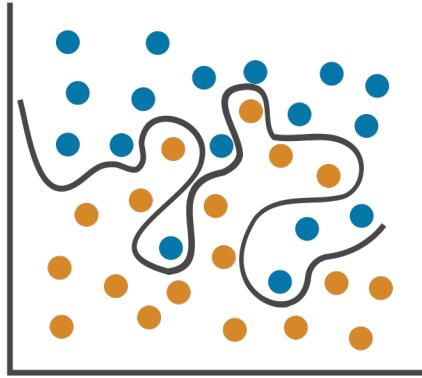
- ▶ **Обучающая выборка (train):** для обучения модели
- ▶ **Тестовая (отложенная) выборка (test):** для измерения качества модели



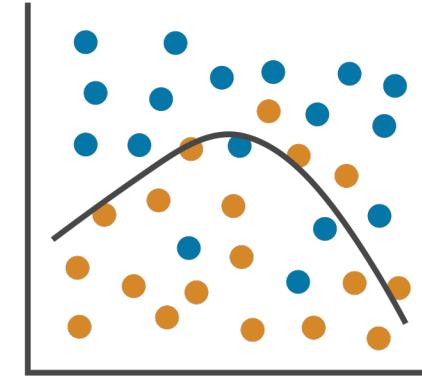
# Переобучение

Classification

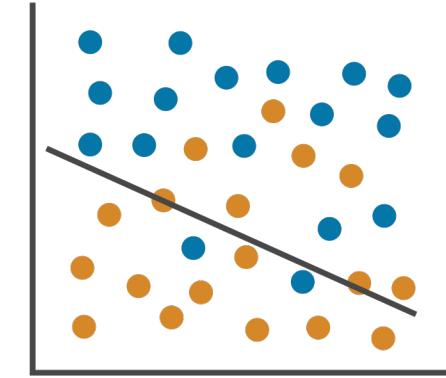
Overfitting



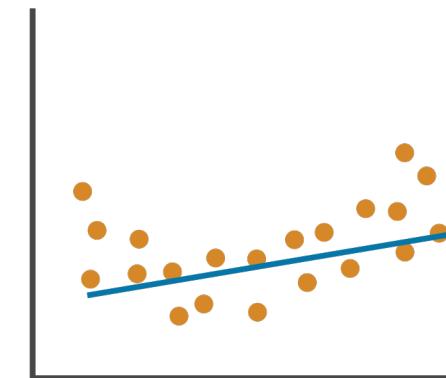
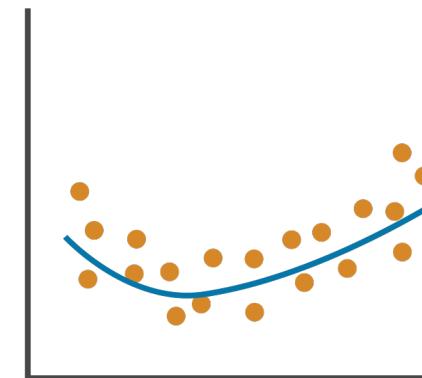
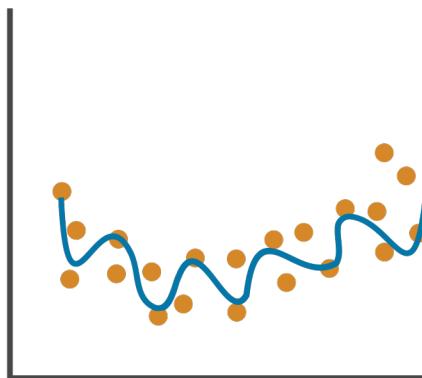
Right Fit



Underfitting

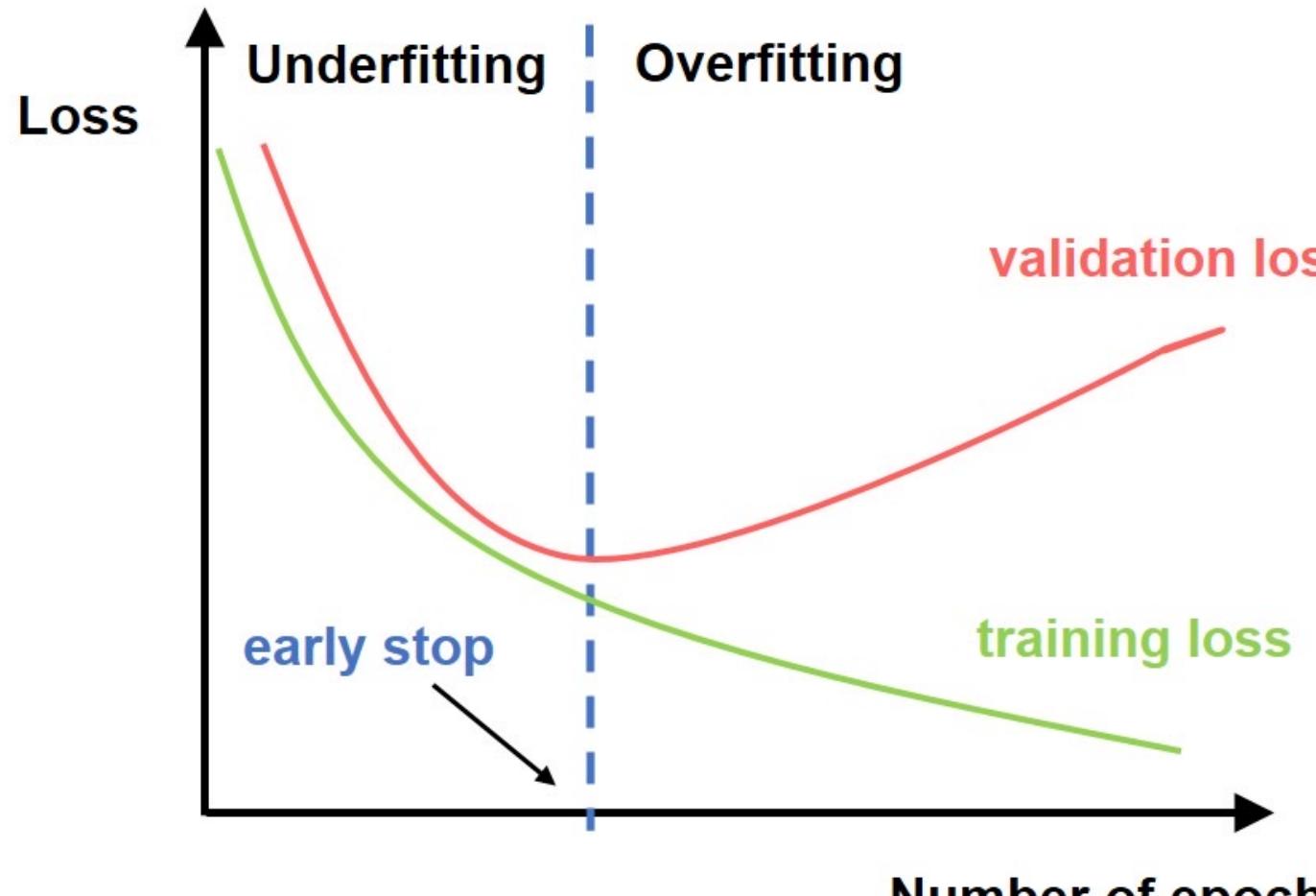


Regression



Источник: <https://www.mathworks.com/discovery/overfitting.html>

# Кривая обучения



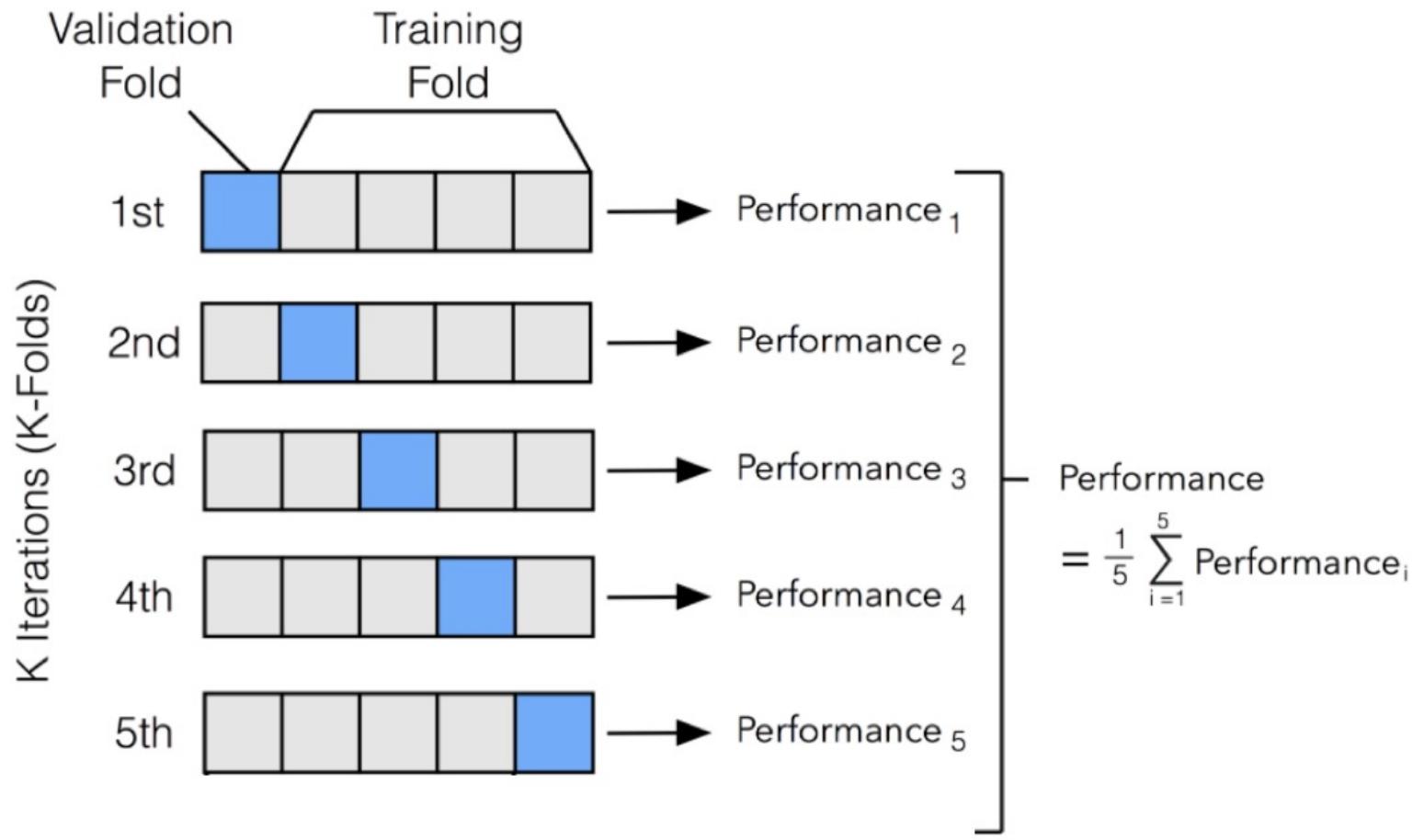
Число итераций  
градиентного  
спуска

Источник: <https://datahacker.rs/018-pytorch-popular-techniques-to-prevent-the-overfitting-in-a-neural-networks/>

# Регуляризация

- ▶ Линейная и логистическая регрессии
  - $L_1$ -регуляризация
  - $L_2$ -регуляризация
- ▶ KNN
  - Число соседей

# K-Fold кросс-валидация



# Кросс-валидация (cross-validation)

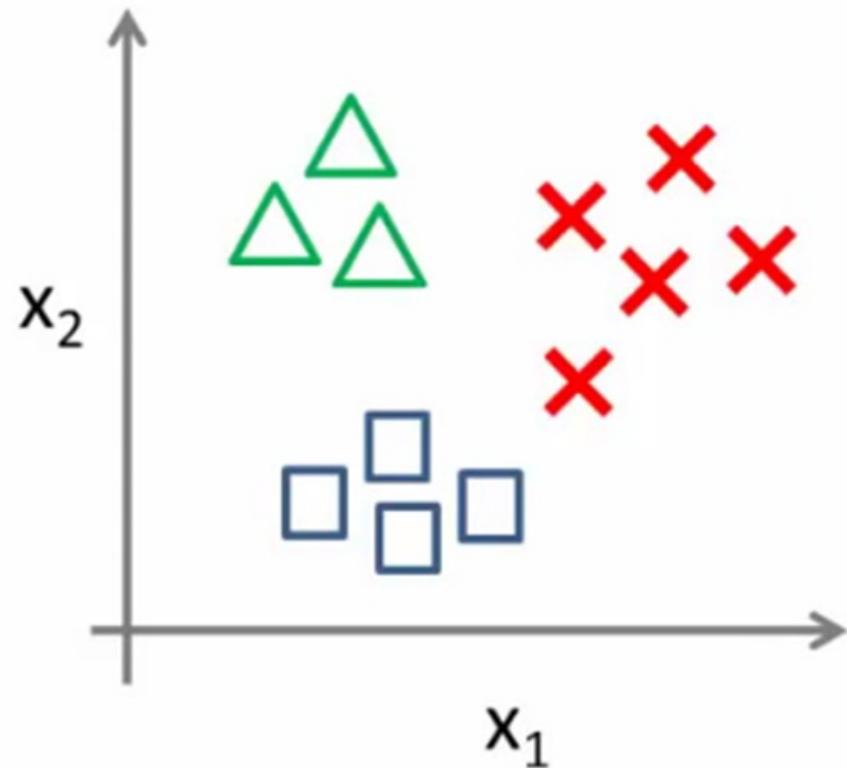
- ▶ Используется для измерения качества моделей в машинном обучении
- ▶ Отложенная выборка (train / test):
  - Делим всю выборку на две подвыборки в пропорции 70:30
  - Большая часть данных не используется для обучения (хуже качество модели)
- ▶ K-Fold кросс-валидация
  - К берем порядка 10
  - Больше данных участвует в обучении отдельной модели
  - Проверяем качество на всех данных
  - Более точная оценка качества

# Многоклассовая классификация

# Задача

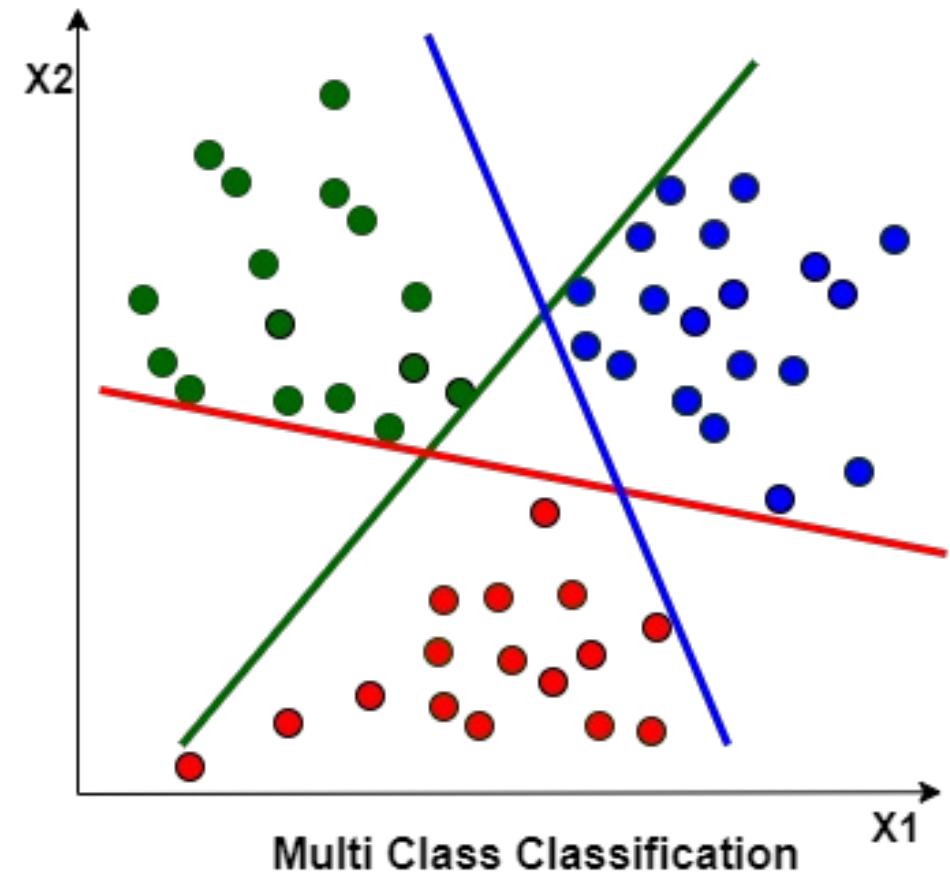
- ▶ Разделить объекты между **несколькими** классами.
- ▶ Многие классификаторы поддерживают несколько классов.
- ▶ Но не все 😞
- ▶ Как **разложить** эту задачу на несколько задач бинарной классификации?

Multi-class classification:



# Один против всех (one-vs-all)

- ▶ Пусть дано **K** классов
- ▶ Для **каждого** класса обучаем свой бинарный классификатор **отделять объекты этого класса от всех остальных**
- ▶ Всего обучаем **K** таких классификаторов



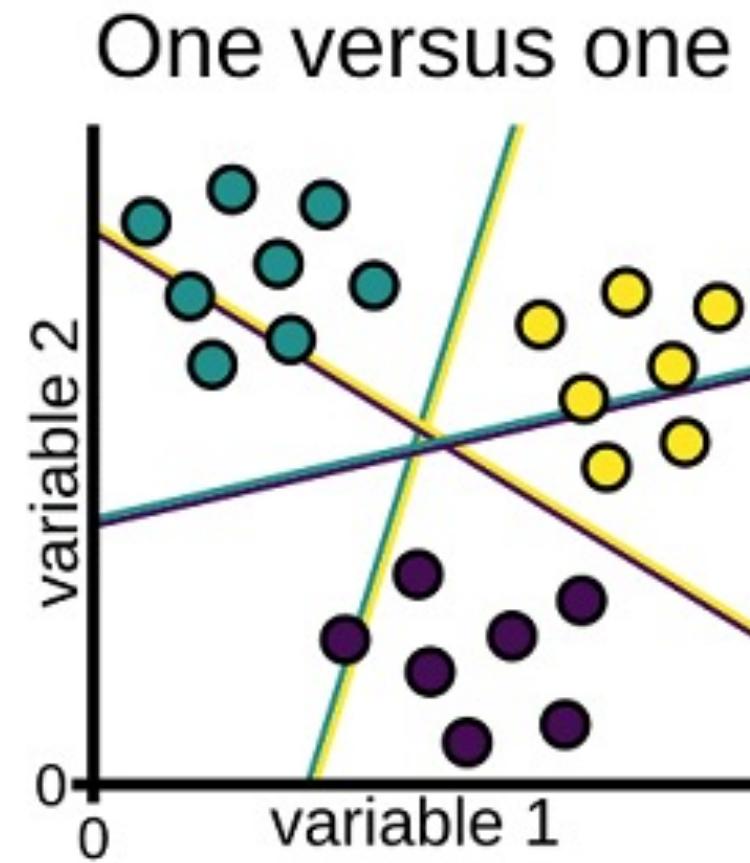
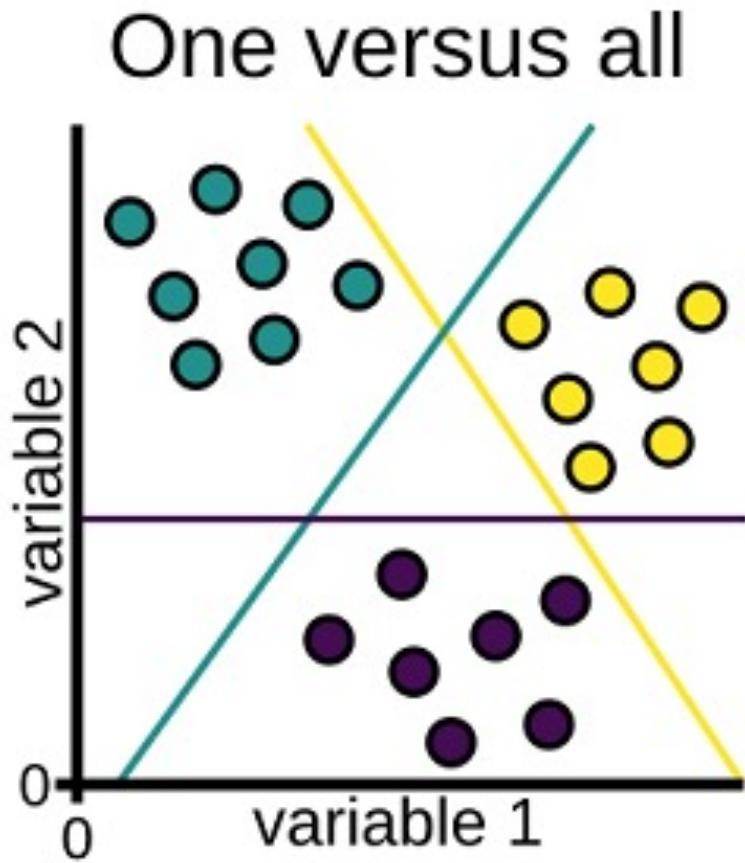
# Один против всех (one-vs-all)

- ▶ Пусть есть К обученных классификаторов:  $\hat{p}_1(x), \hat{p}_2(x), \dots, \hat{p}_K(x)$
- ▶ Пусть дан объект  $x_i$ , для которого делаем прогноз:
  - Получаем К прогнозов:  $\hat{p}_1(x_i), \hat{p}_2(x_i), \dots, \hat{p}_K(x_i)$
  - Находим класс с максимальным прогнозом:

$$\hat{y}(x_i) = \arg \max_{k \in \{1, \dots, K\}} \hat{p}_k(x_i)$$

- Здесь  $\hat{p}_k(x_i)$  – прогноз “вероятности” положительного ( $k$ -го) класса;
- $\hat{y}(x_i)$  - итоговый прогноз метки класса (одного из K)

# Один против одного (one-vs-one)



# Один против одного (one-vs-one)

- ▶ Для каждой пары классов  $i, j$  обучаем свой бинарный классификатор  $\hat{y}_{ij}(x)$
- ▶ Пусть дан объект  $x_t$ , для которого делаем прогноз:

$$\hat{y}(x_t) = \arg \max_{k \in \{1, \dots, K\}} \sum_{i=1}^K \sum_{j \neq i} [\hat{y}_{ij}(x_t) = k]$$

- ▶ Т.е. выбираем класс, за который наберется больше всего голосов

# Логистическая регрессия на K классов

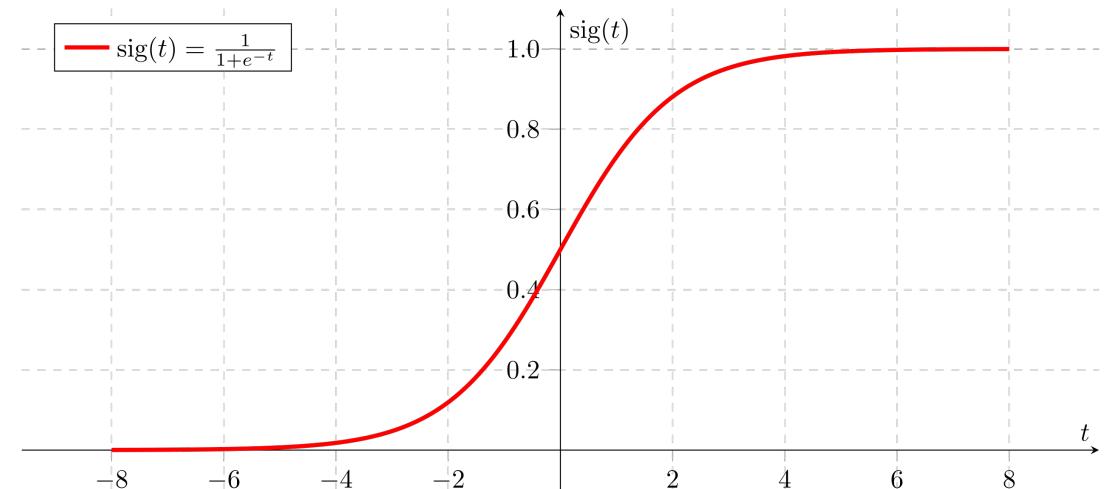
# Логистическая регрессия на 2 класса

- ▶ Вероятность класса 1:

$$p(y = 1|x_i) = \sigma(x_i^T w) = \hat{y}_i$$

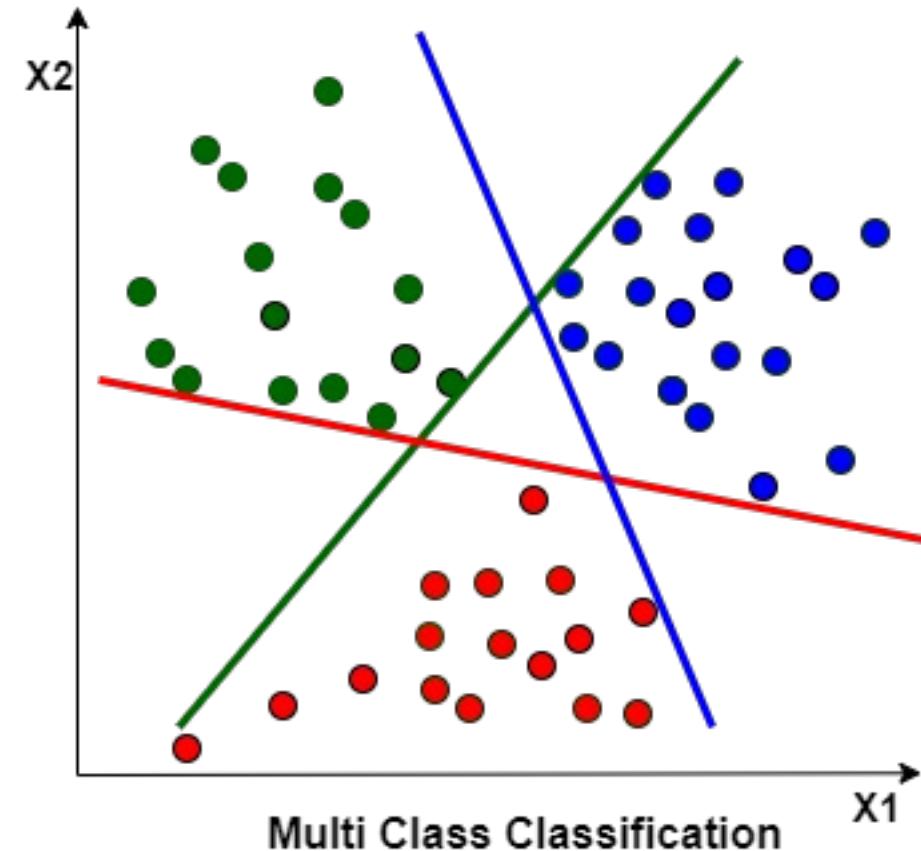
- ▶ Вероятность класса 0:

$$p(y = 0|x_i) = 1 - \sigma(x_i^T w)$$



# Логистическая регрессия на K классов

- ▶ Строим **K один против всех** моделей:
  - Класс 1 против всех:  $z_{i1} = x_i^T w_1$
  - Класс 2 против всех:  $z_{i2} = x_i^T w_2$
  - Класс 3 против всех:  $z_{i3} = x_i^T w_3$
  - Класс K против всех:  $z_{iK} = x_i^T w_K$
- ▶ Получаем **K** векторов весов для обучения



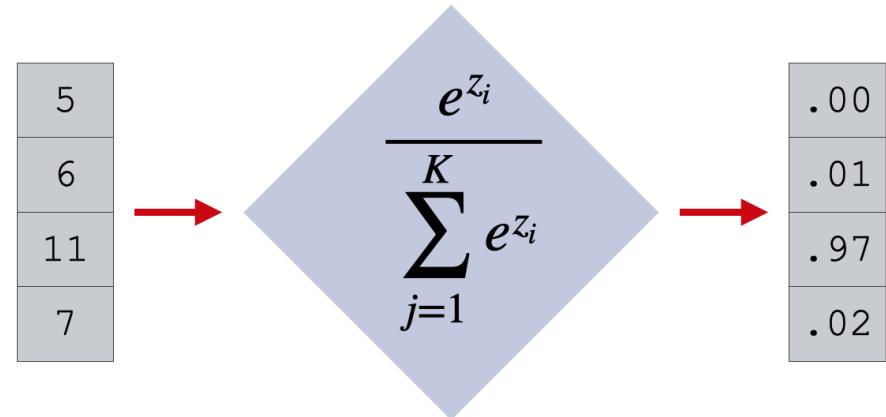
# Логистическая регрессия на K классов

- ▶ SoftMax – многомерный вариант сигмоиды:

$$\hat{y}_{i1} = p(y = \mathbf{1}|x_i) = \frac{e^{z_{i1}}}{\sum_{k=1}^K e^{z_{ik}}}$$

$$\hat{y}_{i2} = p(y = \mathbf{2}|x_i) = \frac{e^{z_{i2}}}{\sum_{k=1}^K e^{z_{ik}}}$$

$$\hat{y}_{iK} = p(y = \mathbf{K}|x_i) = \frac{e^{z_{iK}}}{\sum_{k=1}^K e^{z_{ik}}}$$



# Логарифм правдоподобия для K классов

- ▶ Правдоподобие:

$$\text{Likelihood} = - \prod_{i=1}^n p(y = 1|x_i)^{[y_i=1]} p(y = 2|x_i)^{[y_i=2]} \dots p(y = K|x_i)^{[y_i=k]}$$

- ▶ Логарифм правдоподобия (функция потерь):

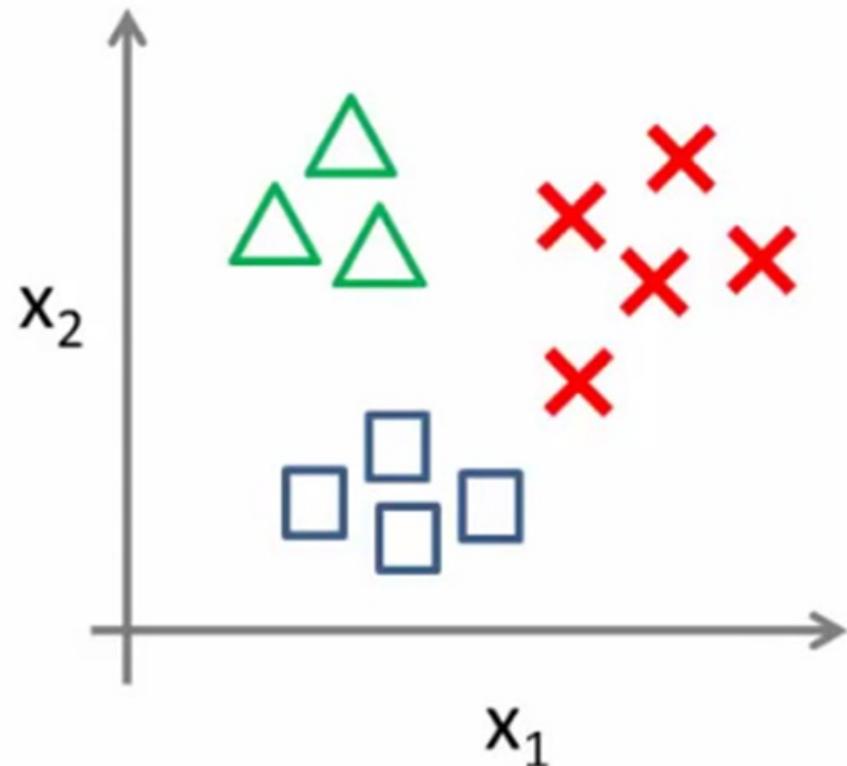
$$L = - \sum_{i=1}^n \sum_{k=1}^K [y_i = k] \log (\hat{y}_{ik}) \rightarrow \min_{w_1, \dots, w_K}$$

# Метрики качества для мультикласса

# Задача

- ▶ Мы знаем как считать метрики для **двух** классов.
- ▶ Что делать в случае К классов?

Multi-class classification:



# Микро-усреднение

- ▶ Пусть дано К классов
- ▶ Рассмотрим К один против всех задач
- ▶ Для каждой задачи считаем  $TP_k, FP_k, FN_k, TN_k$
- ▶ Усредняем эти характеристики по всем классам:

$$\overline{TP} = \frac{1}{K} \sum_{k=1}^K TP_k$$

- ▶ Используем их для подсчета метрик качества:

$$Precision = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$

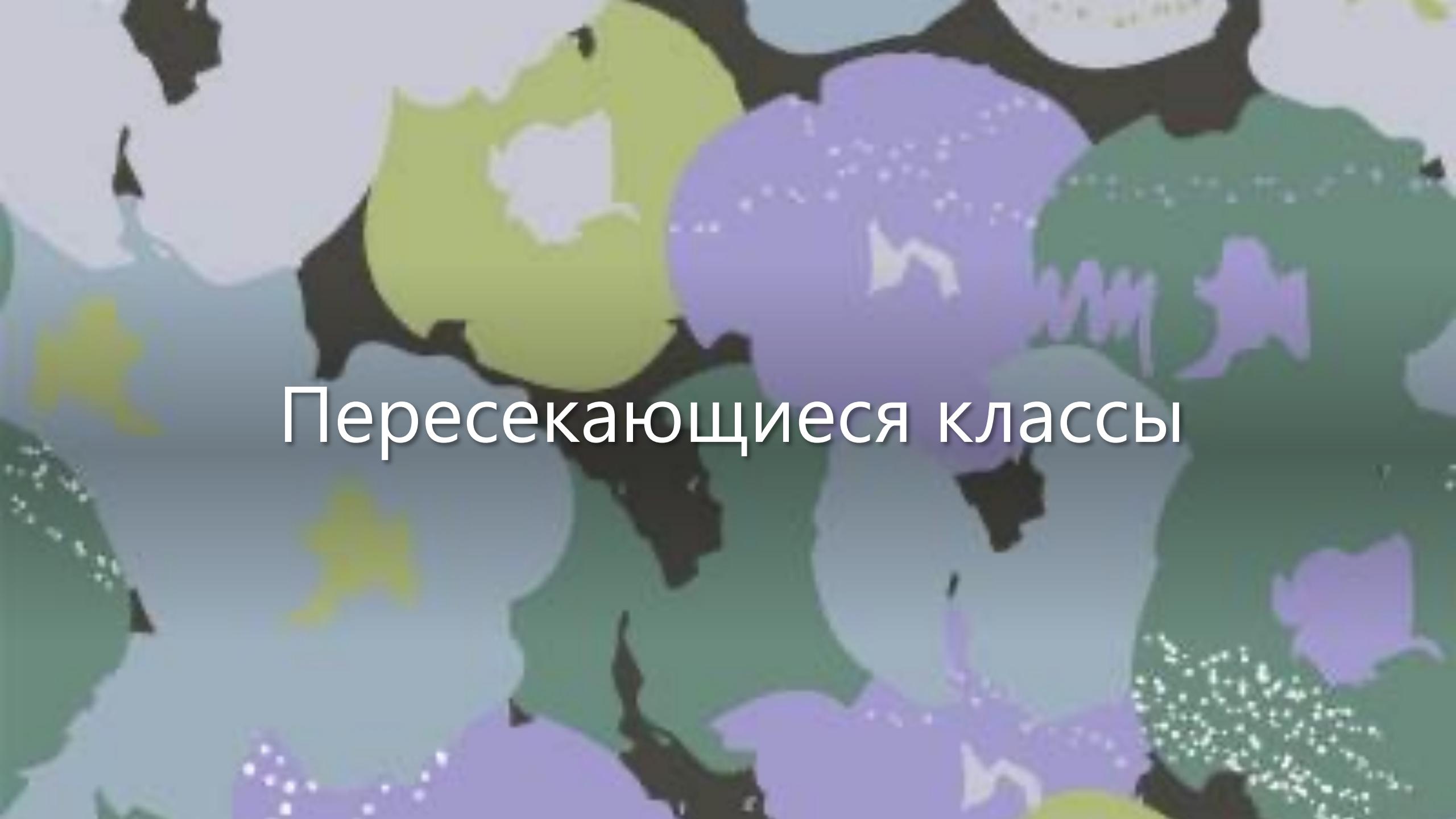
# Макро-усреднение

- ▶ Пусть дано К классов
- ▶ Рассмотрим К один против всех задач
- ▶ Для каждой задачи считаем  $TP_k, FP_k, FN_k, TN_k$
- ▶ Используем их для подсчета метрик качества:

$$Precision_k = \frac{TP_k}{TP_k + FP_k}$$

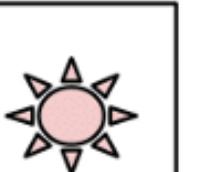
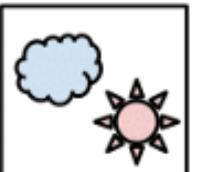
- ▶ Усредняем эти метрики качества по всем классам:

$$\overline{Precision} = \frac{1}{K} \sum_{k=1}^K Precision_k$$



Пересекающиеся классы

# Классификация с пересекающимися классами

	Multi-Class	Multi-Label
$C = 3$   	<p>Samples</p>    <p>Labels (t)</p> <p>[0 0 1] [1 0 0] [0 1 0]</p>	<p>Samples</p>    <p>Labels (t)</p> <p>[1 0 1] [0 1 0] [1 1 1]</p>

# Независимая классификация (one-vs-all)

- ▶ Пусть есть К обученных классификаторов:  $\hat{p}_1(x), \hat{p}_2(x), \dots, \hat{p}_K(x)$
- ▶ Пусть дан объект  $x_i$ , для которого делаем прогноз:
  - Получаем К прогнозов:  $\hat{p}_1(x_i), \hat{p}_2(x_i), \dots, \hat{p}_K(x_i)$
  - Тогда вектор прогнозов:

$$\hat{y}(x_i) = \begin{pmatrix} [\hat{p}_1(x_i) > \tau] \\ [\hat{p}_2(x_i) > \tau] \\ \dots \\ [\hat{p}_K(x_i) > \tau] \end{pmatrix}$$

- Можно взять порог  $\tau = 0.5$
- Здесь  $\hat{p}_k(x_i)$  – прогноз “вероятности” положительного ( $k$ -го) класса;
- $\hat{y}(x_i)$  - итоговый прогноз метки класса (несколько из К)

# Независимая классификация

- ▶ Самое простое решение задачи **multi-label classification**
- ▶ Не учитывает связи между классами

# Заключение



# Вопросы

- ▶ Что такое точность, полнота и F-мера?
- ▶ Что такое AUC-ROC? Опишите алгоритм построения ROC-кривой.
- ▶ В чем состоят преимущества и недостатки использования метрик Mean squared error (MSE) и Mean absolute error (MAE) в задаче регрессии? Запишите формулу метрики Mean absolute percentage error (MAPE).
- ▶ Что такое переобучение и недообучение? Как отличить переобучение от недообучения?
- ▶ Что такое кросс-валидация и для чего она используется? Чем применение кросс-валидации лучше, чем разбиение выборки на обучение и контроль?
- ▶ В чём заключается подход с независимой классификацией в задаче классификации с пересекающимися классами (multilabel classification)?
- ▶ Что такое микро- и марко-усреднение при оценивании качества многоклассовой классификации?