

Искусственный Интеллект

Лекция 6: Оценка модели и регуляризация

Мартынюк Полина Антоновна

telegram: @PAMartynyuk

email: pa-martynyuk@yandex.ru



Разобрались с решением простейших задач МО

- **Линейная регрессия** – задача регрессии

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Логистическая регрессия** – задача классификации

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

$$y = \frac{e^{y_i}}{\sum_{i=0}^{\text{num_of_classes}} e^{y_i}}$$

Функция потерь и метрики: различие

Metric != Loss

Loss :

- Участвует в градиентном спуске (обучении)
- Абстрактна по своей сути
- Не отображает реальное качество модели
- Не более одной



а сколько фруктов было определено верно?

а какая доля конкретно яблок была определена верно?

а насколько компромисс точности на апельсинах может
повысить точность определения яблок?

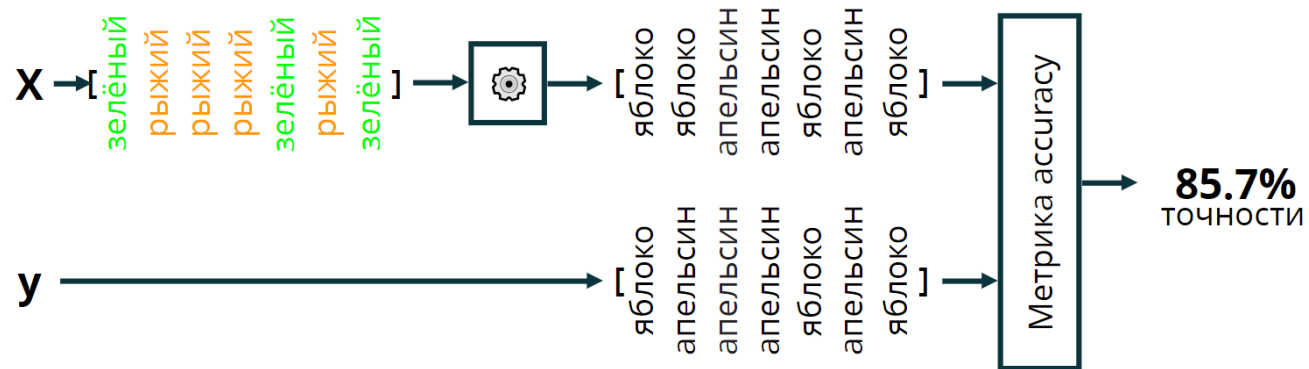
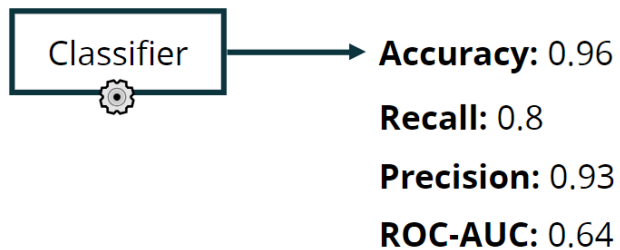
Функция потерь и метрики: различие

Metrics:

- Нужны для определения более качественной оценки **ОТДЕЛЬНЫХ моделей** и изначального определения цели работы
- Сравнивают ответы модели с реальными ответами
- Осмысленные значения
- Не участвует в математике обучения
- Одновременно может быть **несколько разных метрик**

Accuracy: 60% точности (от 0 до 100)

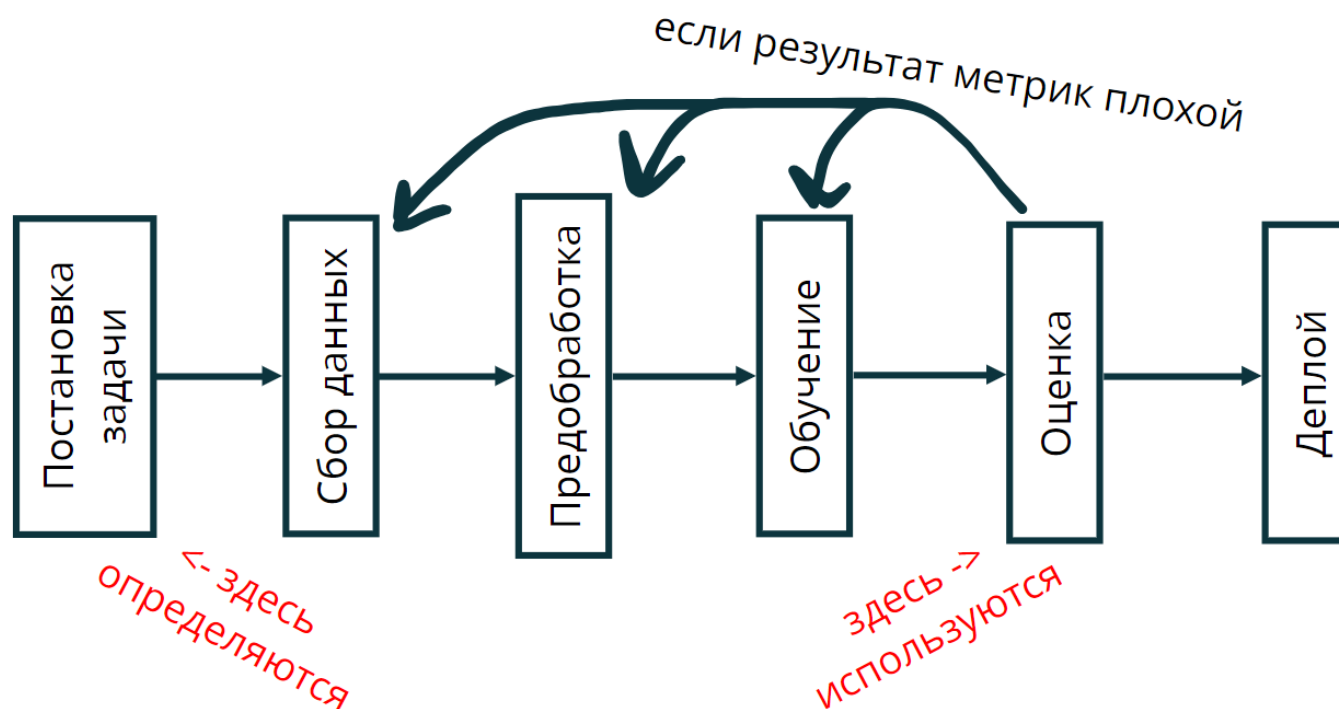
Cosine Similarity: 0.95 схожести (от 0 до 1)



...

Функция потерь и метрики: различие

Метрики в пайплайне



Метрики для классификации

Точность/ Доля верных ответов Accuracy

$$\text{доля правильных ответов} = \frac{\text{количество правильных ответов}}{\text{количество всех ответов}}$$

```
from sklearn.datasets import load_iris
from sklearn.linear_model import
LogisticRegression

X, y = load_iris(return_X_y=True)
clf =
LogisticRegression(random_state=0).fit(X, y)
clf.score(X, y)
>>> 0.97
```

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

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

Метрики для классификации

Точность/ Доля верных ответов

Accuracy

```
from sklearn.metrics import accuracy_score

y_pred = [0, 2, 1, 3]
y_true = [0, 1, 2, 3]

accuracy_score(y_true, y_pred)
>>> 0.5

accuracy_score(y_true, y_pred, normalize=False)
>>> 2
```

Метрики для классификации

Точность/ Доля верных ответов Ассигасу

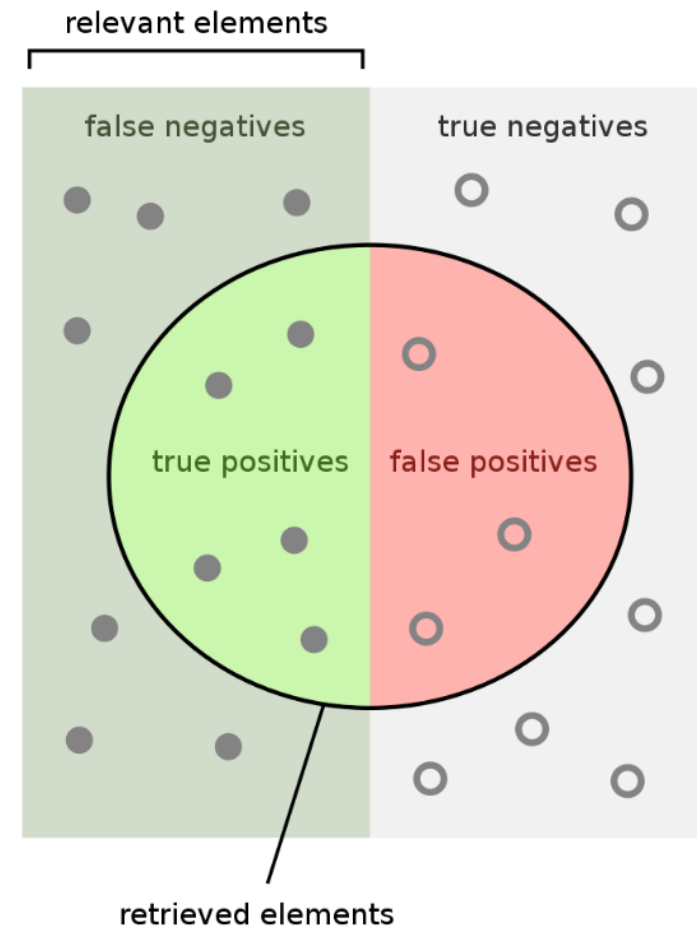
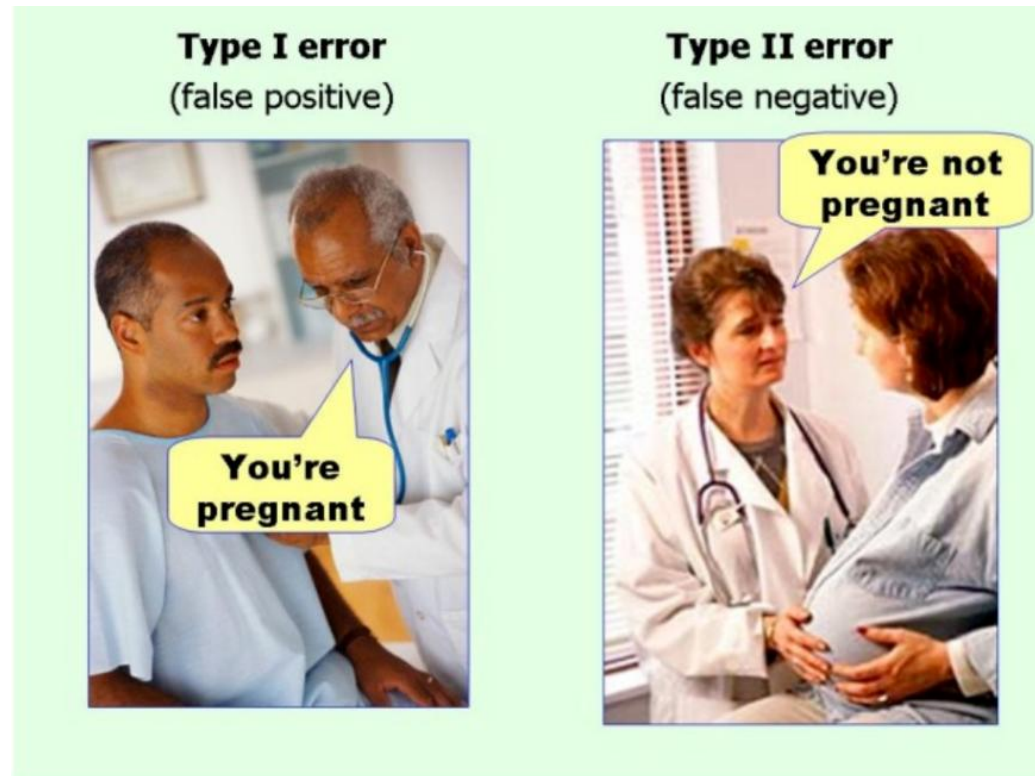
Предположим, что в почтовом ящике 99% обычных писем (класс 0) и 1% спама (класс 1). Тогда, если возвращать всегда 0, то модель будет иметь ассигасу **0.99**

При **неравномерном распределении классов** ассигасу **обманчива**

Метрики для классификации

Precision/recall

Связаны с ошибками разного рода




Метрики для классификации

Precision

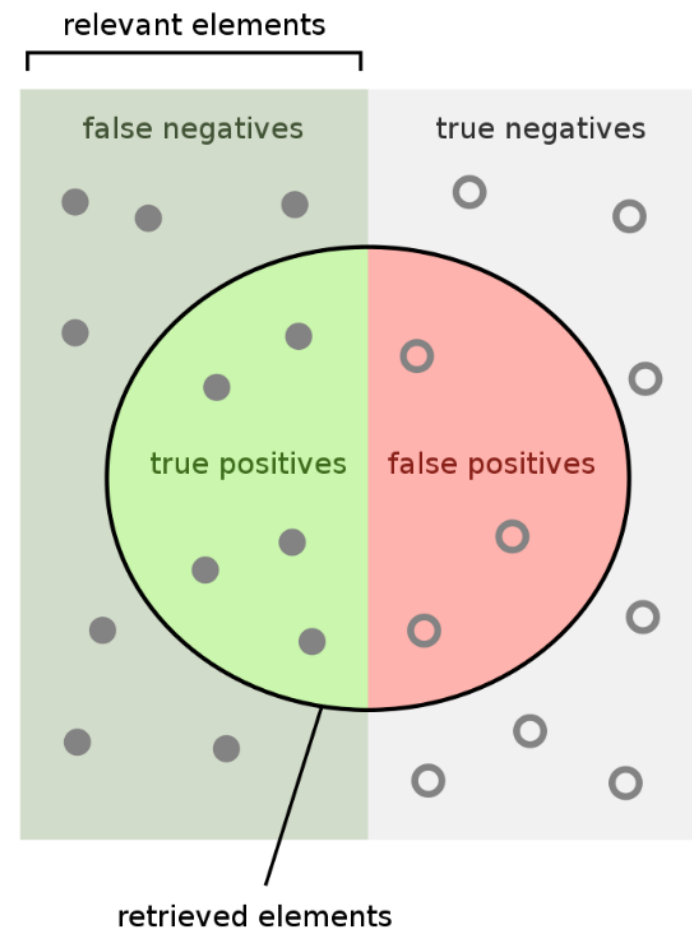
Точность

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

Precision = 

Пример:

Спецагент и сканер
отпечатка пальца

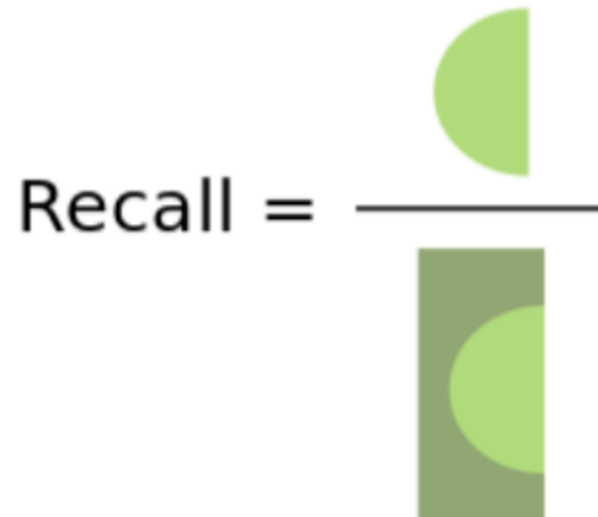


Метрики для классификации

Recall

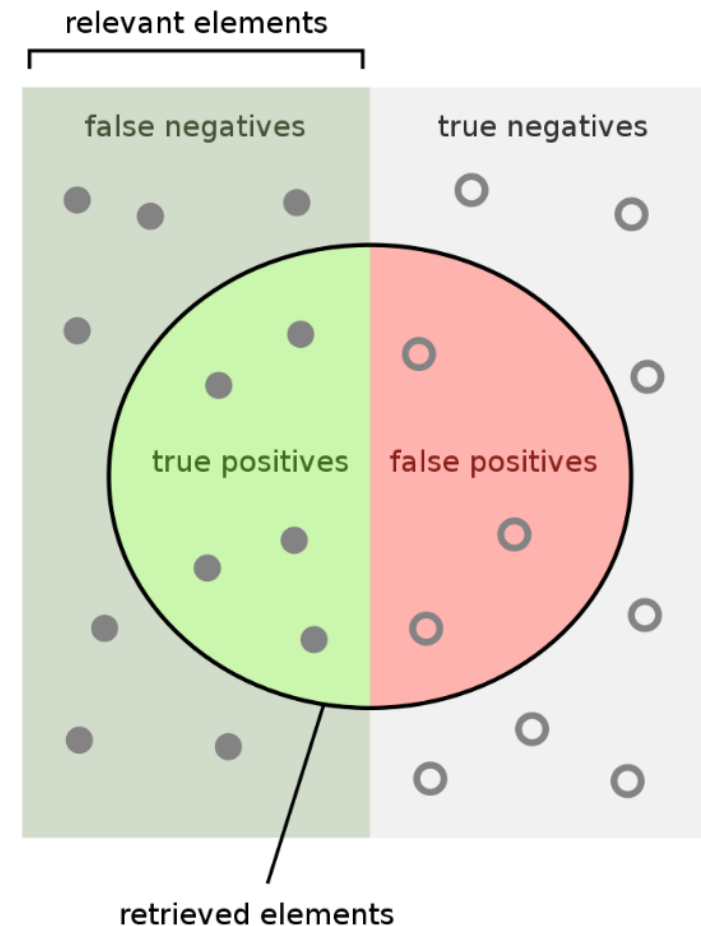
Полнота/Охват

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



Пример:

Больные опасной
болезнью

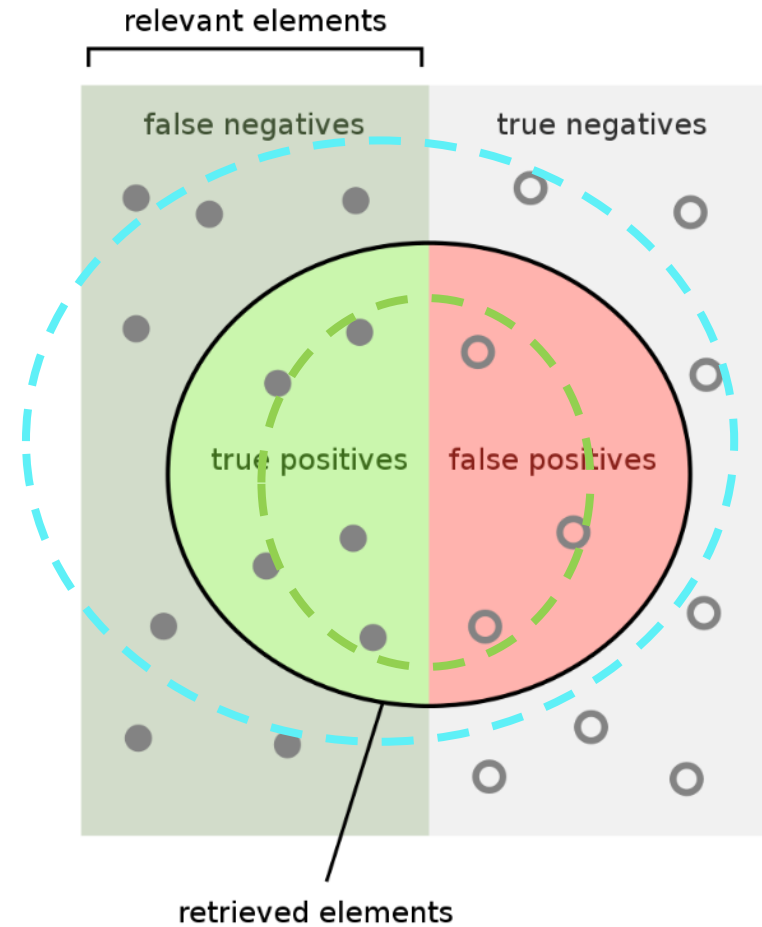
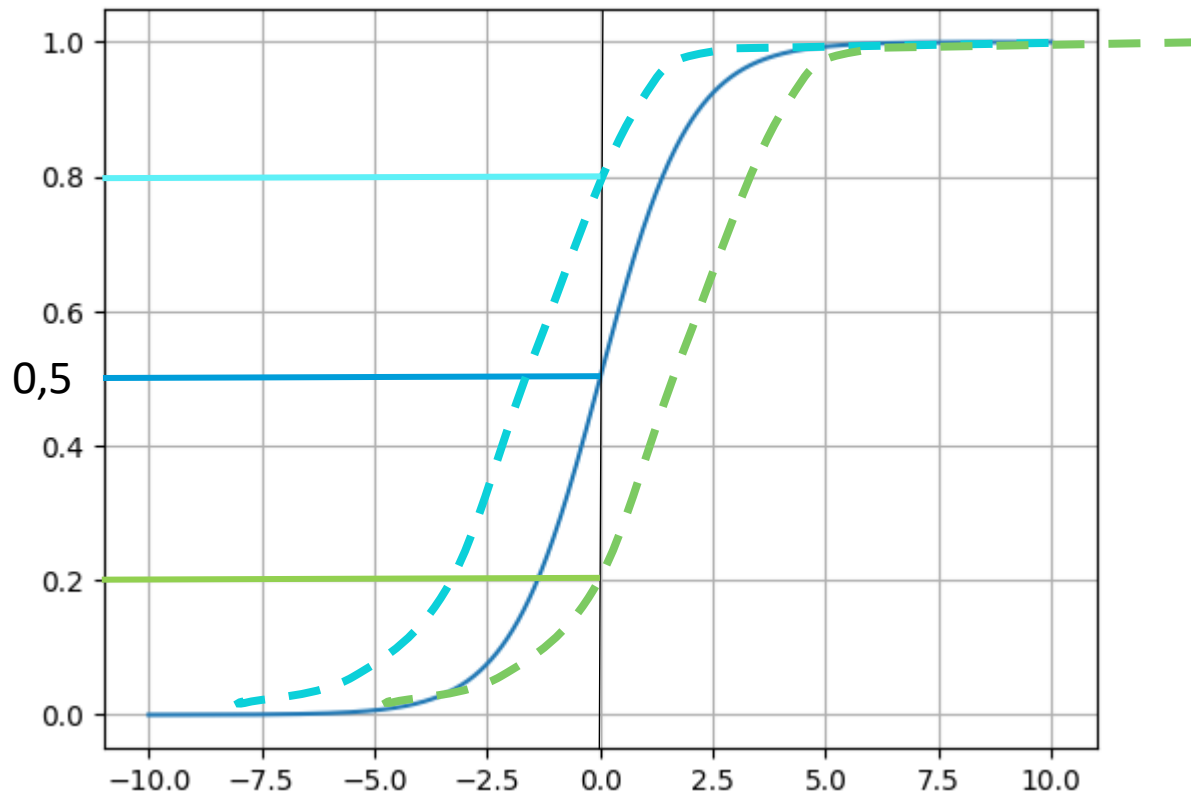


Метрики для классификации

Confusion Matrix Матрица ошибок

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

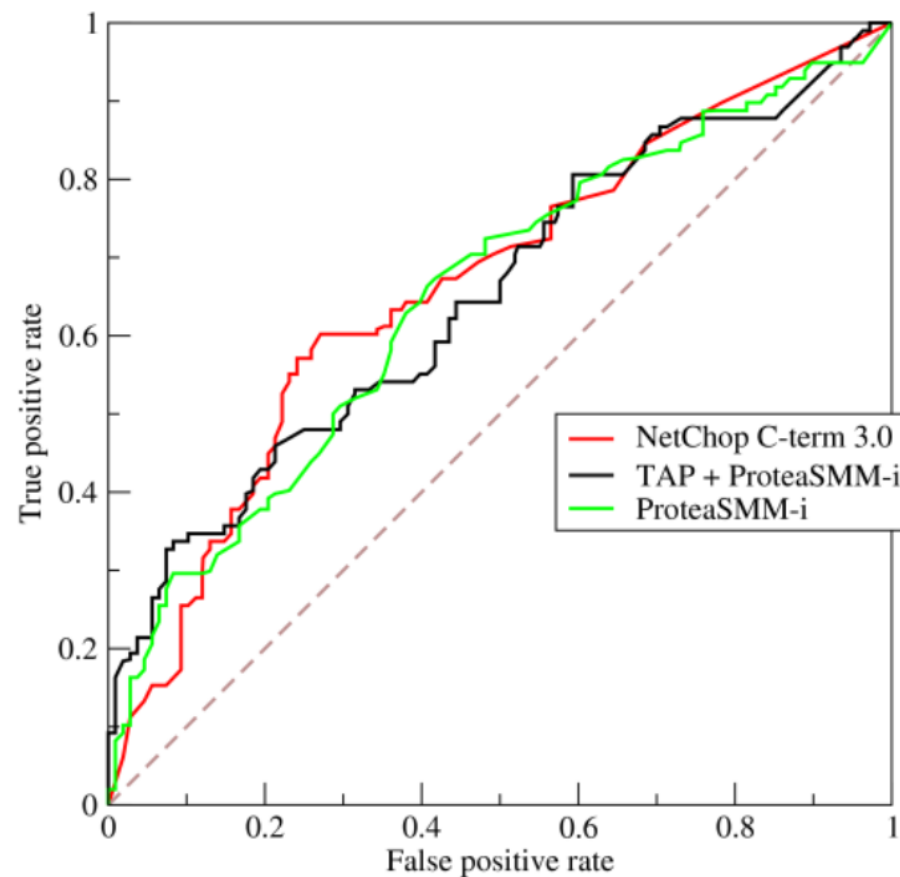
Метрики для классификации



Метрики для классификации

ROC AUC (area under curve)
Площадь под ROC-кривой

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>



Метрики для классификации

Метрика F1

Среднее гармоническое между precision и recall

«Ассурасу без проблем ассурасу»

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Метрики для классификации

Есть и другие метрики

Sources: [5][6][7][8][9][10][11][12] view · talk · edit

		Predicted condition			
		Positive (PP)	Negative (PN)		
Total population = P + N				Informedness, bookmaker informedness (BM) = TPR + TNR - 1	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
Prevalence $= \frac{P}{P+N}$		Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR-) $= \frac{FNR}{TNR}$
Accuracy (ACC) = $\frac{TP+TN}{P+N}$		False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) = $\frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) = PPV + NPV - 1	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$
Balanced accuracy (BA) = $\frac{TPR+TNR}{2}$		F ₁ score $= \frac{2PPV \times TPR}{PPV+TPR} = \frac{2TP}{2TP+FP+FN}$	Fowlkes–Mallows index (FM) = $\sqrt{PPV \times TNR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index = $\frac{TP}{TP+FN+FP}$

Метрики для регрессии

Mean Squared Error (MSE)

Среднеквадратичная ошибка

$$MSE(y, \hat{y}) = \frac{1}{m} \sum_{i=0}^m (y_i - \hat{y}_i)^2$$

Метрики для регрессии

Root Mean Squared Error (RMSE)

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{m} \sum_{i=0}^m (y_i - \hat{y}_i)^2}$$

Метрики для регрессии

Mean Absolute Error (MAE)

$$MAE(y, \hat{y}) = \frac{1}{m} \sum_{i=0}^m |y_i - \hat{y}_i|$$

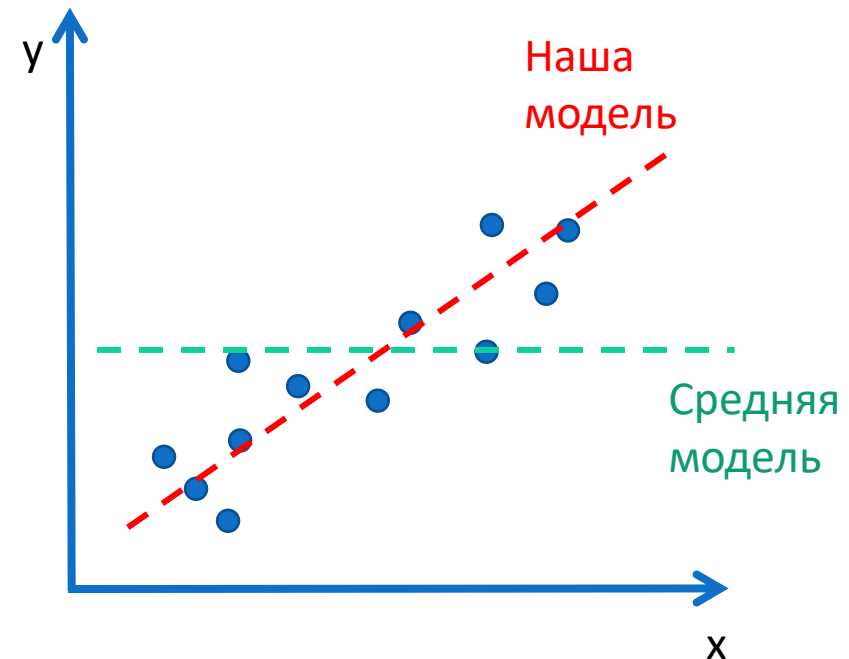
Метрики для регрессии

Коэффициент детерминации
R² score / R-squared

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2$$

$$SS_{tot} = \sum_i (y_i - \bar{y}_i)^2$$

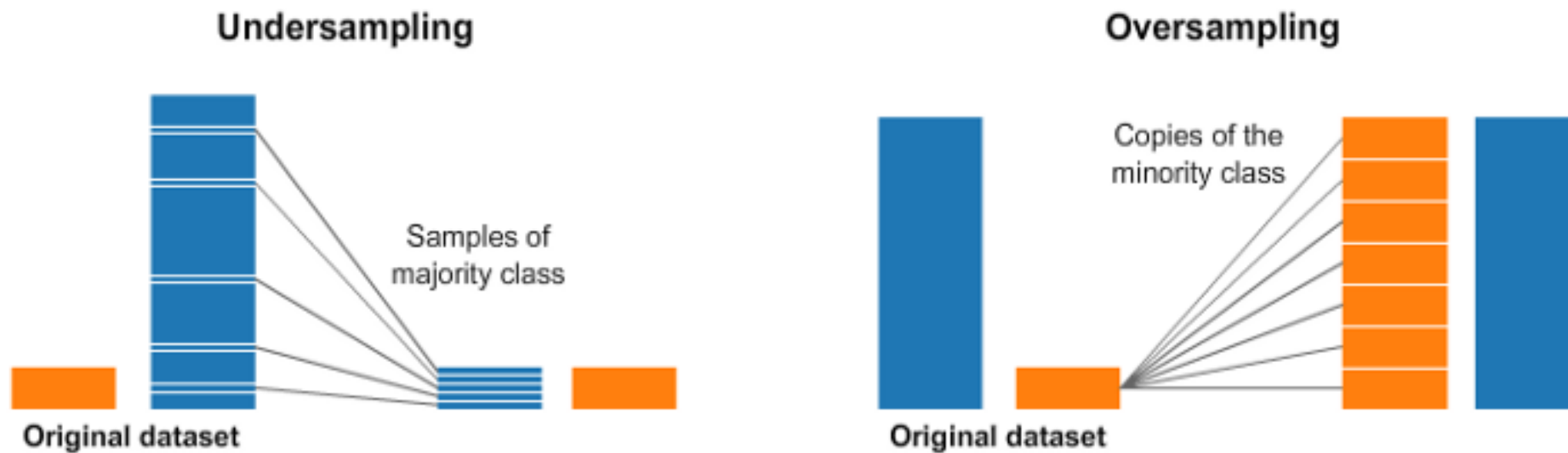
$$R^2 = 1 - \frac{MSE_{model}}{MSE_{avg}}$$



$$E(R^2) \in (-\infty; 1]$$

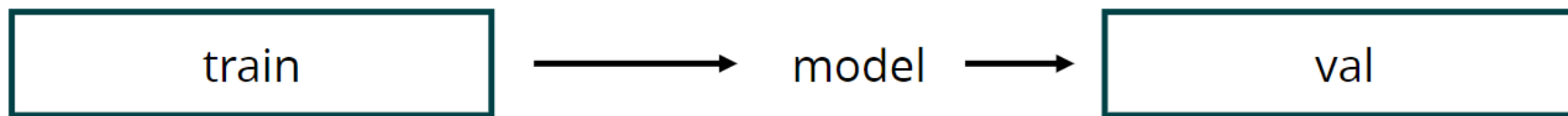
Дополнительно про train/val/test

Необходимо учитывать распределение классов



Дополнительно про train/val/test

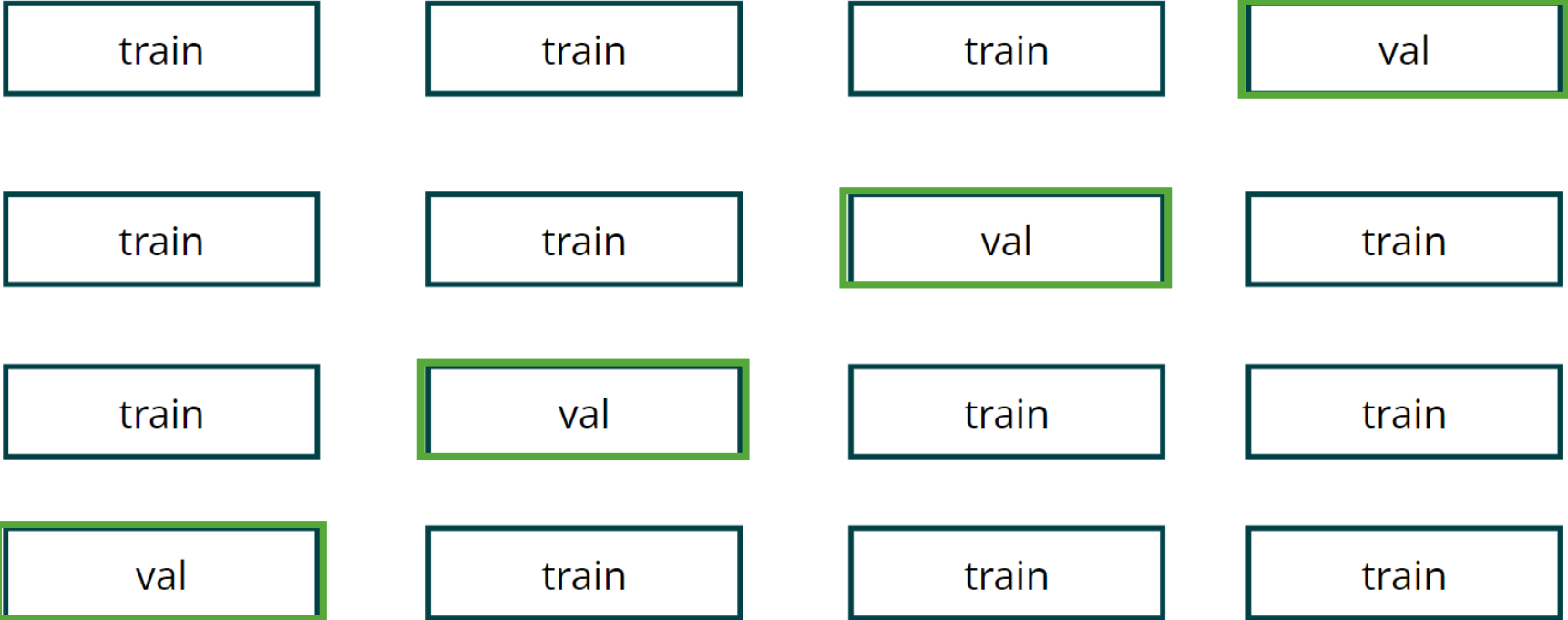
Обычный подход



Что если мало данных в датасете?

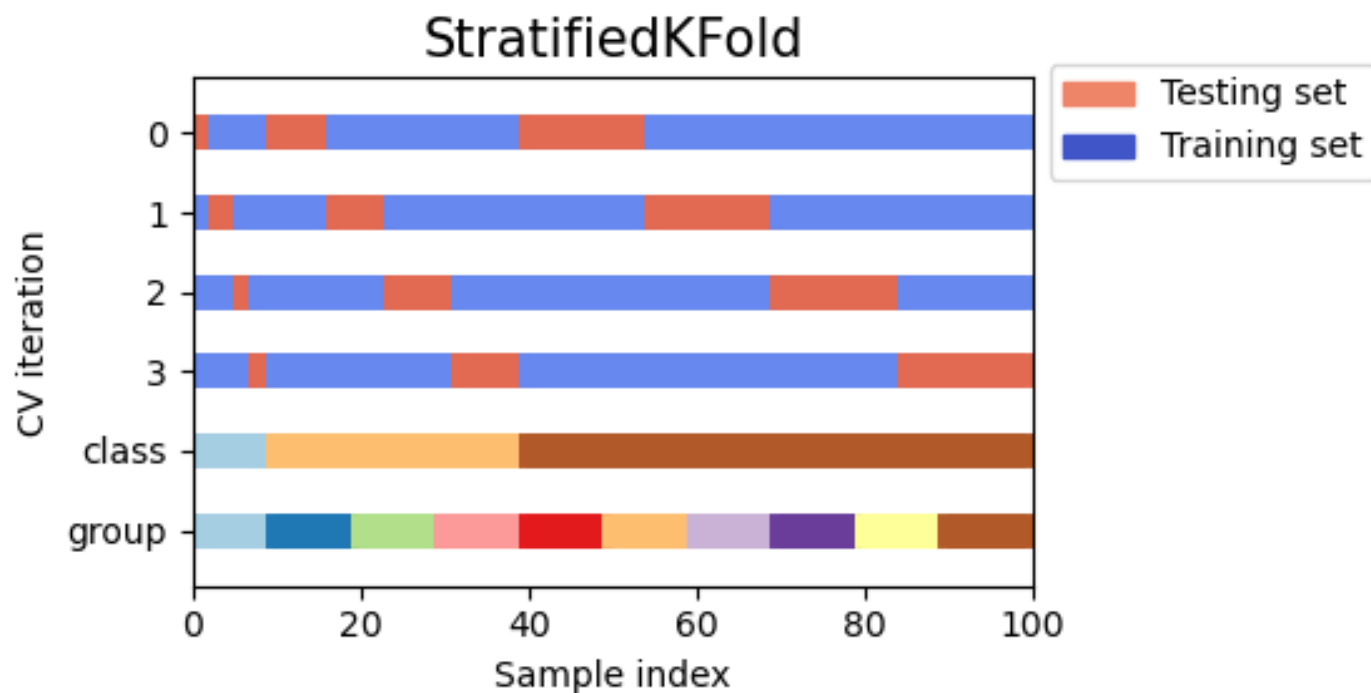
Дополнительно про train/val/test

CrossValidation: усреднение оценки для модели



Дополнительно про train/val/test

Возможно учитывать распределение классов

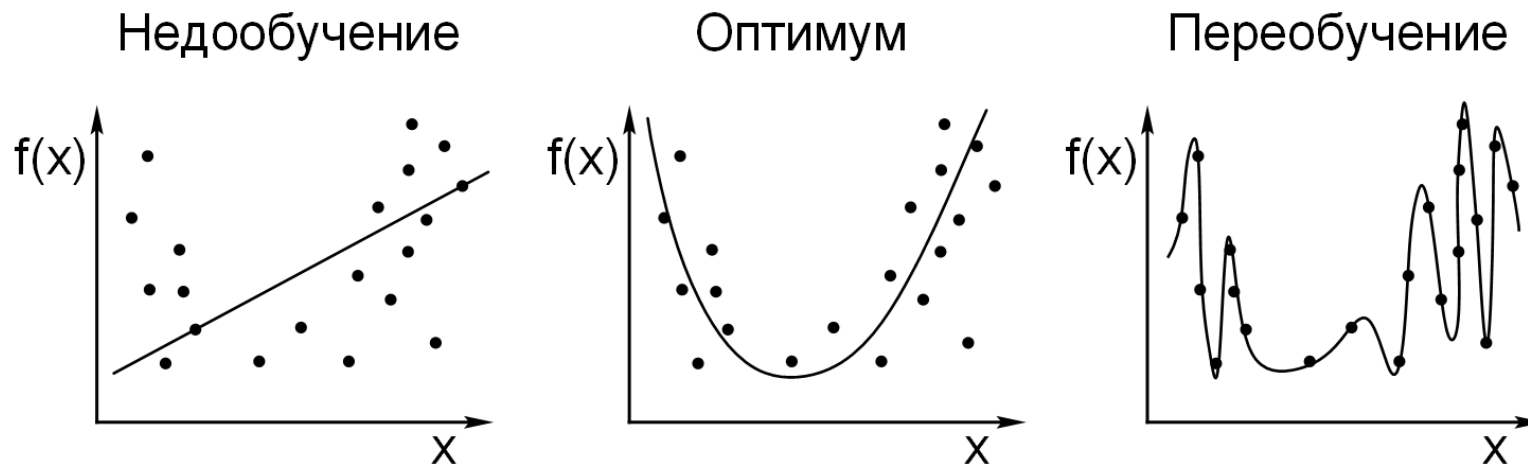


Регуляризация / Regularization

Главная цель модели - **обобщиться** под все экземпляры данных, которые могут встретиться.

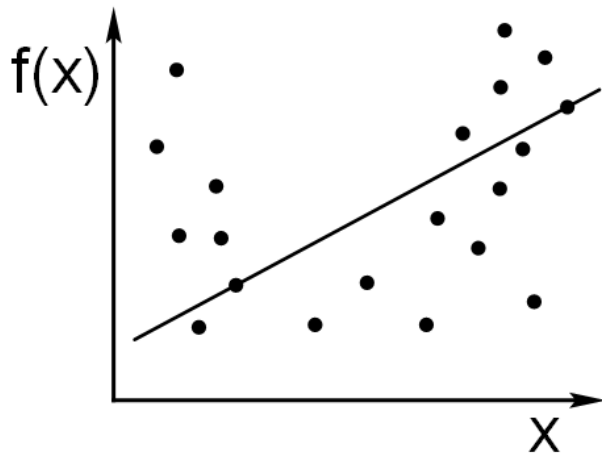
Модель может **переобучиться**, если слишком хорошо выучит тренировочный датасет. Из-за этого результаты на настоящих данных будут хуже.

Регуляризация не позволяет модели быть слишком "сложной" или слишком "самоуверенной".

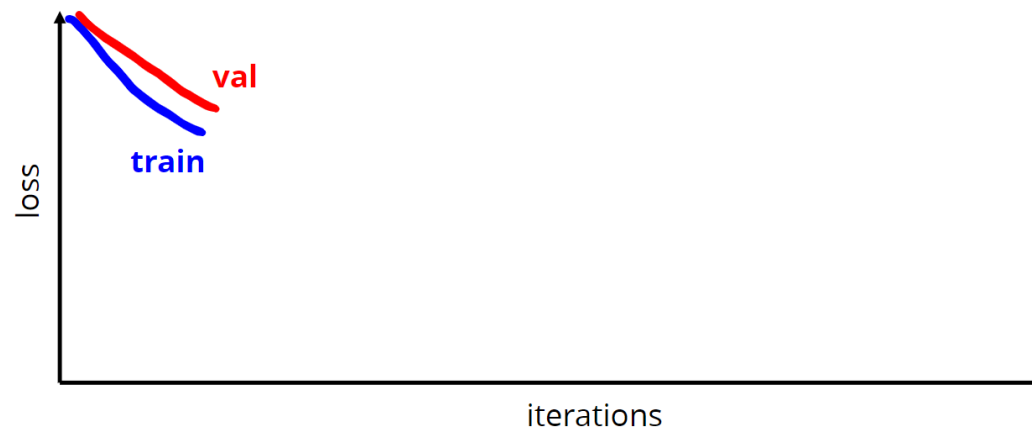


Регуляризация / Regularization

Недообучение



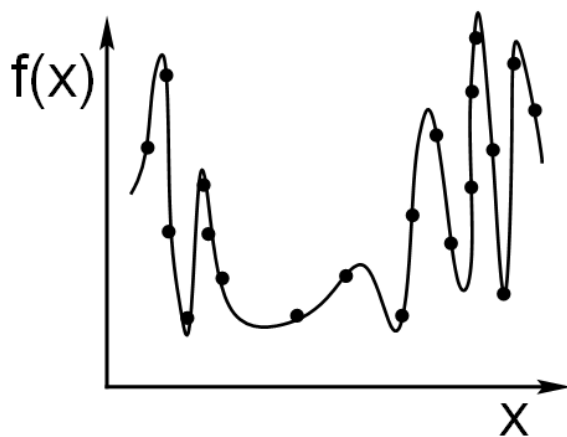
Недообучение



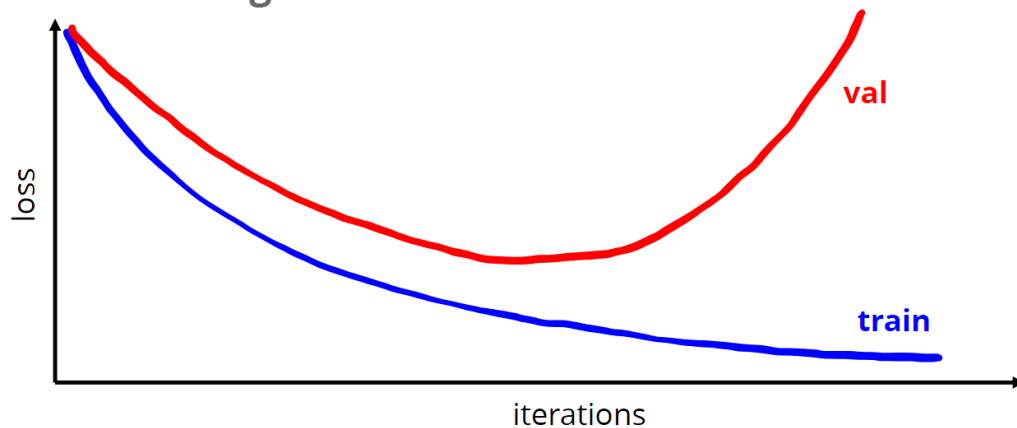
И на тренировочном, и на тестовом датасете значение ошибки велико

Регуляризация / Regularization

Переобучение



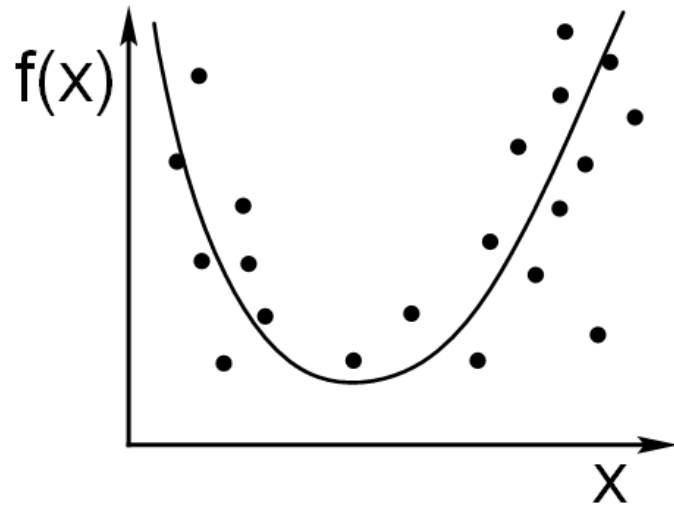
Переобучение
Overfitting



На тренировочном датасете значение ошибки сильно меньше, чем на тестовом

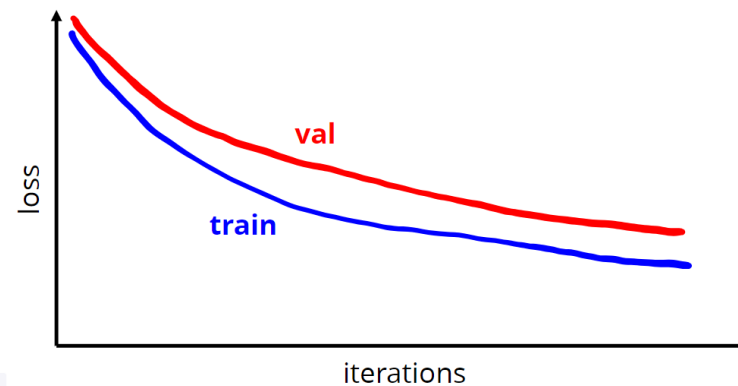
Регуляризация / Regularization

Оптимум



Нормальный вариант

Ошибка на тестовых данных не сильно отличается от ошибки на тестовых данных



Регуляризация / Regularization

Из-за чего возникает переобучение?



Все Марии покупают Волгу?

Имя	Доход	Марка машины
Мария	70 000	Волга
Алексей	150 000	BMW
Михаил	100 000	Dodge
Анастасия	200 000	Renault
Мария	140 000	Волга
Colin McRae	300 000	Subaru
Василий	45 000	Volkswagen
Мария	80 000	Волга
Сергей	100 000	Renault

Регуляризация / Regularization

Проблемы с весами

Переобучение происходит, если модель слишком сильно полагается только на один параметр, то есть вес одной фичи затмевает остальные

$$w_{\text{имя}}x_{\text{имя}} + w_{\text{доход}}x_{\text{доход}} + b = y$$

$$w_{\text{имя}} \rightarrow \infty$$

$$w_{\text{доход}} \rightarrow 0$$

Регуляризация / Regularization

L2 регуляризация
Ridge

$$w_1 := 0$$

$$w_2 := 0$$

for i in range() :

$$w_1 := w_1 - \frac{\delta L(y, \hat{y})}{\delta w_1} \quad - 2 \lambda w_1$$

$$w_2 := w_2 - \frac{\delta L(y, \hat{y})}{\delta w_2} \quad - 2 \lambda w_2$$

$$L_{ridge}(y, \hat{y}) = L(y, \hat{y}) + \lambda \sum_{i=0}^n w_i^2$$

Регуляризация / Regularization

L1 регуляризация

Lasso

(Least Absolute Shrinkage and Selection Operator)

$$L_{lasso}(y, \hat{y}) = L(y, \hat{y}) + \lambda \sum_{i=0}^n |w_i|$$

Параметр регуляризации

λ
(lambda)

В sklearn используется обратный параметр C

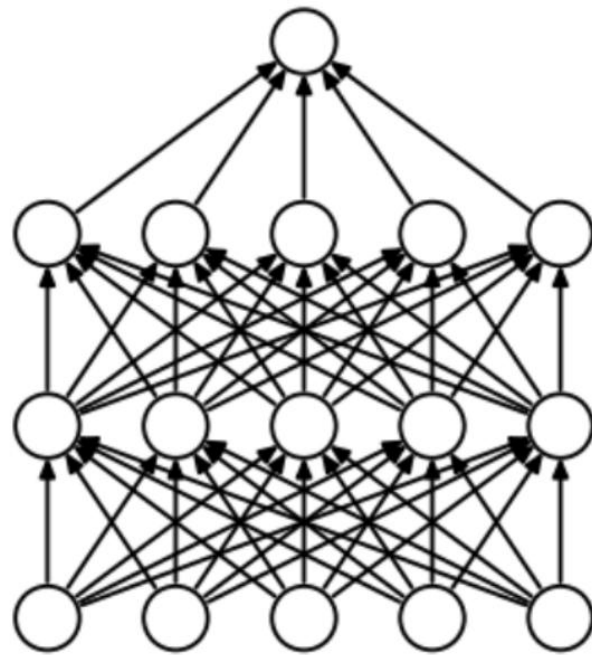
$$C = \frac{1}{\lambda}$$

Регуляризация / Regularization

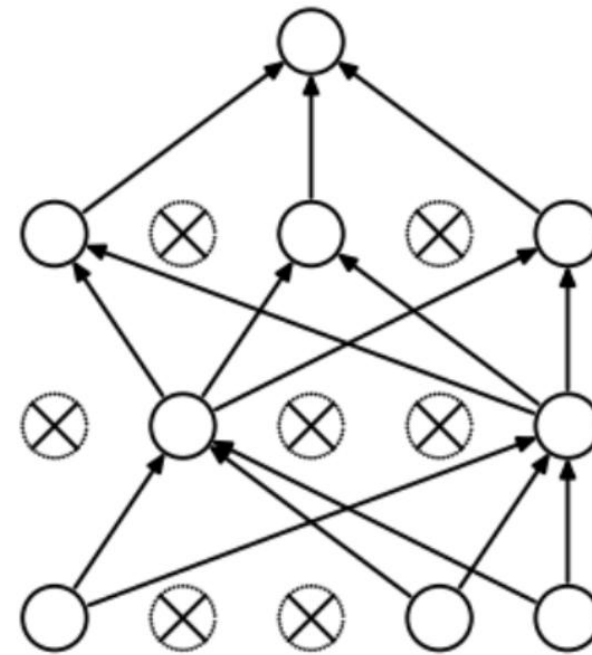
L2 / Ridge	L1 / Lasso
Неразреженный результат	Разреженный результат
Нет отбора фич	Есть отбор фич

Регуляризация в нейронных сетях

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Спасибо за внимание!

Конец Лекции 6