



Интеллектуальные информационные системы

Оценка качества моделей
Выбор модели

Метрики классификации

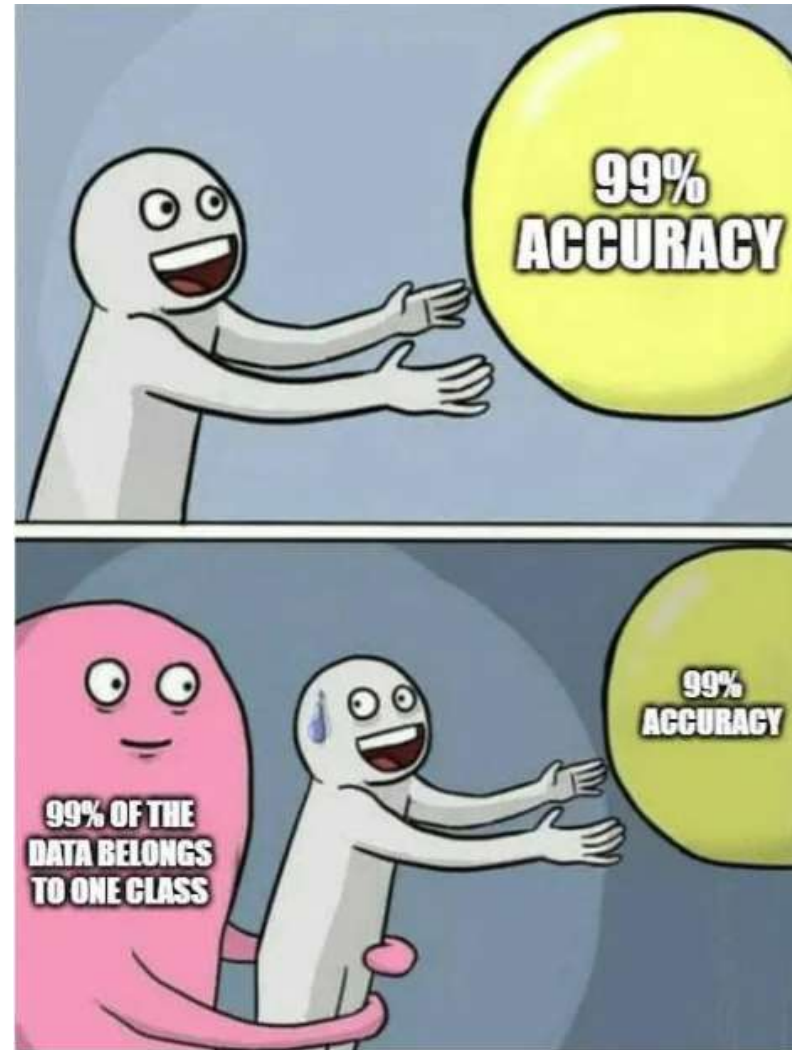
Бинарная классификация

$$Accuracy = \frac{1}{l} \sum_{i=1}^l [y_i = f(x_i)]$$

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

$$Error\ rate = 1 - Accuracy$$

$$Error\ rate = \frac{FP + FN}{FP + FN + TP + TN}$$

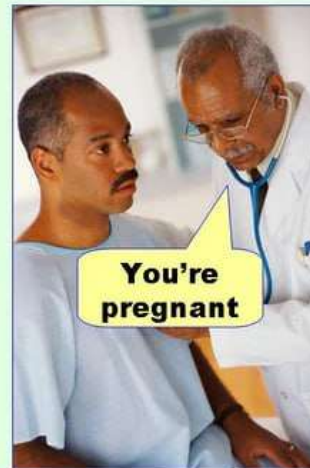


Confusion matrix

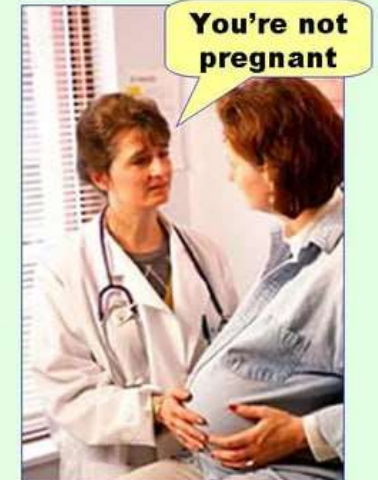
	Ответ классификатора	Правильный ответ
TP, True Positive	$f(x_i) = +1$	$y_i = +1$
TN, True Negative	$f(x_i) = -1$	$y_i = -1$
FP, False Positive	$f(x_i) = +1$	$y_i = -1$
FN, False Negative	$f(x_i) = -1$	$y_i = +1$

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Type I error
(false positive)



Type II error
(false negative)



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

Доля релевантных среди найденных

Доля найденных среди релевантных

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

||

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Доля верных положительных диагнозов

Доля верных отрицательных диагнозов

$$\text{Specificity} = \frac{TN}{TN + FP}$$

relevant elements

false negatives

true negatives

true positives

false positives

retrieved elements

How many retrieved items are relevant?

Precision =



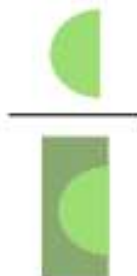
How many relevant items are retrieved?

Recall =



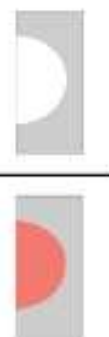
How many relevant items are selected?
e.g. How many sick people are correctly identified as having the condition.

Sensitivity =



How many negative selected elements are truly negative?
e.g. How many healthy people are identified as not having the condition.

Specificity =



$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

$$= \frac{2 \times Recall \times Precision}{Recall + Precision} = \frac{TP}{TP + \frac{FP + FN}{2}}$$

$$F_\beta = (\beta^2 + 1) \frac{Recall \times Precision}{Recall + \beta^2 Precision}$$

Бинарная классификация: вероятности классов

Вероятностная модель бинарной классификации

$$y_i \in \{-1, +1\}: \quad g(x, w) = P(y = +1|x, w)$$

Функция потерь log-loss:

$$L(w) = \sum_{i=1}^l [y_i = +1] \ln g(x, w) + [y_i = -1] \ln(1 - g(x, w)) \rightarrow \max_w$$

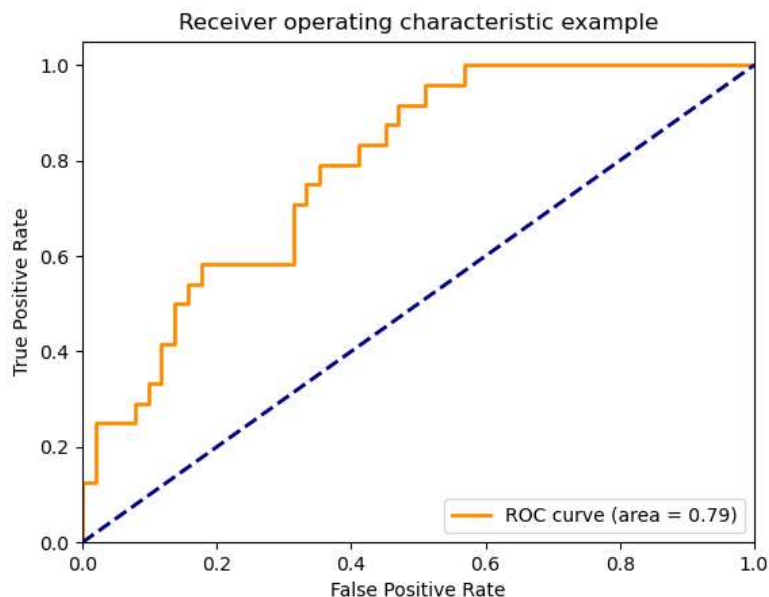
- TPR (true positive rate) – полнота, доля положительных объектов, правильно предсказанных положительными:

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

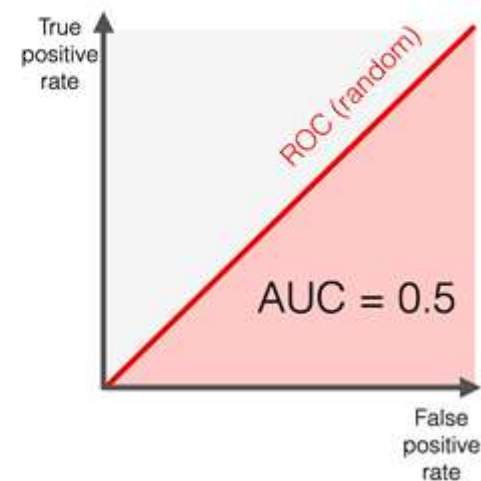
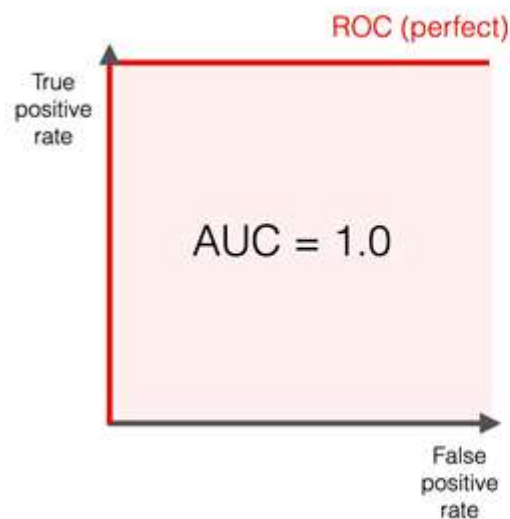
- FPR (false positive rate) – доля отрицательных объектов, неправильно предсказанных положительными:

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

Кривая ошибок - ROC-curve

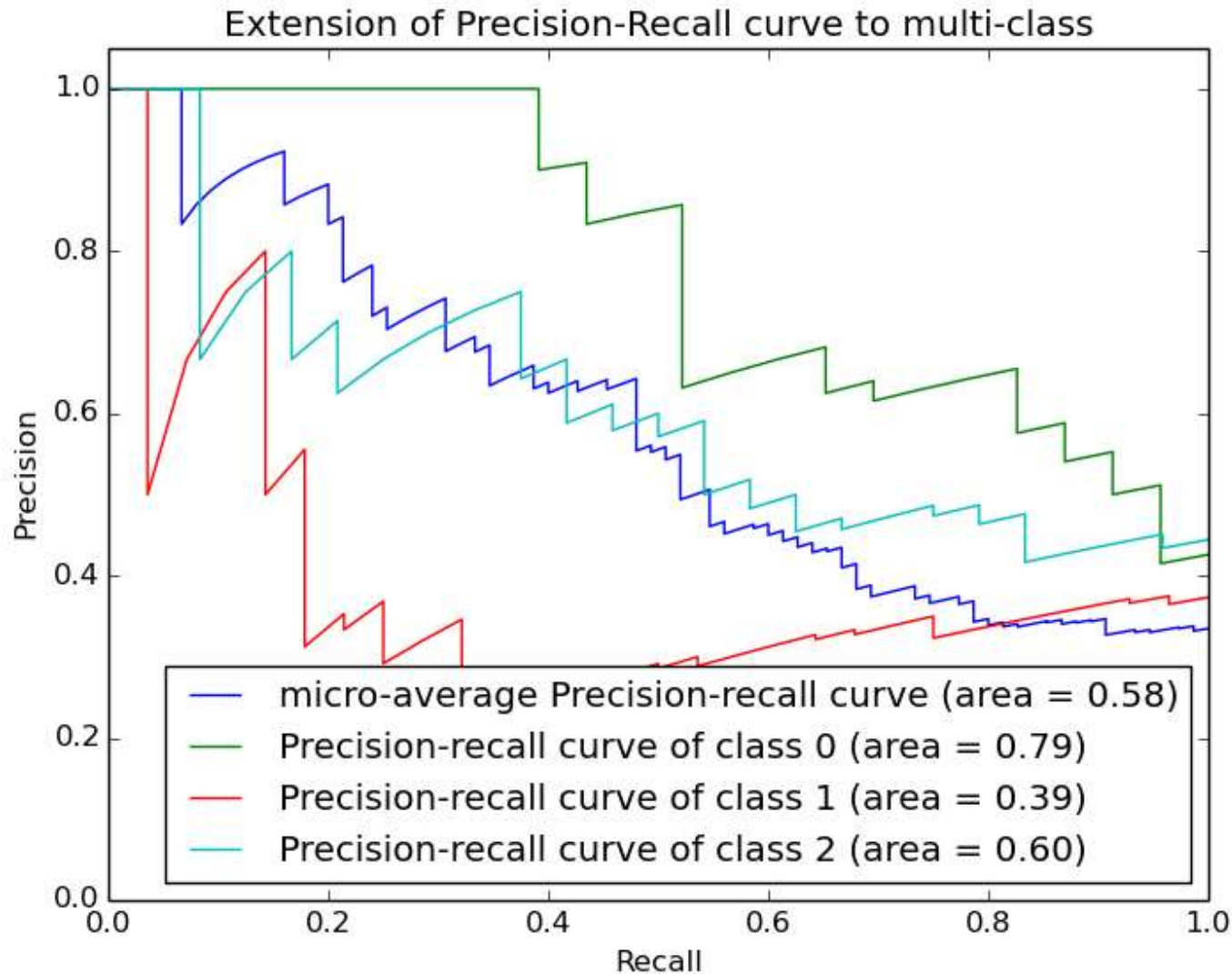


Площадь под кривой - AUC



Precision-Recall curve

AUPRC



Многоклассовая классификация

- Микроусреднение (micro averaging):



Вклад минорного класса в
общую метрику незаметен

$$P = \frac{\sum_y TP_y}{\sum_y (TP_y + FP_y)}$$

$$R = \frac{\sum_y TP_y}{\sum_y (TP_y + FN_y)}$$

- Макроусреднение (macro averaging):



Вклад каждого класса в
общую метрику одинаковый

$$P = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FP_y}$$

$$R = \frac{1}{|Y|} \sum_y \frac{TP_y}{TP_y + FN_y}$$

Метрики регрессии

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

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

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^l (\hat{y}_i - y_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2}$$

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

Метрики регрессии

$$MAPE(y, \hat{y}) = \frac{1}{l} \sum_{i=1}^l \frac{|\hat{y}_i - y_i|}{y_i} \times 100\%$$

$$SMAPE(y, \hat{y}) = \frac{1}{l} \sum_{i=1}^l \frac{|\hat{y}_i - y_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100\%$$

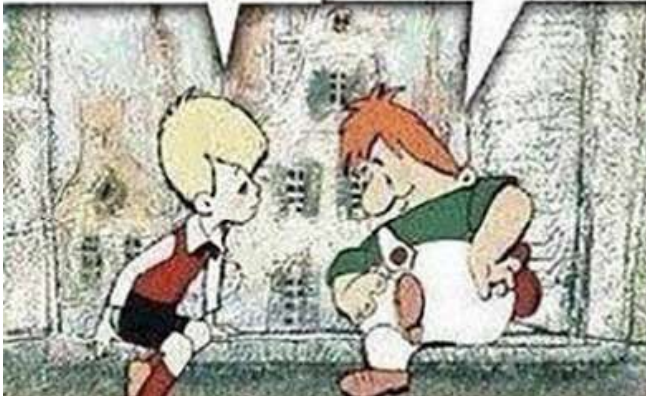
Выбор модели

ПИТОНИСТАМ
МАЛО ПЛАТЯТ...



ЭТО ЗАЧЕМ?

БУДЕШЬ РАЗРАБАТЫВАТЬ
ИСКУССТВЕННЫЙ
ИНТЕЛЕКТ ЗА 300К/СЕК



СТАНОВИСЬ ДАТА
САЕНТИСТОМ КАК Я

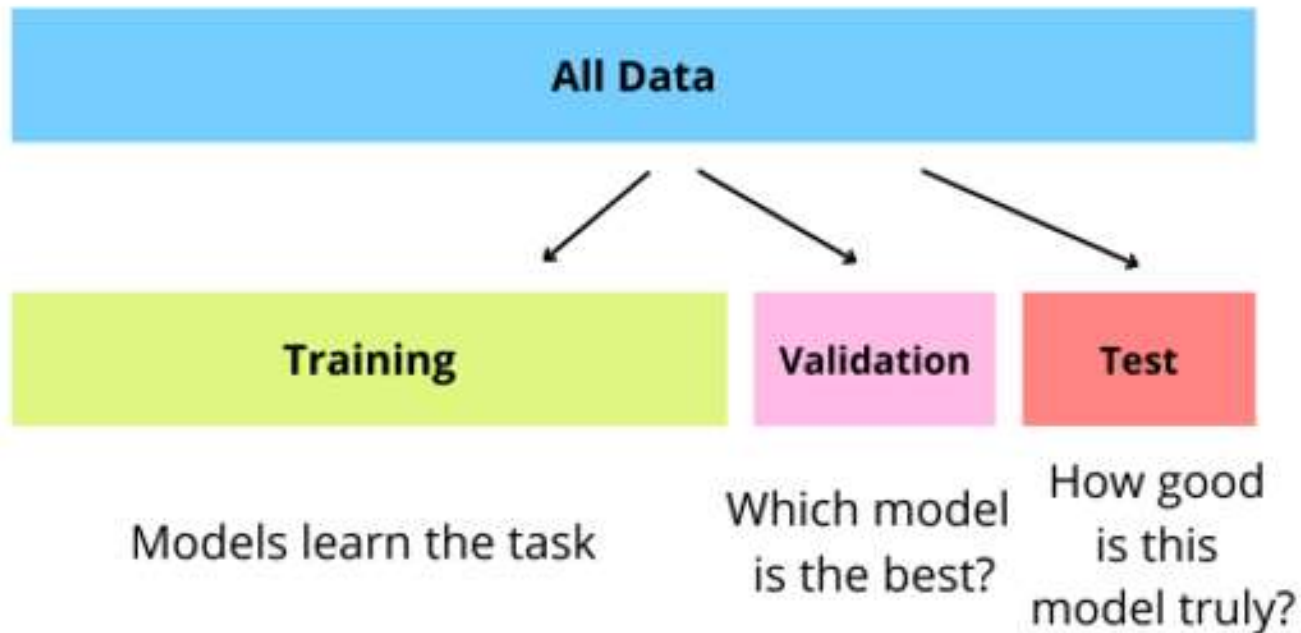


ТЫ ЧЁ ПЁС,
Я МАТЕМАТИК

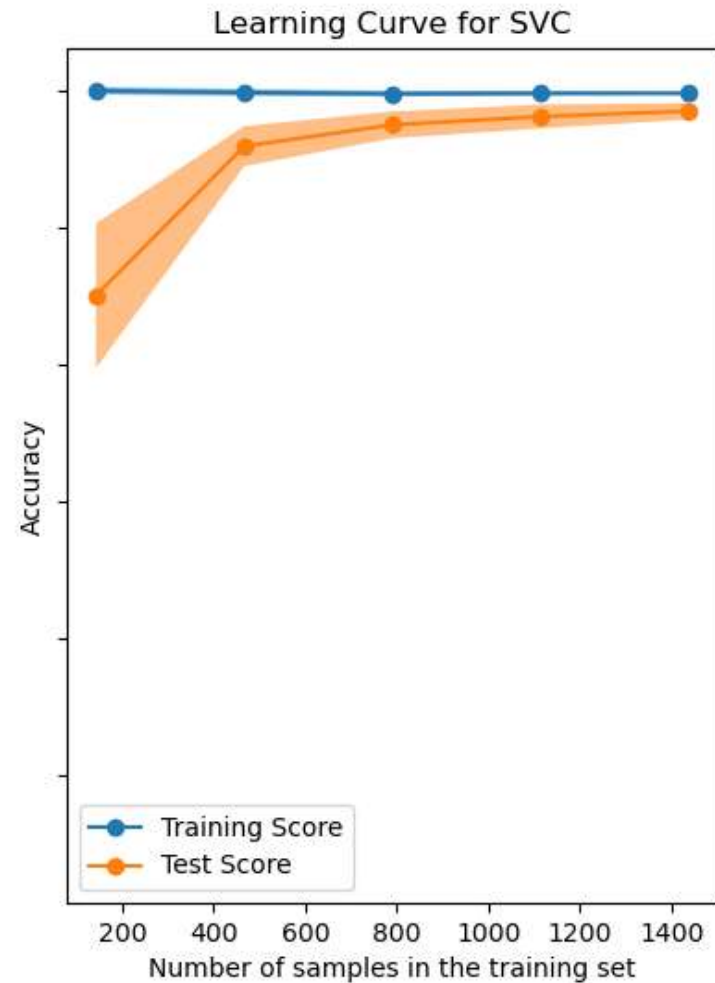
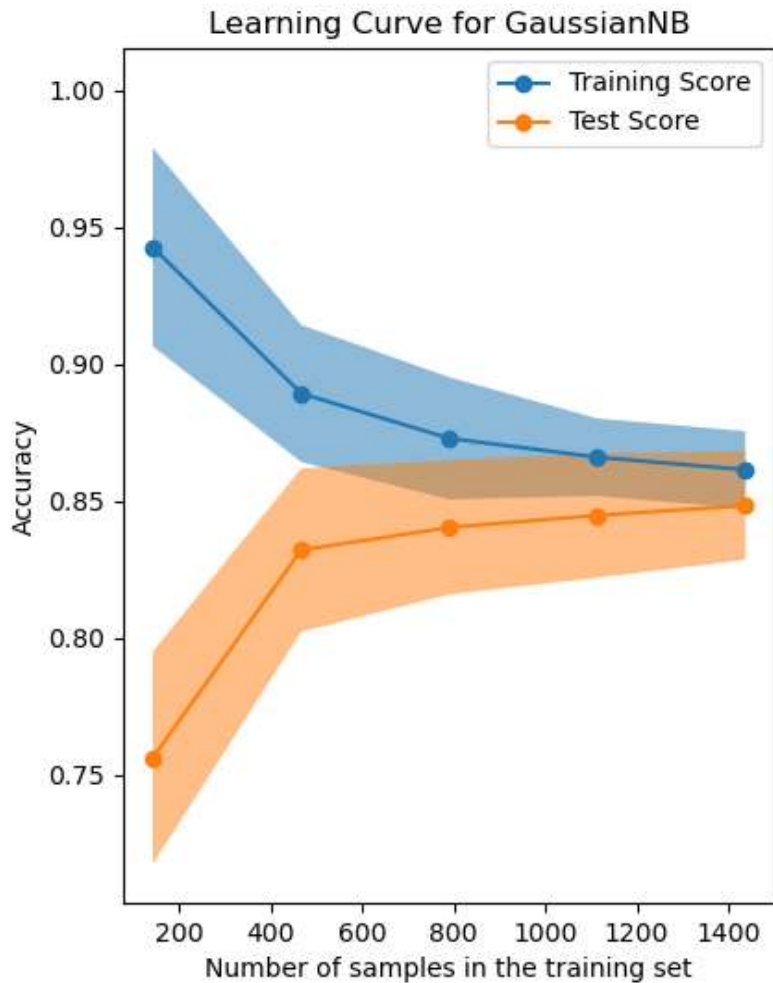
ТАК ТЫ ЖЕ ПРОСТО РАНДОМНО
ПОДБИРАЕШЬ КОЭФФИЦИЕНТЫ
ПОКА КРОСС-ВАЛИДАЦИЯ
НЕ ДАСТ НОРМАЛЬНЫЙ РЕЗУЛЬТАТ



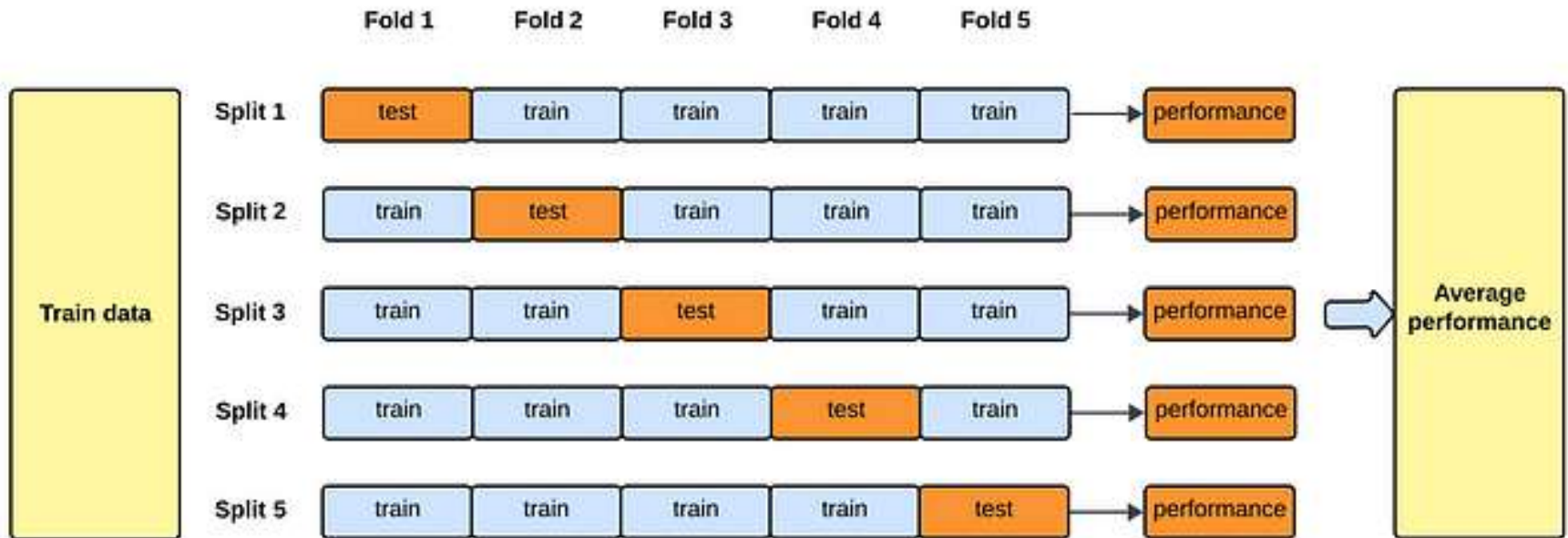
- Hold out (train-test split) – на самом деле train-validation-test split



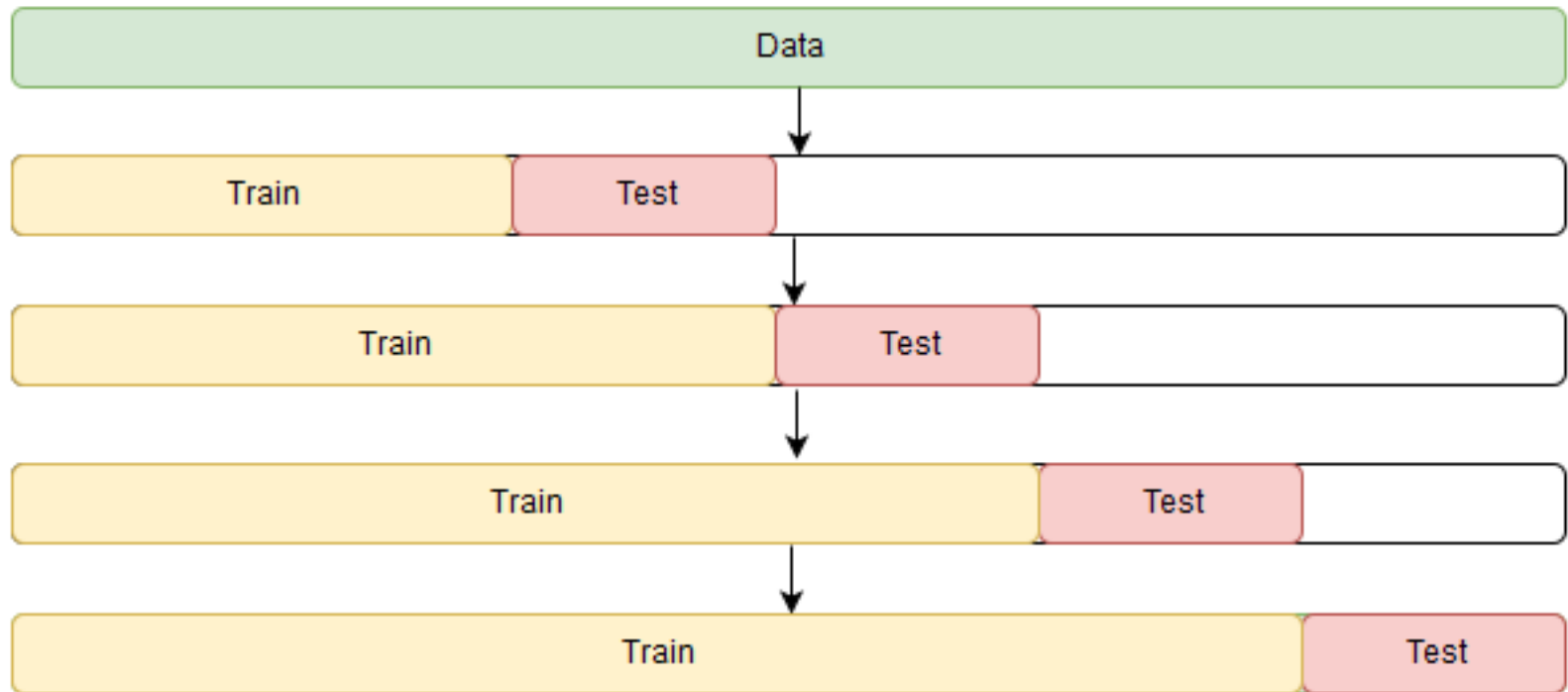
Model scalability



- k-Fold и его частные случаи – stratified k-Fold, leave-one-out, leave-p-out



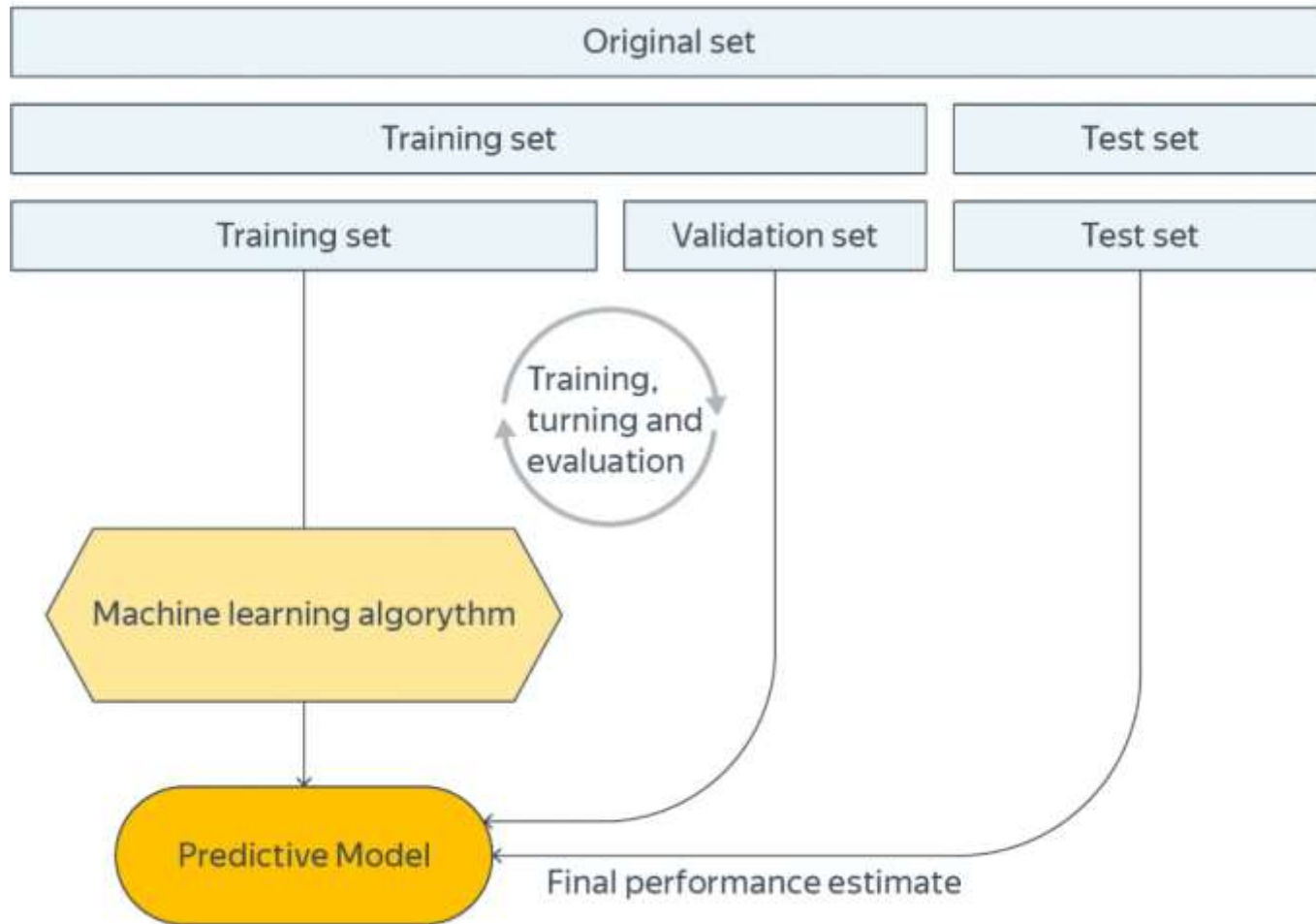
- Time-series cross-validation



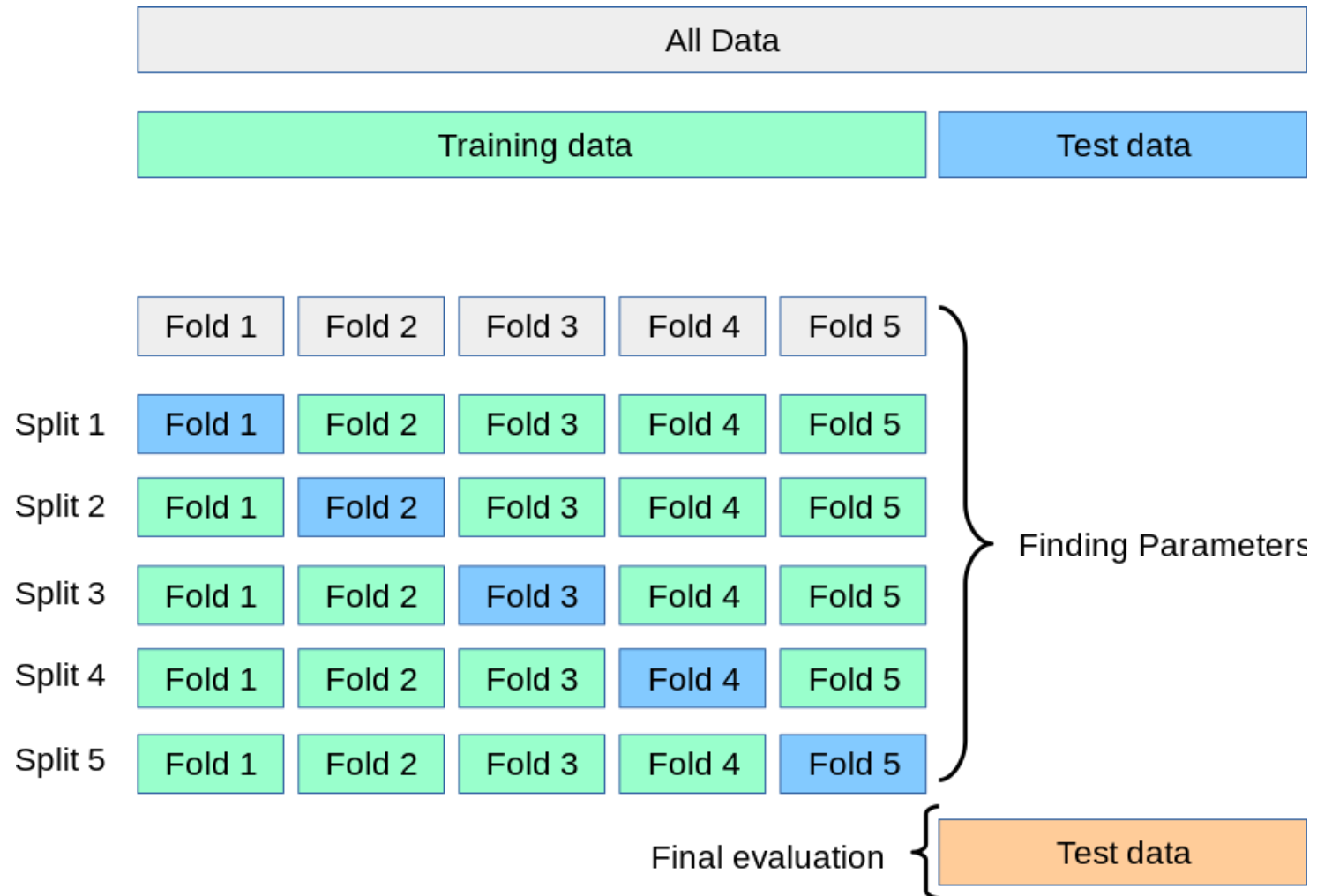
Подбор гиперпараметров

- Параметры – настраиваются в процессе обучения модели (веса, структура решающего дерева и другие)
- Гиперпараметры – характеристики модели, которые фиксируются до начала обучения (глубина решающего дерева, количество ближайших соседей, коэффициенты регуляризации, learning rate и другие)

- Подбор гиперпараметров на валидационной выборке



- Подбор гиперпараметров и кросс-валидация



Методы подбора гиперпараметров

Grid search (поиск по сетке) – полный перебор в заданном подмножестве

Достоинства:

- Интуитивно понятен
- Легко распараллелить

Недостатки:

- Ограничен в выборе заданным подмножеством (сеткой)
- Не использует результаты других итераций
- Долго работает

Random search (случайный поиск в заданном подмножестве) – N комбинаций (итераций) путем семплирования из каждого распределения гиперпараметра

Достоинства:

- Быстрее, чем Grid Search
- Эффективность за счет использования распределений
- Легко распараллелить

Недостатки:

- Ограничен в выборе заданным подмножеством (не так жестко как Grid Search) и количеством итераций
- Не использует результаты других итераций

Байесовская оптимизация – вероятностная модель, приближающая распределение целевой функции (значение валидационных метрик в зависимости от комбинации гиперпараметров) в зависимости от предыдущих итераций

Достоинства:

- Использует результаты предыдущих итераций
- Может расширять заданные изначально границы множества поиска гиперпараметров
- Моделирует внутренние зависимости между гиперпараметрами
- Более эффективен, чем Random Search

Недостатки:

- Сложно распараллелить
- Долго сходится

Tree-structured Parzen Estimator (TPE) – итерационный вероятностный алгоритм, разделяющий гиперпараметры на две группы – те, которые приводят к значению целевой функции ниже порога (успешные) и ниже порога (неуспешные)

Достоинства:

- Использует результаты предыдущих итераций
- Учитывает зависимость между гиперпараметрами, меняет один гиперпараметр только при достижении другим определенного значения
- Имеет линейную сложность по числу гиперпараметров
- Достигает высоких результатов по качеству

Недостатки:

- Может работать довольно медленно

Population based training (PBT) – эволюционный алгоритм, обучает целую популяцию моделей, причем модели периодически обмениваются гиперпараметрами на основе своего успеха

Достоинства:

- Параллельный по определению
- Использует результаты предыдущих итераций

Недостатки:

- Требуется большого количества воркеров, иначе неэффективен