

# План занятия

- Измерение качества
- Задача классификации
  - kNN
  - Naive Bayes
  - Деревья решений
- Кросс валидация

# Материал для самоподготовки, повторения

- Открытый курс по машинному обучению
  - [Текст](#)
  - [Видео](#)
- [Метрики в задаче классификации](#)
- [Метрики в задаче регрессии](#)
- Про модель kNN
  - [вики](#)
  - [хабр](#)
  - [machinelearning.ru](http://machinelearning.ru)

# Материал для самоподготовки, повторения

- Про модель Naive Bayes
  - [ВИКИ](#)
  - [ВЫВОД](#)
- Про деревья решений
  - <https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac>
  - <http://datareview.info/article/derevya-resheniy-i-algoritmyi-ih-postroeniya/>
  - <https://habr.com/ru/company/ods/blog/322534/>

# Постановка задачи машинного обучения

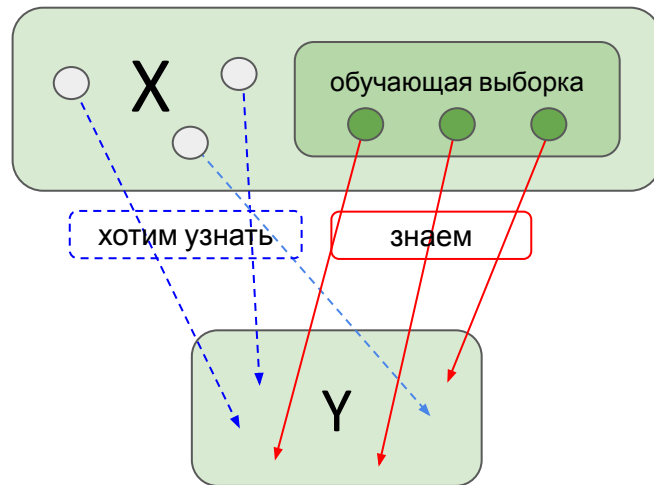
$X$  — множество *объектов*

$Y$  — множество *ответов* (например, два класса или произвольные числа)

$y: X \rightarrow Y$  — неизвестная закономерность

**Дано:** обучающая выборка,  $\{x_1, x_2, \dots, x_n\}$  — подмножество множества  $X$

**Цель:** подобрать *алгоритм*, приближающий функцию  $y(x)$ .



# Метрики

# Измерение качества модели

Чтобы понять, насколько адекватно ведет себя модель, нужно каким-то образом численно оценить ее качество.

Метрика — это функция вида:

$$metric(\mathbf{y}, \hat{\mathbf{y}})$$

где  $\mathbf{y}$  — это правильное значение целевой переменной (**label**),

а  $\hat{\mathbf{y}}$  — значение, предсказанное моделью (**prediction**).

# Примеры метрик

Классификации:

- **accuracy** — процент правильных предсказаний среди всех примеров
- [precision — точность](#)
- [recall — полнота](#)
- [f1 — объединяет полноту и точность](#)
- [ROC-AUC](#) - вероятность правильного ранжирования двух случайных примеров

Регрессии:

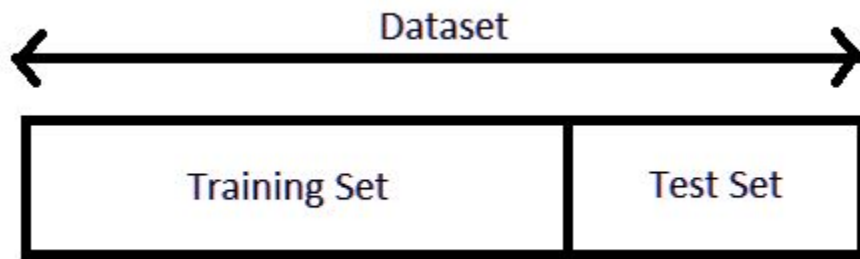
- [MSE — среднеквадратичная ошибка](#)
- [MAE — средний модуль ошибки](#)
- [R2 score — коэффициент детерминации](#)

Полезный материал по [метрикам](#)

# Отложенная выборка

Можно “отложить”, скажем, 20% обучающей выборки для валидации модели. Использовать 80% выборки для обучения и 20% для тестирования.

- Оценка на тестовой выборке будет несмещенной
- Тестовая выборка маленькая - оценка будет иметь погрешность





# Классификация

За формальной постановкой обращаемся [сюда](#)

Пример: KNN

# K Nearest Neighbors

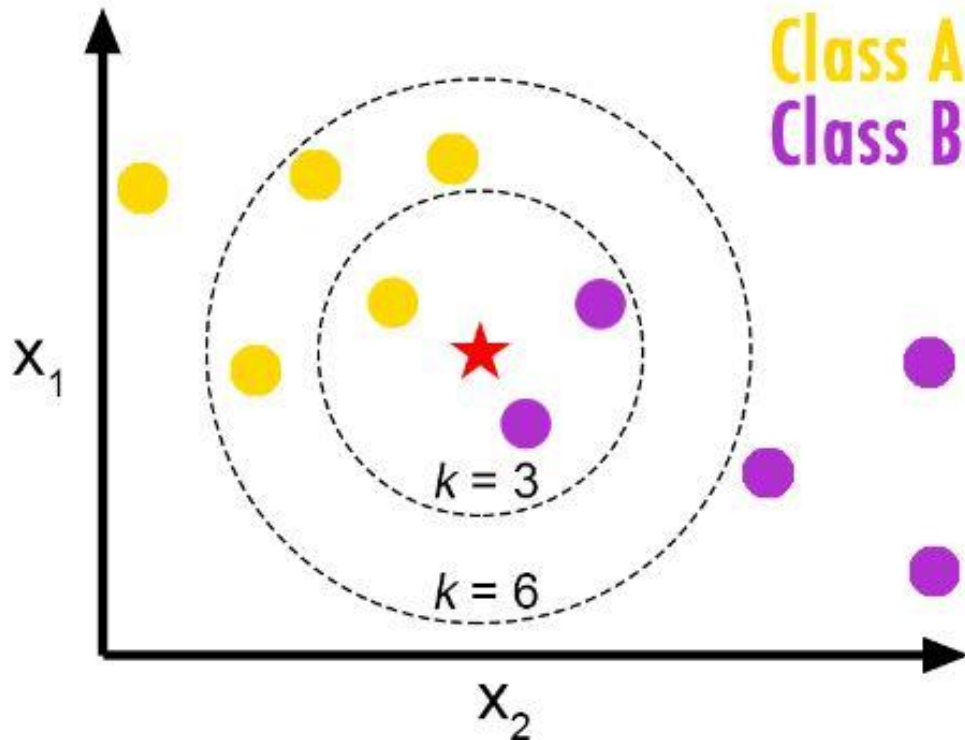
## Метод K ближайших соседей

- На вход подается вектор - признаковое описание какого-то объекта
- Находится K ближайших к нему векторов, для которых ответ известен
- Ответ для новой точки выбирается с помощью
  - Усреднения в случае регрессии
  - Голосования в случае классификации
- Возможно также усреднение/голосование с весами

# KNN классификация

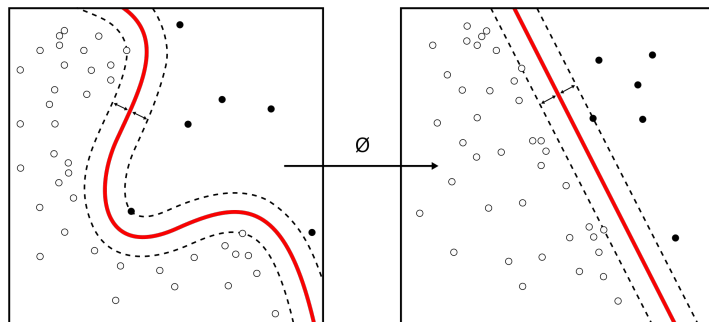
$K$  - внешний параметр. Он подбирается так, чтобы модель работала как можно лучше.

Результат предсказания для некоторых точек может зависеть от  $K$



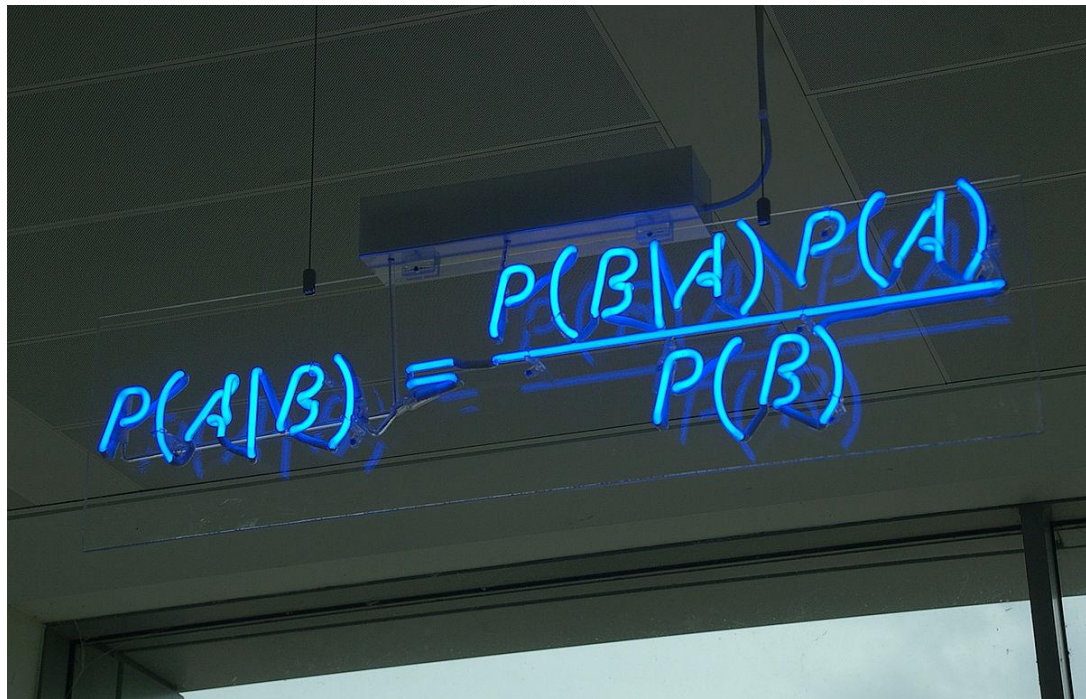
# Тезисы

- Данные нужно превращать в числа — **признаковое описание**
- В данных должна присутствовать **целевая переменная**
- Можно обучить модель предсказывать целевую переменную — это называется **обучение с учителем**
- Если предсказывается число — это **регрессия**, если класс — **классификация**
- Качество модели оценивается с помощью **метрик**



Пример: Naive Bayes

# Теорема Байеса



A photograph of a blue neon sign mounted on a dark wall. The sign displays the formula for Bayes' Theorem in a stylized, handwritten font. The formula is  $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ . The sign is illuminated, and the background is dark.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

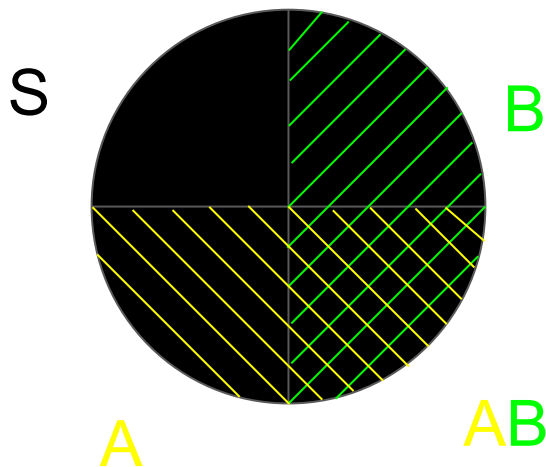


# Теорема Байеса

$$P(B \cap A) = P(A \cap B)$$

$$P(B|A)P(A) = P(A|B)P(B)$$

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$



A photograph of a chalkboard with the formula for Bayes' Theorem written in blue chalk. The formula is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

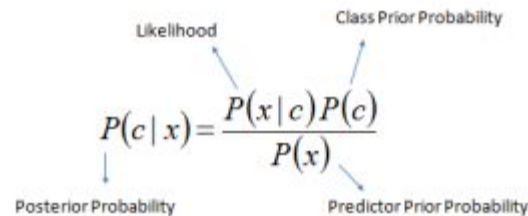


# О чем говорит теорема Байеса?

$x$  — набор признаков

$c$  — метка класса

1. Class Prior Probability априорная вероятность класса (грубо говоря как часто встречается класс)
2. Predictor Prior Probability априорная вероятность признаков (с какой вероятностью получается такой набор признаков)
3. Posterior Probability апостериорная вероятность класса (какова вероятность класса при данном наборе признаков)



The diagram shows the formula  $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$  with arrows pointing from labels to the terms in the formula. 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

# О чем говорит теорема Байеса?



14



Although there are four components listed in Bayes' law, I prefer to think in terms of three conceptual components:

$$\underbrace{P(B|A)}_2 = \frac{P(A|B)}{\underbrace{P(A)}_3} \underbrace{P(B)}_1$$

1. The **prior** is what you believed about  $B$  *before* having encountered a new and relevant piece of information (i.e.,  $A$ ).
2. The **posterior** is what you believe (or ought to, if you are rational) about  $B$  *after* having encountered a new and relevant piece of information.
3. The **quotient of the likelihood divided by the marginal probability of the new piece of information** indexes the *informativeness* of the new information for your beliefs about  $B$ .

# О чем говорит теорема Байеса?

Воспользуемся теоремой для подсчёта вероятности заболевания по симптомам

По наличию симптомов нужно определить с какой вероятностью у пациента covid-19.

Возьмем один из симптомов, например, головную боль  $\rightarrow P(\text{головная боль} \mid + \text{covid}) = 0.7$

$P(+ \text{ covid} \mid \text{головная боль}) = ? = P(\text{головная боль} \mid + \text{covid}) * P(+ \text{ covid}) / P(\text{головная боль})$

# О чем говорит теорема Байеса?

Согласно [ВОЗ](#), можем взять  $P(\text{головная боль}) = 0.5$

Предположим, что covid через какое-то время станет сопоставим с обычным гриппом. Тогда, по данным [ВОЗ](#) в северном полушарии  $P(+ \text{ covid}) = 0.1$

$P(+ \text{ covid} \mid \text{головная боль}) = P(\text{головная боль} \mid + \text{ covid}) * P(+ \text{ covid}) /$

$P(\text{головная боль}) = 0.7 * 0.1 / 0.5 = (0.7/0.5) * 0.1 = 1.4 * 0.1 = 0.14$

# Наивный Байес

Мы учли лишь один фактор. Но в реальности факторов много. Выражение для теоремы Байеса будет выглядеть так

$$P(c_i | \theta_1, \theta_2, \dots, \theta_n) = \frac{P(\theta_1, \theta_2, \dots, \theta_n | c_i) P(c_i)}{P(\theta_1, \theta_2, \dots, \theta_n)}$$

$$P(\theta_1, \theta_2, \dots, \theta_n) = P(\theta_1 \cap \theta_2 \cap \dots \cap \theta_n)$$

# Наивный Байес

Осталось сделать предположение об условной независимости признаков!

$$P(c_i | \theta_1, \theta_2, \dots, \theta_n) = \frac{P(\theta_1, \theta_2, \dots, \theta_n | c_i) P(c_i)}{P(\theta_1, \theta_2, \dots, \theta_n)} \quad P(\theta_1, \theta_2, \dots, \theta_n) = P(\theta_1 \cap \theta_2 \cap \dots \cap \theta_n)$$

$$P(A, B | C) = P(A | C) P(B | C)$$

$$P(c_i | \theta_1, \theta_2, \dots, \theta_n) = \frac{P(\theta_1, \theta_2, \dots, \theta_n | c_i) P(c_i)}{P(\theta_1, \theta_2, \dots, \theta_n)}$$

$$P(c_i | \theta_1, \theta_2, \dots, \theta_n) = \frac{[P(\theta_1 | c_i) P(\theta_2 | c_i) P(\theta_3 | c_i) \dots P(\theta_n | c_i)] P(c_i)}{P(\theta_1, \theta_2, \dots, \theta_n)}$$

$$P(c_i | \theta_1, \theta_2, \dots, \theta_n) = \frac{P(c_i) [\prod_{m=1}^n P(\theta_m | c_i)]}{P(\theta_1, \theta_2, \dots, \theta_n)}$$

$$class = \underset{c_i}{\operatorname{argmax}} \frac{P(c_i) [\prod_{m=1}^n P(\theta_m | c_i)]}{P(\theta_1, \theta_2, \dots, \theta_n)}$$



$$class = \underset{c_i}{\operatorname{argmax}} P(c_i) [\prod_{m=1}^n P(\theta_m | c_i)]$$

# Тонкости

В зависимости от данных и задачи, необходимо моделировать условные вероятности разными способами, поэтому существуют [различные реализации наивного Байесовского классификатора в sklearn](#)

Colab? Colab!



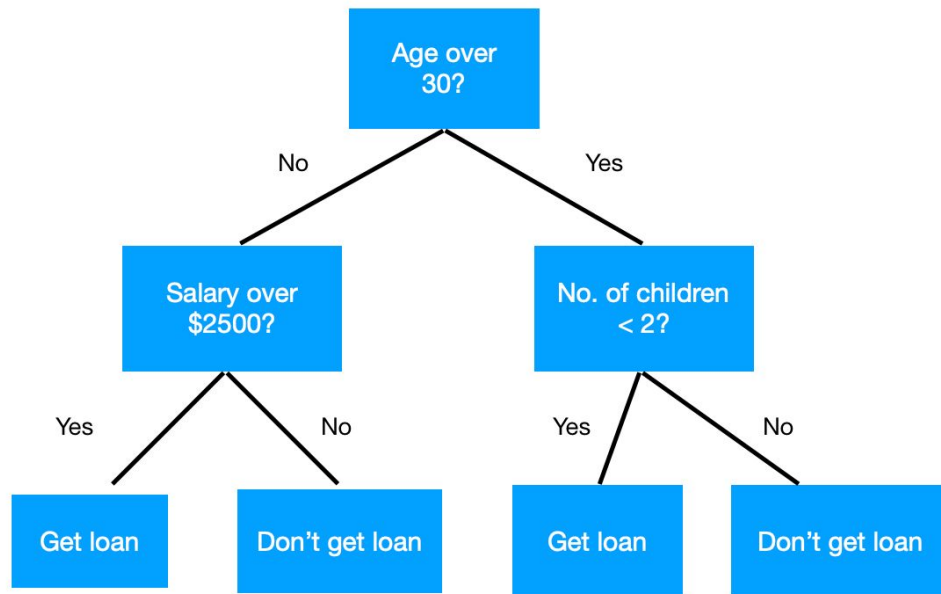
Пример: Decision tree

# Дерево решений

Подробнее об устройстве алгоритма можно почитать по ссылкам

1. <https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac>
2. <http://datareview.info/article/derevya-resheniy-i-algoritmyi-ih-postroeniya/>
3. <https://habr.com/ru/company/ods/blog/322534/>

# Дерево решений



# Дерево решений. Алгоритм CART в паре слов

1. Правила, основанные на значениях переменных, выбираются для получения наилучшего разделения для дифференциации наблюдений на основе зависимой переменной.
2. После того, как правило выбрано и разбивает узел на два, один и тот же процесс применяется к каждому «дочернему» узлу (т.е. это рекурсивная процедура).
3. Разделение останавливается, когда CART обнаруживает, что дальнейшее усиление невозможно, или выполняются некоторые предварительно установленные правила остановки. (В качестве альтернативы данные максимально разделяются, а затем дерево позже обрезается.)

# Что значит наилучшее разбиение?

Используется специальный информационный критерий. Например, [gini](#) или [информационная энтропия](#).

Для набора  $[1,0,1,1,0,1,0,0]$  (представим что 1 и 0 это метки классов)

1.  $[1,0,1,0] \mid [0,0,1,1]$  наилучшее разбиение по признаку A
2.  $[1,1,1,0] \mid [0,0,0,1]$  по признаку B
3.  $[1,1,1,1] \mid [0,0,0,0]$  по признаку C

Какое из разбиений будет наилучшим?

# Что значит наилучшее разбиение?

Используется специальный информационный критерий. Например, [gini](#) или [информационная энтропия](#).

Для набора [1,0,1,1,0,1,0,0] (представим что 1 и 0 это метки классов)

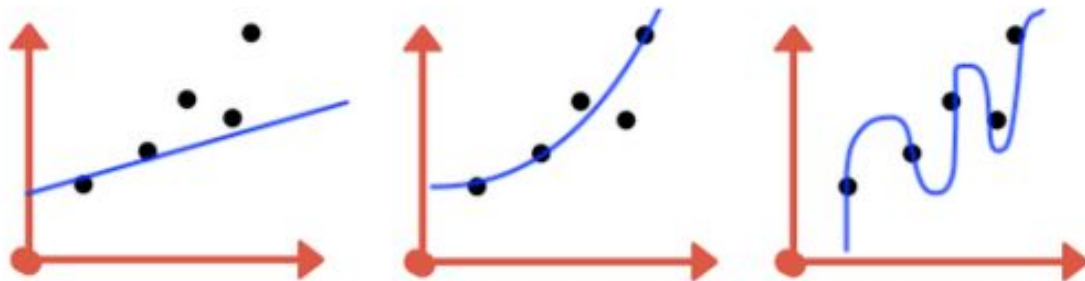
1. [1,0,1,0] | [0,0,1,1] наилучшее разбиение по признаку A
2. [1,1,1,0] | [0,0,0,1] по признаку B
3. [1,1,1,1] | [0,0,0,0] по признаку C

Наилучшее по признаку C, потому что суммарная информационная энтропия такого разбиения = 0 =  $(1 * \log_2(1) + 0 * \log_2(0)) + (0 * \log_2(0) + 1 * \log_2(1))$ .

Для остальных разбиений, посчитайте информационную энтропию самостоятельно

Colab? Colab!

# Несмещенная оценка



Вопрос: какое предсказание лучше по метрикам, а какое на самом деле?

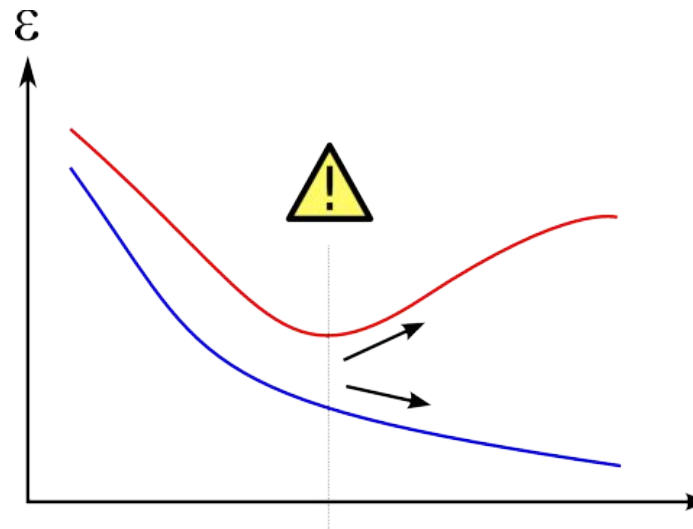
Если тестировать модель на той же выборке, на которой она обучалась, то оценка получится смещенной. В таком случае “самая лучшая” модель - это та, которая просто запомнила все данные.

Хорошая модель должна делать хорошие предсказания на **новых** для себя данных



# Переобучение

- Как обнаружить? - Train/Test split
  - Разделить выборку на обучающую и контрольную
  - Следить за качеством на контрольной выборке
- Минусы?
  - Уменьшение размера обучающей выборки может негативно сказаться на качестве
  - Малый размер тестовой выборки может давать сильное смещение оценки.
  - Можно переобучиться под **тестовую выборку**



# Что делать?

По существу, нам хотелось бы как-то проконтролировать работу нашей модели не только на тестовых данных. Поскольку волшебным образом новые данные не появятся, нужно иначе работать со старыми. Речь идет о механизме **кросс валидации** или кросс проверки.

# Кросс валидация?



## Идея

Давайте разобьем наши данные на несколько обучающих и тестовых выборок! Что нам это даст?

1. Подбор гиперпараметров моделей (например, число соседей в kNN или максимальную глубину дерева в Decision Tree)
2. Страховка от завышенных ожиданий
3. Лучшее представление об устойчивости выбранной модели к разным входным данным
4. Более обоснованный метод сравнения моделей с разными наборами параметров и моделей из разных семейств

# Изменения в общем процессе подготовки модели

## Было

1. Загрузка данных
2. Подготовка данных
3. Разбиение на обучение и тест
4. Обучение модели
5. Сбор метрик на обучении и тесте
6. Принятие решения

## Стало

1. Загрузка данных
2. Подготовка данных
3. Разбиение на обучение и тест
4. Оценка работы модели на кросс валидации
5. Выбор наилучшего набора параметров на основе кросс валидации
6. Обучение модели на всем наборе данных для обучения
7. Сбор метрик на обучении и тесте
8. Принятие решения

# Кросс-валидация

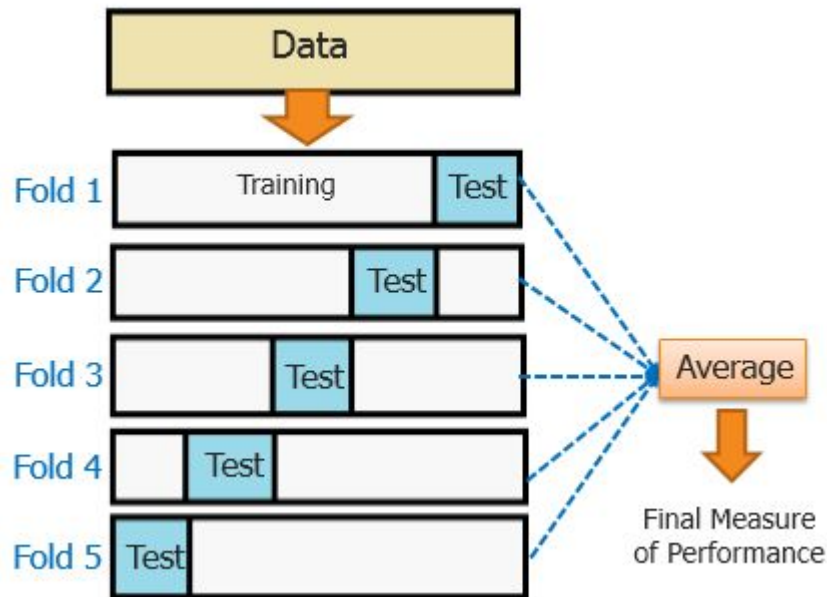
- Разбиваем выборку на  $k$  частей
- $k-1$  частей используются для обучения и одна - для тестирования
- Процесс повторяется  $k$  раз. Каждый раз для тестирования выбирается разная часть
- Результаты тестирования усредняются

Плюсы:

- Погрешность оценки уменьшается, т.к. используется весь набор

Минусы:

- Обучение производится  $k$  раз. Для некоторых моделей это может быть очень долго



# Кросс-Валидация

## Плюсы

- Качество измеряется на всем наборе данных
- Качество не зависит от выбора конкретного тестового набора
- Сложнее переобучиться под тест

## Минусы

- Скорость!

Делаем train test split. Затем

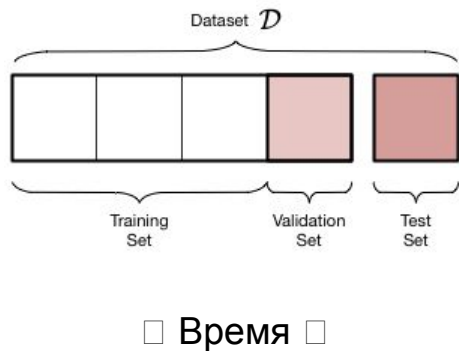
- Мало обучающих данных => Можно попробовать кросс-валидацию, но метрики будут зашумленными
- Много обучающих данных => если есть возможность делать кросс валидацию (по ресурсам), то делаем, если её нет, то дробим выборку на train-validation-test split.

## Не забыть

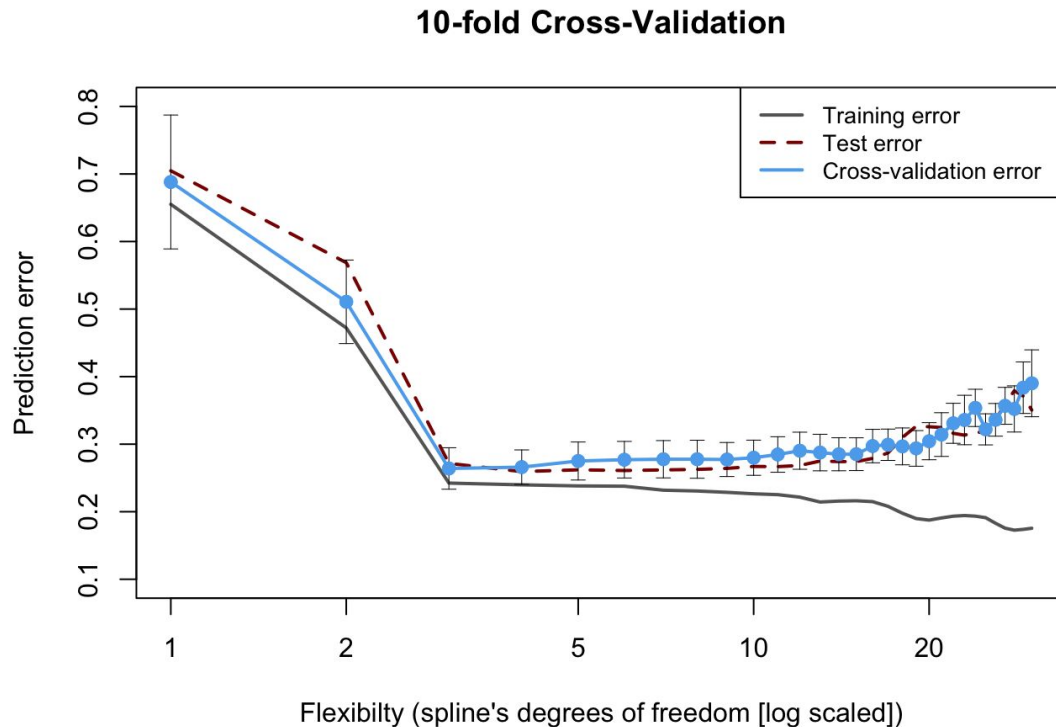
- Отложить Test для замера итогового качества
- Обучить итоговую модель на всех данных

# Кросс-Валидация По Времени

- Используется для анализа временных рядов
  - Тестовый набор выбирается из самых свежих данных. Обучение на более старых
- Полезно в реальных задачах
  - Если в качестве признаков используется множество сигналов, которые могут меняться от времени
  - Есть возможность определить дату наблюдения



# Кросс-Валидация, пример





Colab? Colab!

# Резюме

- Познакомились с задачей классификации
- Научились делать синтетические наборы данных для тренировок
- Познакомились с тремя типами моделей для задачи классификации
- Узнали про общий процесс подготовки моделей машинного обучения

Спасибо за  
внимание!