

# Data Mining in Action

Лекция 1

Примеры и основные понятия



# AI & Data Science

## **Искусственный Интеллект (Artificial Intelligence)**

Наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ

## **Наука о данных (Data Science)**

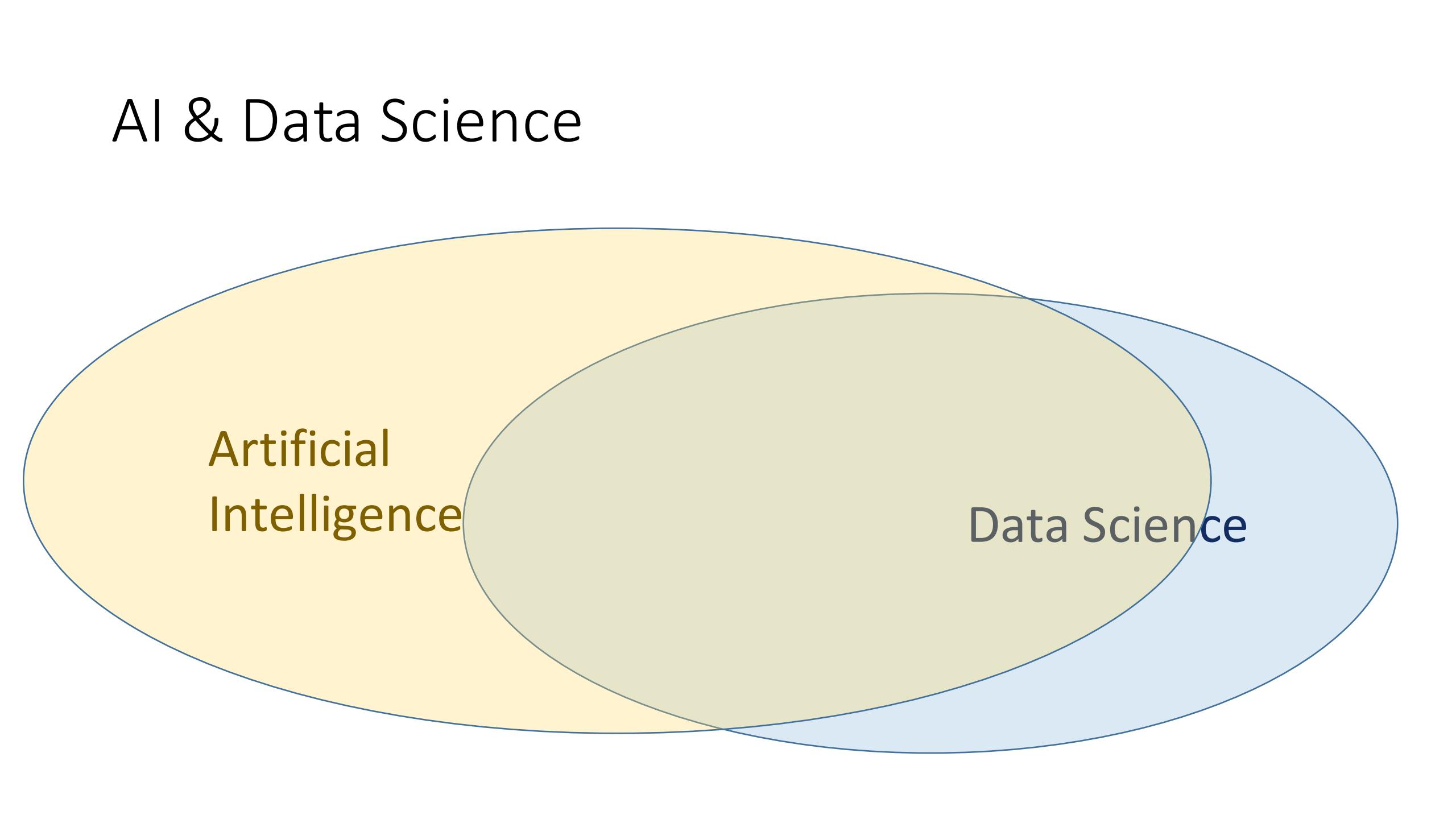
Раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме

# AI & Data Science



Artificial  
Intelligence

# AI & Data Science

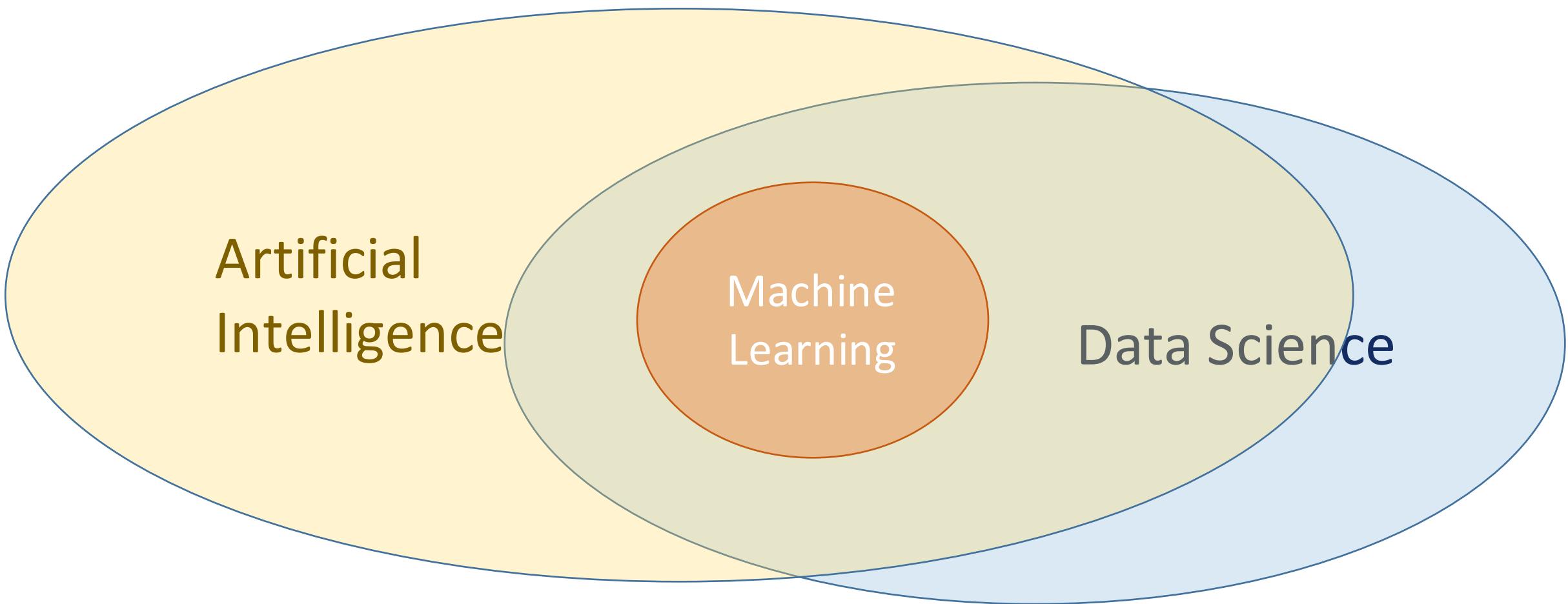


A Venn diagram illustrating the relationship between Artificial Intelligence and Data Science. It consists of two overlapping circles. The left circle is yellow and contains the text "Artificial Intelligence". The right circle is light blue and contains the text "Data Science". The overlapping area is shaded in a darker grey.

Artificial  
Intelligence

Data Science

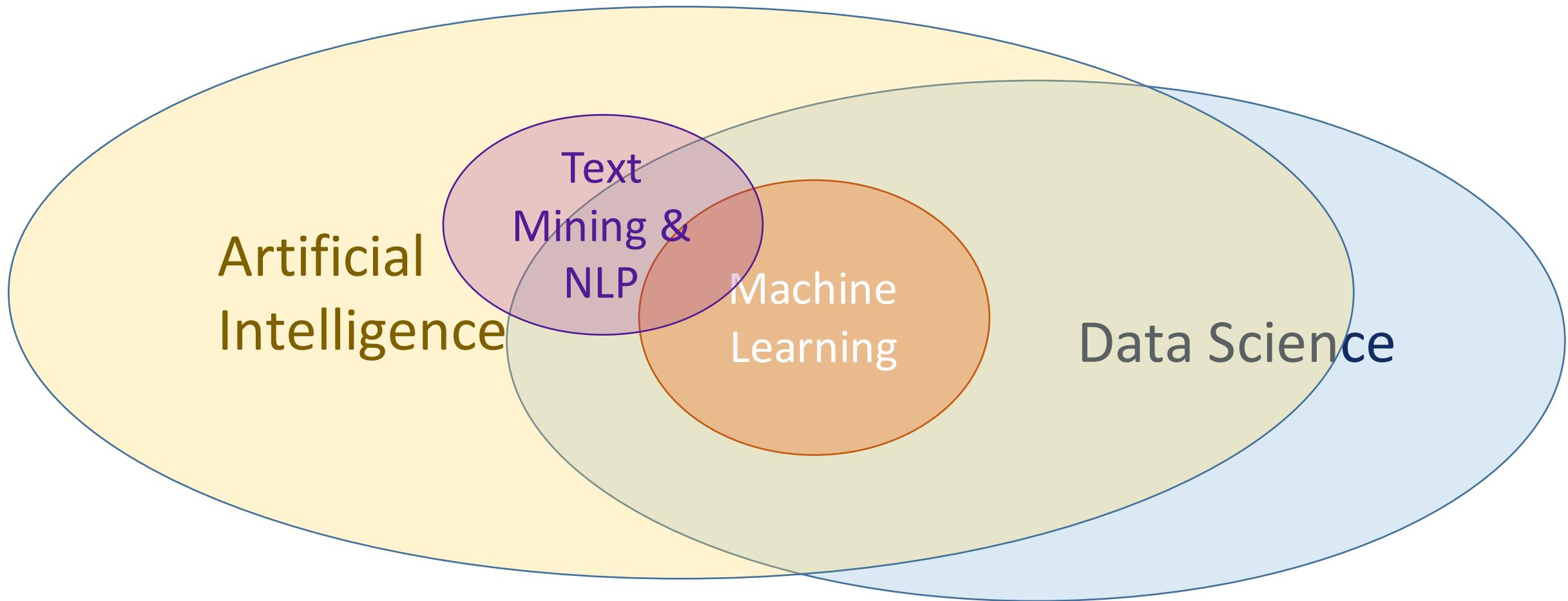
# Machine learning



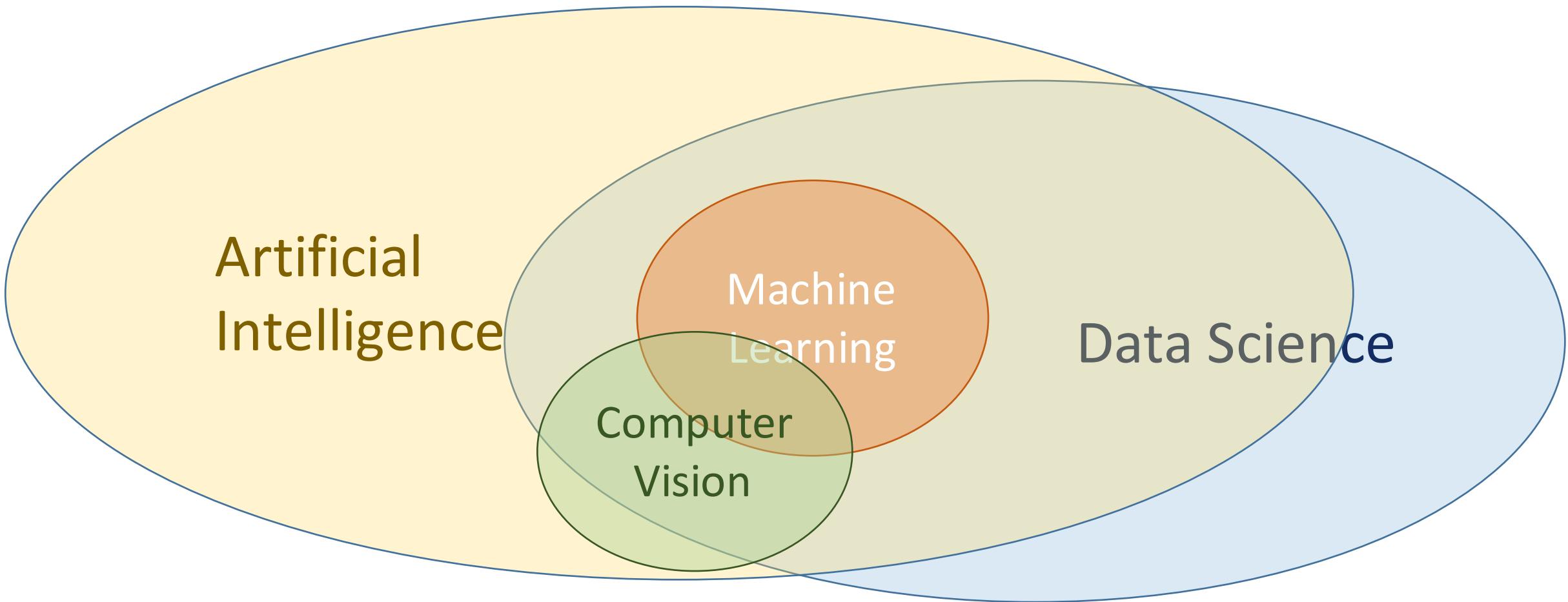
# Machine learning

**Машинное обучение** - класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе решения множества сходных задач.

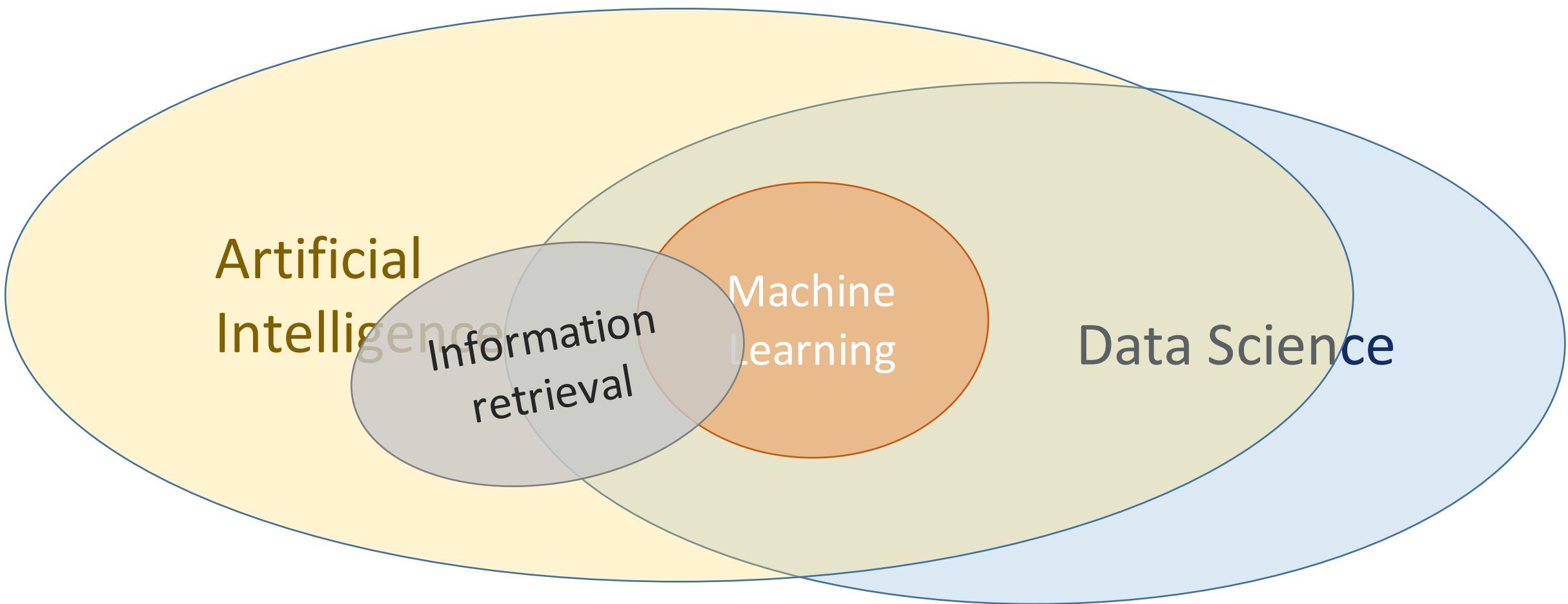
# Приложения



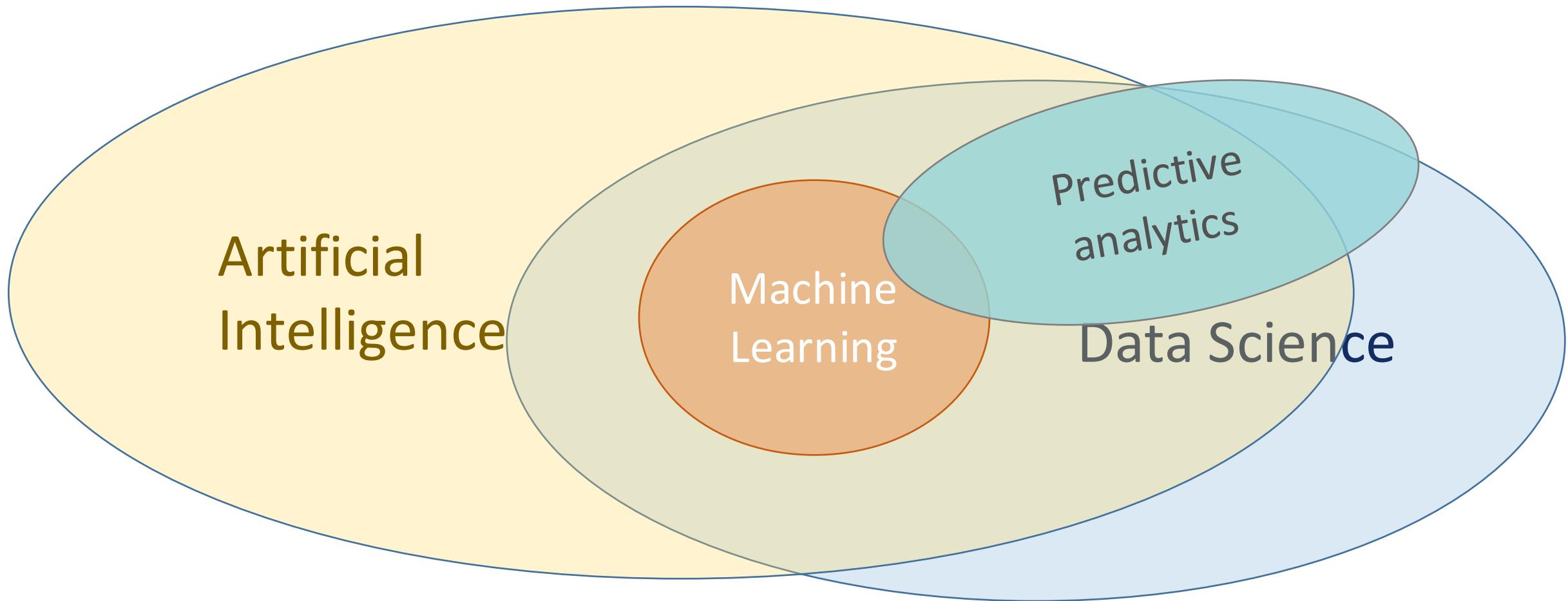
# Приложения



# Приложения



# Приложения



# МЛ ближе, чем кажется



# Курсы ML

Какими бывают курсы машинного обучения?

## «Академический курс»

- Эмпирический риск на тестовых данных (hold-out):

$$\text{HO}(\mu, X^\ell, X^k) = Q(\mu(X^\ell), X^k) \rightarrow \min$$

- Скользящий контроль (leave-one-out),  $L = \ell + 1$ :

$$\text{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^L \mathcal{L}(\mu(X^L \setminus \{x_i\}), x_i) \rightarrow \min$$

- Кросс-проверка (cross-validation),  $L = \ell + k$ ,  $X^L = X_n^\ell \sqcup X_n^k$ :

$$\text{CV}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q(\mu(X_n^\ell), X_n^k) \rightarrow \min$$

# «Практический курс»

```
print('Loading train, prop and sample data')
train = pd.read_csv("../input/train_2016_v2.csv", parse_dates=["transactiondate"])
prop = pd.read_csv('../input/properties_2016.csv')
sample = pd.read_csv('../input/sample_submission.csv')

print('Fitting Label Encoder on properties')
for c in prop.columns:
    prop[c]=prop[c].fillna(-1)
    if prop[c].dtype == 'object':
        lbl = LabelEncoder()
        lbl.fit(list(prop[c].values))
        prop[c] = lbl.transform(list(prop[c].values))

#Create df_train and x_train y_train from that
print('Creating training set:')
df_train = train.merge(prop, how='left', on='parcelid')
```

# «Бигдата за три месяца»



*dmlc*  
**mxnet**

*dmlc*  
**XGBoost**

*dmlc*  
**Spark**

<https://pixelastic.github.io/pokemonorbigdata/>

# Что мы будем обсуждать на лекциях

- 1) Какие задачи решает ML и базовая терминология
- 2) Как работают методы
- 3) Оценка качества алгоритмов
- 4) Представления об особенностях индустриального анализа данных, deep learning и соревнований по анализу данных

# Что ожидать от семинаров

- 1) Рассказы об особенностях выбранного вами направления (индустриального анализа данных, deep learning или соревнований по анализу данных)
- 2) Туториалы о том, как сделать что-то самим на Python
- 3) Практические задания, которые нужно делать самостоятельно на занятиях и дома

# Форматы взаимодействия в курсе

- Лекционное изложение
- Истории из практики
- Демонстрация с кодом
- Обсуждения и мозговые штурмы
- Разбор кейса с преподавателем
- Разбор кейса в командах
- Написание кода на семинаре и в домашнем задании
- Рассказ о своем решении на семинаре
- ...

# На этой лекции

- I. Примеры применения машинного обучения
- II. Задачи и методы
  - Стандартные задачи и простые алгоритмы
  - Наиболее популярные методы
  - Оптимизационные задачи и их решение при обучении
  - Признаки
  - Переобучение
- III. Инструменты

# Часть I: примеры применения

Пример задачи: кредитный скоринг

# Выдача кредита

German credit data set (UCI репозиторий)

Обучающая выборка

1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	1	0	0	1	1		
2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	0	1	0	0	1	0	0	1	2	
4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	0	1	0	0	1	0	0	1	1	
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	0	1	1	
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	0	1	0	0	0	0	0	1	2	
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	0	1	0	0	0	0	0	1	1	
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	0	1	0	0	0	1	0	1	1	
2	36	2	69	1	3	3	2	3	35	3	1	1	2	1	0	0	1	1	0	0	1	0	0	0	1	
4	12	2	31	4	4	1	4	1	61	3	1	1	1	1	0	0	0	1	0	0	0	1	0	1	1	
2	30	4	52	1	1	4	2	3	28	3	2	1	1	1	1	0	0	1	0	0	0	1	0	0	2	
2	12	2	13	1	2	2	1	3	25	3	1	1	1	1	1	0	0	1	0	0	1	0	0	0	1	2
1	48	2	43	1	2	2	4	2	24	3	1	1	1	1	0	0	0	1	0	0	0	1	0	0	1	2
2	12	2	16	1	3	2	1	3	22	3	1	1	2	1	0	0	0	1	0	0	0	1	0	0	1	1
1	24	4	12	1	5	3	4	3	60	3	2	1	1	1	1	0	0	1	0	0	0	1	0	1	2	
1	15	2	14	1	3	2	4	3	28	3	1	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1
1	24	2	13	2	3	2	2	3	32	3	1	1	1	1	1	0	0	0	1	0	0	0	1	0	2	
4	24	4	24	5	5	3	4	2	53	3	2	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1
1	30	0	81	5	2	3	3	3	25	1	3	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1
2	24	2	126	1	5	2	2	4	44	3	1	1	2	1	0	0	1	1	0	0	0	0	0	0	0	2
4	24	2	34	3	5	3	2	3	31	3	1	2	2	1	0	0	0	1	0	0	0	1	0	0	1	1
4	9	4	21	1	3	3	4	3	48	3	3	1	2	1	1	0	0	1	0	0	0	1	0	0	1	1
1	6	2	26	3	3	3	3	1	44	3	1	2	1	1	0	0	0	1	0	0	0	1	0	0	1	1
1	10	4	22	1	2	3	3	1	48	3	2	2	1	2	1	0	0	1	0	0	0	1	0	0	1	1
2	12	4	18	2	2	3	4	2	44	3	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1
4	10	4	21	5	3	4	1	3	26	3	2	1	1	2	0	0	0	1	0	0	0	1	0	0	1	1
1	6	2	14	1	3	3	2	1	36	1	1	1	2	1	0	0	0	1	0	0	0	1	0	0	1	1
4	6	0	4	1	5	4	4	3	39	3	1	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1
3	12	1	4	4	3	2	3	1	42	3	2	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1
2	7	2	24	1	3	3	2	1	34	3	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	1

# Выдача кредита

German credit data set (UCI репозиторий)



1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	1	0	0	1	1	
2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	0	1	0	0	1	0	0	1	2
4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	0	1	0	0	1	0	1	0	1
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	0	1	1
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	0	1	0	0	0	0	0	1	2
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	0	1	0	0	0	0	1	0	1
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	0	1	0	0	0	0	1	1	1
2	36	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	0	1	0	0	0	0	0	1	1
4	12	2	30	2	12	1	48	2	12	1	24	1	15	1	24	4	24	1	30	2	24	4	24	4	9
1	6	2	26	3	3	3	3	1	44	3	1	2	1	1	0	0	1	0	0	1	0	0	0	1	1
1	10	4	22	1	2	3	3	1	48	3	2	2	1	2	1	0	1	0	1	0	0	1	0	1	1
2	12	4	18	2	2	3	4	2	44	3	1	1	1	1	0	1	1	0	0	1	0	0	1	1	1
4	10	4	21	5	3	4	1	3	26	3	2	1	1	2	0	0	1	0	0	1	0	0	1	1	1
1	6	2	14	1	3	3	2	1	36	1	1	1	2	1	0	0	1	0	0	1	0	1	0	1	1
4	6	0	4	1	5	4	4	3	39	3	1	1	1	1	0	0	1	0	0	1	0	1	0	1	1
3	12	1	4	4	3	2	3	1	42	3	2	1	1	1	0	0	1	0	1	0	0	0	1	1	1
2	7	2	24	1	3	3	2	1	34	3	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1

# Выдача кредита

German credit data set (UCI репозиторий)



Attribute 2: Duration in month																									
1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	1	0	0	1	1	
2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	1	0	0	1	0	0	1	2	
4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	1	0	0	1	0	1	0	1	
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	1	1	
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	1	0	0	0	0	0	1	2	
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	1	0	0	0	0	0	1	1	
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	1	0	0	1	0	0	1	1	
2	36	2	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	1	
4	12	2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	
2	30	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	2	
2	12	2	13	1	2	2	1	3	25	3	1	1	1	1	1	1	0	1	0	1	0	0	0	1	2
1	48	2	43	1	2	2	4	2	24	3	1	1	1	1	1	0	0	1	0	1	0	0	0	1	2
2	12	2	16	1	3	2	1	3	22	3	1	1	2	1	0	0	1	0	0	1	0	0	0	1	1
1	24	4	12	1	5	3	4	3	60	3	2	1	1	1	1	0	1	0	0	1	0	0	1	2	
1	15	2	14	1	3	2	4	3	28	3	1	1	1	1	1	0	1	0	1	0	0	0	1	1	
1	24	2	13	2	3	2	2	3	32	3	1	1	1	1	1	0	0	1	0	0	1	0	0	2	
4	24	4	24	5	5	3	4	2	53	3	2	1	1	1	0	0	1	0	0	1	0	0	1	1	
1	30	0	81	5	2	3	3	3	25	1	3	1	1	1	1	0	0	1	0	0	1	0	0	1	1
2	24	2	126	1	5	2	2	4	44	3	1	1	2	1	0	0	1	1	0	0	0	0	0	0	2
4	24	2	34	3	5	3	2	3	31	3	1	2	2	1	0	0	1	0	0	0	1	0	0	1	1
4	9	4	21	1	3	3	4	3	48	3	3	1	2	1	1	0	1	0	0	1	0	0	0	1	1
1	6	2	26	3	3	3	3	1	44	3	1	2	1	1	0	0	1	0	0	1	0	0	0	1	1
1	10	4	22	1	2	3	3	1	48	3	2	2	1	2	1	0	1	1	0	0	1	0	0	1	1
2	12	4	18	2	2	3	4	2	44	3	1	1	1	1	0	0	1	1	0	0	1	0	0	1	1
4	10	4	21	5	3	4	1	3	26	3	2	1	1	2	0	0	1	0	0	0	1	0	0	1	1
1	6	2	14	1	3	3	2	1	36	1	1	1	2	1	0	0	1	0	0	0	1	0	0	1	1
4	6	0	4	1	5	4	4	3	39	3	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1
3	12	1	4	4	3	2	3	1	42	3	2	1	1	1	0	0	1	0	0	1	0	0	0	1	1
2	7	2	24	1	3	3	2	1	34	3	1	1	1	1	0	0	0	0	0	0	1	0	0	1	1

# Выдача кредита

German credit data set (UCI репозиторий)

1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	0	1	0	0	1	1
2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	1	0	0	0	1	0	0	1	2
4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	1	0	0	0	1	0	1	0	1
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	0	1	1
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	1	0	0	0	0	0	0	1	2
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	1	0	0	0	0	0	1	0	1
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1
2	36	2	69	1	3	3	2	3	35	3	1	1	2	1	0	1	1	0	1	0	0	0	0	0	1
4	12	2	31	4	4	1	4	1	61	3	1	1	1	1	0	0	1	0	0	0	1	0	1	0	1
2	30	4	52	1	1	4	2	3	28	3	2	1	1	1	1	0	1	0	0	0	1	0	0	0	2
2	12	2	13	1	2	2	1	3	25	3	1	1	1	1	1	0	1	0	1	0	0	0	1	2	
1	48	2	43	1	2	2	4	2	24	3	1	1	1	1	0	0	1	0	0	1	0	0	0	1	2
2	12	2	16	1	3	2	1	3	22	3	1	1	2	1	0	0	1	0	0	0	1	0	0	1	1
1	24	4	12	1	5	3	4	3	60	3	2	1	1	1	1	0	1	0	0	0	1	0	1	0	2
1	15	2	14	1	3	2	4	3	28	3	1	1	1	1	1	0	1	0	0	0	1	0	0	1	1
1	24	2	13	2	3	2	2	3	32	3	1	1	1	1	1	0	0	1	0	0	0	1	0	2	
4	24	4	24	5	5	3	4	2	53	3	2	1	1	1	1	0	1	1	0	0	0	0	0	1	1
1	30	0	81	5	2	3	3	3	25	1	3	1	1	1	0	0	1	0	0	0	1	0	0	1	1
2	24	2	126	1	5	2	2	4	44	3	1	1	2	1	0	1	1	0	0	0	0	0	0	0	2
4	24	2	34	3	5	3	2	3	31	3	1	2	2	1	0	0	1	0	0	0	1	0	0	0	1
4	9	4	21	1	3	3	4	3	48	3	3	1	2	1	1	0	1	0	0	0	1	0	0	1	1
1	6	2	26	3	3	3	3	1	44	3	1	2	1	1	0	0	1	0	0	1	0	0	0	1	1
1	10	4	22	1	2	3	3	1	48	3	2	2	1	2	1	0	1	0	1	0	0	0	1	0	1
2	12	4	18	2	2	3	4	2	44	3	1	1	1	1	0	1	1	0	0	0	1	0	0	1	1
4	10	4	21	5	3	4	1	3	26	3	2	1	1	2	0	0	1	0	0	0	1	0	0	1	1
1	6	2	14	1	3	3	2	1	36	1	1	1	2	1	0	0	1	0	0	0	1	0	1	0	1
4	6	0	4	1	5	4	4	3	39	3	1	1	1	1	0	0	1	0	0	0	1	0	1	0	1
3	12	1	4	4	3	2	3	1	42	3	2	1	1	1	0	0	1	0	0	1	0	0	0	1	1
2	7	2	24	1	3	3	2	1	34	3	1	1	1	1	0	0	0	0	0	0	1	0	0	1	1

Answer: 1 – Good, 2 - Bad

# Выдача кредита

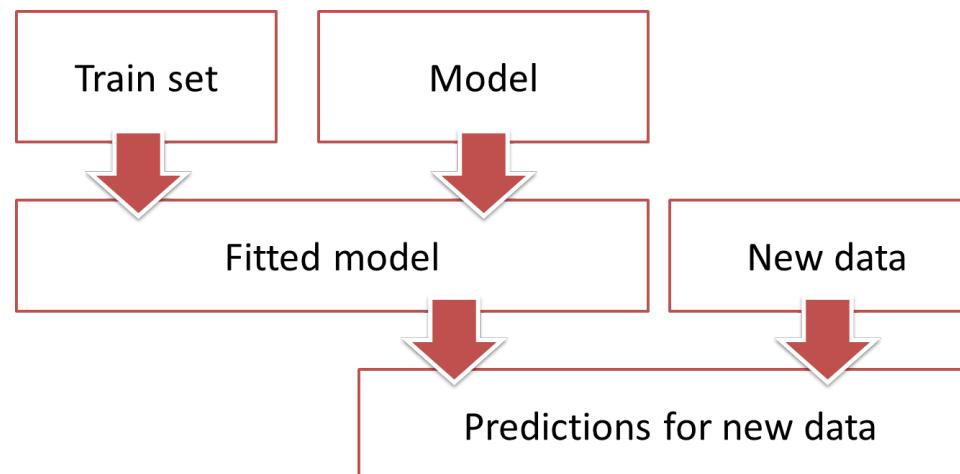
Задача (supervised classification): предсказать класс (1 или 2)

1	60	3	68	1	5	3	4	4	63	3	2	1	2	1	0	0	1	0	0	1	0	0	1	?
2	18	2	19	4	2	4	3	1	36	1	1	1	2	1	0	0	1	0	0	1	0	0	1	?
1	24	2	40	1	3	3	2	3	27	2	1	1	1	1	0	0	1	0	0	1	0	0	1	?
2	18	2	59	2	3	3	2	3	30	3	2	1	2	1	1	0	1	0	0	1	0	0	1	?
4	12	4	13	5	5	3	4	4	57	3	1	1	1	1	0	0	1	0	1	0	0	1	0	?
3	12	2	15	1	2	2	1	2	33	1	1	1	2	1	0	0	1	0	0	1	0	0	0	?
2	45	4	47	1	2	3	2	2	25	3	2	1	1	1	0	0	1	0	0	1	0	1	0	?

Test set

Более глобальная задача:

Придумать алгоритм, генерирующий алгоритм классификации  
("обученную модель") на данной выборке



Пример проекта: рекомендации товаров

# Блок рекомендаций

Товар 1	Товар 2	Товар 3	Товар 4
---------	---------	---------	---------

# Максимизация количества покупок

Товар 1	Товар 2	Товар 3	Товар 4

Вероятность:	$p_1$	$p_2$	$p_3$	$p_4$
--------------	-------	-------	-------	-------

# Максимизация дохода

	Товар 1	Товар 2	Товар 3	Товар 4
Вероятность:	$p_1$	$p_2$	$p_3$	$p_4$
Цена:	$c_1$	$c_2$	$c_3$	$c_4$

# Максимизация дохода



Puma  
Ветровка  
3 490 руб.

Crocs  
Сланцы  
1 990 руб.

Tony-p  
Слипоны  
~~1 999 руб.~~ 1 590 руб.

Champion  
Брюки спортивные  
~~3 599 руб.~~ 1 970 руб.

Вероятность:	0.05	0.02	0.015	0.009
Цена:	3490	1990	1590	1970

# Прогнозирование вероятности

- Объекты: тройки (пользователь, товар, момент времени)
- Классы: 1 - товар будет куплен, 0 – товар не будет куплен
- Признаки: параметры пользователя, товара, момента времени и их «взаимодействие»

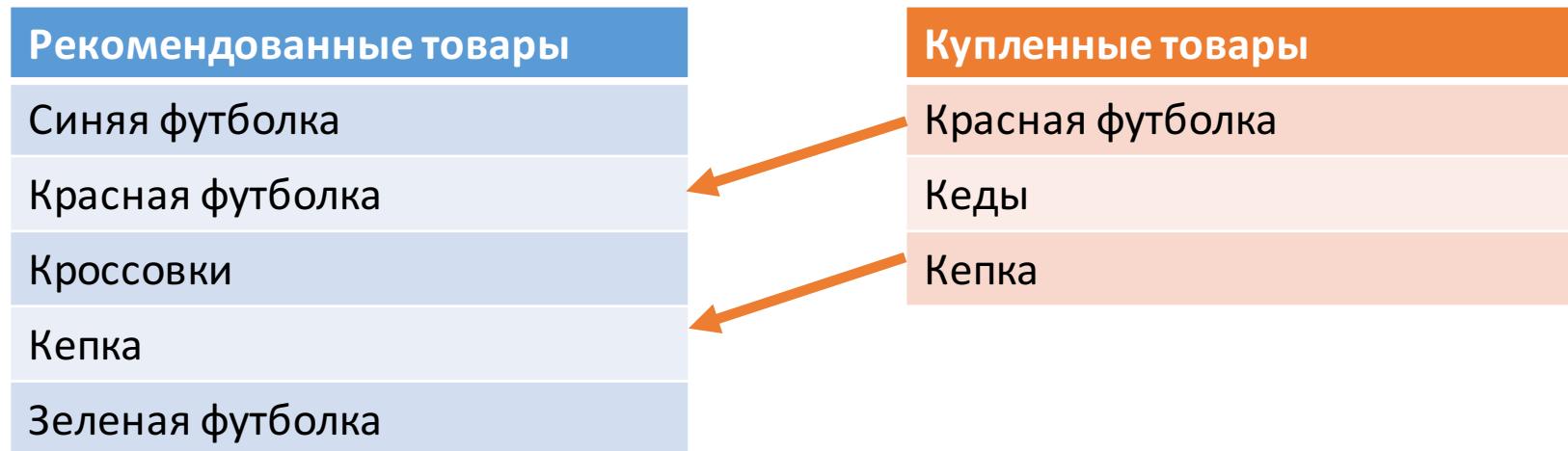
# Отбор кандидатов

- Популярные
- Популярные из тех же категорий
- Часто просматриваемые вместе с теми товарами, которые пользователь уже видел
- Из заранее подготовленных списков похожих товаров

# Генерация негативных примеров

- Добавить к каждому позитивному примеру весь каталог как негативный (не реально)
- Случайные с равномерным распределением
- Случайные, с вероятностями, пропорциональными популярности объекта
- Самые популярные примеры
- Те объекты, которые рекомендовал бы какой-то алгоритм, но они не были куплены

# Точность (Precision@k)

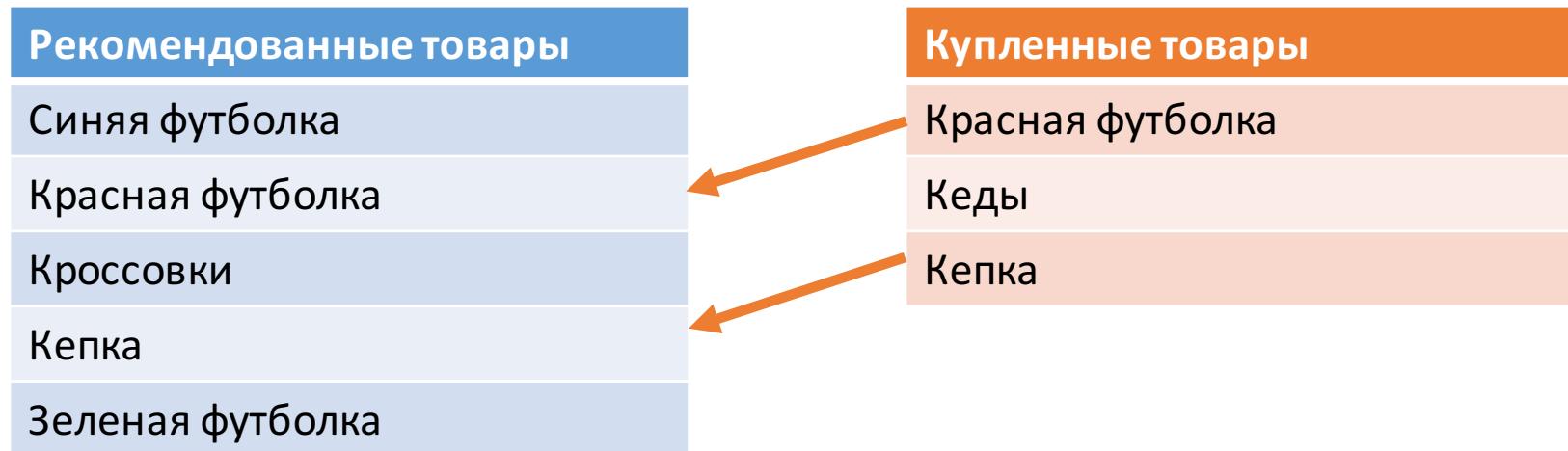


$k$  – количество  
рекомендаций

$$\text{Precision}@k = \frac{\text{купленное из рекомендованного}}{k}$$

AveragePrecision@k - усредненный по сессиям Precision@k

# Полнота (Recall@k)



$k$  – количество  
рекомендаций

$$\text{Recall}@k = \frac{\text{купленное из рекомендованного}}{\text{количество покупок}}$$

AverageRecall@k - усредненный по сессиям Recall@k

# Взвешенный ценами recall@k

Рекомендованные товары	Купленные товары
Синяя футболка – 1000р	Красная футболка – 1200р
Красная футболка – 1200р	Кеды – 3000р
Кроссовки – 3500р	Кепка – 900р
Кепка – 900р	
Зеленая футболка – 800р	

Взвешенный ценами Recall@k =  $\frac{\text{стоимость купленного из рекомендованного}}{\text{стоимость покупок}}$

AverageRecall@k - усредненный по сессиям Recall@k

# Качество классификации против качества рекомендаций

Пример – 2 решения для прогноза купит/не купит товар

	Алгоритм 1	Алгоритм 2
Качество классификации	0.52	0.85
Recall@5	0.72	0.71

# История из практики: сравнение методов

- Интегрировали чужое решение, чтобы сравнить качество со своим
- Оценили качество у обоих
- Совпало до тысячных долей
- Не стали использовать чужое решение
- Позже – выяснили, в чем дело :)

# Онлайновая оценка качества

Допустим, на исторических данных качество алгоритма высокое, а будет ли оно высоким в реальности?

# Онлайновая оценка качества

Допустим, на исторических данных качество алгоритма высокое, а будет ли оно высоким в реальности?

Идеи:

1. А/В тест
2. Оценка статзначимости результата

# A/B тест

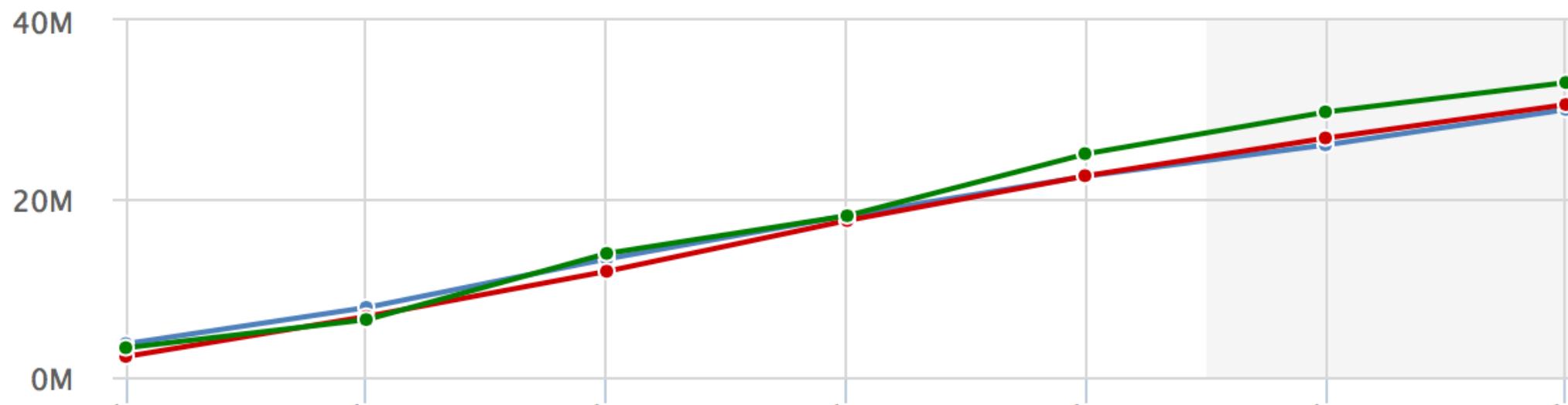
1. Случайным образом делим пользователей на равные группы
2. Измеряем целевые метрики (например, количество заказов или доход) в каждой группе за длительный период времени
3. Получаем какое-то число для каждой группы
4. Что дальше?

# Истории из практики: разбиение на группы

- Предложено:
  - Брать hash от user\_id
  - Смотреть на остаток от деления на 2
- Сделано:
  - Брать hash от user\_id+user\_email
  - Смотреть на остаток от деления на 2

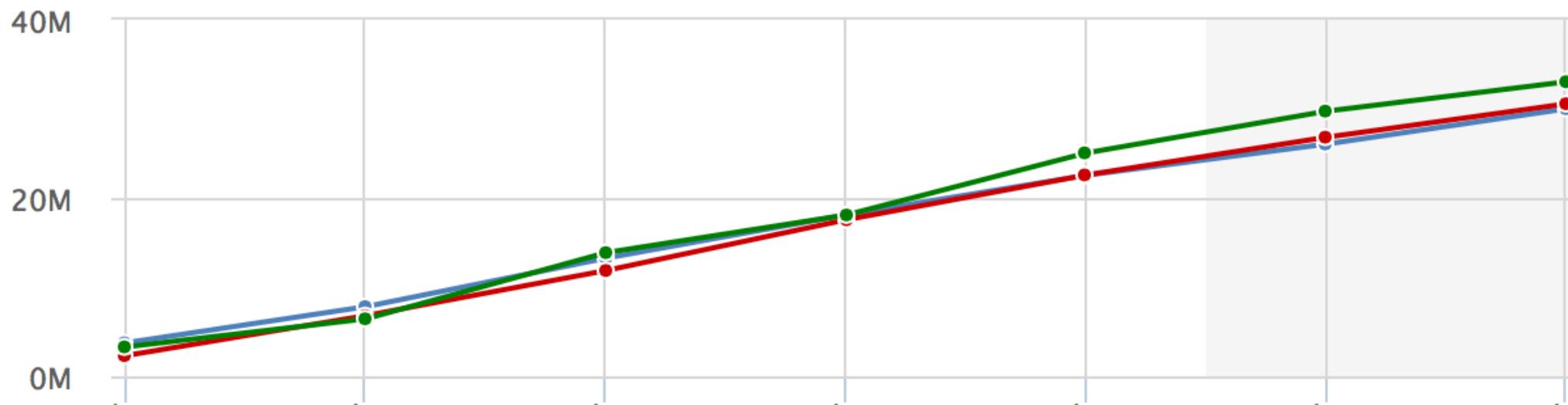
# Статистическая значимость: пример

**Суммарная выручка**



# Статистическая значимость: пример

Суммарная выручка



Одна кривая отличается от других на 10%  
Но разбиение на самом деле – случайное

# На какие метрики смотрят в онлайне

- Доход в группе
- Доход с пользовательской сессии
- Средняя стоимость купленного товара
- Средний чек
- Конверсия в покупку
- Клики
- Различные модели атрибуции: last click, first click

# Итог: о чём нужно позаботиться

- Высокоуровневая постановка задачи - от экономического эффекта
- Оценка возможного экономического эффекта
- Оценка реализуемости проекта
- Оффлайновая оценка качества
- Онлайновая оценка качества
- Решение задачи – декомпозиция на подзадачи, выбор признаков, выбор моделей

Еще примеры

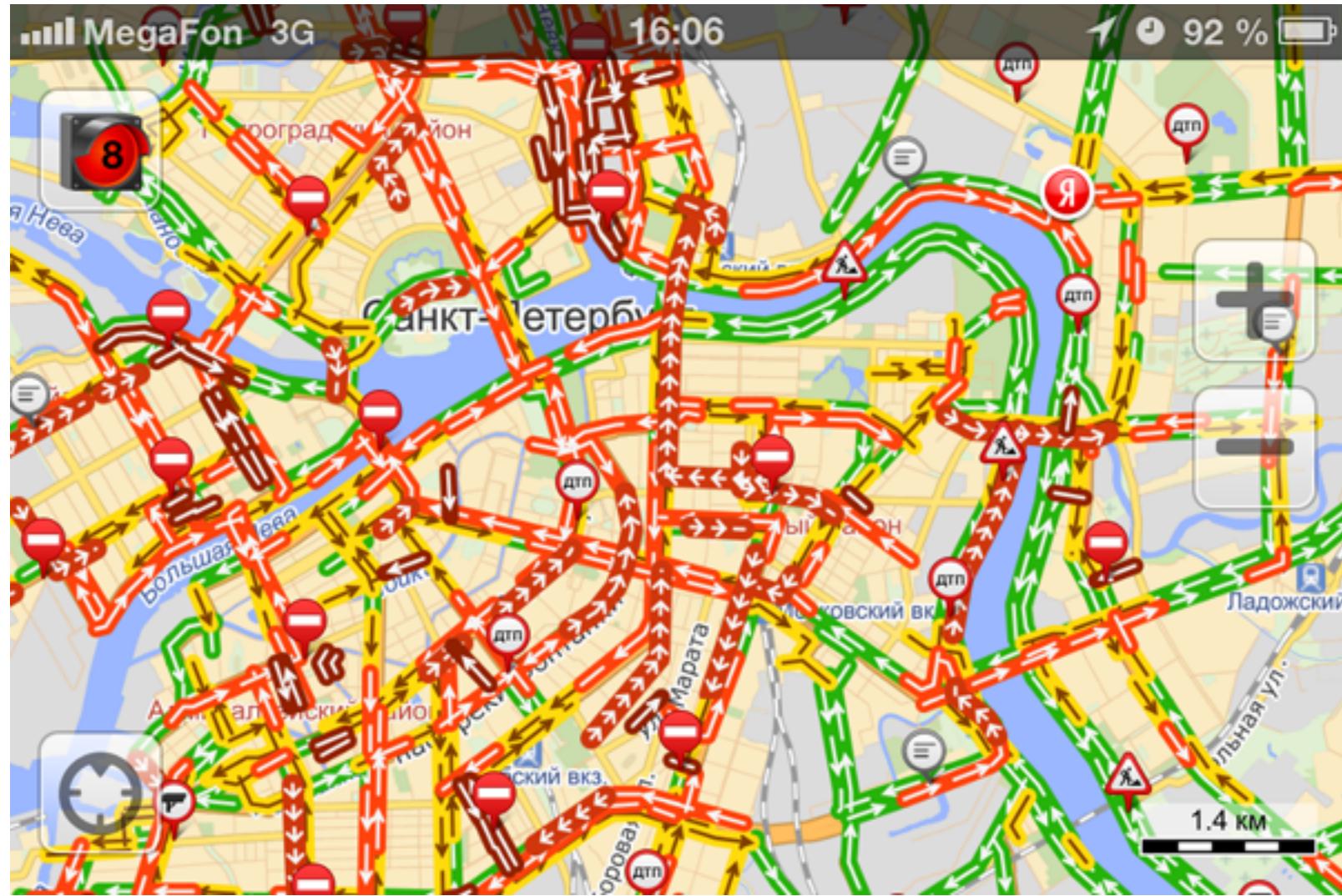
[ПОИСК](#) [КАРТИНКИ](#) [ВИДЕО](#) [КАРТЫ](#) [МАРКЕТ](#) [ДИСК](#) [МУЗЫКА](#) [ЕЩЁ](#)**VK Data Mining in Action | ВКонтакте**[vk.com > data\\_mining\\_in\\_action ▾](#)

Москва, Россия Денис Семененко. Администратор сообщества. Data Mining in Action. So it begins. Местоположение: Москва, Россия. . Data Mining in Action запись закреплена. 6 мая в 23:04.

**Нашлось 8 млн результатов**[Добавить объявление](#) [Показать все](#)**H Process Mining: знакомство / Хабрахабр**[habrahabr.ru > post/244879/ ▾](#)

Статья подготовлена на основе материалов онлайн курса **Process Mining: Data Science in Action**, являющихся собственностью Технического университета Эйндховена.

**Coursera Process Mining: Data science in Action... | Coursera**[coursera.org > learn/process-mining ▾](#)





## Часть II: стандартные задачи и методы

Стандартные постановки задач  
и простые методы их решения

# Классификация



*Iris setosa*



*Iris versicolor*



*Iris virginica*

Вход (обучающая выборка):

Признаки N объектов с известными классами

Выход:

Классификатор (алгоритм, прогнозирующий классы новых объектов по их признакам)

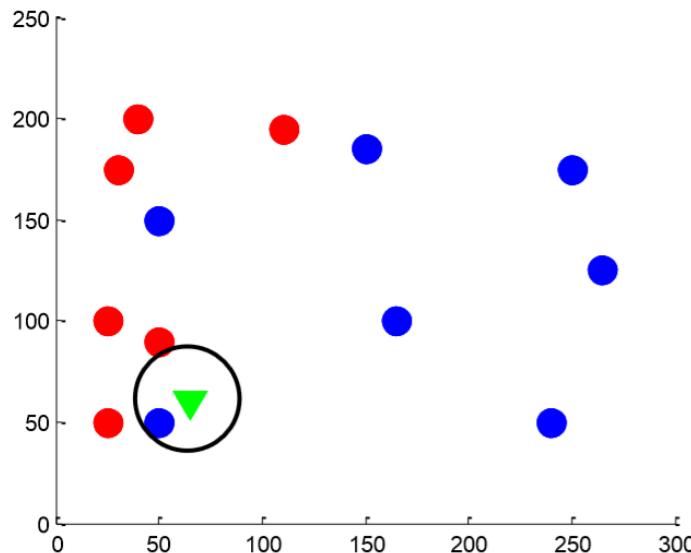
# Классификация: обучающая выборка

Fisher's Iris Data

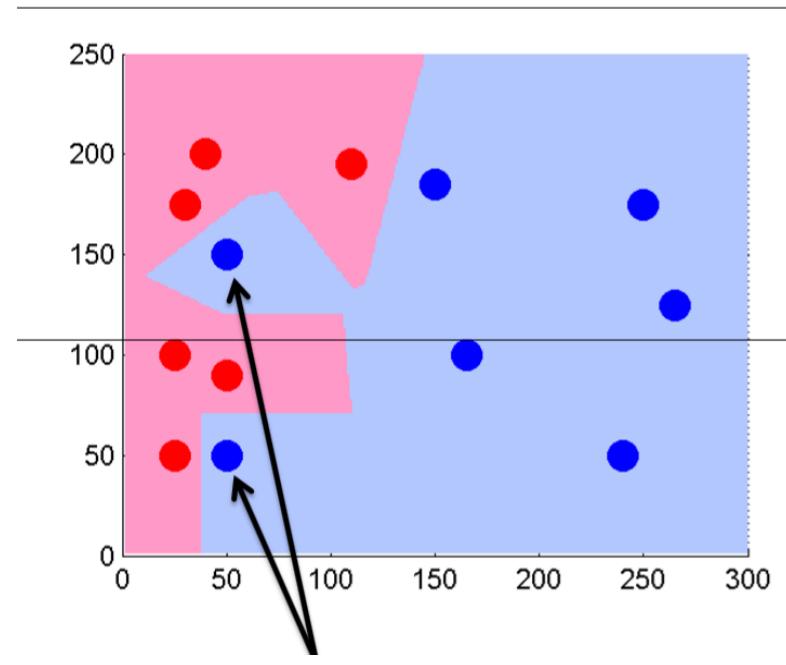
Sepal length	Sepal width	Petal length	Petal width	Species
5.0	2.0	3.5	1.0	<i>I. versicolor</i>
6.0	2.2	5.0	1.5	<i>I. virginica</i>
6.2	2.2	4.5	1.5	<i>I. versicolor</i>
6.0	2.2	4.0	1.0	<i>I. versicolor</i>
6.3	2.3	4.4	1.3	<i>I. versicolor</i>
5.5	2.3	4.0	1.3	<i>I. versicolor</i>
5.0	2.3	3.3	1.0	<i>I. versicolor</i>
4.5	2.3	1.3	0.3	<i>I. setosa</i>
5.5	2.4	3.8	1.1	<i>I. versicolor</i>
5.5	2.4	3.7	1.0	<i>I. versicolor</i>
4.9	2.4	3.3	1.0	<i>I. versicolor</i>
6.7	2.5	5.8	1.8	<i>I. virginica</i>
5.7	2.5	5.0	2.0	<i>I. virginica</i>
6.3	2.5	5.0	1.9	<i>I. virginica</i>
6.3	2.5	4.9	1.5	<i>I. versicolor</i>
4.9	2.5	4.5	1.7	<i>I. virginica</i>

# Простой классификатор: kNN

k nearest neighbours



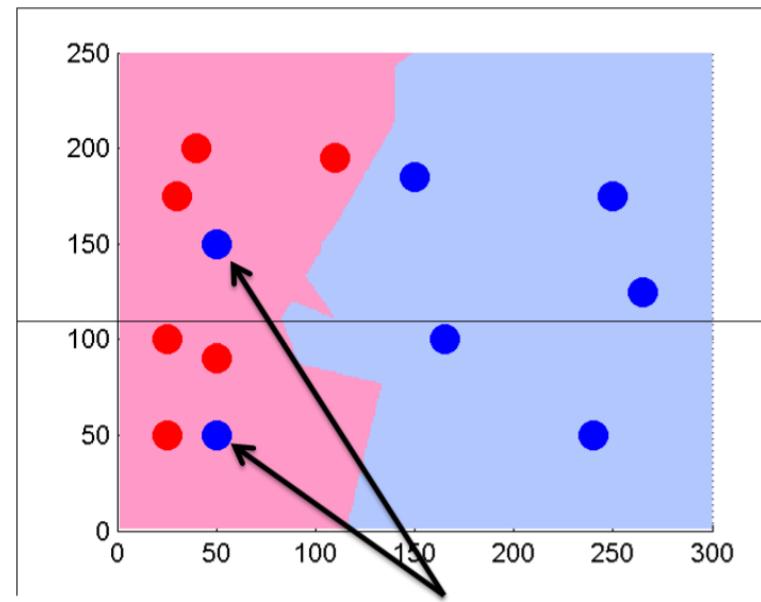
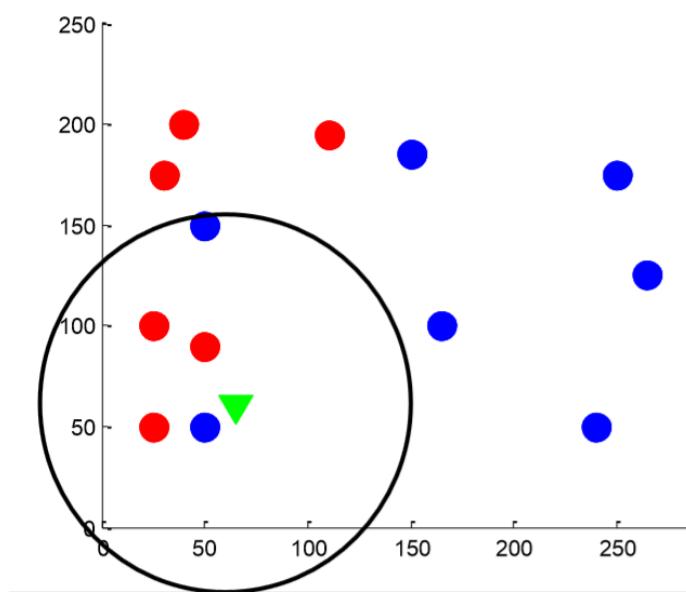
$k = 1$



Шумы? (outliers)

# Простой классификатор: kNN

k nearest neighbours

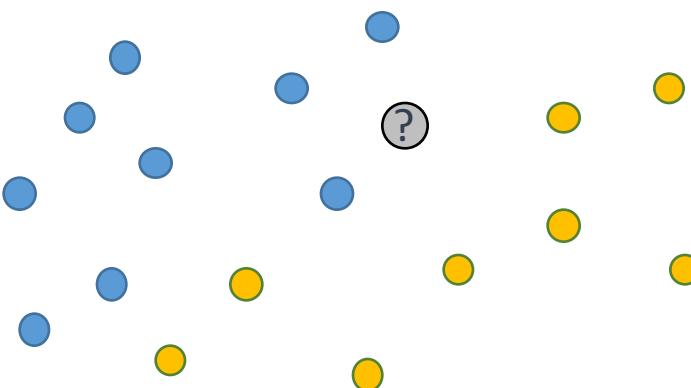


Ошибки?

$$k = 5$$

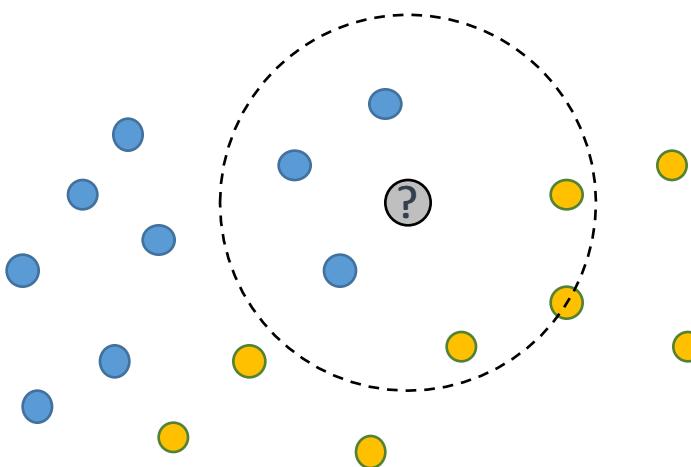
# Взвешенный kNN

Пример классификации ( $k = 6$ ):



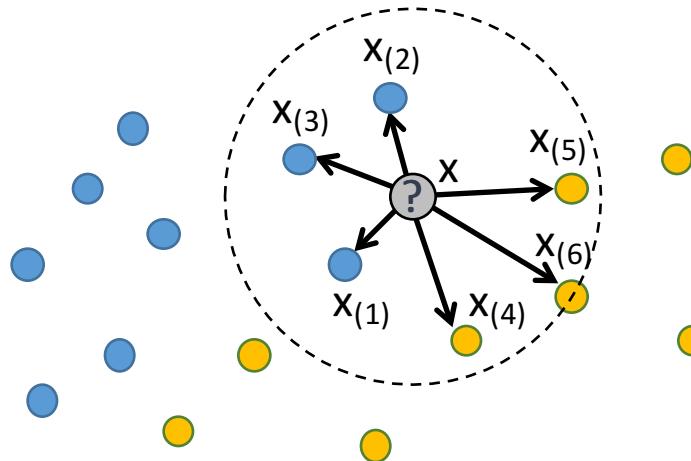
# Взвешенный kNN

Пример классификации ( $k = 6$ ):



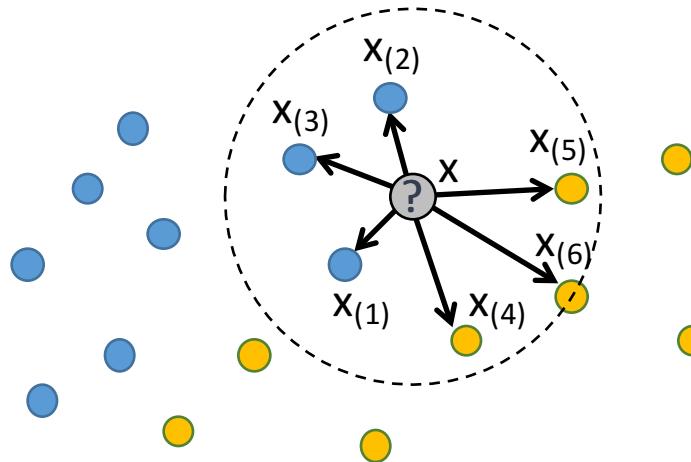
# Взвешенный kNN

Пример классификации ( $k = 6$ ):



# Взвешенный kNN

Пример классификации ( $k = 6$ ):

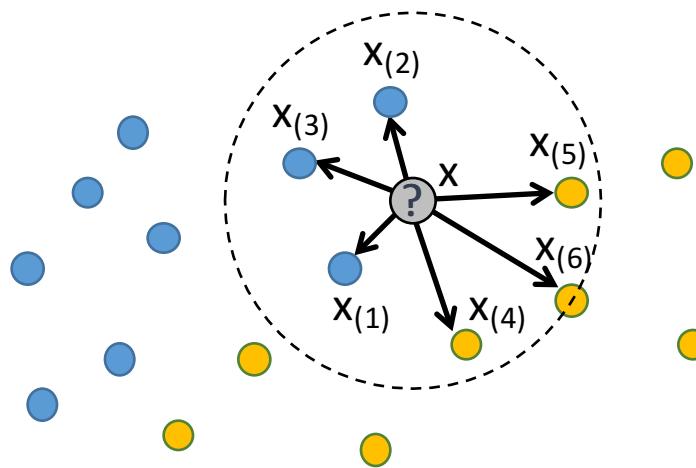


Веса можно определить как функцию от соседа или его номера:

$$w(x_{(i)}) = w(i)$$

# Взвешенный kNN

Пример классификации ( $k = 6$ ):



Веса можно определить как функцию от соседа или его номера:

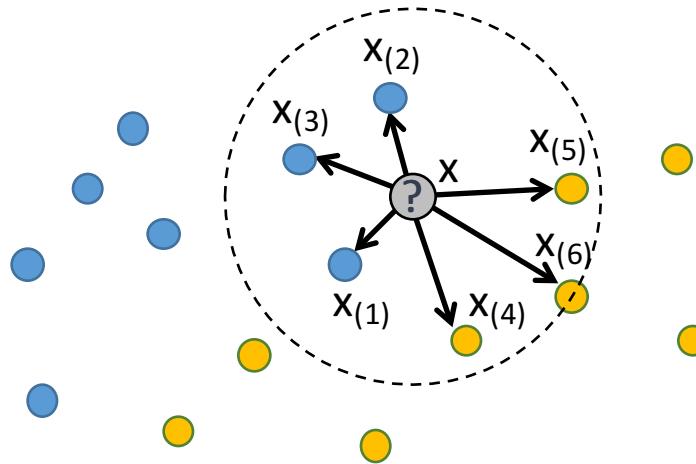
$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

$$w(x(i)) = w(d(x, x_{(i)}))$$

# Взвешенный kNN

Пример классификации ( $k = 6$ ):



$$z_{\text{blue}} = \frac{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

Веса можно определить как функцию от соседа или его номера:

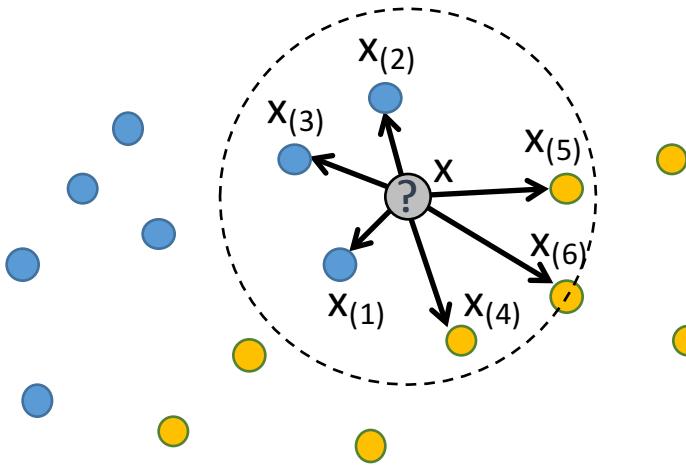
$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

$$w(x(i)) = w(d(x, x_{(i)}))$$

# Взвешенный kNN

Пример классификации ( $k = 6$ ):



Веса можно определить как функцию от соседа или его номера:

$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

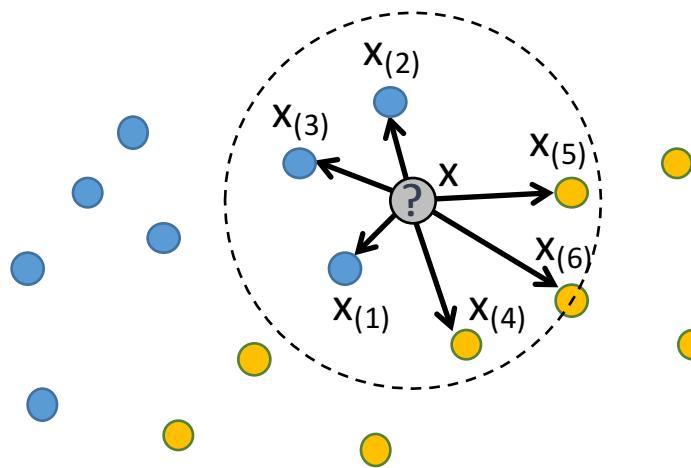
$$w(x(i)) = w(d(x, x_{(i)}))$$

$$z_{\text{blue}} = \frac{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

$$z_{\text{yellow}} = \frac{w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

# Взвешенный kNN

Пример классификации ( $k = 6$ ):



Веса можно определить как функцию от соседа или его номера:

$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

$$w(x(i)) = w(d(x, x_{(i)}))$$

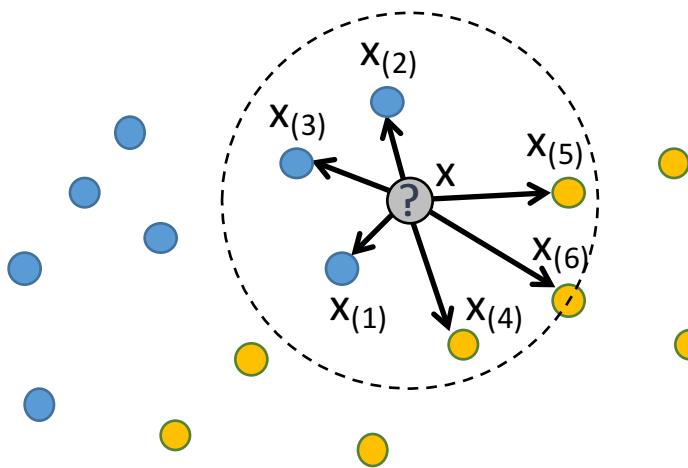
$$Z_{\text{blue}} = \frac{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

$$\text{?} = \operatorname{argmax}_{\text{color}} Z_{\text{color}}$$

$$Z_{\text{yellow}} = \frac{w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

# Взвешенный kNN

Пример классификации ( $k = 6$ ):



$$Z_{\text{blue}} = \frac{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

$$Z_{\text{yellow}} = \frac{w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

Веса можно определить как функцию от соседа или его номера:

$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

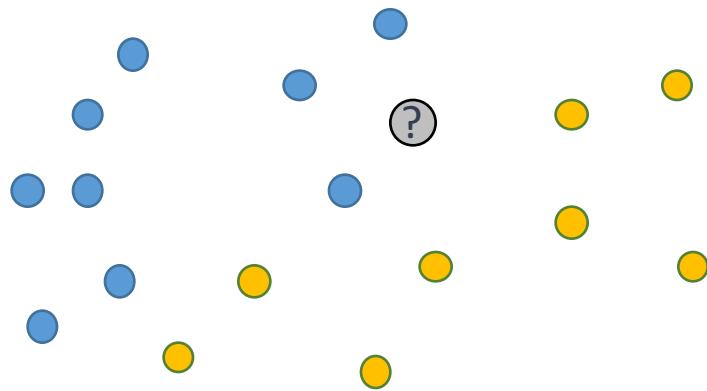
$$w(x(i)) = w(d(x, x_{(i)}))$$

$$\text{?} = \operatorname{argmax}_{\circlearrowleft} Z_{\circlearrowleft}$$

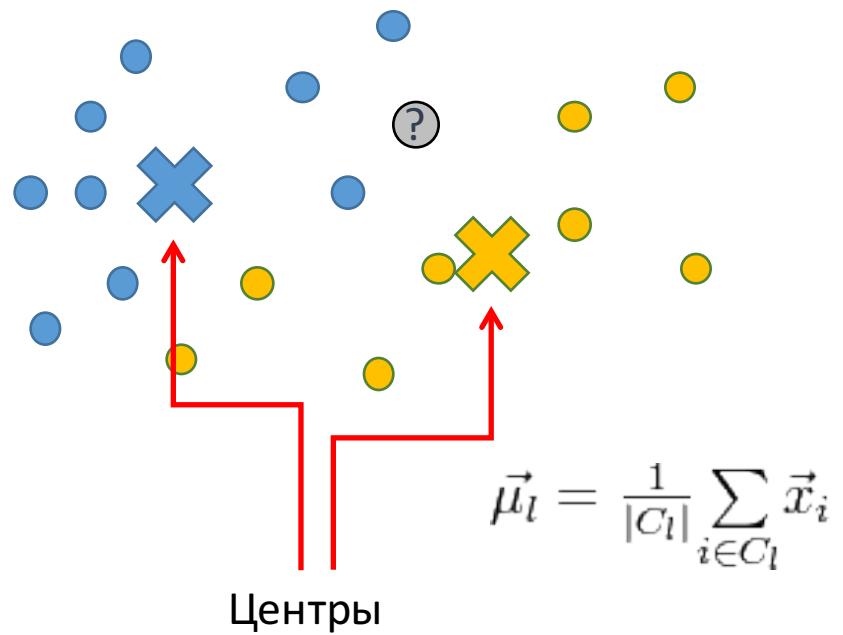
$$\text{if } Z_{\text{yellow}} > Z_{\text{blue}} : \quad \text{?} = \text{yellow}$$

$$\text{if } Z_{\text{yellow}} < Z_{\text{blue}} : \quad \text{?} = \text{blue}$$

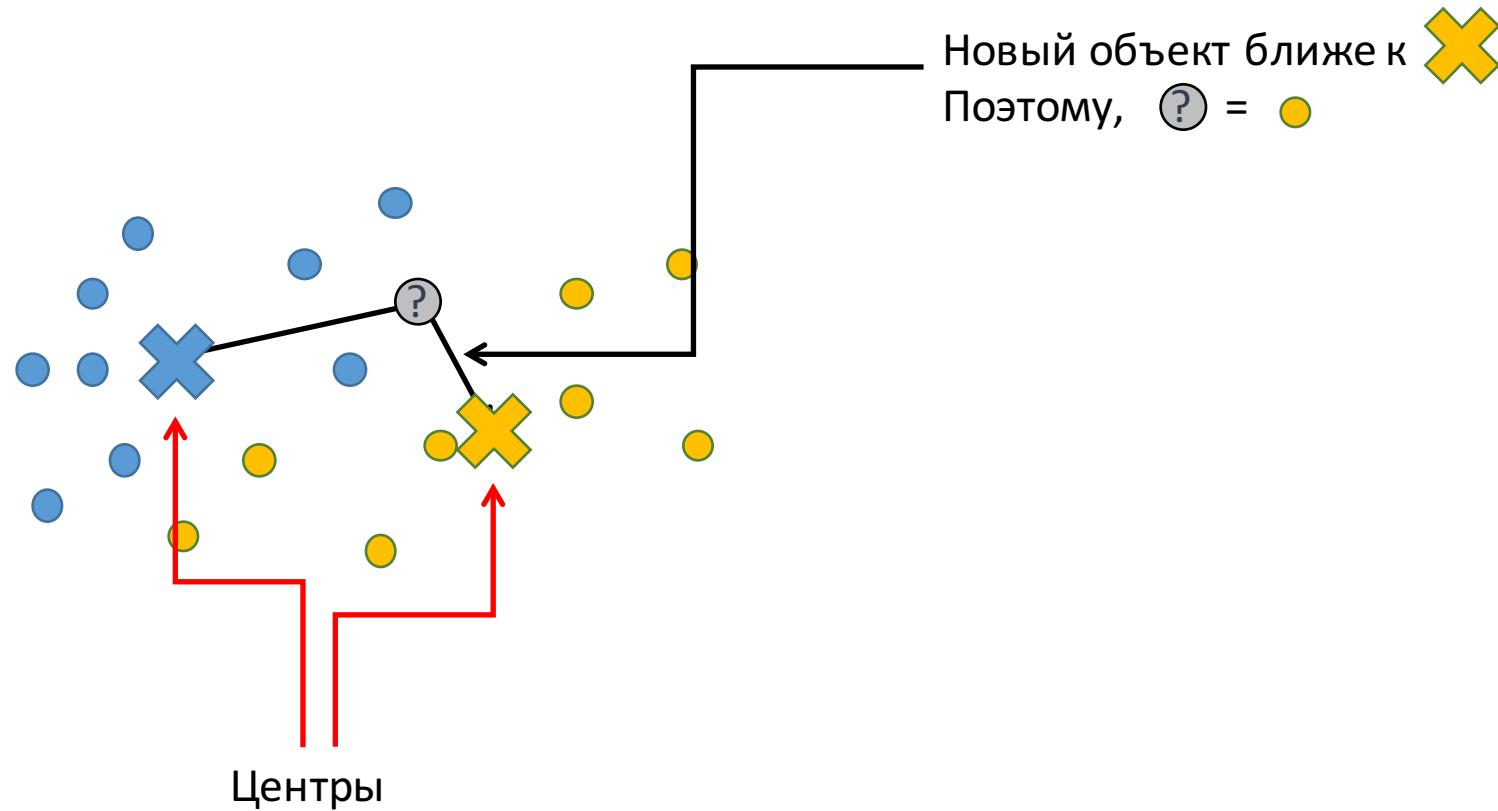
# Центроидный классификатор



# Центроидный классификатор



# Центроидный классификатор



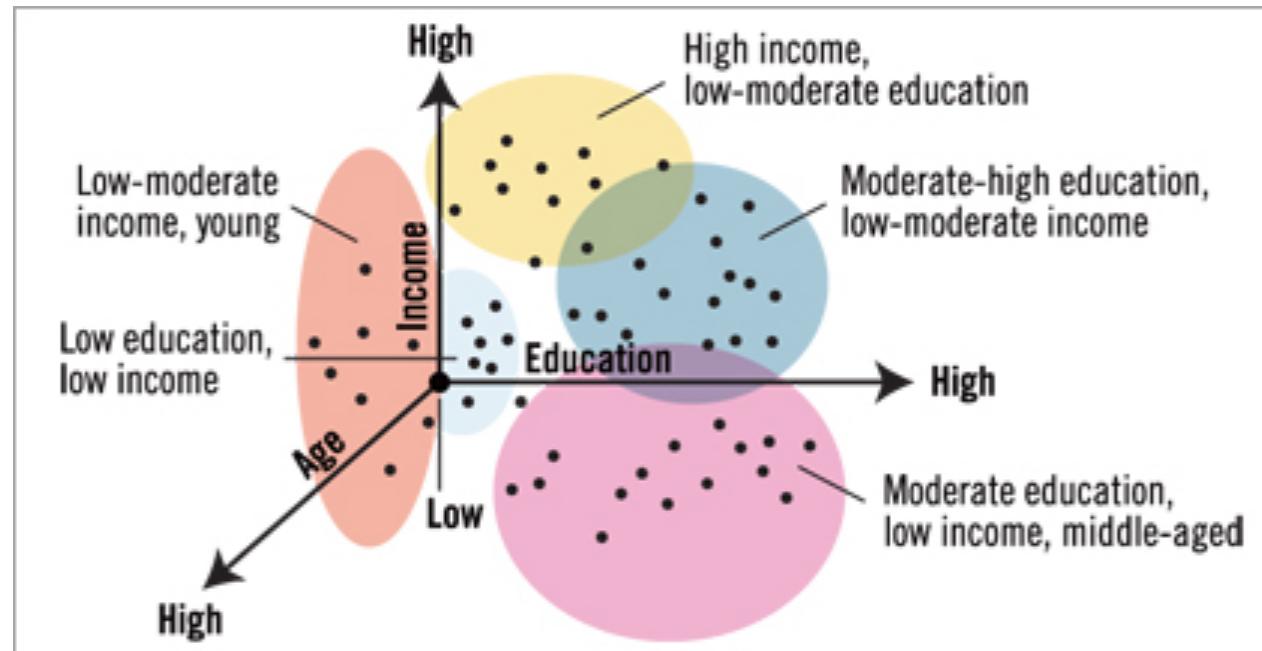
# Кластеризация

Вход (обучающая выборка):

Признаки  $N$  объектов

Выход:

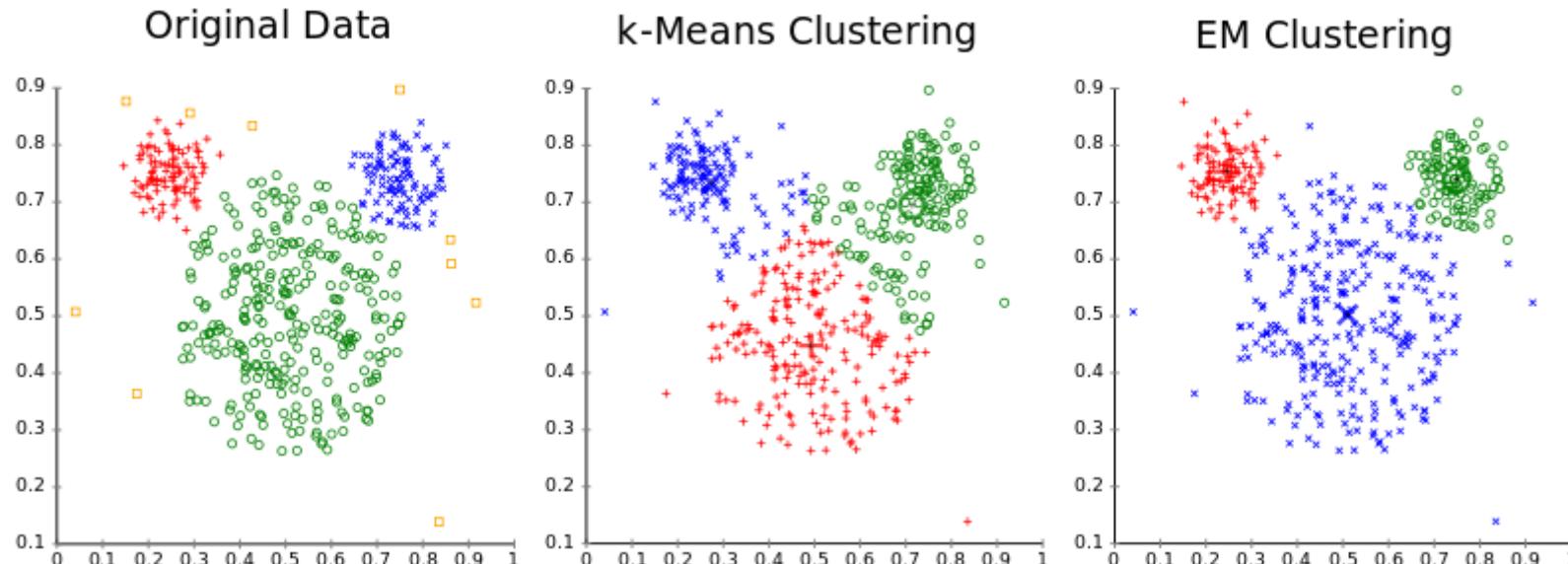
Найденные в выборке классы (кластеры), метки кластеров для объектов из обучающей выборки и алгоритм отнесения новых объектов к кластеру



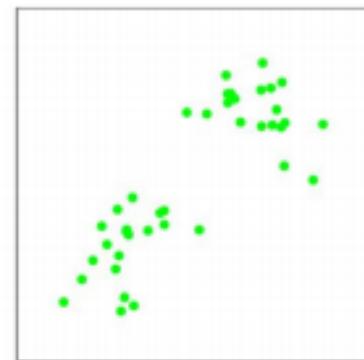
Пример: сегментация рынка

# Кластеризация

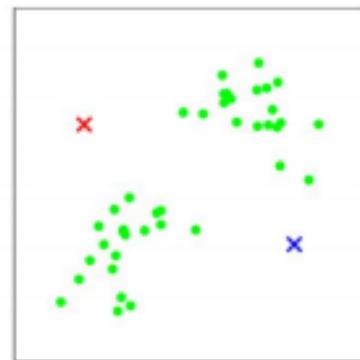
Different cluster analysis results on "mouse" data set:



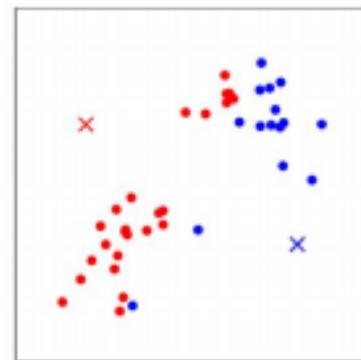
# Простой алгоритм кластеризации: kMeans



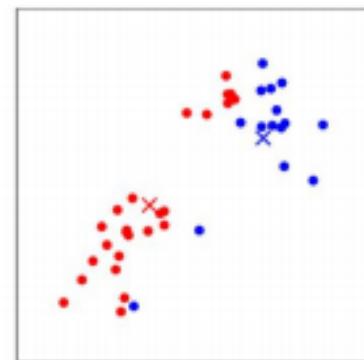
(a)



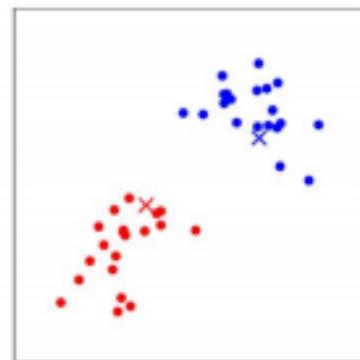
(b)



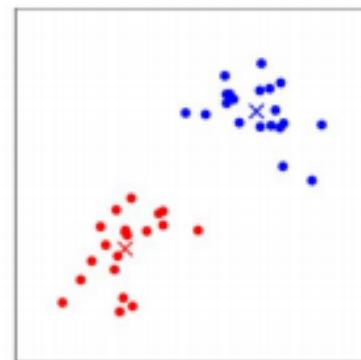
(c)



(d)



(e)



(f)

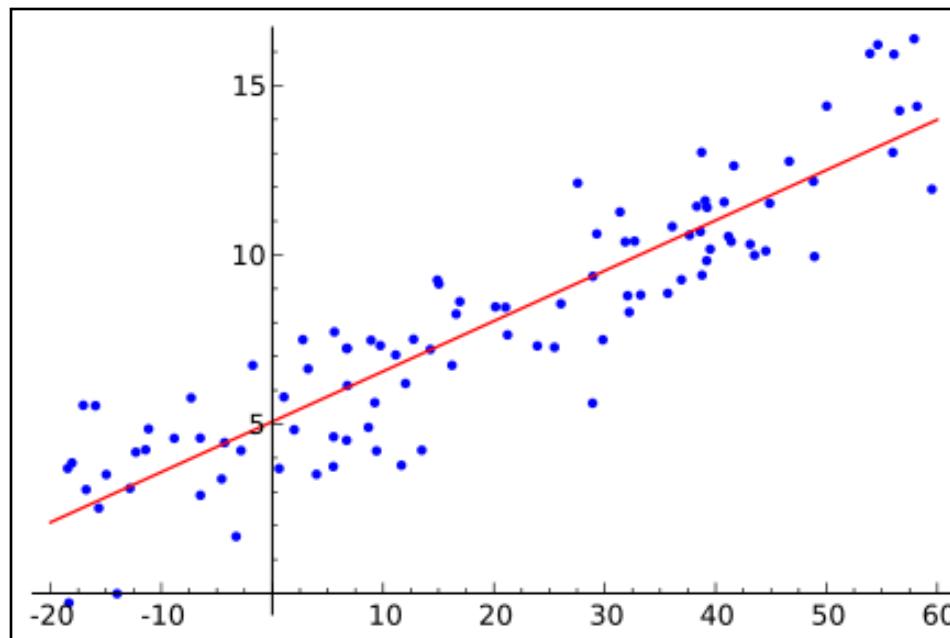
# Регрессия

Вход (обучающая выборка):

Признаки  $N$  объектов с известными значениями прогнозируемого вещественного параметра объекта

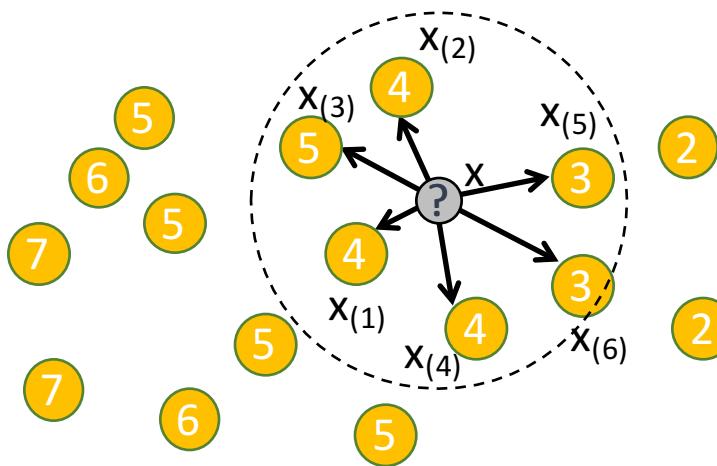
Выход:

Алгоритм, прогнозирующий значение вещественной величины по признакам объекта



# Взвешенный kNN для регрессии

Пример ( $k = 6$ ):



Веса можно определить как функцию от соседа или его номера:

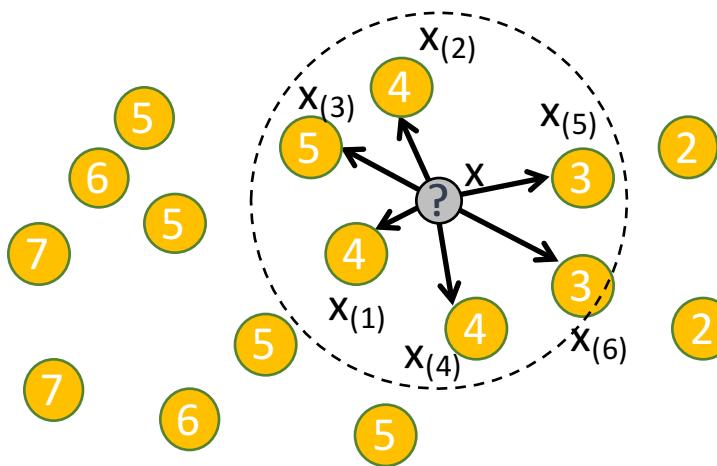
$$w(x_{(i)}) = w(i)$$

или как функцию расстояния:

$$w(x(i)) = w(d(x, x_{(i)}))$$

# Взвешенный kNN для регрессии

Пример ( $k = 6$ ):



Веса можно определить как функцию от соседа или его номера:

$$w(x_{(i)}) = w(i)$$

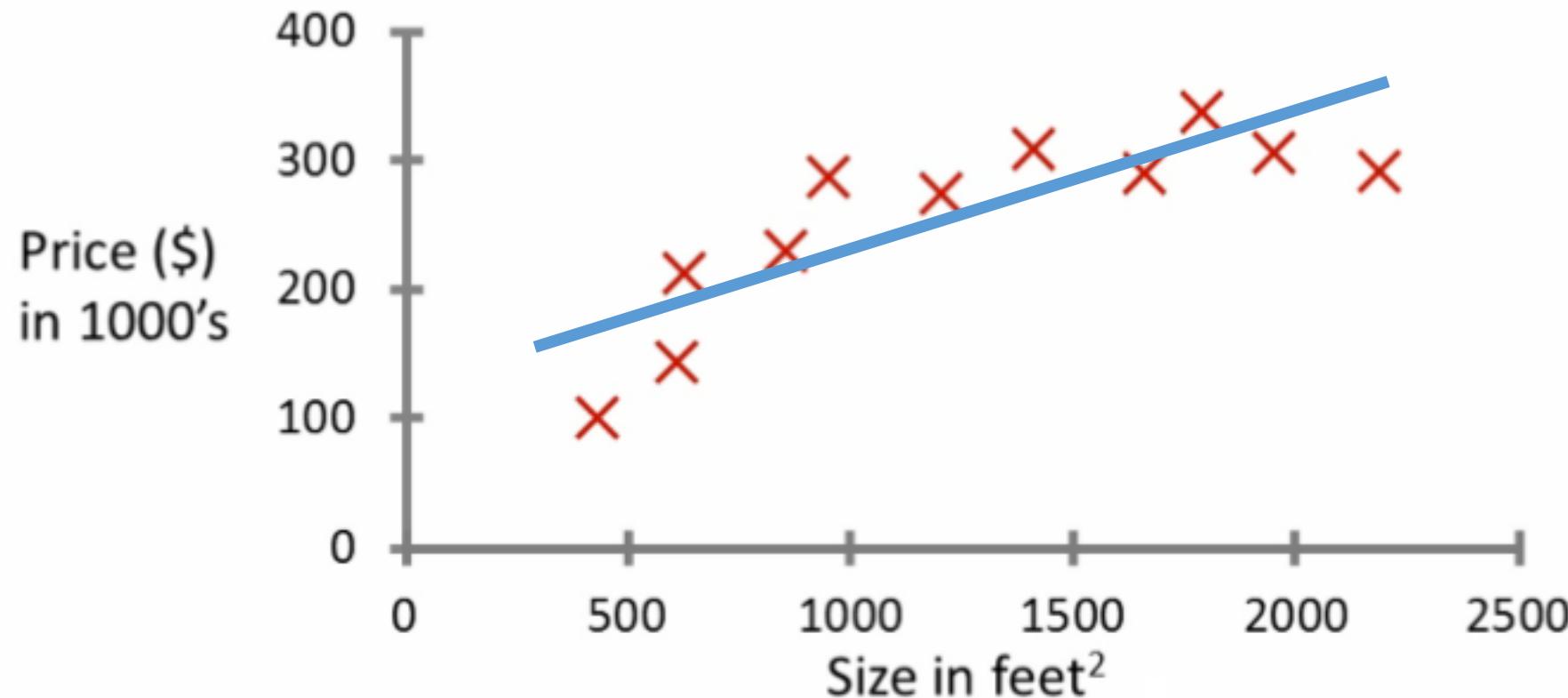
или как функцию расстояния:

$$w(x(i)) = w(d(x, x_{(i)}))$$

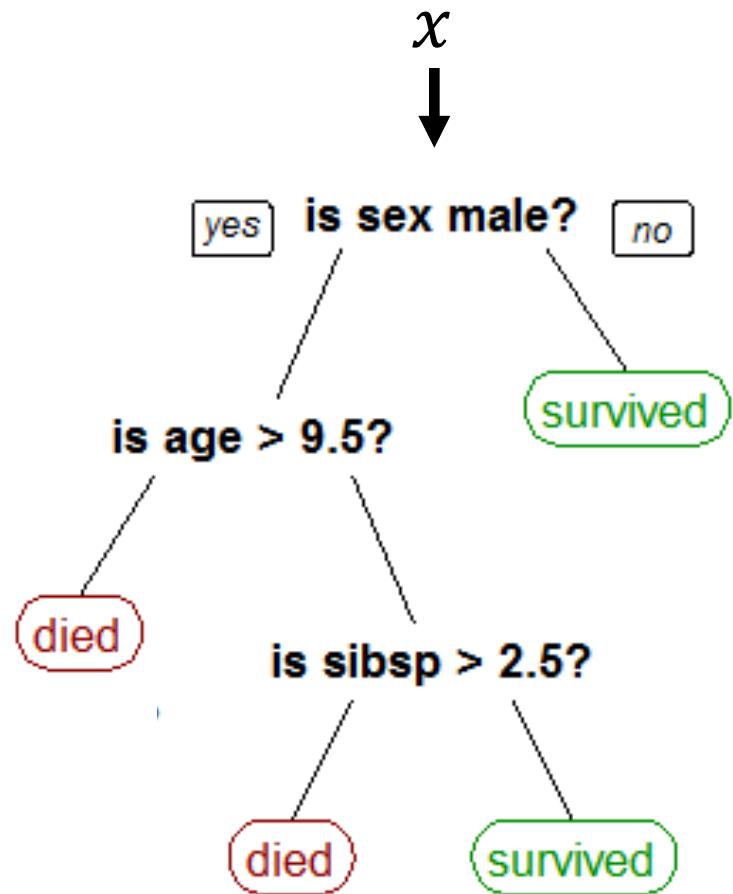
$$\textcircled{?} = \frac{4 \cdot w(x_{(1)}) + 4 \cdot w(x_{(2)}) + 5 \cdot w(x_{(3)}) + 4 \cdot w(x_{(4)}) + 3 \cdot w(x_{(5)}) + 3 \cdot w(x_{(6)})}{w(x_{(1)}) + w(x_{(2)}) + w(x_{(3)}) + w(x_{(4)}) + w(x_{(5)}) + w(x_{(6)})}$$

Наиболее часто используемые методы

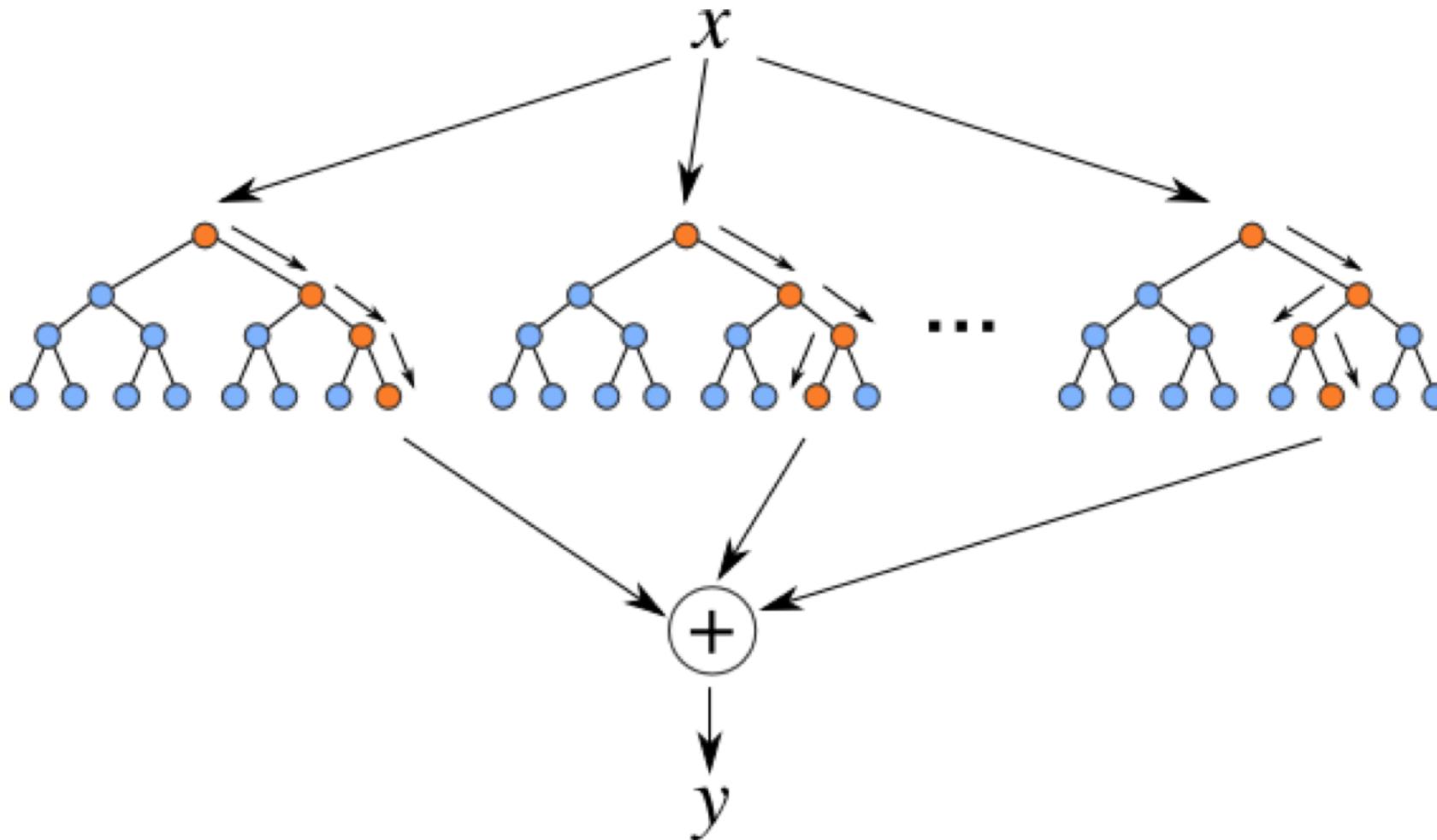
# Линейные модели



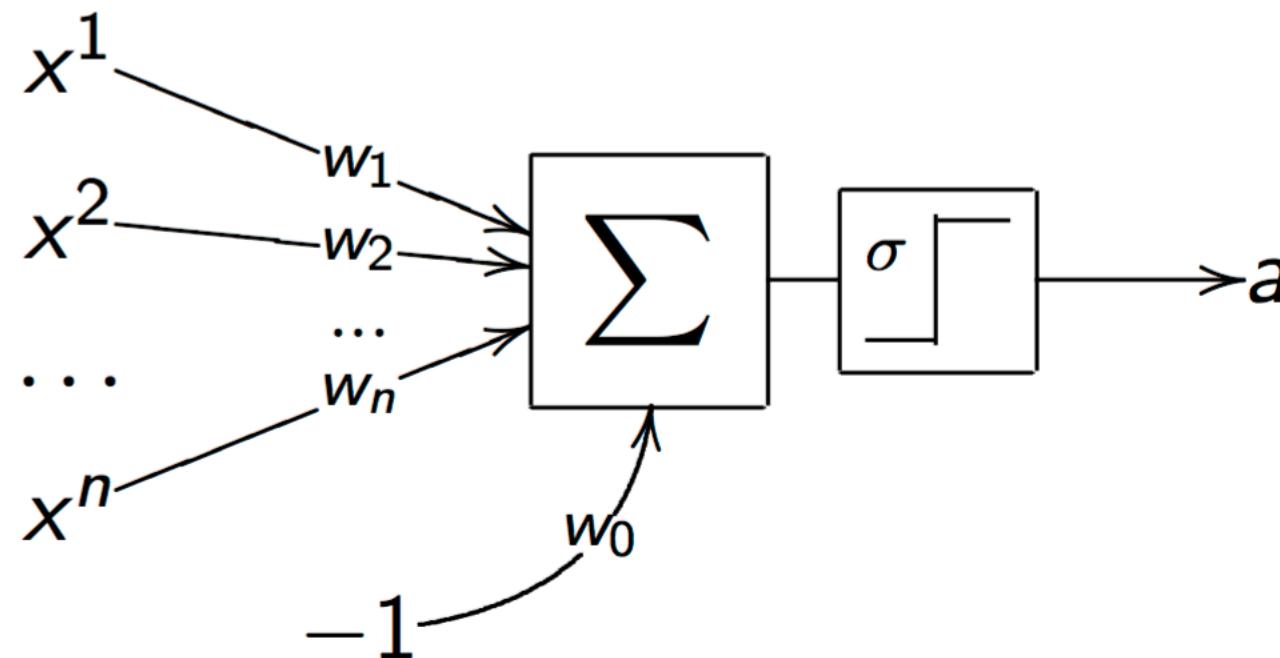
# Решающие деревья



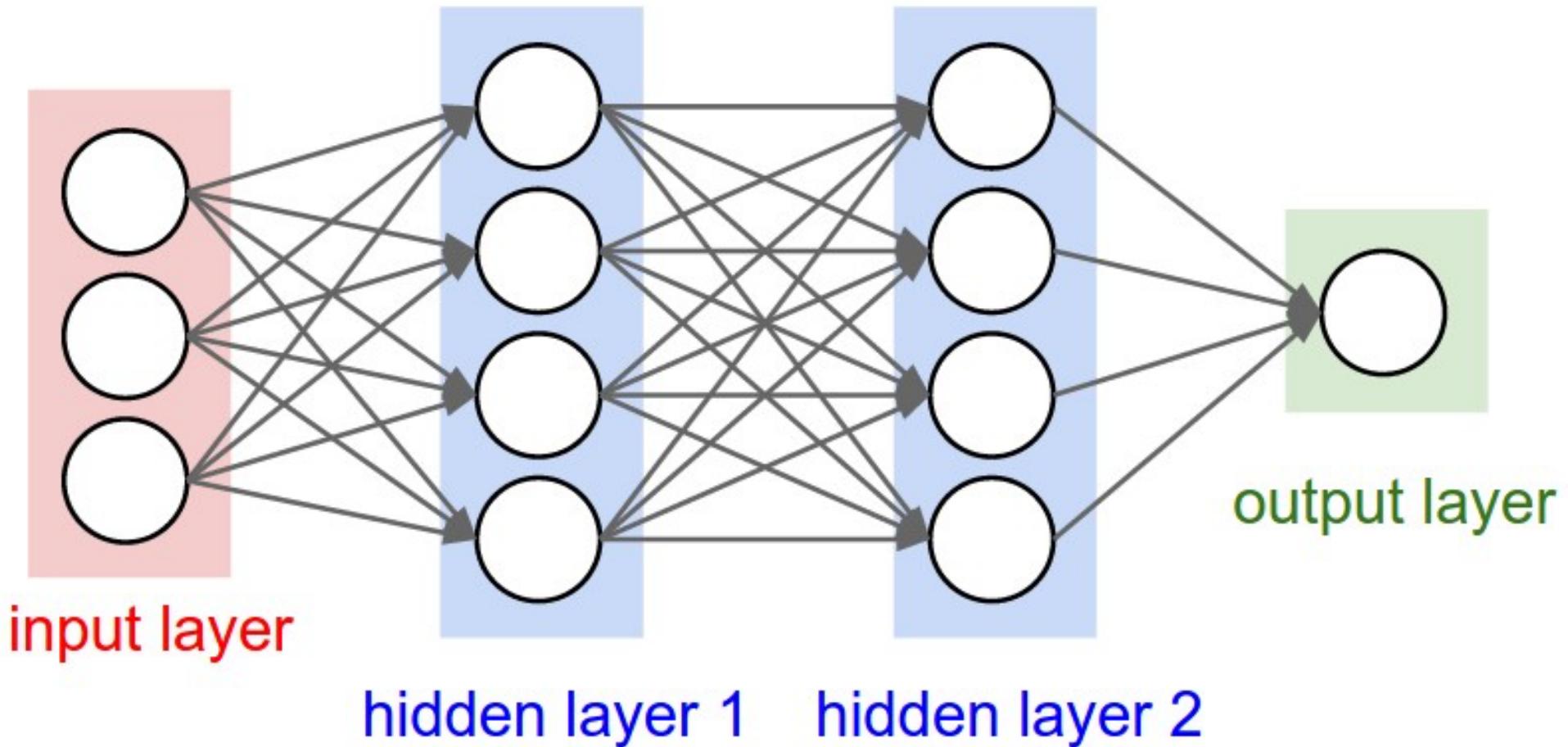
# Ансамбли решающих деревьев



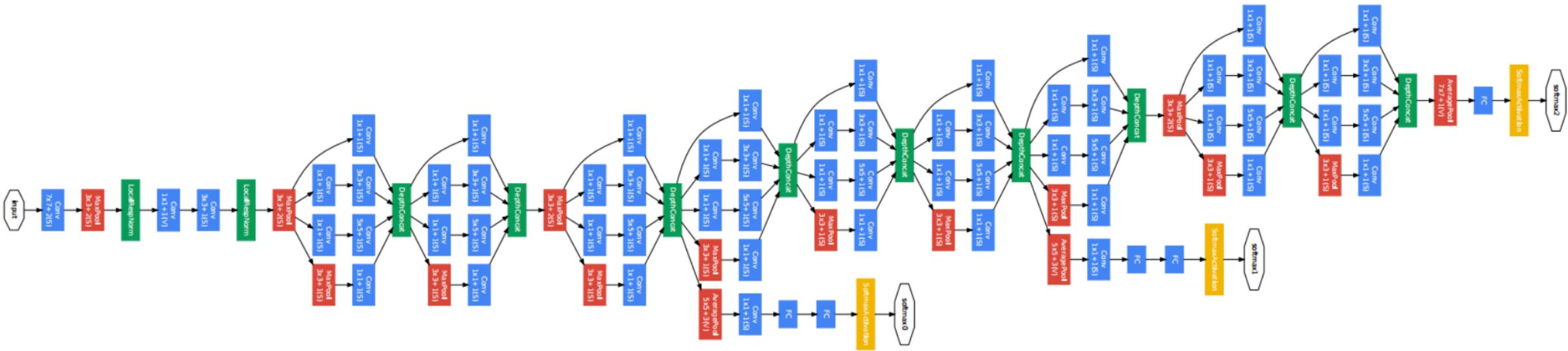
# Нейронные сети



# Нейронные сети



# Нейронные сети



# GoogLeNet

# Функционалы качества и методы оптимизации

# Задача регрессии

$x_1, x_2, \dots, x_l$  - точки, в которых известны значения некоторой величины:

$$y_1, y_2, \dots, y_l$$

Мы строим прогнозирующий алгоритм:

$$y_i \approx a(x_i)$$

Но что значит «примерно равно»?

# Задача регрессии

$x_1, x_2, \dots, x_l$  - точки, в которых известны значения некоторой величины:

$$y_1, y_2, \dots, y_l$$

Мы строим прогнозирующий алгоритм:

$$y_i \approx a(x_i)$$

Но что значит «примерно равно»?

Например, это:

$$\sum_{i=1}^l (y_i - a(x_i))^2 \rightarrow \min$$

# Задача регрессии

$x_1, x_2, \dots, x_l$  - точки, в которых известны значения некоторой величины:

$$y_1, y_2, \dots, y_l$$

Мы строим прогнозирующий алгоритм:

$$y_i \approx a(x_i)$$

Но что значит «примерно равно»?

В общем случае:

$$\sum_{i=1}^l L(y_i, a(x_i)) \rightarrow \min$$

# Задача классификации

$x_1, x_2, \dots, x_l$  - объекты, для которых известны их классы:

$$y_1, y_2, \dots, y_l$$

Мы строим прогнозирующий алгоритм:

$$y_i = a(x_i)$$

Как выразить то, что он должен угадывать класс как можно чаще?

# Задача классификации

$x_1, x_2, \dots, x_l$  - объекты, для которых известны их классы:

$$y_1, y_2, \dots, y_l$$

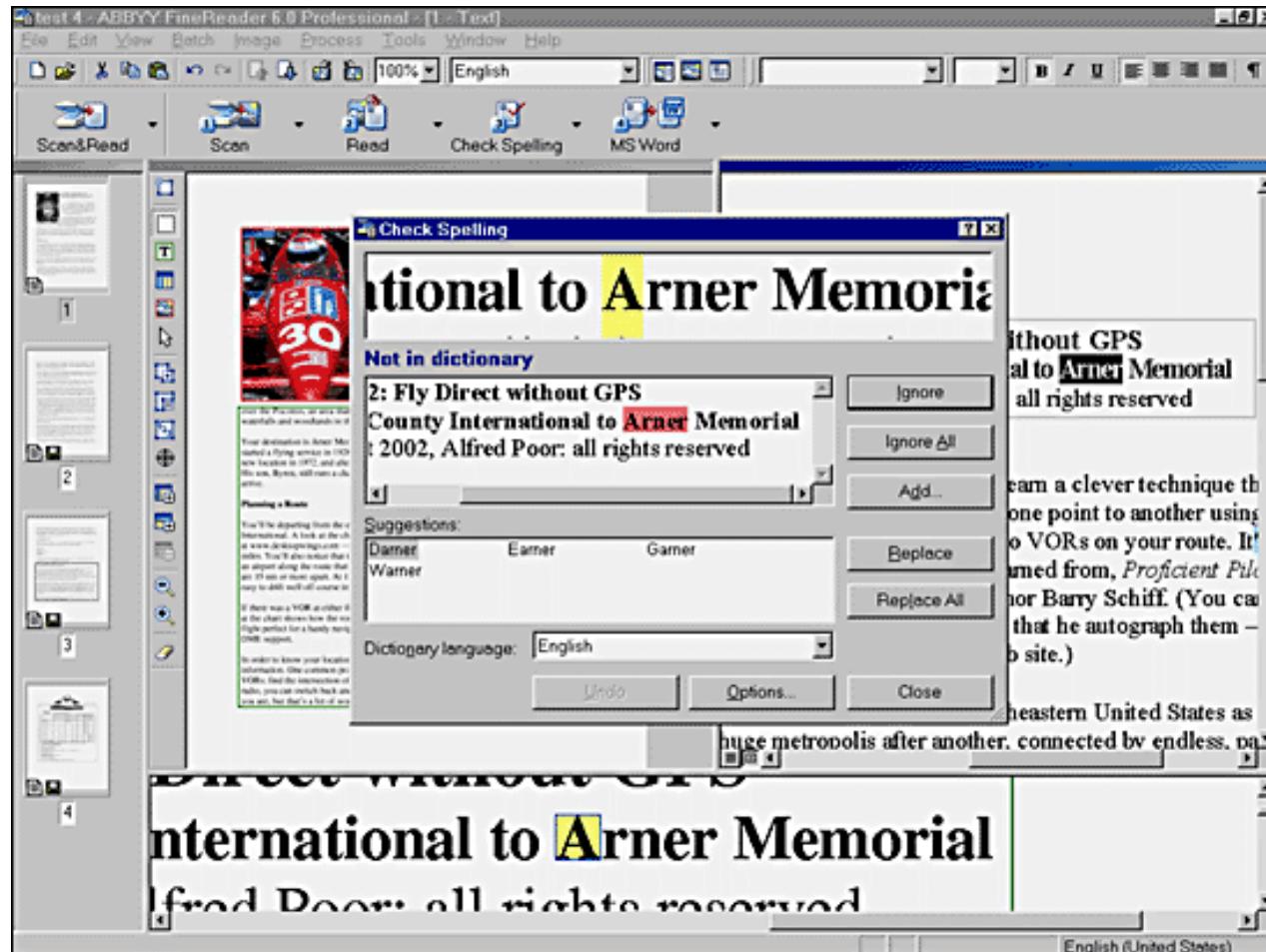
Мы строим прогнозирующий алгоритм:

$$y_i = a(x_i)$$

Как выразить то, что он должен угадывать класс как можно чаще?

$$\sum_{i=1}^l [y_i \neq a(x_i)] \rightarrow \min$$

# Сложный пример: исправление опечаток



# Сложный пример: исправление опечаток

$$Suggest(w) = [w_1, w_2, \dots, w_k]$$

В алгоритме есть параметры, которые когда-то были заданы «вручную». Хочется настроить их так, чтобы *suggest* был как можно «адекватней».

Есть выборка:

w (слово с опечаткой), cw(правильное написание)

Как сформулировать «адекватность» *suggest'*a,  
как настроить параметры?

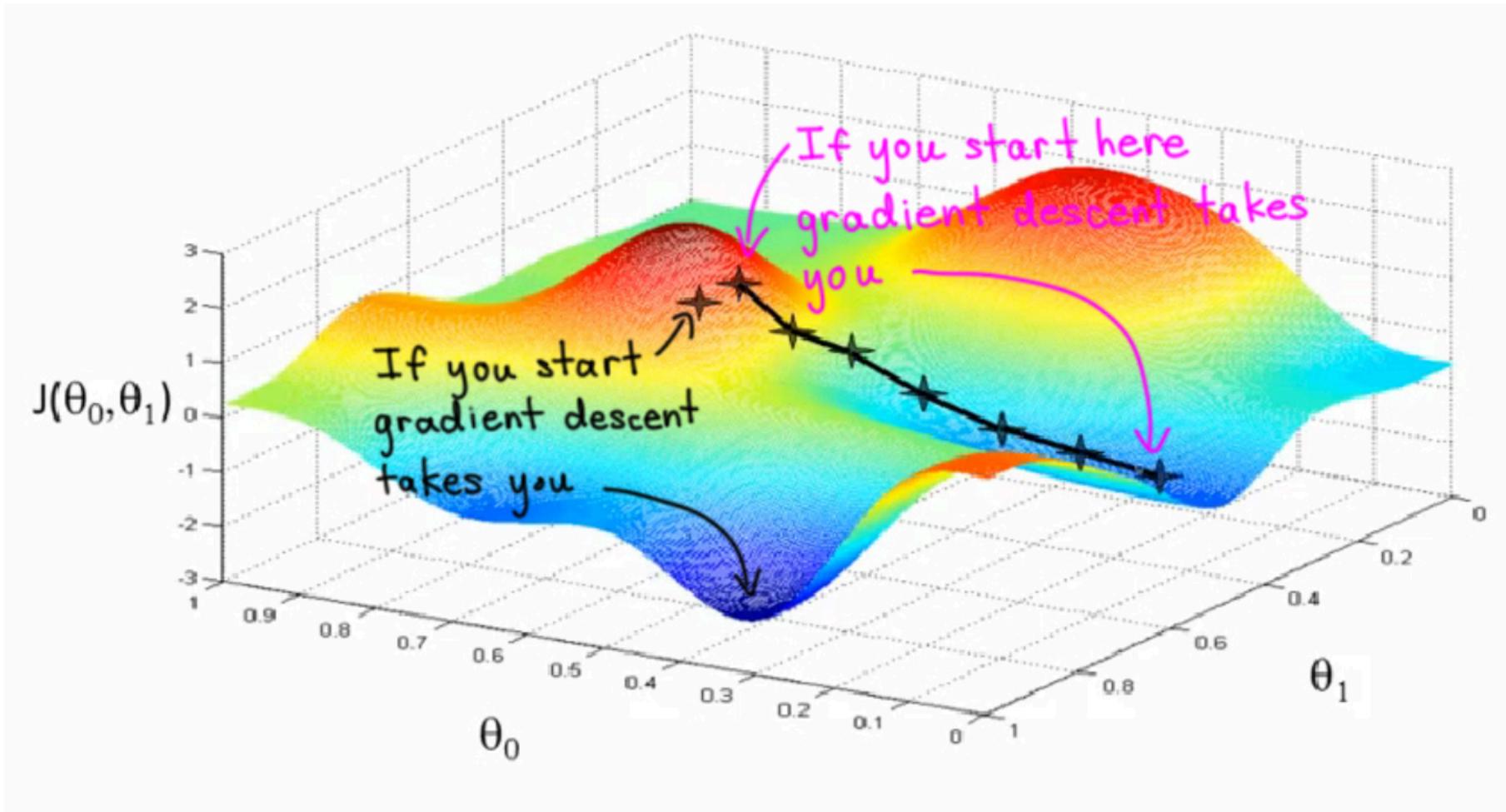
# Сложный пример: исправление опечаток

Возможное решение:

$$\begin{aligned}Suggest(w) &= [w_1, w_2, \dots, w_k] \\ Pos(w_j, [w_1, w_2, \dots, w_k]) &= j\end{aligned}$$

$$\sum_{i=1}^l Pos(cw_i, Suggest(w_i)) \rightarrow \min$$

# Градиентные методы оптимизации



# Методы глобальной оптимизации

$$P(\overline{x^*} \rightarrow \overline{x_{i+1}} \mid \overline{x_i}) = \begin{cases} 1, & F(\overline{x^*}) - F(\overline{x_i}) < 0 \\ \exp\left(-\frac{F(\overline{x^*}) - F(\overline{x_i})}{Q_i}\right), & F(\overline{x^*}) - F(\overline{x_i}) \geqslant 0 \end{cases}.$$

# Методы глобальной оптимизации



$$P(\overline{x^*} \rightarrow \overline{x_{i+1}} | \overline{x_i}) = \begin{cases} 1, & F(\overline{x^*}) - F(\overline{x_i}) < 0 \\ \exp\left(-\frac{F(\overline{x^*}) - F(\overline{x_i})}{Q_i}\right), & F(\overline{x^*}) - F(\overline{x_i}) \geq 0 \end{cases}.$$

# Признаки

# Feature engineering

- Выделение признаков (feature extraction) – генерация признаков по известным данным
- Отбор признаков (feature selection) – ранжирование признаков по «полезности» и выкидывание наименее полезных (или даже наоборот «вредных»)
- Преобразование признаков (feature transform) – создание новых признаков на основе имеющихся

# Пример 1: текстовые признаки

- Dataset: 20news\_groups
- Электронные письма, разбитые по 20 темам (классам)
- Попробуем придумать классификатор, который различает две темы:  
**auto и politics.mideast**

# Извлечение текстовых признаков

- Пример письма 1:

From: carl\_f\_hoffman@cup.portal.com

Newsgroups: rec.autos

Subject: 1993 Infiniti G20

Message-ID: <78834@cup.portal.com>

Date: Mon, 5 Apr 93 07:36:47 PDT

Organization: The Portal System (TM)

Lines: 26

I am thinking about getting an Infiniti G20.

In consumer reports it is ranked high in many  
categories including highest in reliability index for compact cars.  
Mitsubishi Galant was second followed by Honda Accord.)

# Извлечение текстовых признаков

- Пример письма 2:

From: Bob.Waldrop@f418.n104.z1.fidonet.org (Bob Waldrop)

Subject: Celebrate Liberty! 1993

Message-ID: <1993Apr5.201336.16132@dsd.es.com>

Followup-To:talk.politics.misc

Announcing... Announcing... Announcing... Announcing...

CELEBRATE LIBERTY!  
1993 LIBERTARIAN PARTY NATIONAL CONVENTION  
AND POLITICAL EXPO

THE MARRIOTT HOTEL AND THE SALT PALACE  
SALT LAKE CITY, UTAH

# Текстовые признаки: bag-of-words



the world of **TOTAL**

**all about the company**

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

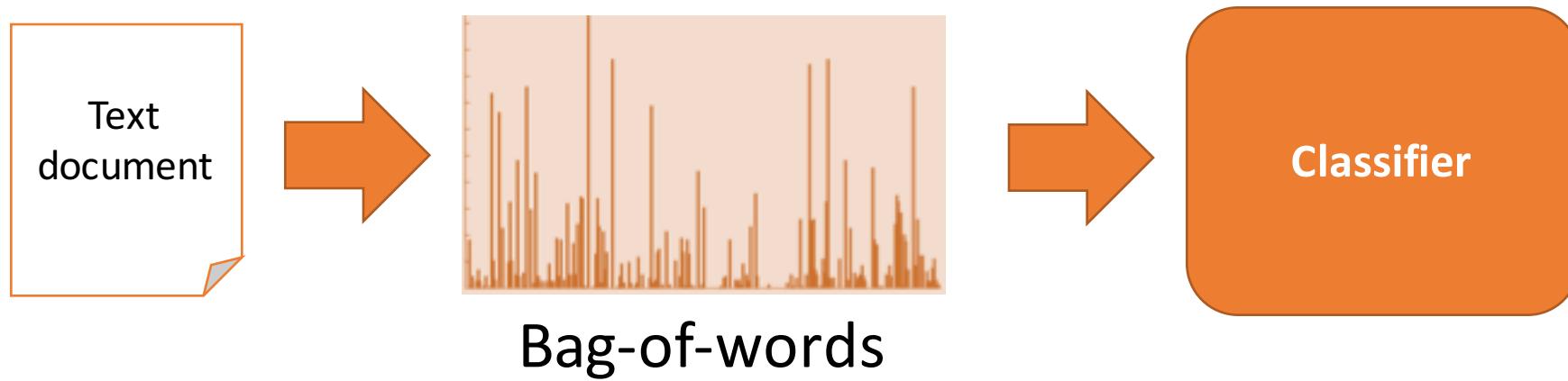
Our expanding refining and marketing operations in Asia and the Mediterranean Basin complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

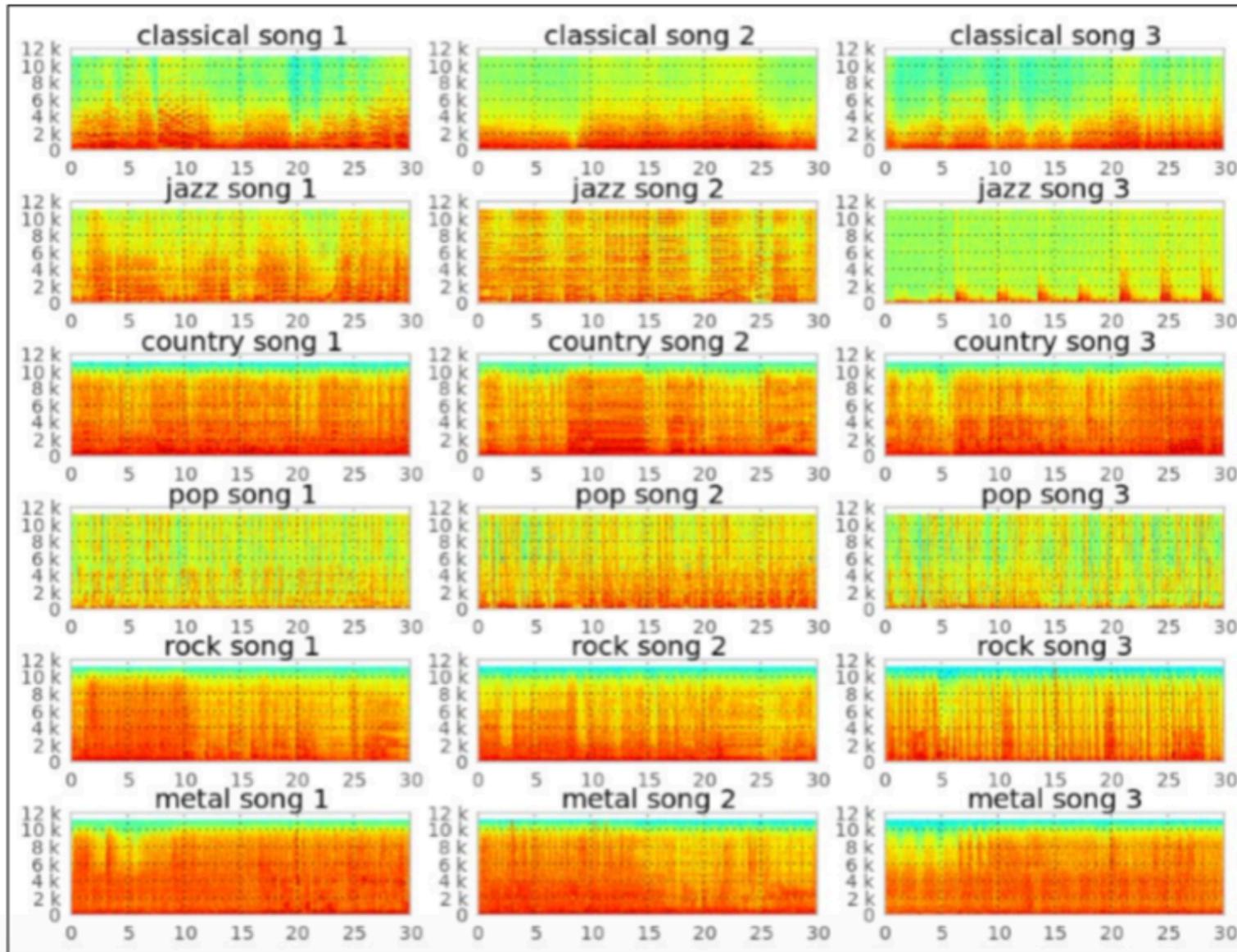


aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
...	
Zaire	0

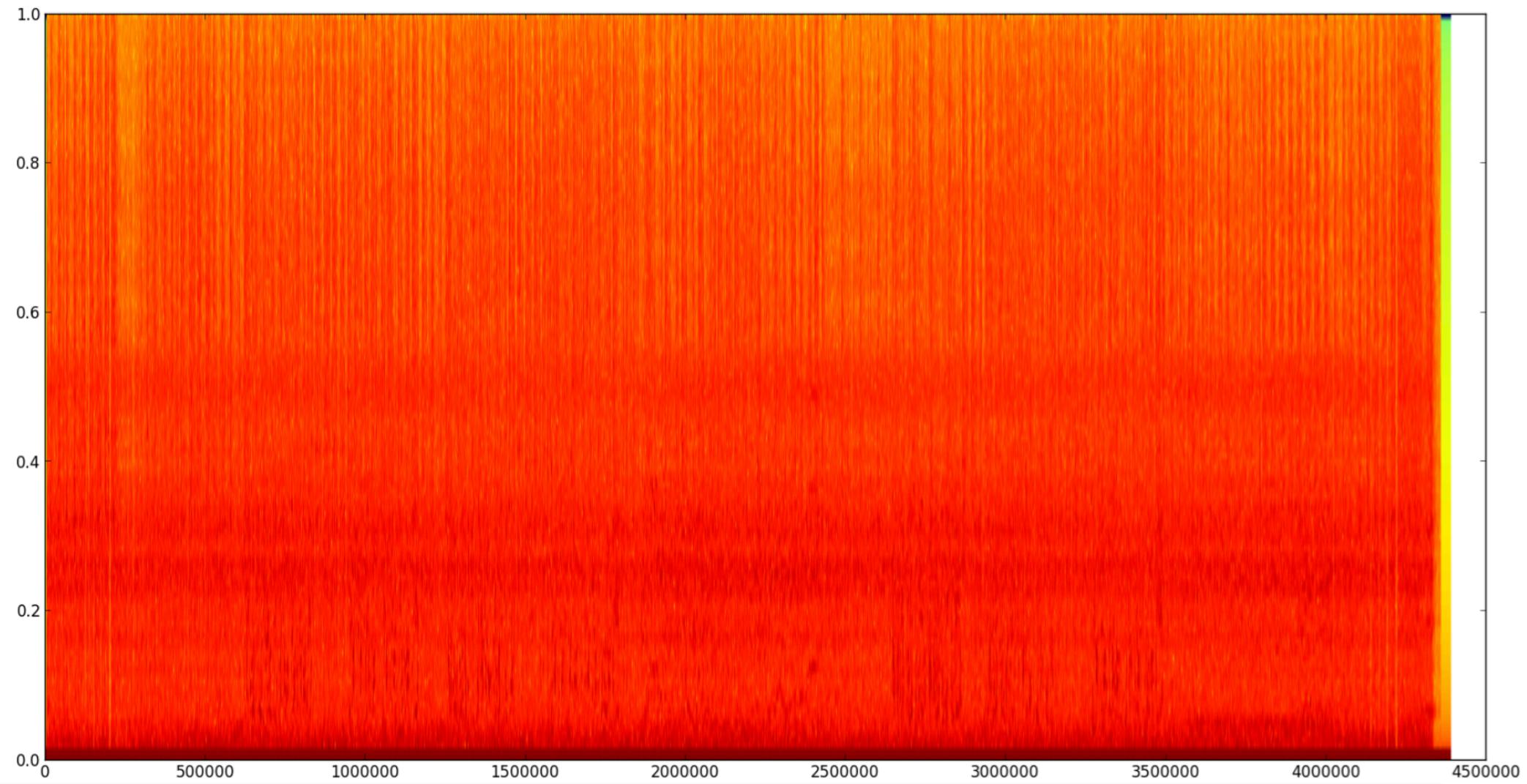
# Самый простой классификатор текстов



## Пример 2: признаки аудиофайла



## Пример 2: признаки аудиофайла

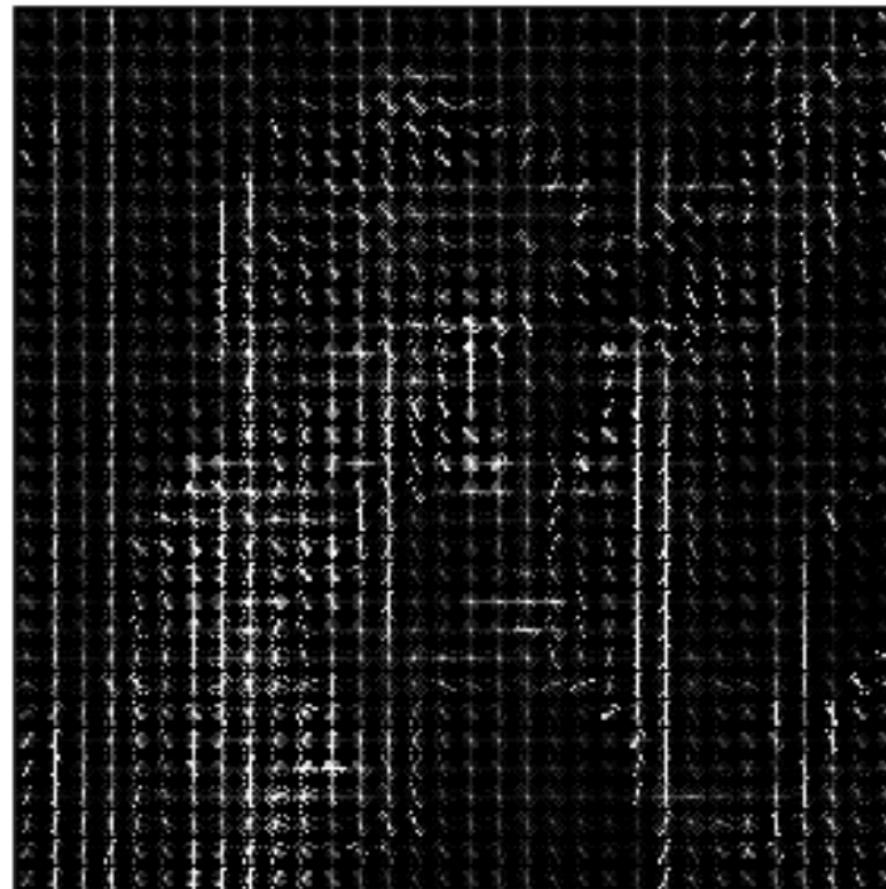


# Пример 3: признаки изображения

Input image



Histogram of Oriented Gradients



## Пример 4: категориальные признаки

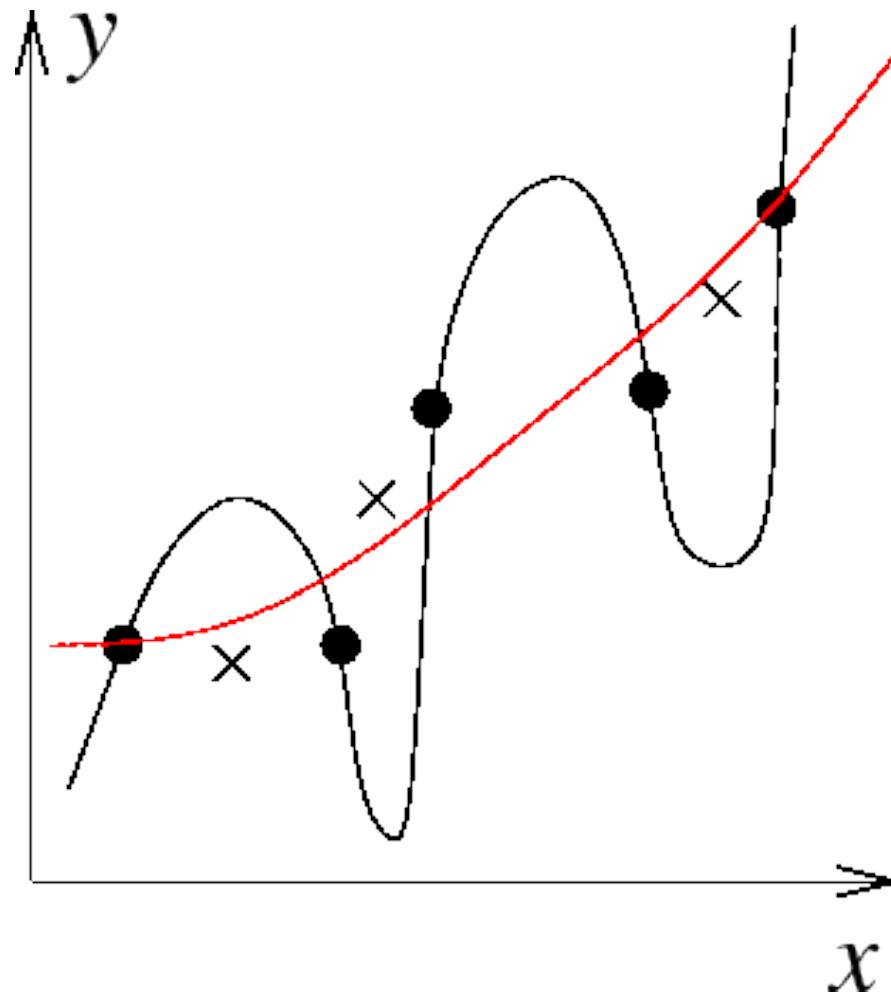
№ склада	Город	...	Продано вина (ящиков)
2343	Москва	...	56000
185	Самара	...	10500
121	Ростов	...	11300
...	...	...	...

## Пример 4: категориальные признаки

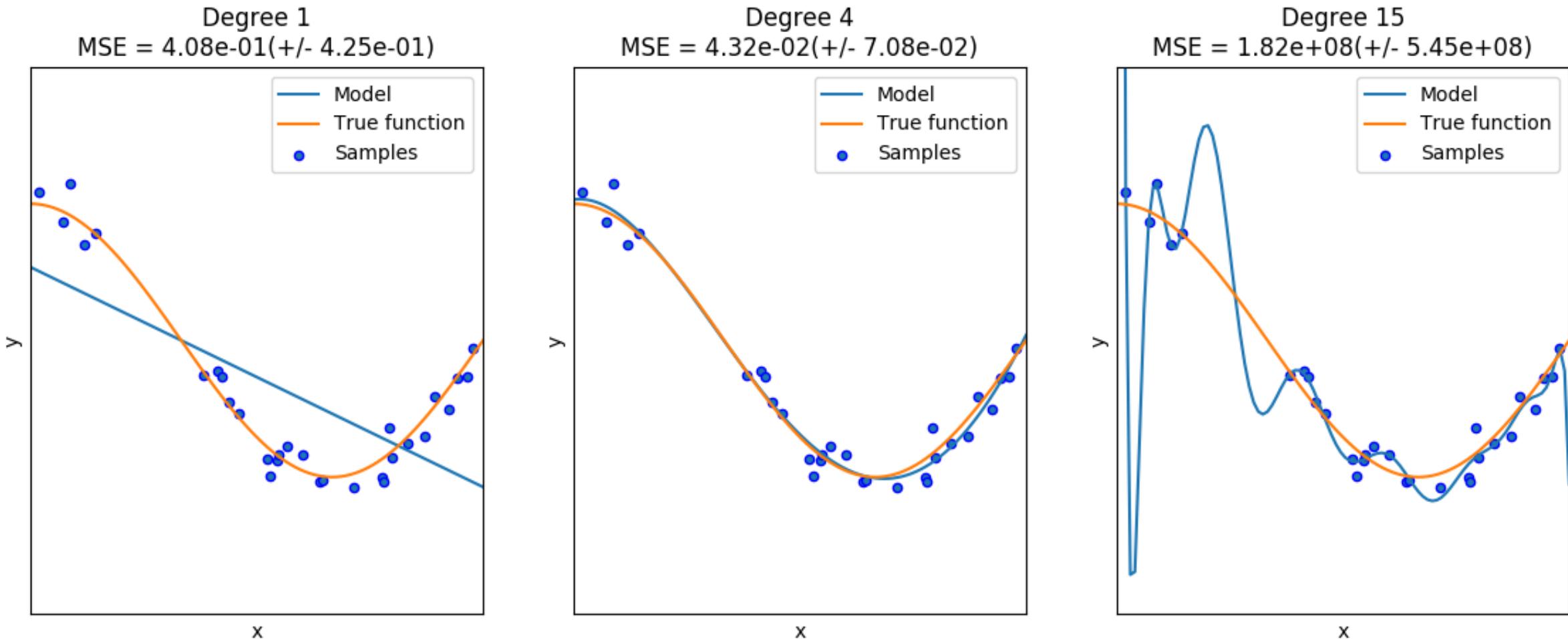
№ склада	В среднем продают в городе	...	Продано вина (ящиков)
2343	59000	...	56000
185	11200	...	10500
121	12100	...	11300
...	...	...	...

# Переобучение

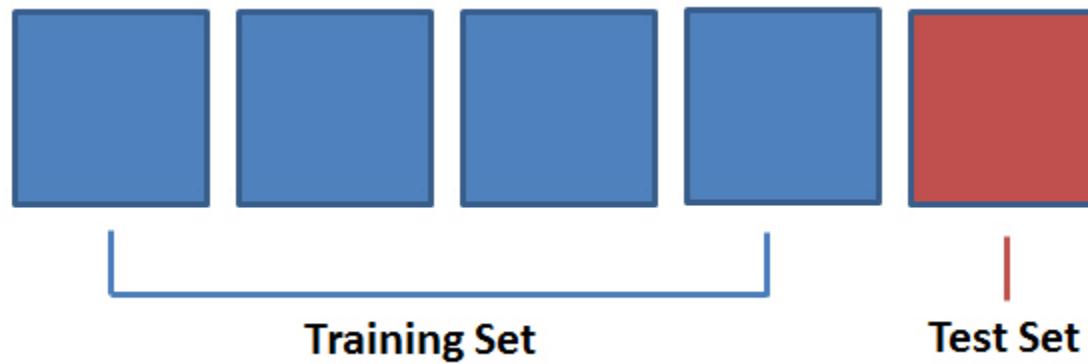
# Переобучение на примере регрессии



# Сложность модели, недо- и переобучение

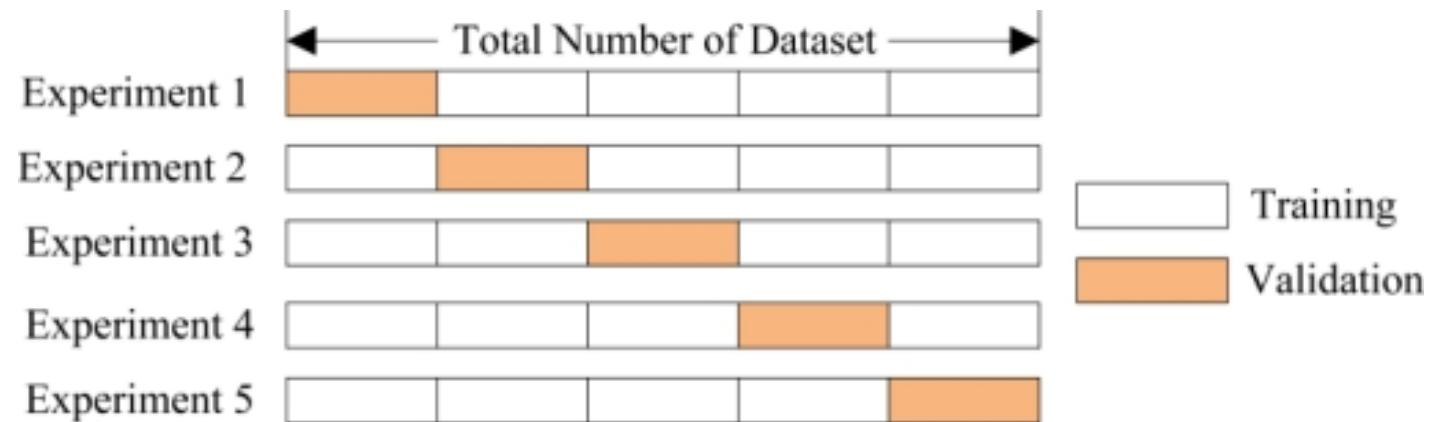


# Оценка качества



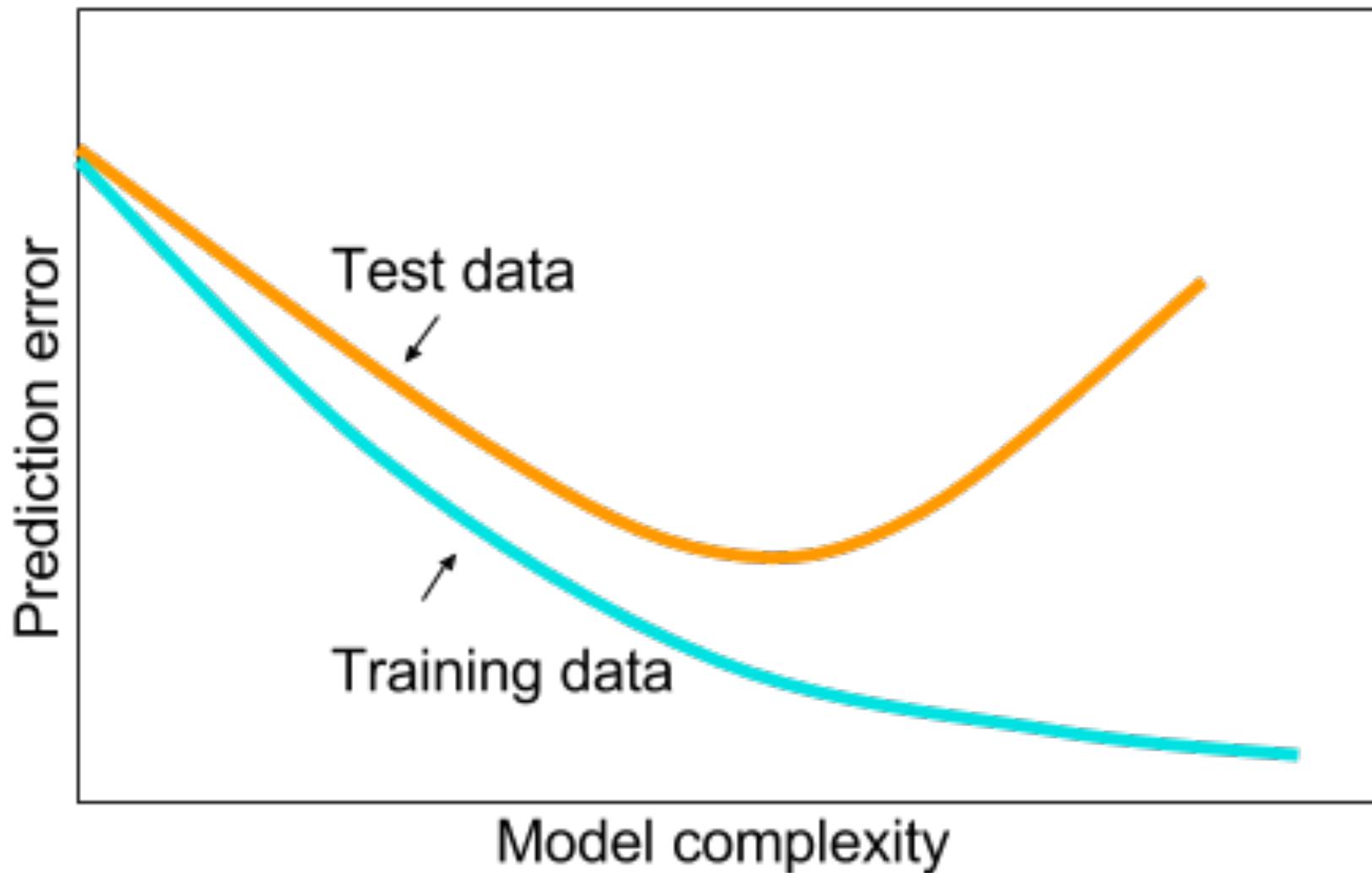
# Кросс-валидация

K-Fold cross validation:



На картинке  $k = 5$ , обычно такое  $k$  и используют. Другие частые варианты – 3 и 10.

# Кривые обучения (learning curves)

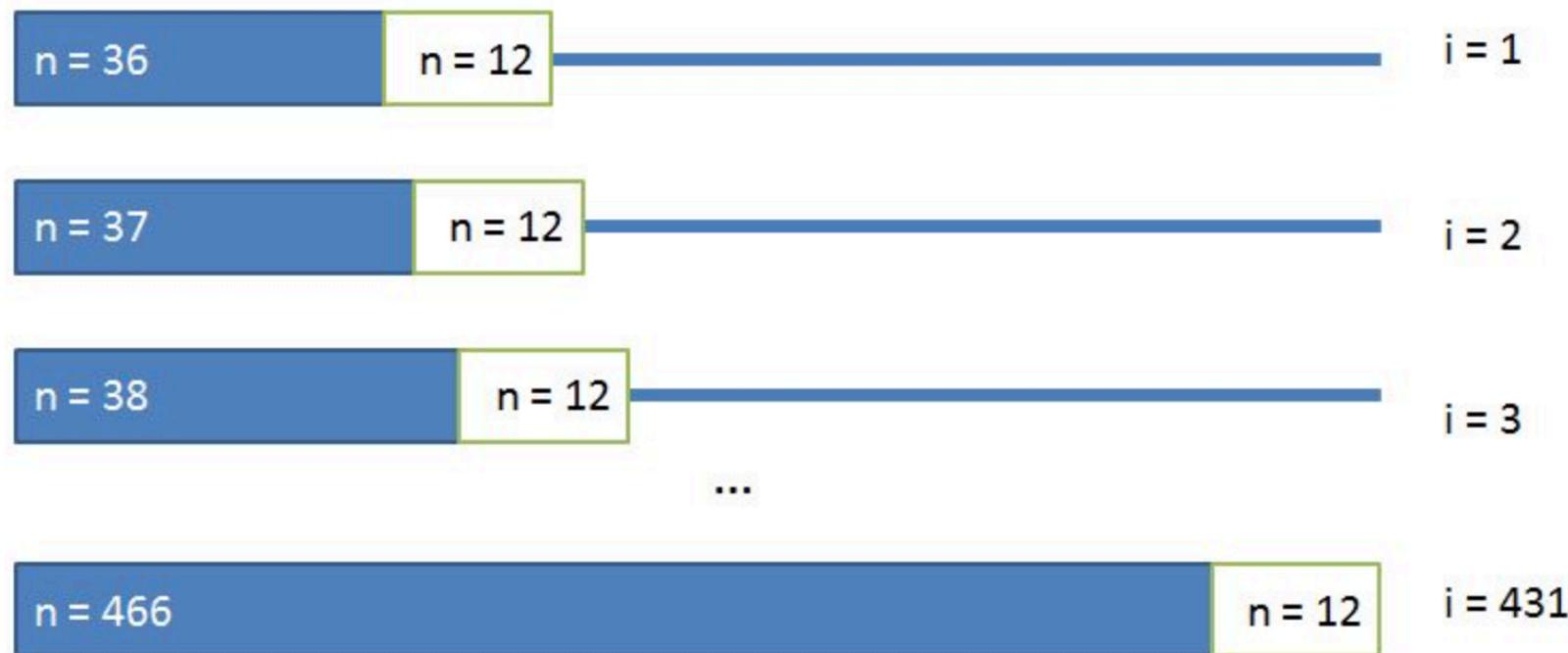


# Кросс-валидация и данные «из будущего»

N = 478 (month-end data)

June 1967

March 2007



# История про танки



Классификатор: есть танки на снимке или нет

# История про танки



Классификатор: есть танки на снимке или нет

# Часть III: инструменты

# Python

# На чем будут примеры

- Python, библиотеки: numpy, scipy, sklearn, matplotlib, pandas
- Почему Python? Потому что можно всего в 5 - 30 строк очень простого кода продемонстрировать интересные явления.
- Что использовать на практике – ваш выбор
- Под Windows проще всего установить Anaconda Python



PyCharm



python

# Scikit-learn

# Scikit-learn

The screenshot shows the official scikit-learn website. At the top, there is a navigation bar with links for Home, Installation, Documentation (with a dropdown menu), Examples, a Google Custom Search bar, and a Search button. Below the navigation bar, there is a large banner featuring the scikit-learn logo and the text "Machine Learning in Python". To the left of the banner, there is a grid of 27 small plots illustrating various machine learning models. The first row contains plots for "Input data", "Nearest Neighbors", "Linear SVM", "RBF SVM", "Gaussian Process", "Decision Tree", "Random Forest", and "Neural N". The subsequent rows show the results of these models on the same dataset, with numerical scores (e.g., 97, 98, 95) indicating accuracy or performance.

scikit-learn  
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

# Машинное обучение в несколько строк

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression()  
model.fit(X_train, y_train)  
predictions = model.predict(X_test)
```

Спасибо за внимание!

Далее – представление направлений

# Направление «Индустрія»

# Содержание направления

- Постановка задач
- Оценка качества в ML задачах
- Часто используемые на практике ML методы
- Инструменты для анализа данных
- Создание прототипов

- Постановка задач



## • Оценка качества в ML задачах

- Обзор существующих метрик качества
- Переход от бизнес задачи к метрикам качества
- Оценка потенциального экономического эффекта
- Онлайн оценка качества

• АБ тестирование



## • Основные алгоритмы ML

- Градиентный бустинг
- Случайный лес
- Линейные модели

- Инструменты для анализа данных

- XGBoost
- LightGBM
- CatBoost
- Vowpal Wabbit
- Spark
- И много много МНОГО библиотек для python

- Создание прототипов



# Приходите на направление и вы научитесь:

- Делать правильные с точки зрения бизнеса постановки задач
- Оценивать качество ваших решений как в оффлайн, так и в онлайн экспериментах
- Разбираться в часто используемых на практике ML методах
- Владеть инструментами для анализа данных
- Создавать прототипы продукта

Направление «Тренды»

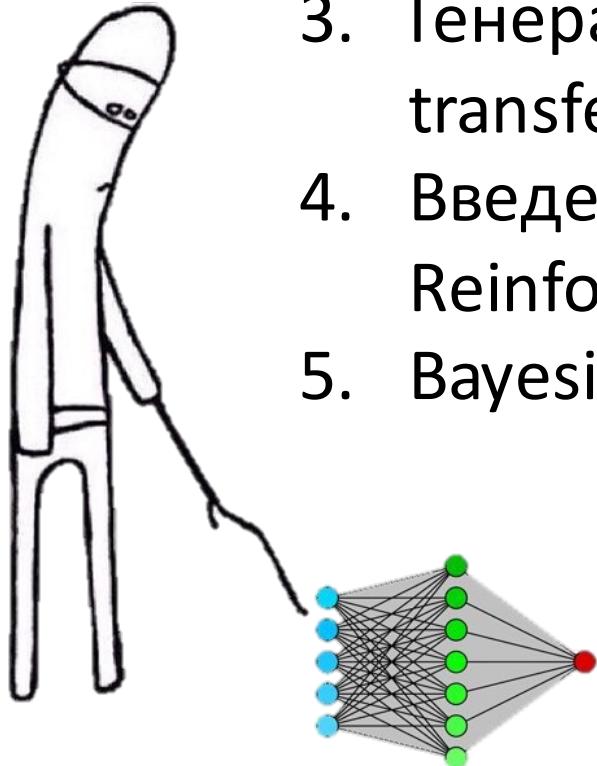
# Темы направления

1. Базовые блоки нейросетей (Dense, Convolution, RNN)
2. Обзор современных DL-фреймворков. Введение в Tensorflow.
3. Анализ изображений. Задача классификации. Обучение глубоких сетей и интерпретация весов. Задачи сегментации и детектирования объектов.
4. Анализ текстов. Задача классификации. Различные подходы и текущий state-of-the art.
5. Рекуррентные нейросети. GRU, LSTM. Seq2Seq. Задачи Machine translation, Image captioning.



# Темы направления

1. Механизм Attention, Stack-RNN. Differentiable neural computers
2. Transfer learning, One-shot learning
3. Генеративные модели: VAE, GAN и модификации. Style transfer.
4. Введение в Reinforcement Learning и Deep Reinforcement Learning.
5. Bayesian neural networks



Направление «Спорт»

# Этапы решения задач по анализу данных

1. постановка задачи, выбор метрики
2. сбор данных, предобработка
3. решение задачи, достижение приемлемого качества
4. использование модели в продакшене
5. анализ фактических результатов и доработка модели

# Этапы решения задач по анализу данных

1. постановка задачи, выбор метрики
2. сбор данных, предобработка
3. решение задачи, достижение ~~приемлемого~~ максимального качества
4. использование модели в продакшене
5. анализ фактических результатов и доработка модели

# Этапы решения задач по анализу данных

1. постановка задачи, выбор метрики
2. сбор данных, предобработка
3. решение задачи, достижение ~~приемлемого~~ максимального качества
  - a. Создавать бейзлайны
  - b. Анализировать данные
  - c. Улучшать решение
  - d. Разбираться в чужих решениях
4. использование модели в продакшене
5. анализ фактических результатов и доработка модели

# Причины участвовать в соревнованиях

## 1. Причины участвовать. Соревнования - это:

- a. Эффективный и интересный способ учиться практическому анализу данных
- b. Интересные нетривиальные задачи и современные подходы к их решению
- c. Возможность стать известным в сообществе и получить предложение о работе

## 2. Причины не участвовать:

- a. Хочется разобраться в постановке задач / сборе данных / продакшне
- b. Нужно много времени и ресурсов

Спасибо за внимание,  
до встречи через неделю :)