

**«Машинное обучение»**

# **Постановка основных задач**

**Александр Дьяконов**

**06 сентября 2021 года**

## План лекции

### **Обучение с учителем (с размеченными данными / метками)**

термины: целевая функция, объект, метка, признак, функции ошибки, эмпирический риск, обучающая выборка, модель, алгоритм, обучение, обобщающая способность

#### **Классификация**

#### **Прогнозирование**

#### **Задачи оптимизации в обучении**

#### **Схема решения задачи машинного обучения**

### **Обучение без учителя / с неразмеченными данными**

### **Обучение с подкреплением**

Другие виды обучения: с частично размеченными данными, трансдуктивное обучение, структурный вывод, активное обучение, онлайн-обучение, Transfer Learning, Multitask Learning, Feature Learning, привилегированное обучение

### **Проблемы в машинном обучении**

## Обучение с учителем

**Supervised Learning, с размеченными данными / метками**



## Обучение с учителем

$$X_{\text{train}} = \{(x_1, y_1), \dots, (x_m, y_m)\}$$

$$y: X \rightarrow Y$$

$y$  – **целевая функция (переменная)**

**target / response / outputs / dependent variable**

$$y(x_1) = y_1$$

...

$$y(x_m) = y_m$$

$x_i$  – **объект (наблюдение)**

**observation / example / instance / object**

$X$  – **пространство объектов (входов)**

$Y$  – **пространство меток / значений целевого признака (выхода)**

## Цели

### 1. Восстановление целевой зависимости

Уметь восстанавливать метки новых объектов  $y(x)$

– найти зависимость целевой переменной от остальных

### 2. Интерпретация

Как устроена  $y(x)$

### 3. Оценка качества полученного решения

Например, на сколько ошибаемся в среднем,  
что ждать при использовании нашего прогноза

## Типы задач обучения с учителем

### Классификация (Classification)

$$|Y| = k \ll \infty$$

**бинарная**

$$Y = \{0, 1\} \text{ или } Y = \{-1, +1\}$$

**скоринговая бинарная**

$$a(x) \in [0, 1]$$

**на  $k$  непересекающихся классов**

**(multiclass classification)**

$$Y = \{1, 2, \dots, k\}$$

**на  $k$  пересекающихся классов**

**(multi-label classification)**

$$Y = \{0, 1\}^k$$

### Регрессия (Regression)

$$Y = \mathbb{R}$$

**Многомерная регрессия**

$$Y = \mathbb{R}^n$$

### Прогнозирование (Forecasting)

$$X_{\text{train}} = \{(x_1, t_1, y_1), \dots, (x_m, t_m, y_m)\}$$

$$t_1 \leq t_2 \leq \dots \leq t_m$$

### Ранжирование (Learning to Rank)

**$Y$  – ЧУМ**



## Пространство объектов

**Практически какое угодно:**

- **медицинские истории**
- **тексты**
- **сигналы / временные ряды / последовательности**
- **изображения**
- **векторы / множества / графы**
- **...**

**Для удобства-простоты-теории-практики:**

$$X = \mathbb{R}^n$$

**n-мерное признаковое пространство**

$x_i = (x_{i1}, \dots, x_{in})$  – **объект в признаком описании**

$x_{ij}$  –  **$j$ -й признак**

**inputs / attributes / repressors / properties / covariates / features / variables**

Задача в признаковой постановке

матрица «объект-признак» (data matrix)

плохой_клиент	линии	возраст	поведение_30-59_дней	Debt_Ratio	доход	число_кредитов
0	0.111673	46	0	1.329588	800.0	8
0	0.044097	69	0	0.535122	3800.0	10
0	0.047598	77	0	0.169610	3000.0	7
0	0.761149	58	1	2217.000000	NaN	4
0	0.690684	55	0	0.432552	12416.0	7

По строкам – признаковые описания объектов  
по столбцам – значения конкретных признаков



## Признаки (features)

**Задачи классификации – целевой признак категориальный**

**Задачи регрессии – целевой признак вещественный**

Вообще говоря, целевой признак тоже м.б. любым, например графом!

### Замечание

Целевой признак «условен»

Часто просто дана матрица (целевой признак приходится формировать)

## Генерация признаков

**объект может быть не задан в признаковом пространстве  
или задан в «плохом» признаковом пространстве**

**⇒ извлечение признаков:**

$$X \rightarrow \mathbb{R}^n$$

**м.б. производится автоматически**

**чем лучше генерация признаков, тем более простое ML нужно;)**

**mmp@cs.msu.ru →**

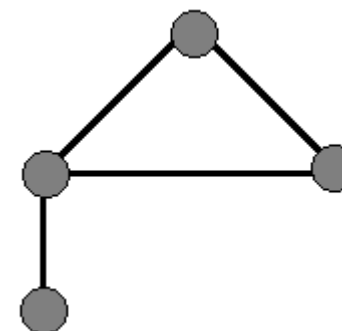
**длина = 3**

**доменов = 3**

**«1 уровень=ru» = 1**

**«1 уровень=com» = 0**

**«1 уровень=org» = 0**



**число вершин = 3**

**число рёбер = 4**

**число компонент связности = 1**

**максимальная степень = 3**

## Примеры

### Классификация спама

$X$  – письма

$Y = \{\text{спам, норма}\}$

признаки = длина письма, число вхождений слова, отправитель, ...

### Медицинская диагностика

$X$  – пациенты

$Y$  – диагнозы

признаки = результаты анализов, возраст, пол и т.п.

вариант постановки: предсказать вероятности болезней  $Y = [0, 1]^l$

### Прогнозирование цен акций

$X$  – ситуация на рынке

$Y$  – цена на акцию через час

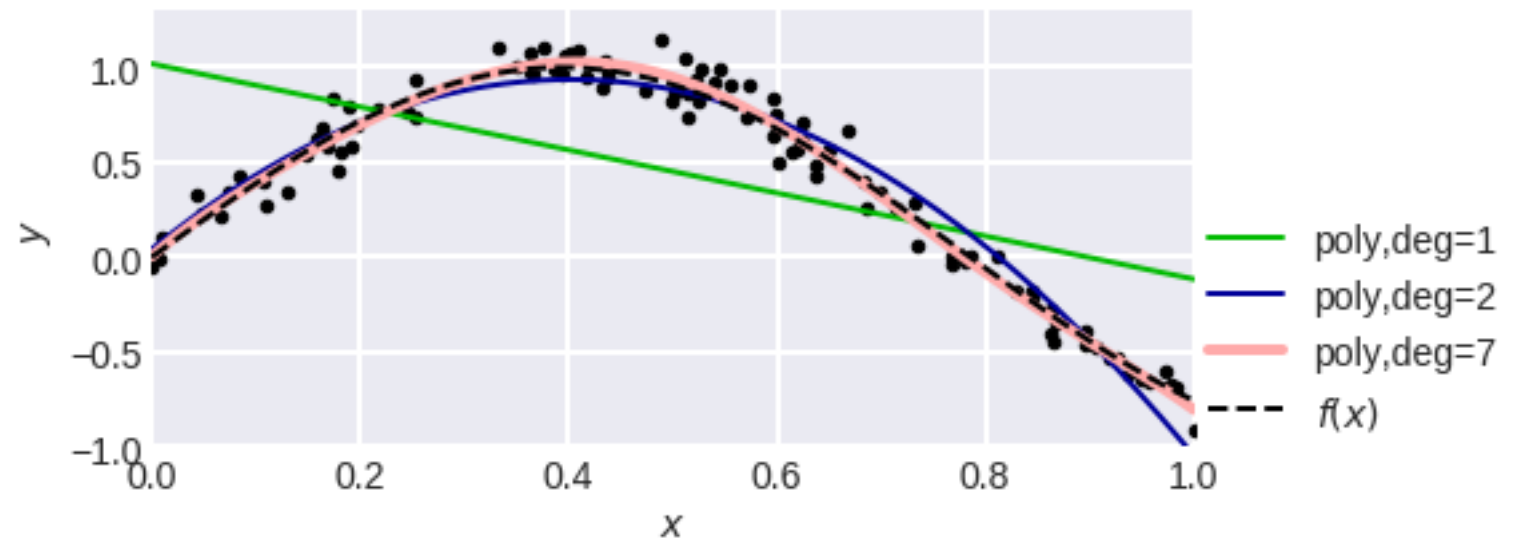
вариант постановки: множественная регрессия –

– цены нескольких акций  $Y = \mathbb{R}^l$

## Визуализация задач

### Задача регрессии

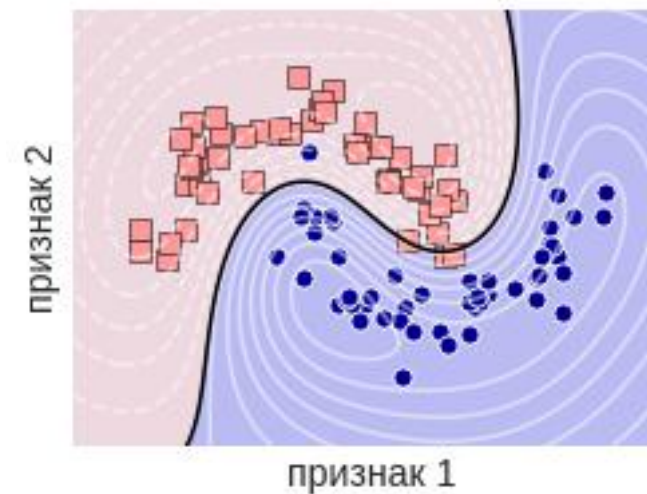
график модели



### Задача классификации

разделяющая поверхность  
(decision boundary)

+ м.б. линии уровня вероятностей



## Что значит «восстановление целевой зависимости» (меток)

**Строим «алгоритм» (гипотезу)  $a(x)$ ,  
который выдаёт предполагаемые метки**

**Формализация качества:**

**$L(y, a)$  – функция ошибки (error / loss function)**

**ошибка на объекте  $x$**

$$L(y(x), a(x))$$

**$a(x)$  – ответ нашего алгоритма  $a$**

**Примеры:**

**в задаче регрессии –  $L(y, a) = |y - a|$**

**в задаче классификации –  $L(y, a) = I[y \neq a]$**

## Что значит «восстановление целевой зависимости» (меток)

Если объекты имеют вероятностную природу, то

$$\int_{X \times Y} L(y, a(x)) \partial P(x, y) \rightarrow \min$$

теоретический риск

На практике не знаем меры

можем вычислить лишь «эмпирический риск»

Обучающая выборка (обучение – не путать с процессом)

$$X_{\text{train}} = \{(x_1, y_1), \dots, (x_m, y_m)\}$$

Ошибка на выборке (один из вариантов):

$$L(a, X_{\text{train}}) = \frac{1}{m} \sum_{i=1}^m L(y(x_i), a(x_i))$$

$$a^* = \arg \min L(a, X_{\text{train}})$$

На самом деле, интересна не ошибка на обучении (Training Error)!

## Как минимизируется ошибка

**Минимизация производится в рамках модели**

**Модель** – параметрическое семейство алгоритмов

$$A = \{a(x; w)\}_{w \in W}$$

**пример:**  $A = \{a(x; w) = w^T x : \mathbb{R}^n \rightarrow \mathbb{R}\}_{w \in \mathbb{R}^n}$

**Обучение** – определение параметров алгоритма,

как правило, производится с помощью оптимизации значения функции ошибки (функционала качества) или их модификаций на обучающей выборке

**По сути, интеллектуальный перебор алгоритмов...**

**Как – дальше!**



## Обобщающая способность (Generalization)

**Какое качество (ошибка) алгоритма на новых данных?**

$$L(a, X_{\text{train}}) \vee L(a, X_{\text{test}})$$

**Ошибка на тестовой выборке (Generalization Error / Test Error)**

более строго: матожидание ошибки на новых данных

**обучение  $\neq$  запоминание**

**потом: недообучение, переобучение, сложность...**

**потом: отложенная выборка, контроль и т.п.**

## Что такое алгоритм

Мы под этим понимаем функцию

$$a(x) : X \rightarrow Y,$$

которую можно эффективно реализовать в виде программы

1. Допускает вычисление за приемлемое время
2. Использует ограниченный набор ресурсов
3. Есть специфика, связанная с вычислениями на компьютере

## Требования к модели

- **Качество (Predictive Accuracy)**
- **Эффективность (Efficiency)**
- **Робастность (Robustness)**
- **Масштабируемость (Scalability)**
- **Интерпретируемость (Interpretability)**
- **Компактность (Compactness)**

см. выше

время обучения и использования

устойчивость к шуму/пропускам ...

использование при увеличении объёма данных

объяснение результатов модели

затраты на хранение модели

## Почему МО не оптимизация

### 1. Не знаем меру в

$$\int_{X \times Y} l(y(x), a(x)) dP \rightarrow \min$$

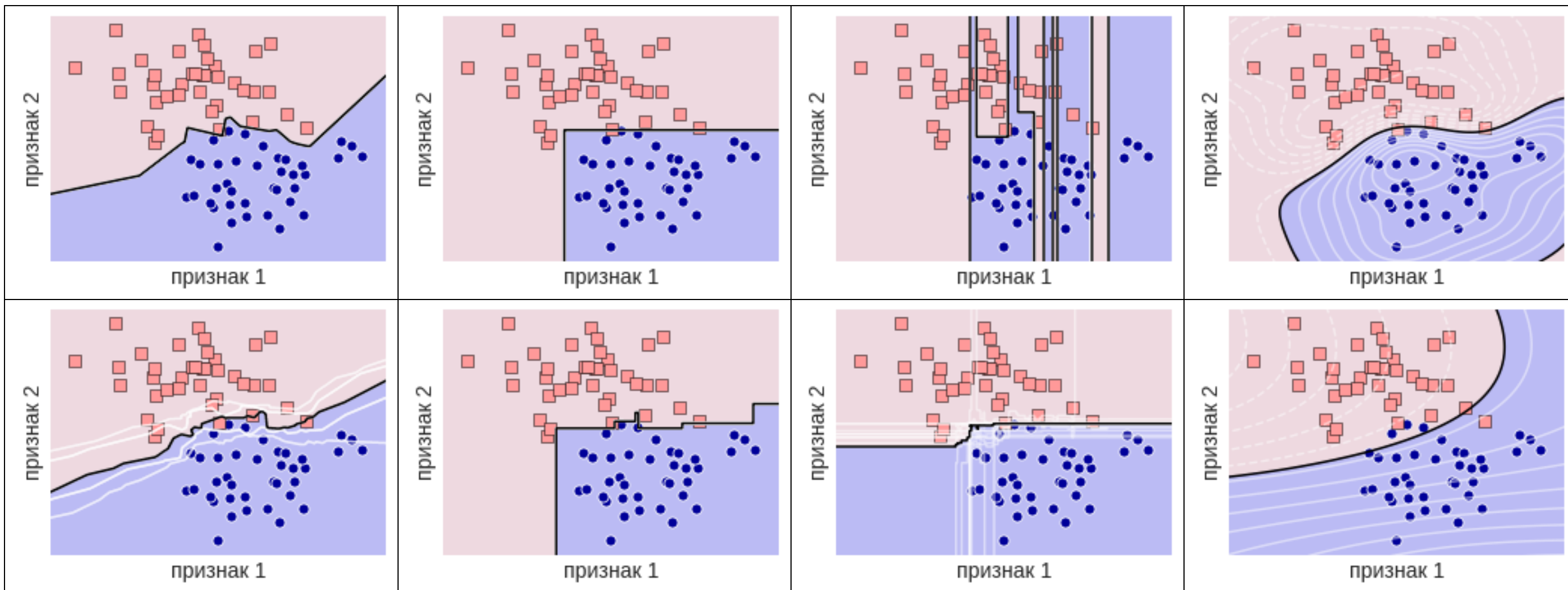
т.е. решаем «неправильную задачу оптимизации»  
и правильный выбор неправильности – особое умение  
(регуляризация, проблемно-ориентированные модели и т.п.)

### 2. Оптимизация не в классе функций, а в классе алгоритмов дополнительные требования на решение

### 3. Есть контекст

поэтому много неоптимизационных приёмов, например, аугментация

## Чем различаются алгоритмы



**Кроме сложности, ресурсоёмкости, времени работы и т.п. – функциональной  
выразимостью и геометрией решения!**

## Схема решения задачи

### 1. Уточнение и постановка задачи (Problem Definition)

понимание бизнес-задачи

### 2. Сбор, подготовка и анализ данных (Data Mining)

понимание исходных данных

сбор данных (Data collection)

предобработка данных (Data cleaning) / подготовка данных для модели

разведочный анализ (Exploratory Data Analysis)

### 2. Выбор

- Алгоритма
  - модели (Algorithm selection)
  - способа обучения: гиперпараметры, методы оптимизации (Parameter optimization)
- Контроля
  - функции ошибки (Metric selection)
  - способа контроля (разбиение train/test/valid)
- Признаков
  - генерация (Data coding = feature engineering)
  - селекция

### 3. Обучение (fit-predict)

### 4. Предсказание (м.б. Post-processing) → Проверка качества

### 5. Deploy / Release / Online evaluation / Debug / Monitoring / Maintenance (алг. переоб. на всех данных)

### 6. Отчётность, презентация, коммуникация (визуализация данных, модели, результатов)

**Машинное  
обучение  
(Machine  
Learning)**

## CRISP-DM (Cross-Industry Standard Process for Data Mining)

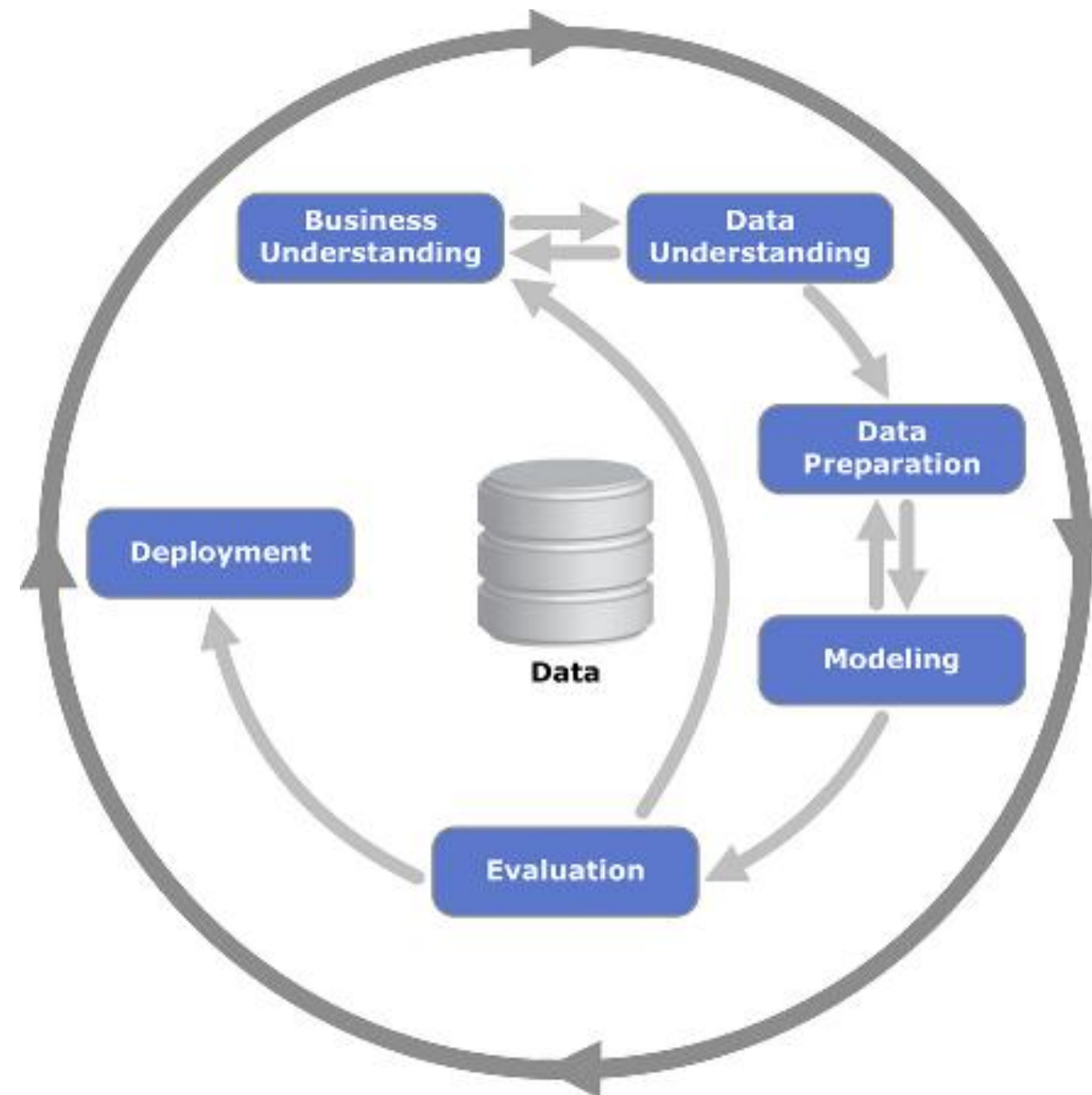
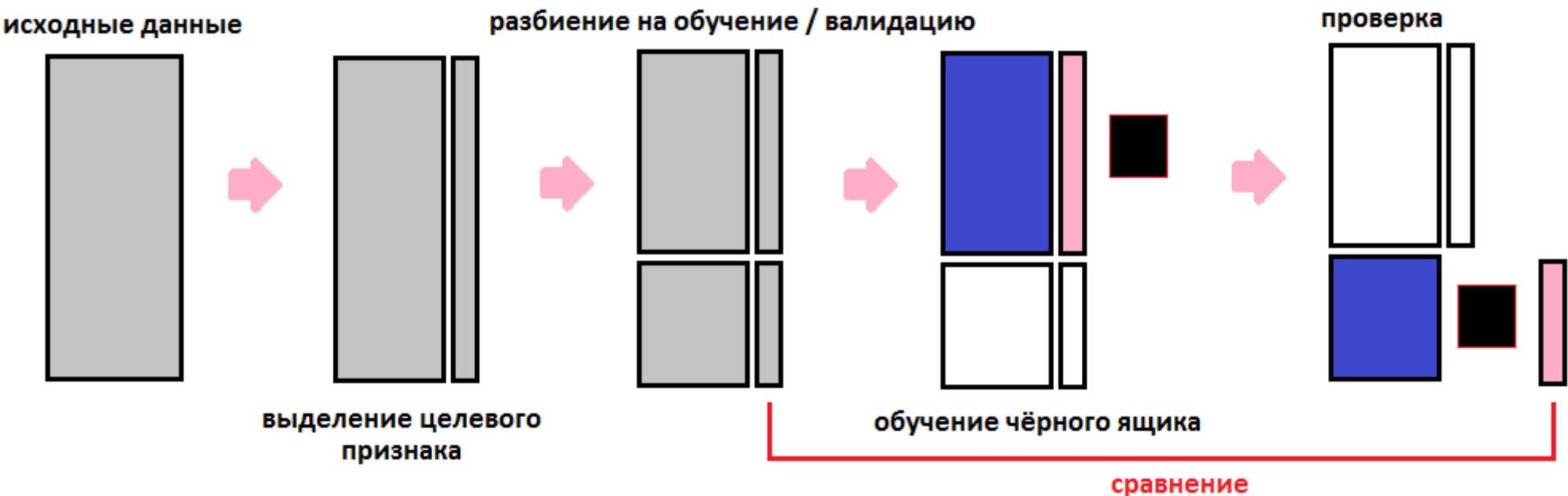


Схема проверки алгоритма





## Пример работы в Scikit-Learn

```
# данные
from sklearn.datasets import make_blobs
X, y = make_blobs(centers=2, random_state=0)

# разбивка: обучение - контроль
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

# обучение модели и предсказание
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(X_train, y_train)
prediction = classifier.predict(X_test)

# качество
print (classifier.score(X_test, y_test)) # 0.8
print (classifier.score(X_train, y_train)) # 0.93
```

## Как решаются задачи

Пусть  $y = f(X_1, \dots, X_n) + \varepsilon$

$y$  – продажи,

$X_1$  – затраты на рекламу по TV,

$X_2$  – затраты на рекламу в Интернете,

$X_3$  – затраты на рекламу на радио, и т.д.

Надеемся

$$a(X_1, \dots, X_n) \approx f(X_1, \dots, X_n),$$

$a \sim$  алгоритм (алгоритмически реализуемая функция)

Ищем в параметризованном семействе  $a \in \{a\}$  (модели)

$\varepsilon$  – неустранимая ошибка (irreducible error)

## Подход основанный на близости

$$a(x) = \text{mean}(y_i \mid x_i = x)$$

**но если в тестовой выборке нет именно таких объектов**

$$a(x) = \text{mean}(y_i \mid x_i \in N(x))$$

$N(x)$  – **окрестность (neighborhood) объекта  $x$**   
**(похожие на него объекты)**

**Параметризация может определять размер окрестности**

Но что такое окрестность при больших размерностях...

**curse of dimensionality** (след. лекция)

## Параметрические модели

### Линейная модель

$$a(X_1, \dots, X_n) = w_0 + w_1 X_1 + \dots + w_n X_n$$

**Параметры оцениваются с помощью подгонки на данных обучения**

**(fitting the model to training data)**

$$a(x_i) = y_i, i = 1, 2, \dots, m$$

$w_i$  – **веса (weights) / параметры (parameters) модели**

$w_0$  – **смещение (bias)**

**Линейная модель – простая,  
можно усложнить – полиномиальная модель.**

$$a(X_1, \dots, X_n) = w_0 + \sum_t w_t X_t + \dots + \sum_{i,j} w_{ij} X_i X_j$$

## Переобучение / переподгонка

**Чем сложнее модель, тем проще настроиться на данные,  
но возникает проблема – переобучение (overfitting) –  
качество на контроле существенно ниже чем на обучении**

**Линейная модель хорошо интерпретируемая**

- легко объяснить, как работает
- легко объяснить, почему получен такой ответ

**простая  $\Rightarrow$  надёжная**

**(оценка ошибки, как правило, соответствует действительности)**

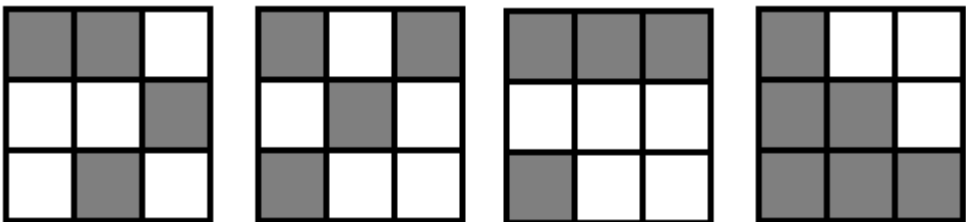
## Пример задачи машинного обучения

	x_0	x_1	x_2	x_3	x_4	y
0	1.5	7.4	2.6	5.3	0.1	3.8
1	9.2	9.0	0.3	9.6	1.4	6.2
2	2.8	6.1	9.4	8.5	0.0	6.1
3	5.2	5.5	4.9	7.7	1.6	5.2
4	7.6	0.2	1.4	1.2	3.1	3.1
5	6.7	4.7	8.2	2.9	7.3	6.5
6	7.0	3.3	3.3	9.8	6.2	4.5
7	9.5	7.7	8.3	4.1	4.5	8.5
8	4.0	10.0	1.8	9.6	4.2	5.3
9	4.2	4.6	3.7	4.7	0.4	4.2

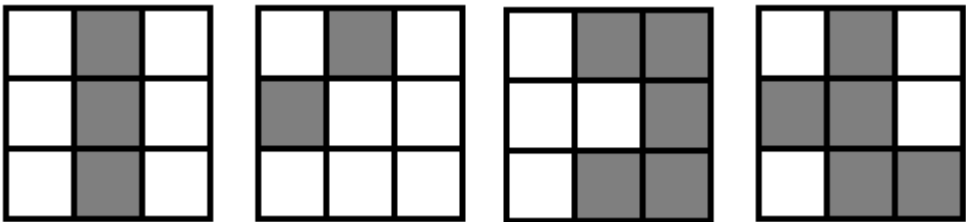
**Как зависит целевая переменная от остальных?**

Пример задачи машинного обучения

класс 1



класс 2

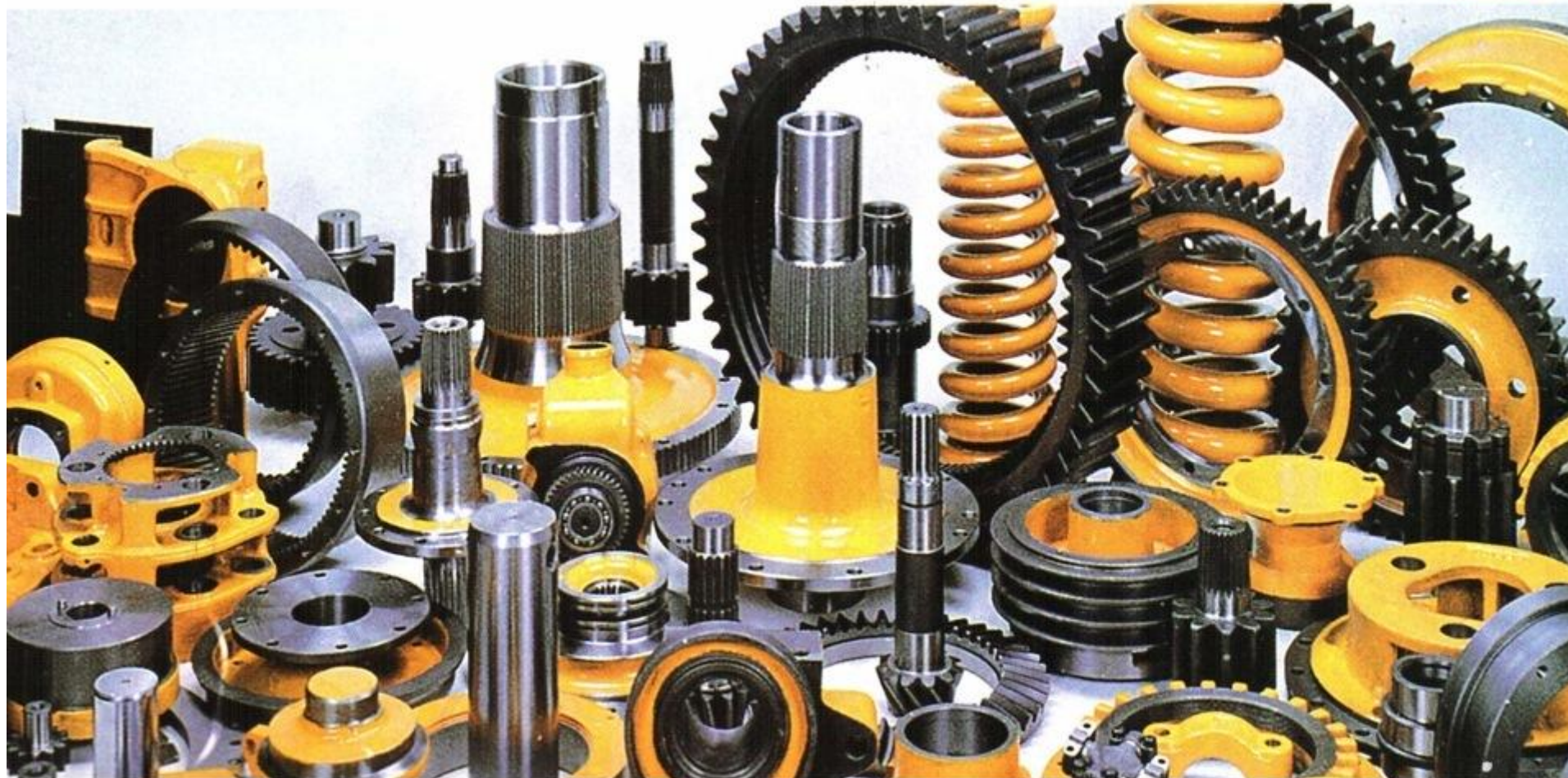


Как определяется класс?



## Другие виды обучения

## Обучение без учителя (unsupervised Learning) с неразмеченными данными, без меток



## Обучение без учителя

$$X_{\text{train}} = \{x_1, \dots, x_m\} \subseteq X$$

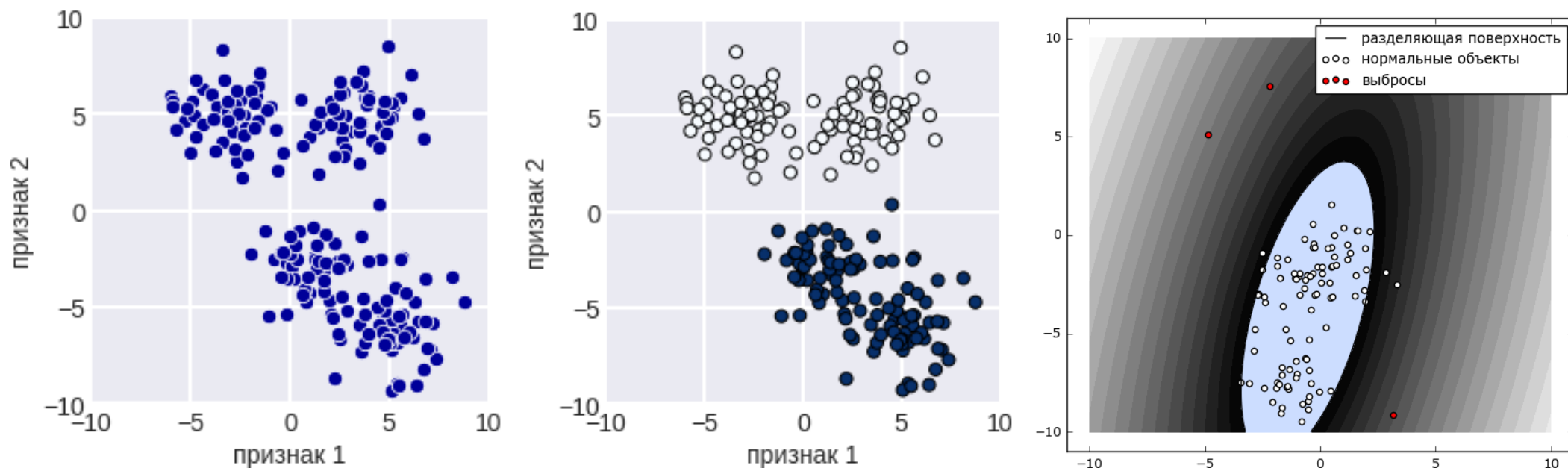
**Понять «структуру» пространства объектов  $X$**

Как на нём распределены объекты?

Можно ли его разделить на подпространства похожих объектов?

Можно ли эффективно описать объекты/пространство?

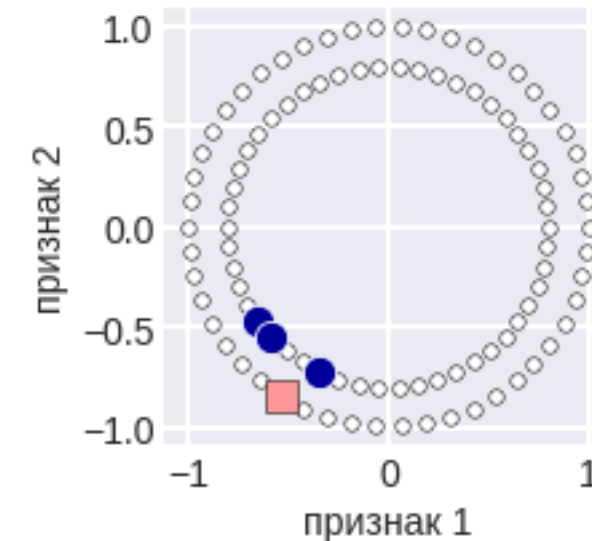
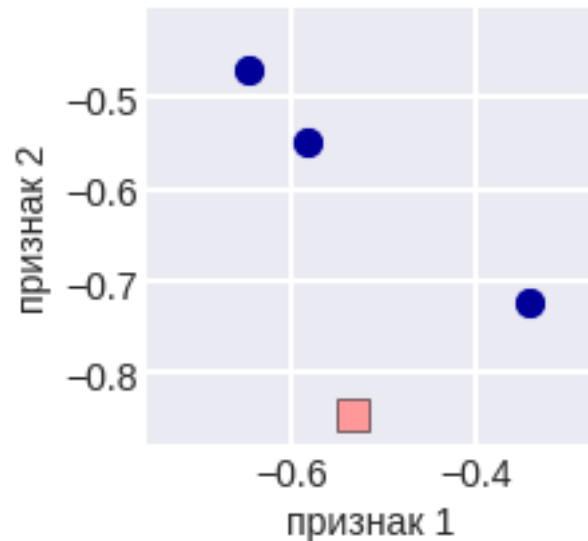
**Часто нет понимания, насколько хорошо решается задача**



## Обучение с частично размеченными данными (Semi-Supervised Learning)

$$X_{\text{train}} = \{(x_1, y_1), \dots, (x_k, y_k), x_k, \dots, x_m\}$$

Если заранее известна контрольная выборка  $x'_1, \dots, x'_q$ ,  
то это **трандуктивное обучение (transductive learning)**

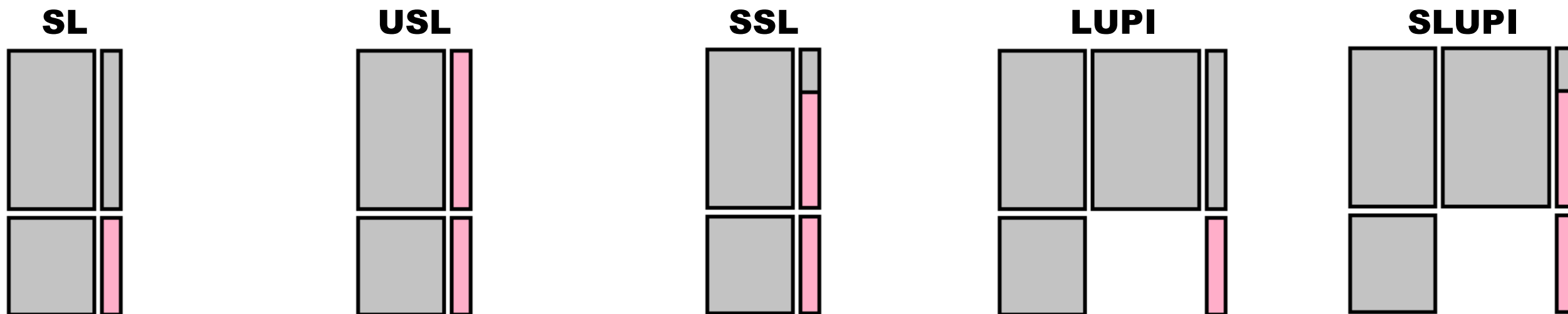


## Привилегированное обучение (Learning Using Privileged Information)

$$X_{\text{train}} = \{(x_1, \tilde{x}_1, y_1), \dots, (x_m, \tilde{x}_m, y_m)\}$$

$$X_{\text{test}} = \{x'_1, \dots, x'_q\}$$

**только на обучении есть дополнительные признаки**



Vapnik V, Vashist A. A new learning paradigm: learning using privileged information. Neural Netw. 2009 Jul-Aug;22(5-6):544-57. <https://pubmed.ncbi.nlm.nih.gov/19632812/>

## Другие виды обучения

**Обучение с подкреплением (Reinforcement Learning)**  
обучение агента, который взаимодействует со средой  
и получает награду за взаимодействие

**Структурный вывод (Structured output)**  
на выходе набор значений со связями между ними,  
примеры:

- Грамматический разбор (parsing): текст → дерево
- Аннотирование изображений (Image Captioning): изображение → текст
  - Транскрипция (Transcription):  $X \rightarrow$  текст
- Машинный перевод (Machine translation): текст → текст
  - Синтез: выборка → выборка



## **Другие виды обучения**

### **Активное обучение (Active Learning)**

**влияем на формирование обучающей выборки**

### **Онлайн-обучение (Online Learning)**

**в каждый момент времени нам доступна небольшая группа объектов (м.б. один объект)**

**~ Инкрементное обучение (incremental learning)**

**постоянное непрерывное обучение**

### **Обучение с переносом опыта (Transfer Learning)**

**решение новых задач с помощью решения старых**

### **Multitask Learning**

**решение одновременно несколько схожих задач**



## Другие виды обучения

### Обучение представлений (Representation Learning)

оптимальное представление объектов, в частности, **выучивание признаков (Feature Learning)** – автоматическое получение хороших признаков из сырых данных, **обучение многообразий (Manifold Learning)**, **матричные и тензорные разложения (Matrix and Tensor Factorization)** и т.п.

### Обучение глубоких сетей (Deep Learning)

решение задач ML с помощью глубоких нейросетей

### Мета-обучение (meta-learning)

~ обучение обучаться (например, подбор оптимальных параметров с помощью машинного обучения)

## Сложности в ML

- **переобучение – основная теоретическая проблема**

- **проблема формализации**

надо переформулировать бизнес-задачу в математическую задачу выявления зависимости, выбор адекватного функционала качества

- **размеры данных**

много объектов (низкого уровня – транзакций,  
высокого – клиентов)  
много признаков (обработка текстов)

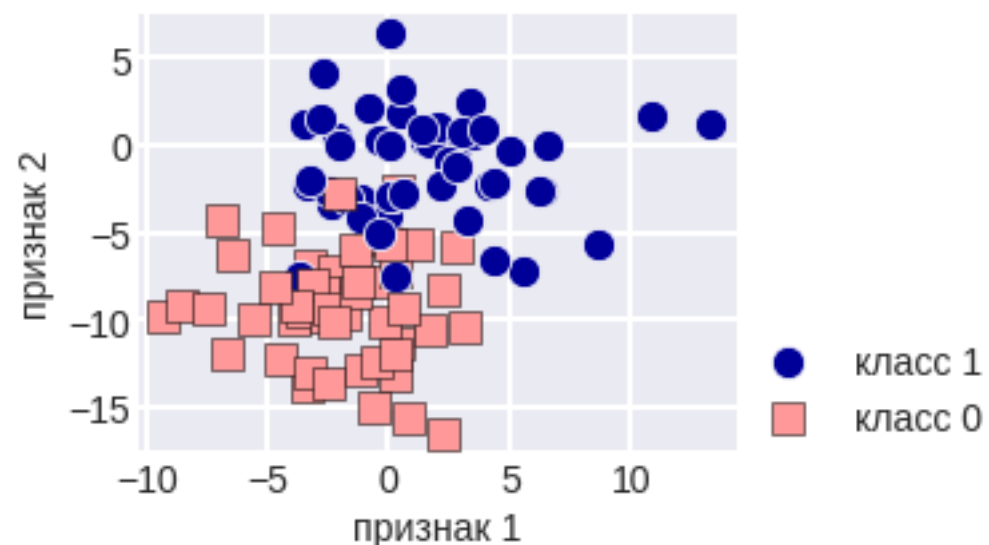
- **качество данных**

невыполнение всех свойств (полнота, корректность, правдивость, ясность и т.п.)

- **несоответствие обучения и контроля**

это больше, чем проблема репрезентативности выборки – это проблема прогноза / адаптации (распознавание голоса, спама)

## Примеры модельных задач

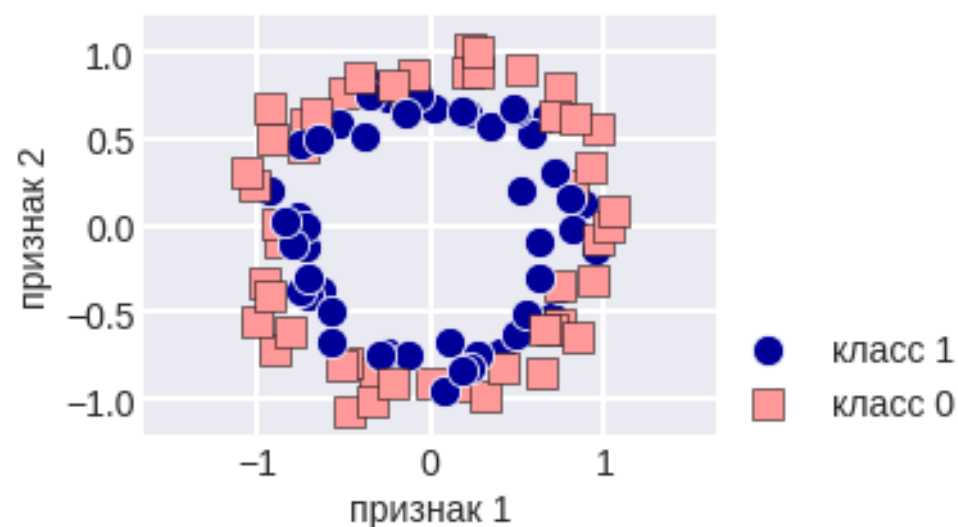


## «Кучки»

```
from sklearn.datasets import make_blobs
```

```
X, y = make_blobs(centers=2,  
                  random_state=2)
```

```
plt.scatter(X[:, 0], X[:, 1],  
           c=y, s=75)
```

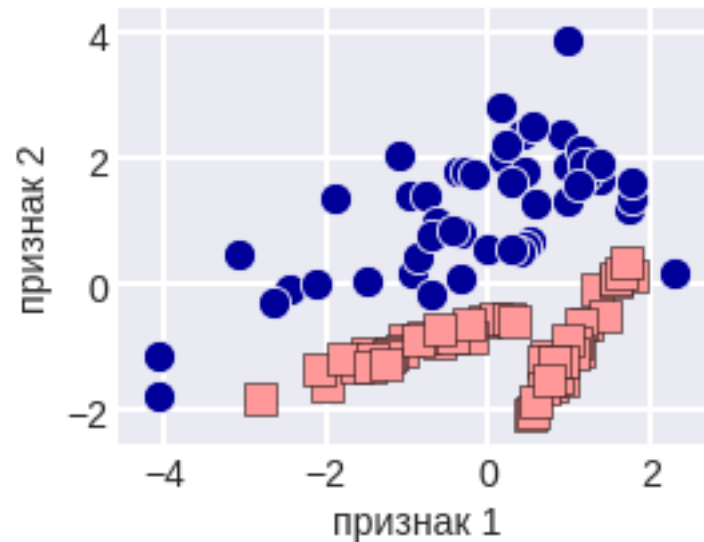


## «Кольца»

```
from sklearn.datasets import  
make_circles
```

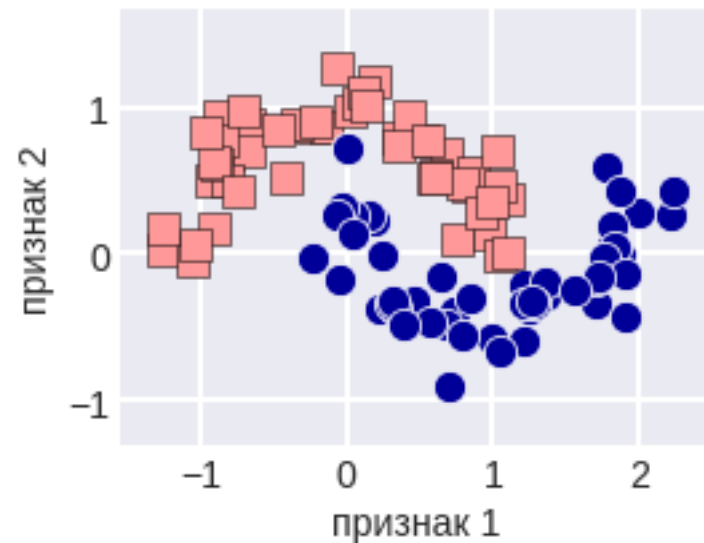
```
X, y = make_circles(noise=0.1,  
                   random_state=1)
```

## Примеры модельных задач



## «Классификация»

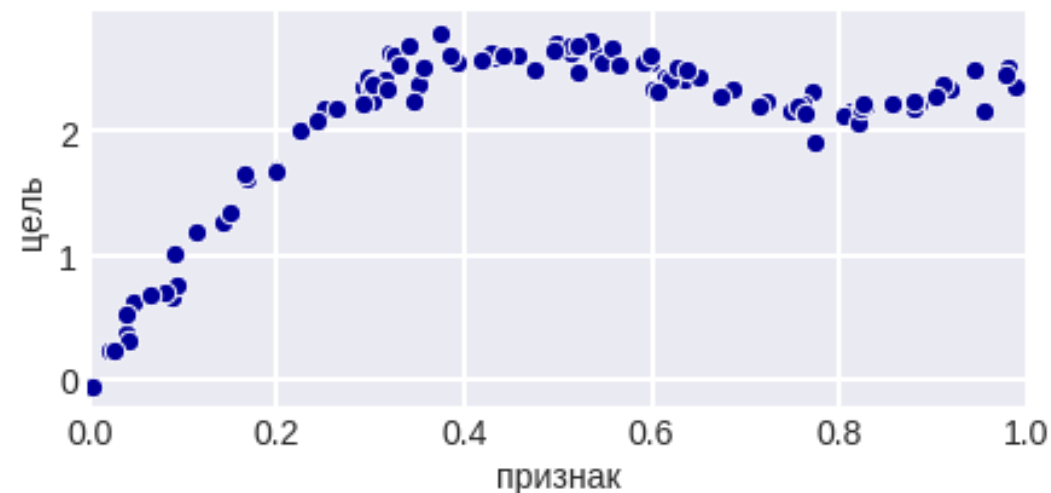
```
make_classification(n_samples=100,  
                  n_features=2,  
                  n_informative=2,  
                  n_redundant=0,  
                  n_repeated=0,  
                  n_clusters_per_class=2)
```



## Два месяца

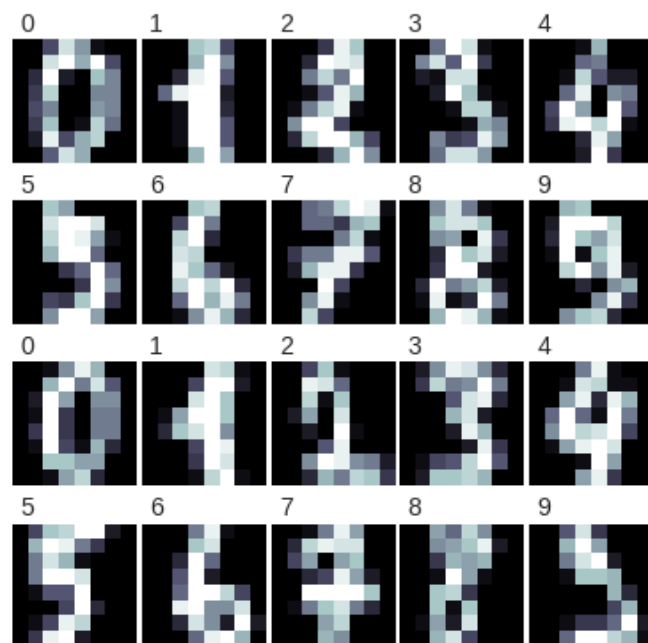
```
make_moons(n_samples=100,  
           noise=0.15,  
           random_state=1)
```

## Ручная генерация данных



```
n_samples = 100
np.random.seed(10)
X = np.random.rand(n_samples)
Y = np.sin(5 * X) + 5 * np.log1p(X) + 0.1 *
    np.random.randn(n_samples)
```

## Классические датасеты



```
from sklearn.datasets import load_digits
digits = load_digits()
X_digits, y_digits = digits.data, digits.target
```

## Итог

**Обучение с учителем – восстановление целевой зависимости  
формализуется с помощью функции ошибки**

**Объекты произвольны, но мы дальше рассматриваем признаковые описания**

**Минимизируем эмпирический риск в рамках модели,  
необходима обобщающая способность**

**В отличие от оптимизации ответ – алгоритм  
(много дополнительных требований)**

**Схемы решений задач вполне естественны,  
алгоритмы просты, например линейный**

**Есть много видов машинного обучения,  
начнём с обучения по размеченным данным**

## Ссылки

**Trevor Hastie, Robert Tibshirani, Jerome Friedman «The Elements of Statistical Learning: Data Mining, Inference, and Prediction», 2nd Edition, Springer, 2009 //**

**<https://web.stanford.edu/~hastie/Papers/ESLII.pdf>**

### **Лекции К.В. Воронцова**

**[https://www.machinelearning.ru/wiki/index.php?title=Машинное\\_обучение\\_\(курс\\_лекций%2С\\_К.В.Воронцов\)](https://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2С_К.В.Воронцов))**

**Andrew Glassner «Deep Learning, Vol. 1-2: From Basics to Practice» //**

**<http://www.glassner.com/portfolio/deep-learning-from-basics-to-practice/>**

**использована**

- **лекция «Библиотека языка Питон Scikit-Learn»**

**[https://github.com/Dyakonov/IML/blob/master/IML2018\\_06\\_scikitlearn\\_10.pdf](https://github.com/Dyakonov/IML/blob/master/IML2018_06_scikitlearn_10.pdf)**