



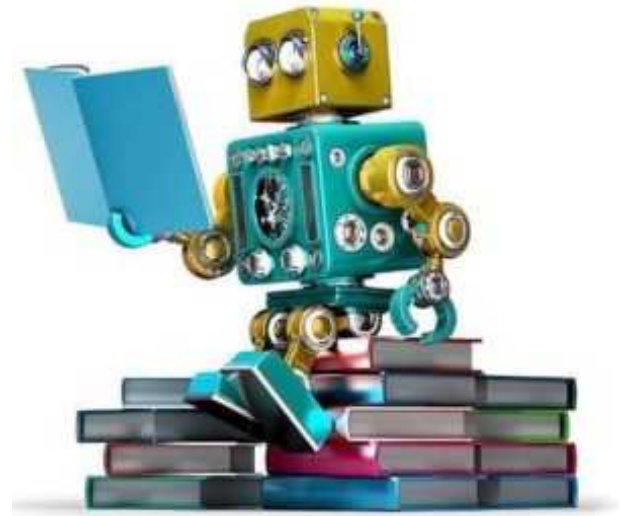
# Методы машинного обучения

*Солодовников Владимир Игоревич*

*к.т.н, каф. ИУ-7*

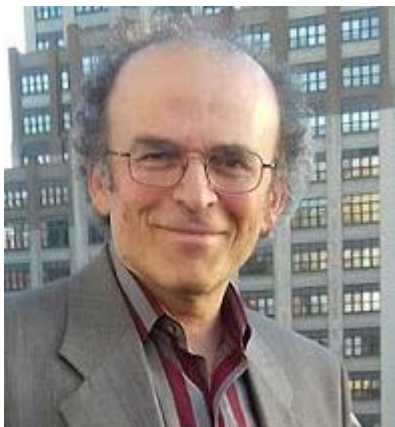
# Основные понятия, цели, задачи, методы

- Постановка задачи обучения;
- Типы и примеры задач;
- Открытые наборы данных для обучения;
- Предобработка (подготовка) исходных данных;
- Классификация методов машинного обучения;
- Выбор метода и настройка гиперпараметров;
- Алгоритмы обучения моделей;
- Метрики оценки качества полученной модели;
- Интерпретация полученного результата.



# Knowledge Discovery in Databases (KDD)

## Обнаружение знаний в базах данных



Основоположниками концепции KDD считаются Пятецкий-Шапиро (*Gregory I. Piatetsky-Shapiro* - слева) и Усама М. Файад (*Usama M. Fayyad* - справа).

*Data scientists and the co-founders of the KDD conferences and the Association for Computing Machinery SIGKDD group for Knowledge Discovery, Data Mining and Data Science.*



**KDD** представляет собой нетривиальный процесс обнаружения корректных, новых, потенциально полезных и интерпретируемых шаблонов в больших массивах данных.

Данные - множество фактов предметной области, представленных в виде записей базы данных.

Шаблон — это выражение на некотором языке, описывающее подмножество данных или применяемую к нему модель, т.е. поиск шаблонов подразумевает подгонку моделей к данным, обнаружение в них зависимостей, закономерностей и структур.

KDD не предписывает, какие методы и алгоритмы обработки следует использовать при решении конкретной задачи, а определяет последовательность действий, которую необходимо выполнить для того, чтобы из исходных данных получить знания. Этот подход универсальный и не зависит от предметной области.

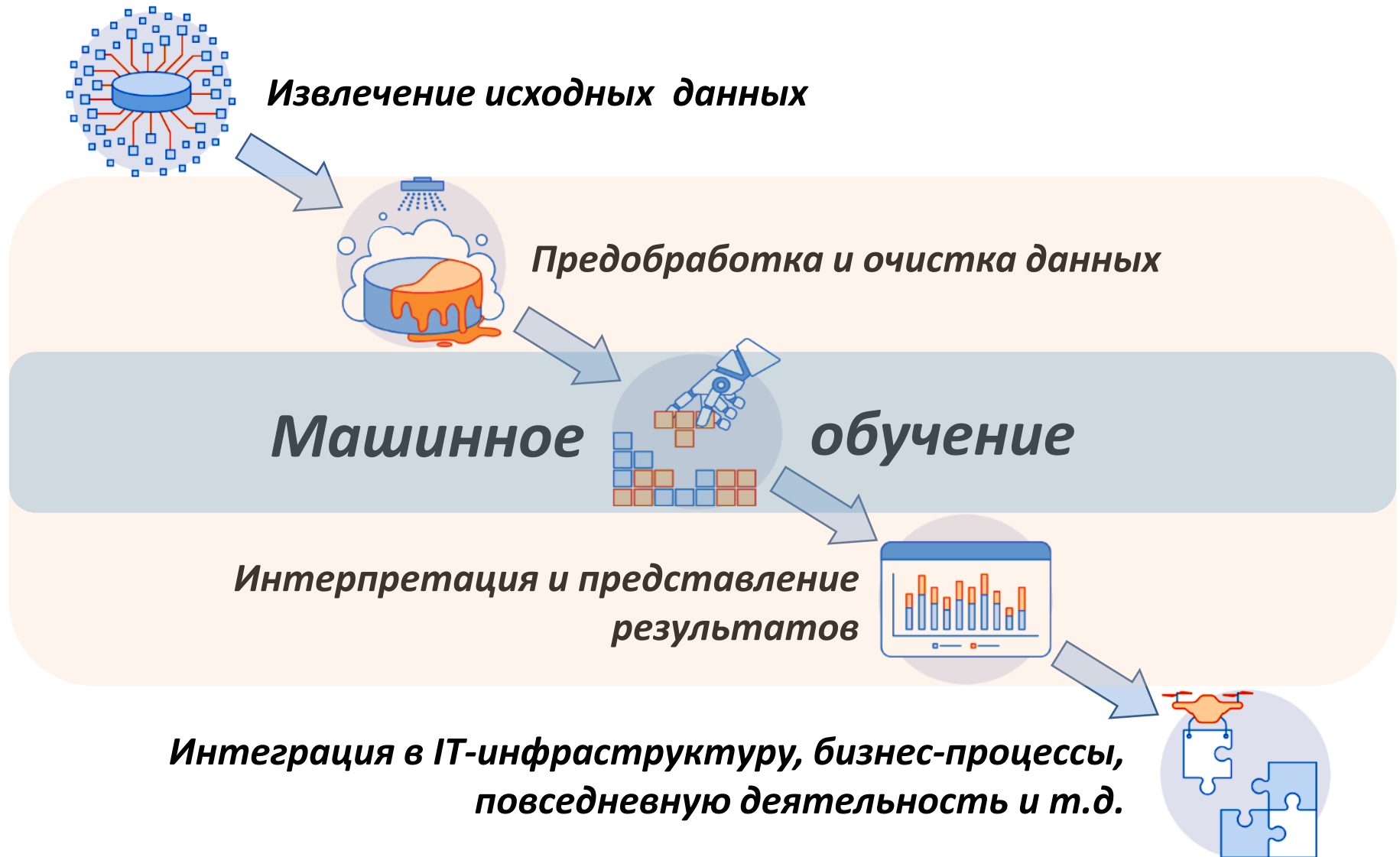
# Интеллектуальный анализ данных (*Data Mining*)

Направлен на обнаружение в исходных данных предварительно неизвестных, практически полезных и доступных интерпретации знаний, шаблонов, закономерностей, отражающих фрагменты многоаспектных взаимоотношений в данных. Особенностью является отсутствие ограничительных рамок априорных предположений о структуре выборки и виде распределений значений анализируемых показателей.



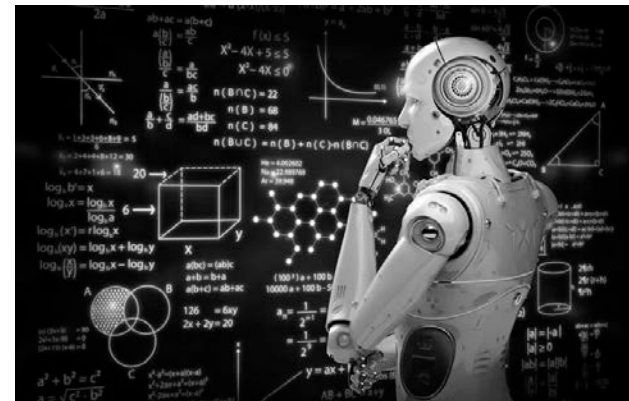
Машинное обучение — это направление искусственного интеллекта, связанное с разработкой и построением аналитических моделей, которые способны автоматически обнаружить в данных скрытые и ранее неизвестные закономерности, а также самостоятельно приобретать свойства, необходимые для распознавания этих закономерностей.

# Процесс интеллектуального анализа данных



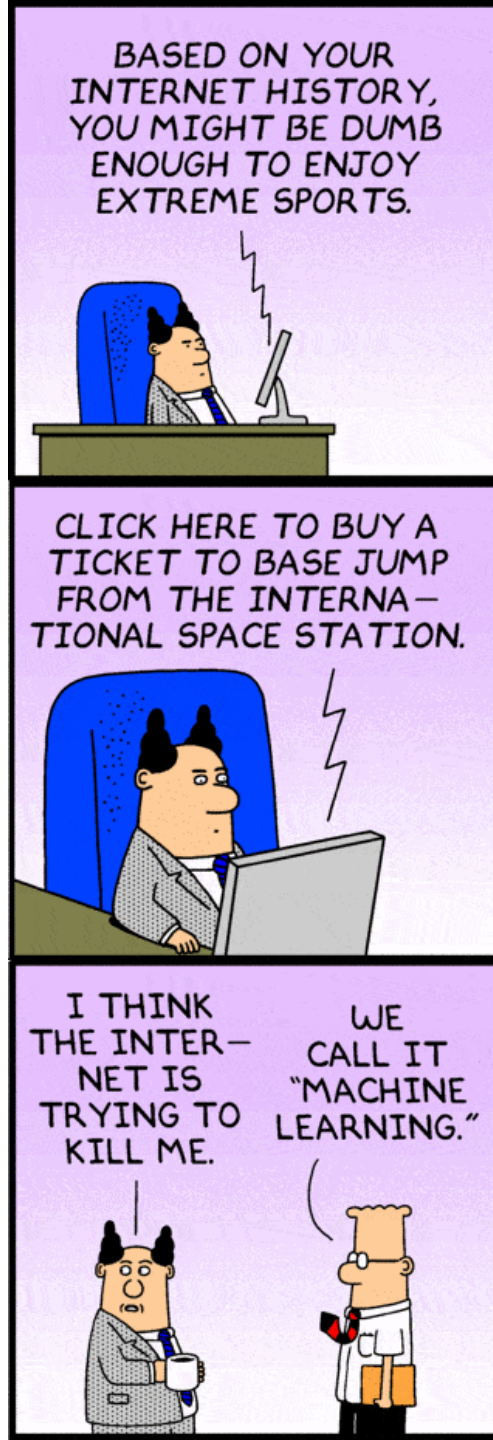
# Классические задачи, решаемые с помощью машинного обучения

- Классификация (*classification*)
- Кластеризация (*clustering/cluster analysis*)
- Регрессия (*regression*)
- Прогнозирование (*forecasting*)
- Поиск субоптимальных решений или стратегий
- Понижение размерности (*dimensionality reduction*)
- Визуализация данных (*data visualization*)
- Восстановление плотности распределения вероятности
- Поиск ассоциативных правил (*association rules learning*)
- Обнаружение аномалий / фильтрация выбросов (*outliers detection*)
- Одноклассовая классификация / Идентификация



# Основные сферы применения

1. Медицинская диагностика.
2. Техника:
  - 2.1. Автоматизация и управление.
  - 2.2. Техническая диагностика.
  - 2.3. Робототехника.
  - 2.4. Компьютерное зрение.
  - 2.5. Распознавание речи.
3. Экономика:
  - 3.1. Кредитный скоринг.
  - 3.2. Предсказание ухода клиентов .
  - 3.3. Обнаружение мошенничества.
  - 3.4. Биржевой технический анализ.
  - 3.5. Биржевой надзор.
4. Офисная автоматизация:
  - 4.1. Распознавание текста.
  - 4.2. Обнаружение спама.
  - 4.3. Категоризация документов.
  - 4.4. Распознавание рукописного ввода.





# Машинное обучение - Machine Learning(ML)

Множество математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных. Процесс поиска закономерностей называют обучением.

Различают два типа обучения:

- **Обучение по прецедентам**, или **индуктивное обучение** - основано на выявлении эмпирических закономерностей в данных.
- **Дедуктивное обучение** - предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний.

**Особенность:** Машинное обучение — не только математическая, но и практическая, инженерная дисциплина, имеющая собственную специфику, связанную с проблемами вычислительной эффективности и переобучения.





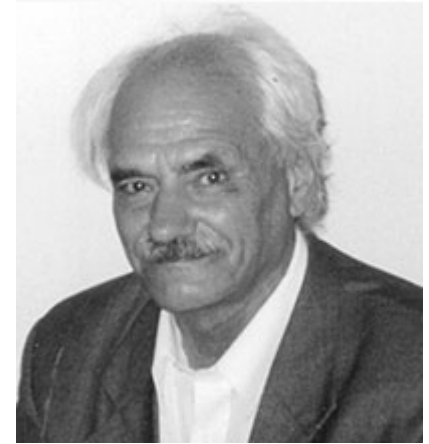
# Теория вычислительного обучения

## (Computational Learning Theory, COLT)



Вапник  
Владимир Наумович

Работы Вапника и Червоненкиса по статистической теории **восстановления зависимостей по эмпирическим данным** в конце 60-х — начале 70-х послужили основой для создания теории вычислительного обучения (*Computational Learning Theory - COLT*), которая изучает методы построения и анализа алгоритмов, обучаемых по прецедентам.



Червоненкис  
Алексей Яковлевич

Основная задача теории вычислительного обучения — дать строгие обоснования алгоритмов обучения по прецедентам. Теория COLT претендует на роль теоретического базиса всего машинного обучения.

**Эмпирические данные** (от др.-греч. *εμπειρία* [*empeiría*] «опыт») — данные, полученные через органы чувств, в частности, путём наблюдения или эксперимента. В философии после Канта полученное таким образом знание принято называть **апостериорным**. Оно противопоставляется **априорному**, доопытному знанию, доступному через чисто умозрительное мышление.

# Обучение по прецедентам (индуктивное обучение)

Дано конечное множество **прецедентов** (объектов)  $\{X_1, \dots, X_m\} \subset X$ , по каждому из которых собраны некоторые данные (описание), а также соответствующие им выходные значения (результаты) из множества допустимые ответов  $Y$ .

Совокупность всех имеющихся описаний прецедентов называется **обучающей выборкой (training sample)**, которая обычно является случайной выборкой объектов из генеральной совокупности  $\Omega$ .

Обучающая выборка имеет вид  $S^m = \{ (X_1, Y_1), \dots, (X_m, Y_m) \}$ , где:

- пары «объект–ответ»  $(X_i, Y_i)$  – прецедент  $S_i^m$  из выборки  $S^m$ ;
- $X_i$  – описание  $i$ -ого объекта из  $S^m$ ,  $i = 1, \dots, m$ ;
- $Y_i$  – значение переменной  $Y$  для  $i$ -ого объекта из  $S^m$ ,  $i = 1, \dots, m$ ;
- $m$  – число объектов в  $S^m$ .

Существует неизвестная **целевая функция (Target function)**  $t^*: X \rightarrow Y$ , значения которой известны только для конечного множества объектов обучающей выборки.

Необходимо построить алгоритм, который выдаст достаточно точный результат (выходное значение) для любых возможных входов (наборов значений признаков), в том числе таких, которые ещё не наблюдались.

# Постановка задачи обучения

**Дано:**

- Обучающая выборка  $S^m = \{ (X_1, Y_1), \dots, (X_m, Y_m) \}$ ;
- Предполагается, что существует некоторая неизвестная зависимость  $t^*: X \rightarrow Y$  (целевая функция).

**Задача обучения по прецедентам:**

По обучающей выборке  $S^m$  построить некоторую **решающую функцию (decision function)**  $a: X \rightarrow Y$ , которая приближала бы целевую функцию  $t^*$ , причём не только на объектах обучающей выборки, но и на всём множестве возможных объектов генеральной совокупности  $\Omega$ .

# Признаковое описание объектов (*feature vector*)

**Признак (*feature*)** — результат измерения некоторой характеристики объекта, т.е. отображение:  $f: X \rightarrow D_f$ , где  $D_f$  — множество допустимых значений признака.

Для всех прецедентов (объектов) выборки фиксируется совокупность из  $n$  признаков. Если для объекта  $X_i$  заданы признаки  $f_1, \dots, f_n$ , то вектор  $(f_1(X_i), \dots, f_n(X_i)) = (x_i^1, \dots, x_i^n)$  называется признаковым описанием объекта  $X_i \in X$ .

**Признаковое описание объекта (*feature vector*)** — это вектор, который составлен из значений, соответствующих фиксированному набору признаков (характеристик) для данного объекта.

В машинном обучении признаковые описания допустимо отождествлять с самими объектами. При этом все множество  $X$  называют признаковым пространством.

Матрицей объектов-признаков (матрицей/таблицей исходных данных) называется совокупность признаковых описаний объектов обучающей выборки  $S^m$  длины  $m$ , записанной в виде матрицы размера  $m \times n$  ( $m$  строк,  $n$  столбцов). Столбцы соответствуют признакам, строки — признаковым описаниям объектов.

	$f_1$	...	$f_n$
$X_1$	$x_1^1$	...	$x_1^n$
...	...	...	...
$X_m$	$x_m^1$	...	$x_m^n$

# Типы признаков (Атрибутов)

Значения в исходном признаковом описании **прецедентов** (объектов) могут подразделяться на **количественные** и **качественные**.

**Количественным (quantitative)** называется признак, который имеет числовое представление и они могут быть:

- дискретными (discrete data) - выражаемые ограниченным набором значений (обычно целыми числами)
- непрерывными (cantinuaes data) - принимающие значения на непрерывной шкале значений.

**Качественные (attribute, qualitative)** признаки выражаются нечисловыми значениями и подразделяются на:

- альтернативные (бинарные) – имеют только два варианта значений.
- атрибутивные (неупорядоченные/категориальные) - имеет более двух вариантов, которые при этом выражаются в виде понятий или наименований.
- порядковые (ординальные) - имеют несколько ранжированных, т.е. упорядоченных по возрастанию или убыванию, качественных вариантов.

В результате, после предобработки и кодирования, в зависимости от множества допустимых значений  $D_f$  признаки делятся на следующие типы:

- бинарный признак:  $D_f = \{0, 1\}$ ;
- номинальный признак:  $D_f$  — конечное множество;
- порядковый признак:  $D_f$  — конечное упорядоченное множество;
- количественный признак:  $D_f$  — множество действительных чисел.

# Модель алгоритмов

Любая дисциплина, использующая математический аппарат, так или иначе, занимается математическим моделированием – заменой реального объекта его абстрактным, идеализированным представлением и использованием полученного представления для изучения объекта и добычи знаний.

**Моделью алгоритмов (predictive model)** называется параметрическое семейство функций (отображений), аппроксимирующих связь входных данных  $X$  и выходных данных  $Y$  на обучающей выборке:

$$A = \{g(X, \theta) | \theta \in \Theta\}$$

где  $g: X \times \Theta \rightarrow Y$  – некоторая фиксированная функция,  $\Theta$  – множество допустимых значений набора параметров  $\theta$ , называемое пространством параметров или пространством поиска (search space). При построении моделей элементы множества  $\theta$ , как правило, заранее неизвестны и требуют нахождения.

Типы математических моделей в зависимости от:

- Математического вида функции - линейные / нелинейные.
- Количества переменных - сосредоточенные / распределенные.
- Присутствия случайности - детерминированные / стохастические.
- Изменчивости во времени – статические / динамические.
- Используемых параметров и переменных - дискретные / непрерывные.

# Метод обучения модели (learning algorithm)

Метод обучения модели - это отображение  $\mu: S^m \rightarrow A$ , которое произвольной конечной выборке  $S^m = \{ (X_1, Y_1), \dots, (X_m, Y_m) \}$  ставит в соответствие некоторый алгоритм  $a \in A$ . Говорят также, что метод  $\mu$  строит алгоритм  $a$  по выборке  $S^m$ .

Пусть  $\lambda[Y_i, a(X_i)]$  – величина/функция «потерь» (loss function), произошедших в результате использования  $a(X_i)$  в качестве прогноза значения  $Y$ . Классический метод обучения (empirical risk minimization, ERM), заключается в том, чтобы найти в заданной модели  $A$  алгоритм  $a$ , который позволит **минимизировать функционал эмпирического риска (функционал качества)** на обучающей выборке:

$$Q(S^m, a) = \frac{1}{m} \sum_{i=1}^m \lambda[Y_i, a(X_i)] \rightarrow \min_{a \in A}$$

Примеры функции потерь, при  $Y \subseteq \mathbb{R}$ :

- $\lambda[Y_i, a(X_i)] = [a(X_i) \neq t^*(X_i)]$  - индикатор ошибки;
- $\lambda[Y_i, a(X_i)] = |a(X_i) - t^*(X_i)|$  - модуль величины ошибки;  
функционал  $Q$  называется средней ошибкой;
- $\lambda[Y_i, a(X_i)] = (a(X_i) - t^*(X_i))^2$  - квадратичная функция потерь;  
функционал  $Q$  называется средней квадратичной ошибкой.



# Свойства алгоритма обучения

Алгоритм обучения принимает на входе конечную обучающую выборку прецедентов и настраивает модель. Настроенная (обученная) модель затем используется для предсказания будущих прецедентов. Алгоритм должен обладать свойством **обучаемости** в следующих двух смыслах.

- Во-первых, алгоритм обучения должен обладать способностью к **обобщению** данных. Построенная им модель должна выдавать в среднем достаточно точные предсказания будущих прецедентов, т.е. обобщение определяет адекватный отклик на данные, выходящие за пределы имеющейся обучающей выборки. Оценки обобщающей способности, как правило, основываются на гипотезе, что прошлые и будущее прецеденты поступают случайно и независимо из одного и того же неизвестного вероятностного распределения. Эта гипотеза позволяет применить статистические методы для получения верхних оценок ожидаемой в будущем ошибки.
- Во-вторых, процесс обучения должен завершиться за приемлемое время. Обычно исследуется вопрос, является ли время обучения модели полиномиальным или экспоненциальным по длине выборки. Таким образом, проблематика вычислительного обучения тесно связана также и с вопросами **вычислительной сложности** алгоритмов.

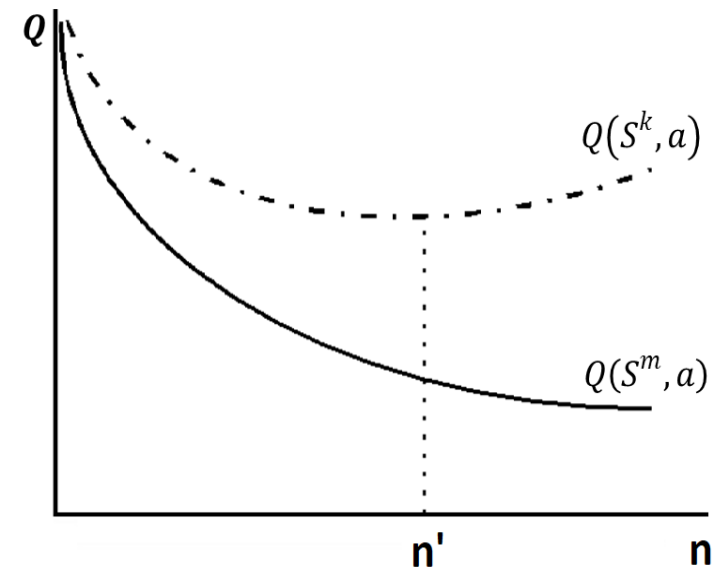
# Обобщающая способность (*generalization ability*)

Обобщающая способность – это свойство модели отражать исходные данные в требуемые результаты ( $X \rightarrow Y$ ) на всем множестве возможных объектов генеральной совокупности (во всех сценариях, а не только на тренировочных примерах).

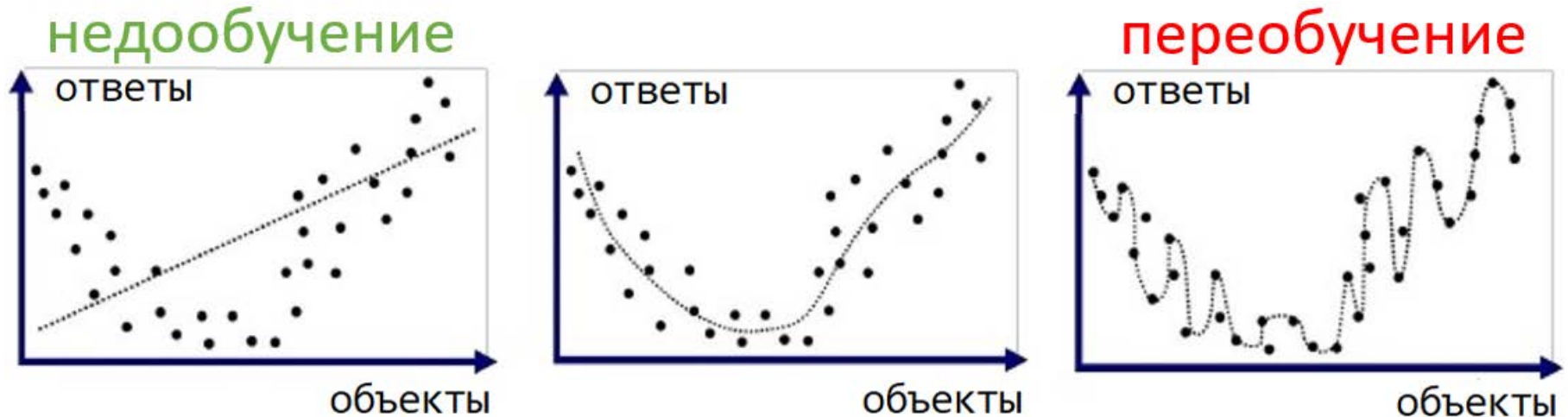
Если минимум функционала эмпирического риска (функционал качества)  $Q(S^m, a)$  достигается на алгоритме  $a$ , то это не гарантирует, что  $a$  хорошо приближает целевую зависимость на произвольной контрольной выборке  $S^k = (X'_i, Y'_i)_{i=1}^k$ . **Обобщающая способность (*generalization ability*)** метода  $\mu$  характеризуется величиной  $Q(S^k, a) = Q(S^k, \mu(S^m))$  при условии, что выборки  $S^k$  и  $S^m$  являются представительными.

Метод обучения  $\mu$  называется состоятельным, если при заданных достаточно малых значениях  $\varepsilon$  и  $\eta$  справедливо для любых простых выборок  $S^k$  и  $S^m$  оценка  $Q(S^k, \mu(S^m)) \leq \varepsilon$  справедлива с вероятностью не менее  $1 - \eta$ .

Когда качество работы алгоритма на новых объектах, не вошедших в состав обучения, оказывается существенно хуже, чем на обучающей выборке, говорят об эффекте переобучения (overtraining) или переподгонки (overfitting).



# Проблема недообучения и переобучения



**Недообучение** (underfitting) возникает в том случае, когда модель слишком проста и содержит недостаточное число параметров  $n$ .

**Переобучение** (overfitting) возникает в том случае, когда модель слишком сложная и содержит избыточное число параметров  $n$ .

Как бороться:

- Уменьшить число настраиваемых параметров модели;
- По возможности увеличить число обучающих примеров;
- Уменьшить число итераций алгоритма обучения;
- Использовать эмпирические оценки обобщающей способности.

# Эмпирические оценки обобщающей способности

Пусть дана выборка  $S^m = \{(X_1, Y_1), \dots, (X_m, Y_m)\}$ . Разобьём её  $N$  различными способами на две непересекающиеся подвыборки - обучающую  $S_n^\ell$  длины  $\ell$  и контрольную  $S_n^k$  длины  $k = m - \ell$ . Для каждого разбиения  $n=1, \dots, N$  построим алгоритм  $a_n = \mu(S_n^\ell)$  и вычислим значение функционала эмпирического риска  $Q_n = Q(S_n^k, a_n)$ .

Среднее арифметическое значений  $Q_n$  по всем разбиениям называется оценкой скользящего контроля (**cross-validation**, CV):

$$CV(\mu, S^L) = \frac{1}{N} \sum_{n=1}^N Q(S_n^k, \mu(S_n^\ell))$$

Стандартом «де факто» считается методика  $t \times q$ -кратного скользящего контроля ( **$t \times q$ -fold cross-validation**), когда выборка случайным образом разбивается на  $q$  блоков равной (или почти равной) длины, каждый блок по очереди становится контрольной выборкой, а объединение всех остальных блоков – обучающей выборкой. Выборка  $S^m$  по-разному  $t$  раз разбивается на  $q$  блоков. Итого получается  $N = t \times q$  разбиений.

# Основные проблемные вопросы машинного обучения

Опираясь на понятие обобщающей способности, можно выделить следующие основные проблемные вопросы машинного обучения:

- Достаточно ли данных для нахождения в них полезных знаний?
- Может ли в принципе данная модель обучиться на имеющихся данных?
- Будет ли полученная модель приближать требуемый «закон природы» на всем возможном множестве  $\Omega$ ?
- Насколько хорошо будет работать обученная модель или насколько часто и насколько сильно будет ошибаться модель на реальных (контрольных) данных?

# Общий порядок действий при обучении по прецедентам

- Анализ постановки задачи и исходных данных;
- Формулировка решения на математическом языке (задача формализуема, а результаты работы модели могут быть проверены);
- Предобработка данных и выделение ключевых признаков;
- Выбирается и фиксируется модель восстанавливаемой зависимости;
- Вводится функционал качества, значение которого показывает, насколько хорошо модель описывает наблюдаемые данные;
- Алгоритм обучения (learning algorithm) ищет такой набор параметров модели, при котором функционал качества на заданной обучающей выборке принимает оптимальное значение;
  - Процесс настройки (fitting) модели по выборке данных в большинстве случаев сводится к применению численных методов оптимизации;
- Эксплуатация модели при достижении требуемого качества, либо возврат к одному из предыдущих шагов (перенастройка модели, добыча новых данных и т. п.).

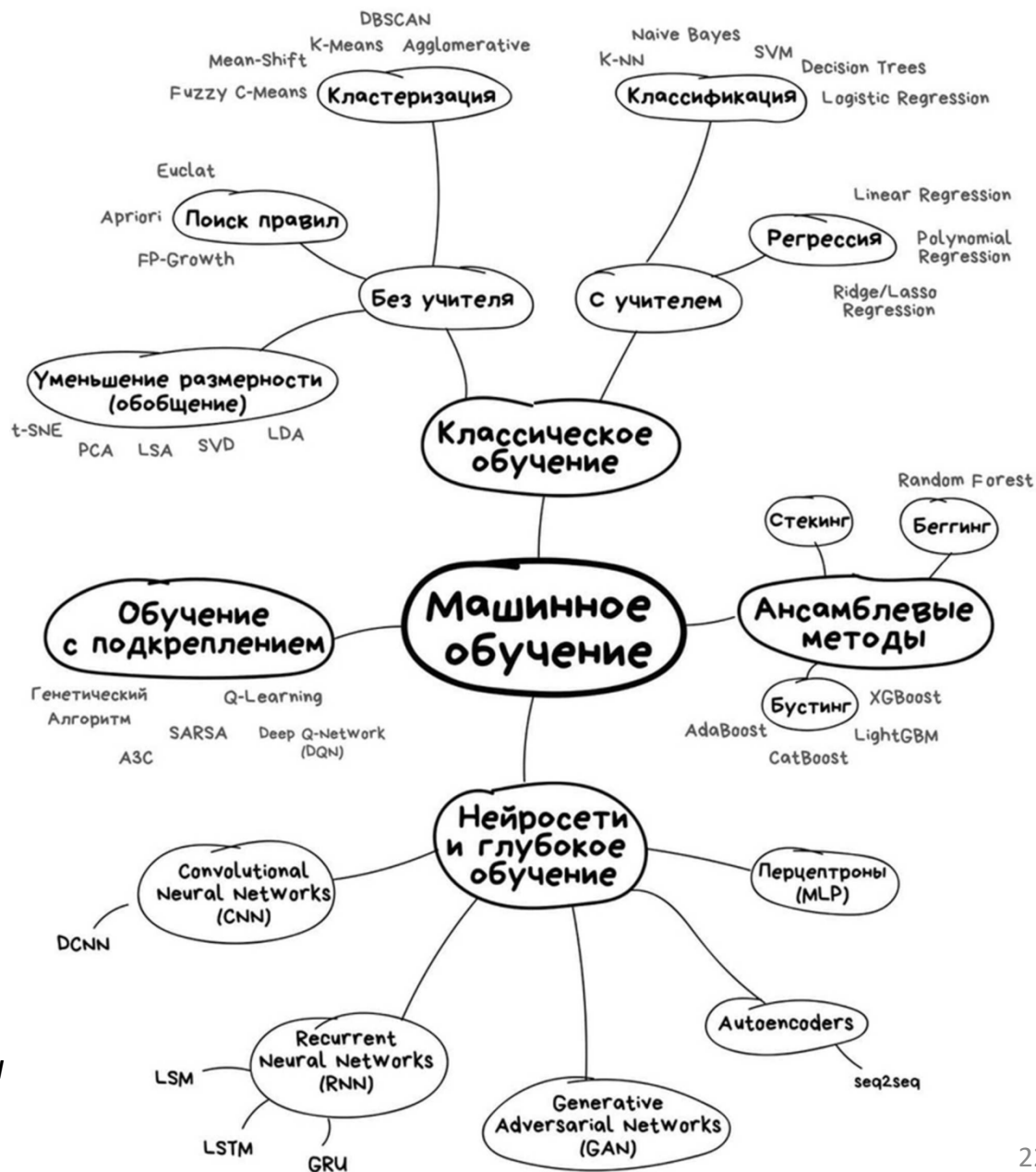
Можно сказать, что машинное обучение реализует подход **Case Based Reasoning (CBR)** — метод решения проблем рассуждением по аналогии, путем предположения на основе подобных случаев (прецедентов).

# Типы машинного обучения

- Классическое обучение:
  - Обучение с учителем (supervised learning)
  - Обучение без учителя (unsupervised learning)
- Обучение с подкреплением (reinforcement learning)
- Ансамблевые методы (Ensemble of models)
- Нейронные сети и глубокое обучение

Дополнительно выделяют:

- Частичное обучение (semi-supervised learning)
- Трансдуктивное обучение (transductive learning)
- Динамическое обучение (online learning)
- Активное обучение (active learning)
- Метаобучение (meta-learning или learning-to-learn)



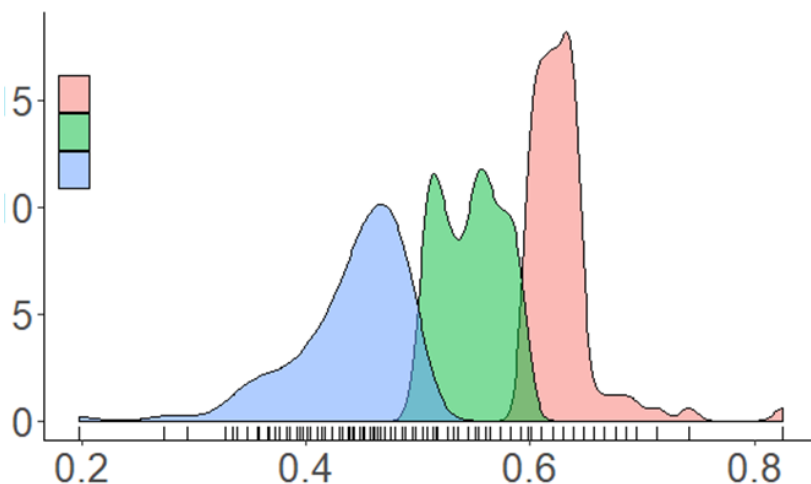


# Тема: «Основные понятия машинного обучения»

- Основные понятия: модель, признаки (непрерывные и дискретные), выборка, меры близости, метрические пространства.
- Неопределенность, виды неопределенности.
- Основы статистического анализа.
- Предобработка данных.

# Основы математической статистики

- Случайная природа входных данных.
- Возможные распределения значений признаков: нормальное, равномерное и др.
- Независимость признаков, матрица корреляции.



	1	2	3	4	5	6	7	8	9	10	11	12
1		0,27	-0,28	0,38	-0,45	0,45	-0,08	-0,02	-0,16	-0,25	0,05	-0,18
2	0,27		-0,43	0,45	-0,29	0,26	0,41	-0,30	0,32	0,27	-0,25	0,12
3	-0,28	-0,43		-0,91	0,40	-0,37	-0,47	0,49	-0,47	-0,28	0,50	-0,29
4	0,38	0,45	-0,91		-0,48	0,45	0,45	-0,49	0,44	0,11	-0,45	0,13
5	-0,45	-0,29	0,40	-0,48		-0,94	-0,12	0,25	-0,18	0,20	0,15	0,27
6	0,45	0,26	-0,37	0,45	-0,94		0,01	-0,10	0,07	-0,23	0,00	-0,31
7	-0,08	0,41	-0,47	0,45	-0,12	0,01		-0,71	0,82	0,44	-0,65	0,28
8	-0,02	-0,30	0,49	-0,49	0,25	-0,10	-0,71		-0,75	-0,52	0,81	-0,42
9	-0,16	0,32	-0,47	0,44	-0,18	0,07	0,82	-0,75		0,55	-0,58	0,28
10	-0,25	0,27	-0,28	0,11	0,20	-0,23	0,44	-0,52	0,55		-0,47	0,62
11	0,05	-0,25	0,50	-0,45	0,15	0,00	-0,65	0,81	-0,58	-0,47		-0,65
12	-0,18	0,12	-0,29	0,13	0,27	-0,31	0,28	-0,42	0,28	0,62	-0,65	

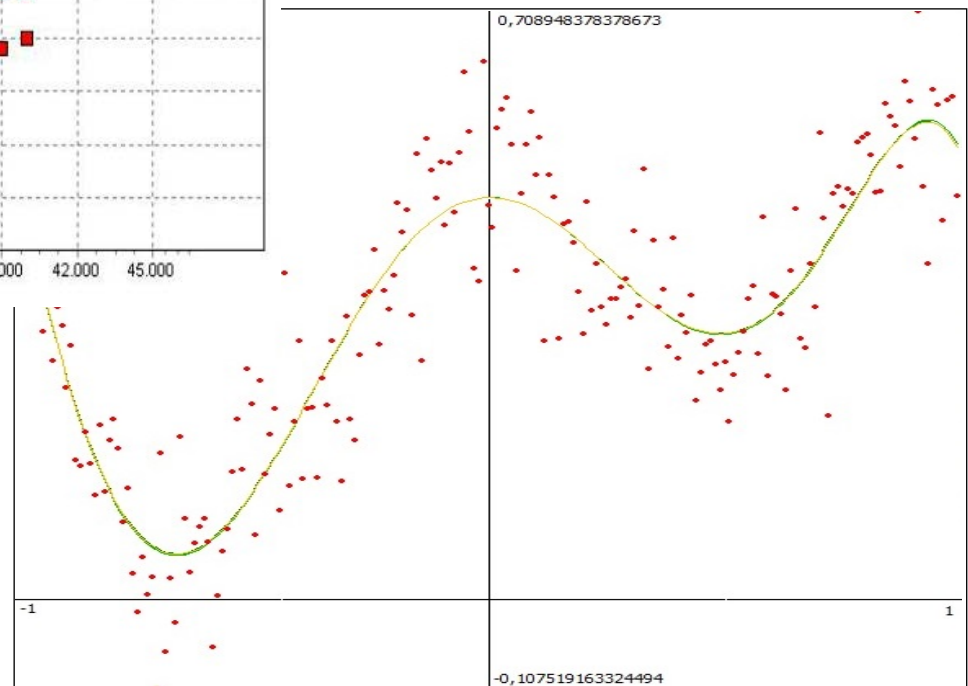
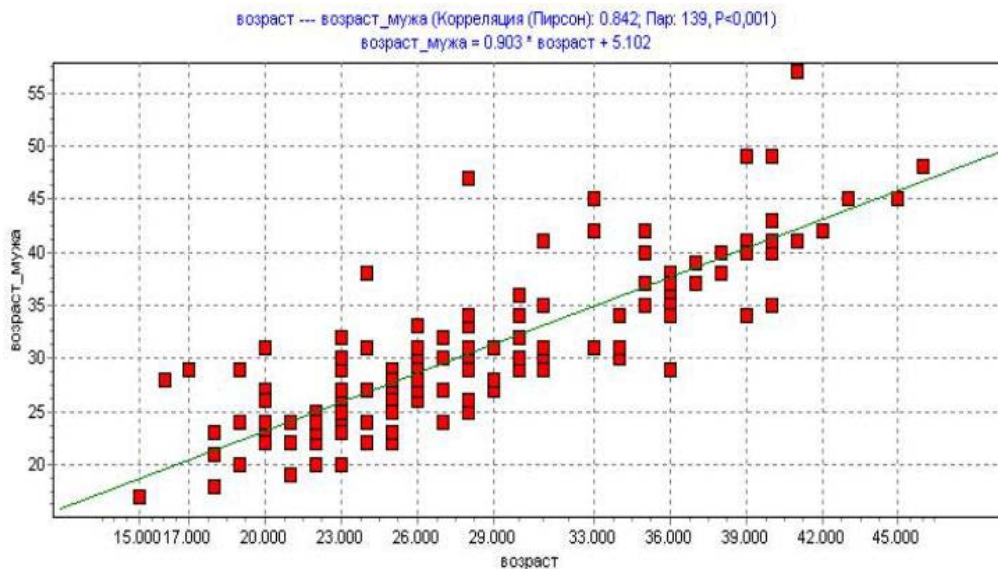
# Тема: «Классическое машинное обучение»

## Обучение с учителем (supervised learning)

- Постановка задачи регрессии, линейная регрессия, логистическая, основные метрики качества регрессии.
- Постановка задачи классификации, методы классификации, основные метрики качества классификации.

# Регрессионные модели

- Постановка задачи;
- Линейная и нелинейная регрессия;
- Основные метрики оценки качества.



# Задачи классификации

- Постановка задачи;
- Методы классификации;
- Основные метрики оценки качества.



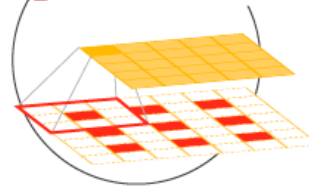
**Дерево  
решений**



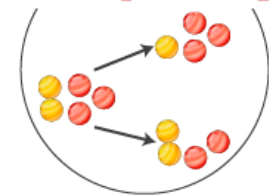
**Метод опорных  
векторов**



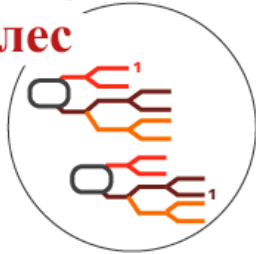
**Сверточные  
нейронные сети**



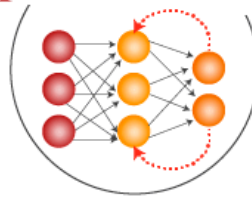
**Наивный  
байесовский  
классификатор**



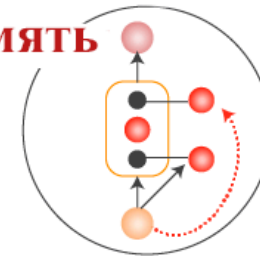
**Случайный  
лес**



**Рекуррентные  
нейронные сети**



**Долгая  
краткосрочная  
память**

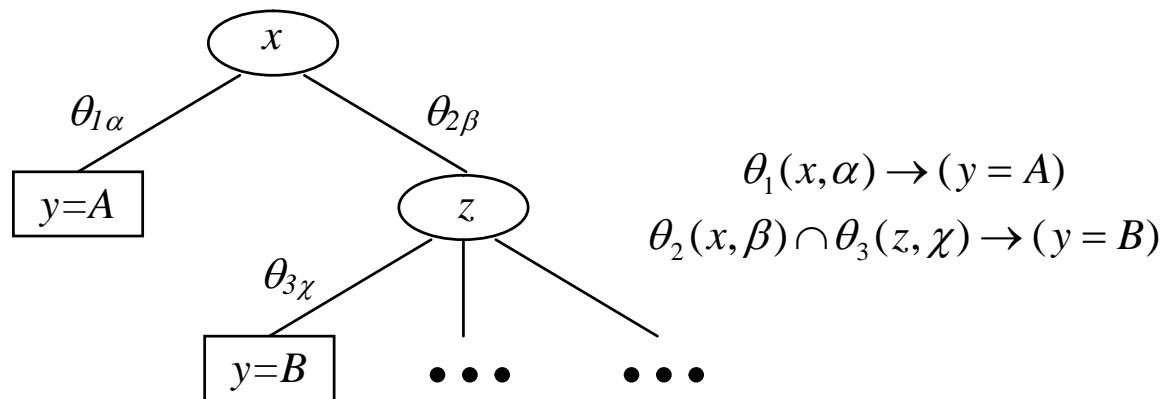


# Деревья решений

Деревья решений относятся к методам поиска логических закономерностей в данных, а также являются основным подходом, применимым в теории принятия решений. Их основным достоинством является простота и наглядность описания процесса поиска решения.

Представление правил в виде продукций «если... то...»:

**Если** (условие 1)  $\wedge$  ...  $\wedge$  (условие N) **то** (значение вершины вывода).



Построения деревьев решений:

- На основе экспертных оценок;
- С использованием алгоритмов обработки примеров (CLS, ID3 (Interactive Dichotomizer), C4.5, CART (classification and regression trees) и др. );
- Генетические алгоритмы и эволюционное программирование.

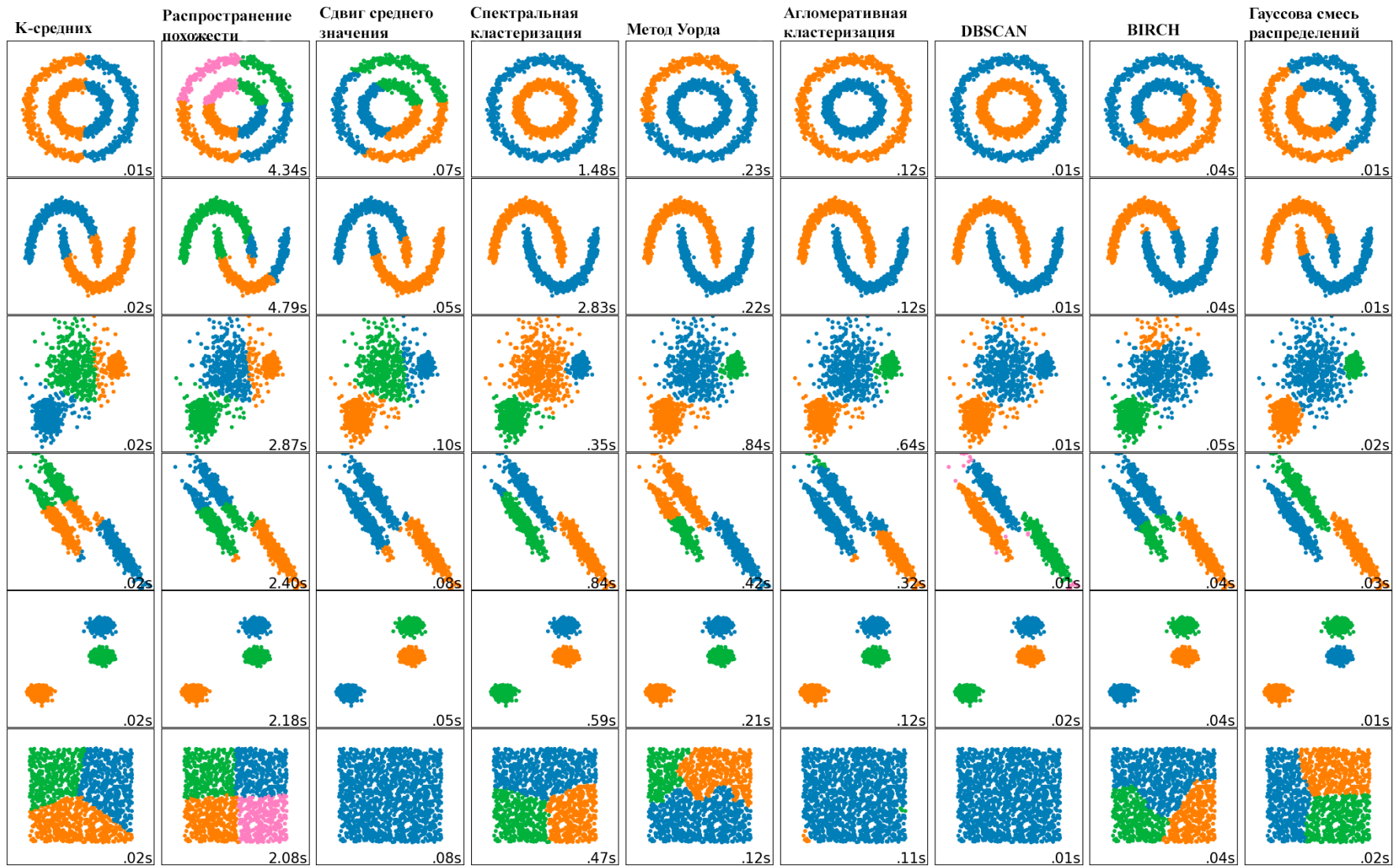
# Тема: «Классическое машинное обучение»

## Обучение без учителя (unsupervised learning)

- Методы кластеризации: иерархическая, k-средних, DBSCAN. Применение кластеризации в анализе данных.
- Методы понижения размерности: PCA, T-SNE, UMAP.

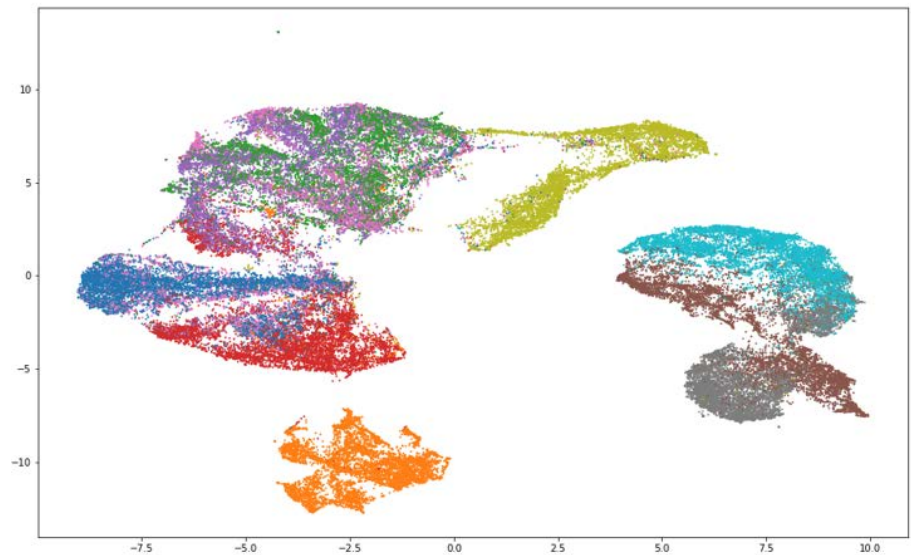
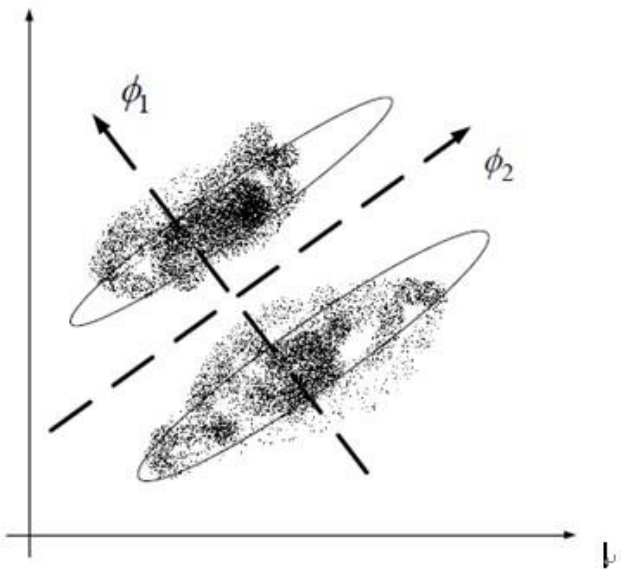


# Методы кластеризации



# Методы понижения размерности

- Алгоритм анализа главных компонент (PCA);
- Линейный дискриминантный анализ (линейный дискриминант Фишера) (LDA);
- Локально линейное вложение (LLE);
- t-distributed stochastic neighbor embedding (T-SNE);
- Uniform Manifold Approximation and Projection (UMAP).

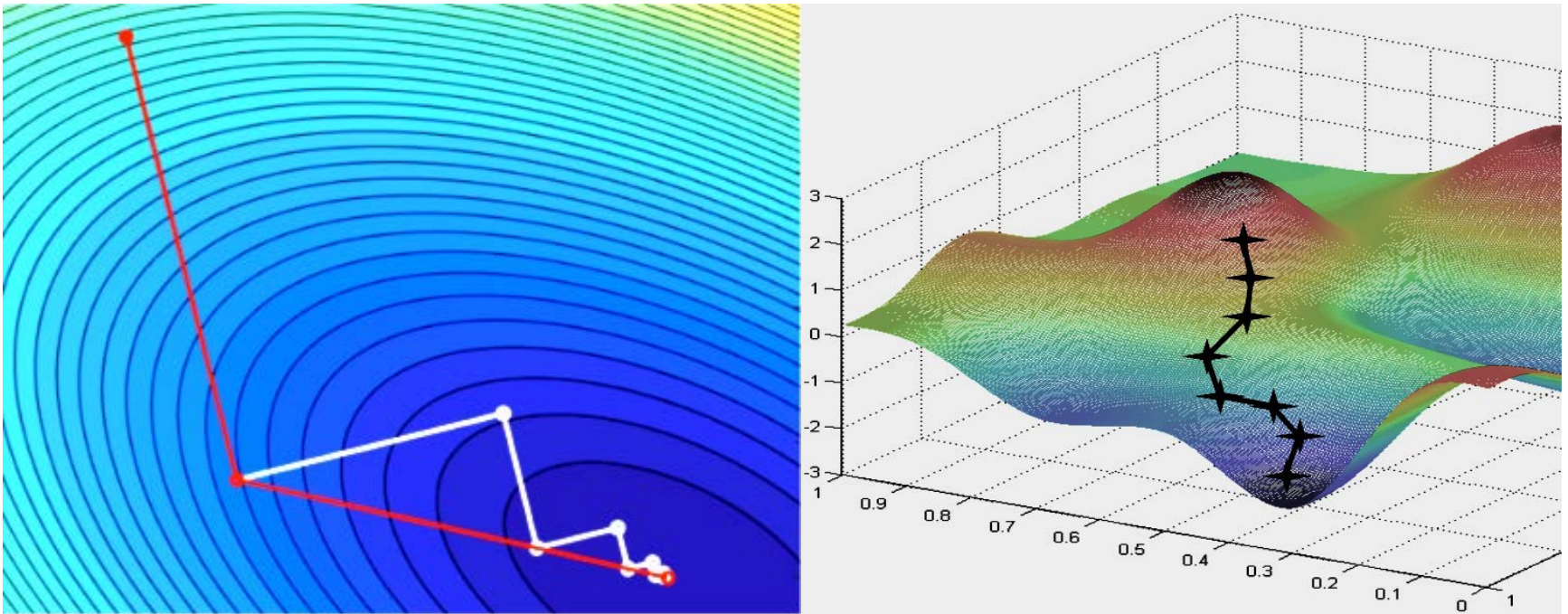


# Тема: «Ансамбли классификаторов»

- Методы оптимизации в приложении к задачам машинного обучения. Градиентный спуск и его адаптации.
- Слабые и сильные классификаторы (байесовский классификатор, метод опорных векторов и др.), их ансамбли. Алгоритмы бустинга (AdaBoost, CatBoost и др.).

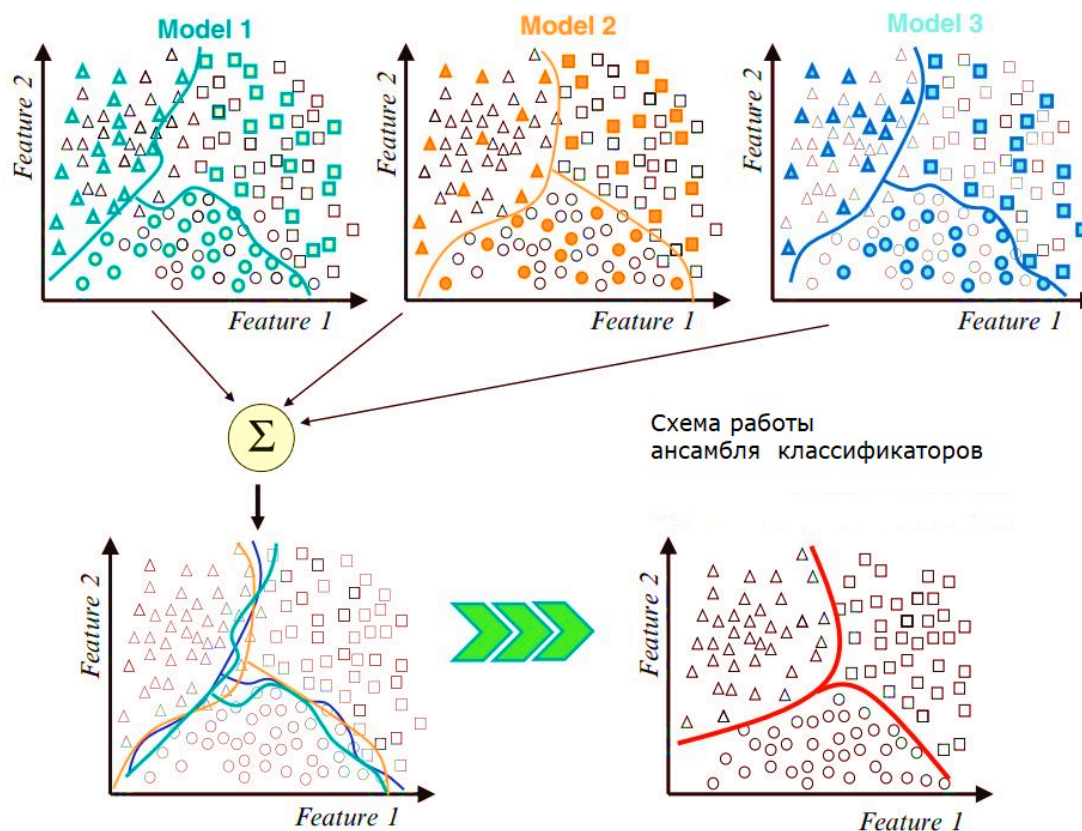
# Методы оптимизации применительно к задачам машинного обучения

- Обзор методов численной оптимизации;
- Скорость сходимости;
- Градиентный спуск и его адаптации;
- Метод сопряженных градиентов.



# Слабые и сильные классификаторы

- Байесовский классификатор;
- Метод опорных векторов;
- Ансамблевые классификаторы;
- Алгоритмы бустинга (AdaBoost, CatBoost и др.)

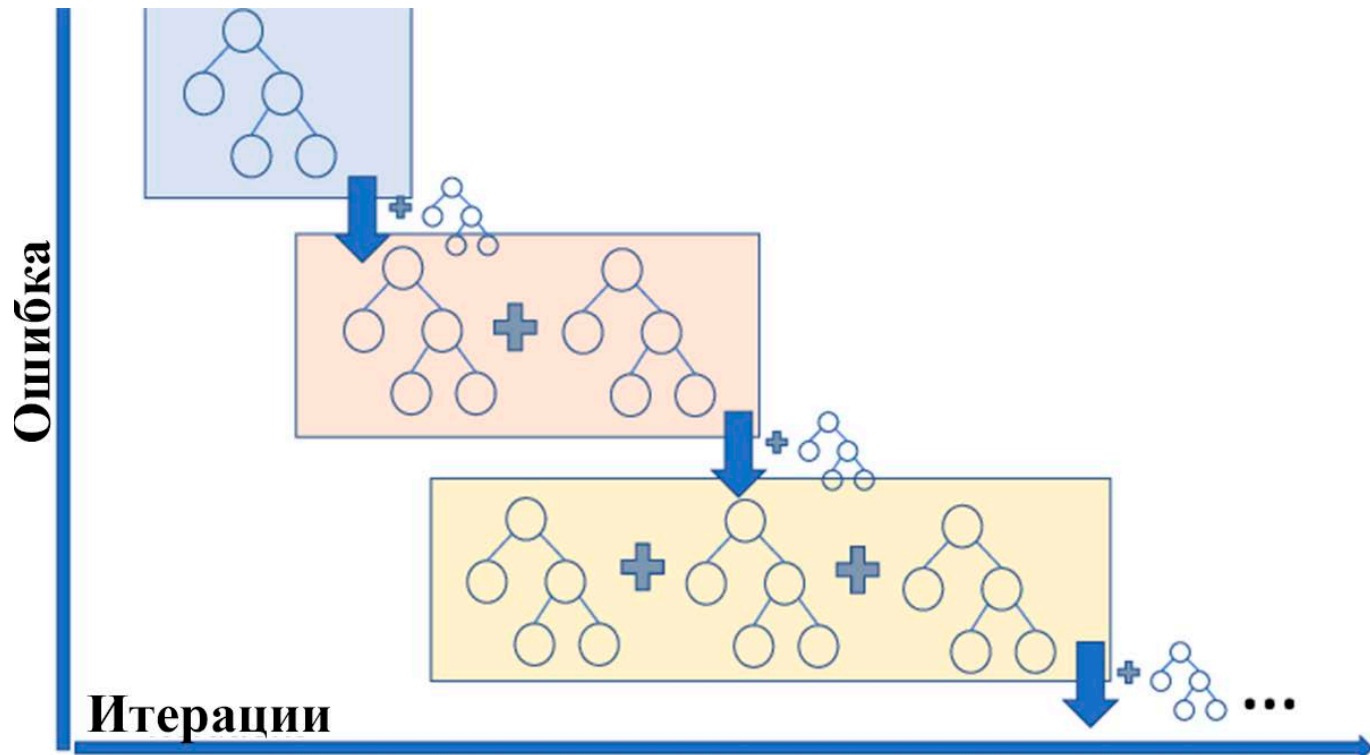




# Алгоритмы бустинга

Особенности градиентного бустинга:

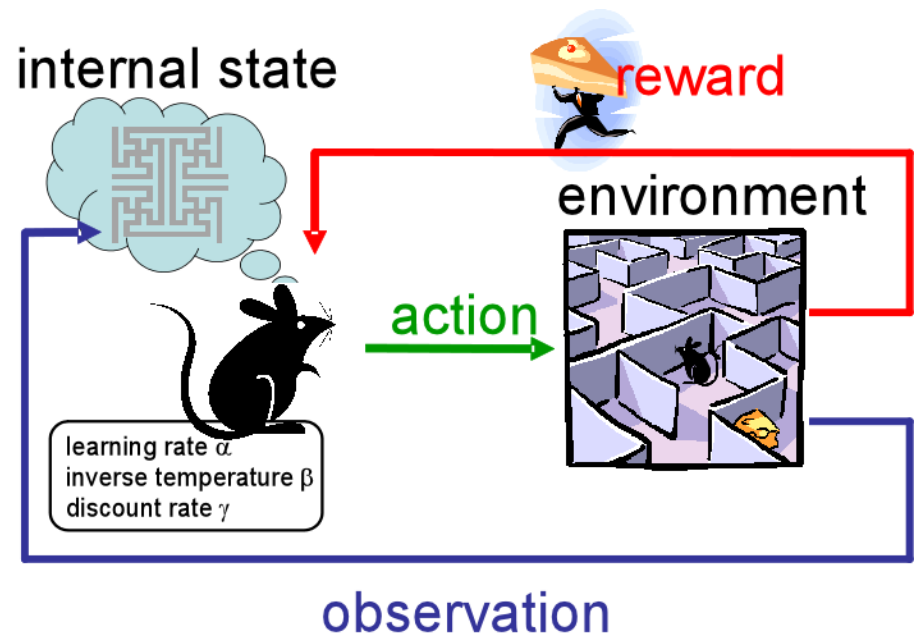
- Использование ансамблей деревьев;
- Возможность использования для выборок с несбалансированными классами;
- Требуется тщательная настройка параметров.



# Тема: «Обучение с подкреплением» (*reinforcement learning*)

Способ машинного обучения, при котором система обучается, взаимодействуя с некоторой средой.

- Марковские процессы. Марковские модели. Агент (*agent*), среда (*environment*), обратная связь, состояние, функции ценности состояния (Value function), качества действия (Q-function);
- Многорукие бандиты;
- Способы взаимодействия и оптимизационные задачи с помощью агентов;
- Генетические алгоритмы.



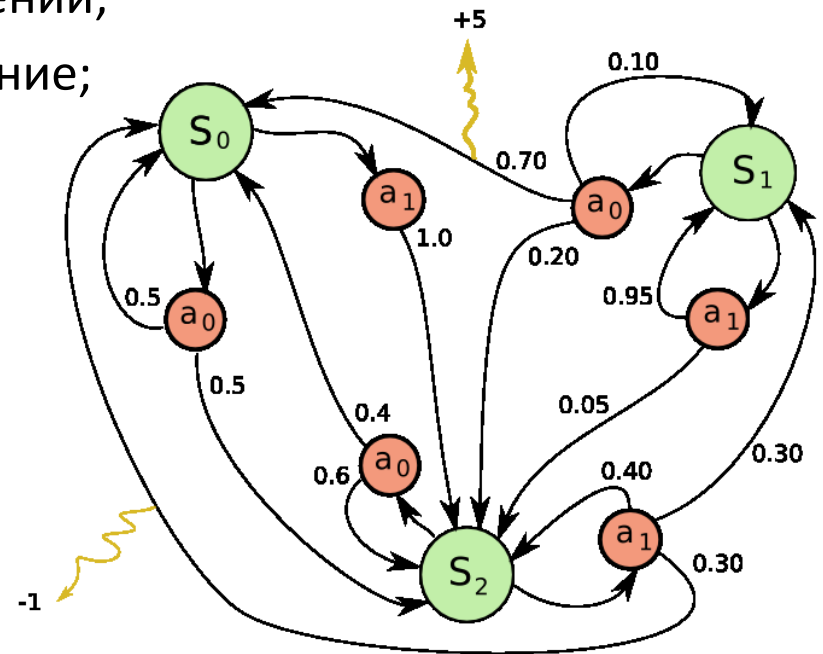


# Марковские процессы

**Марковский процесс** — случайный процесс, эволюция которого после любого заданного значения временного параметра  $t$  *не зависит* от эволюции, предшествовавшей  $t$ , при условии, что значение процесса в этот момент фиксировано («будущее» процесса зависит от «прошлого» лишь через «настоящее»).

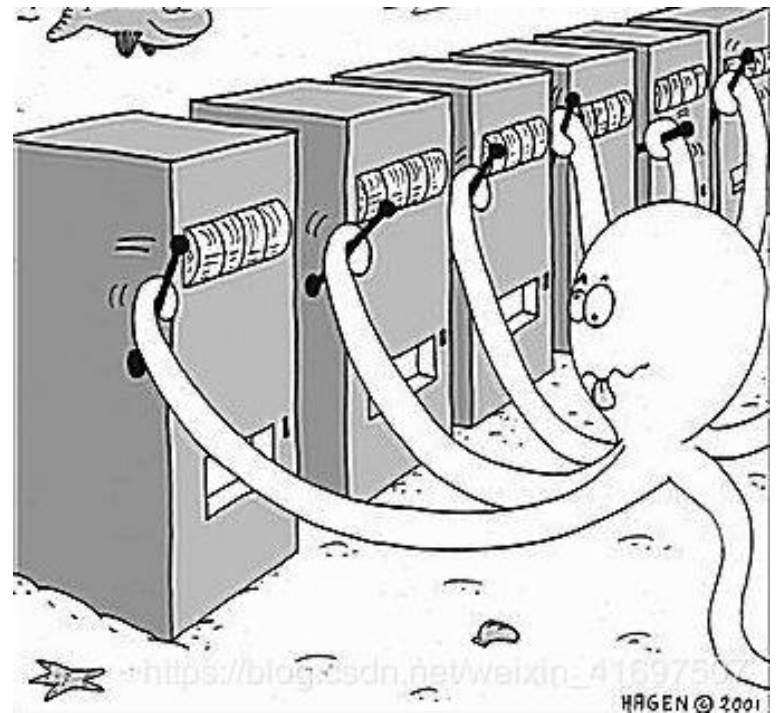
**Марковское свойство** — в теории вероятностей и статистике термин, который относится к памяти случайного процесса.

- Марковский процесс принятия решений;
- Агент, среда, обратная связь, состояние;
- Функция полезности состояния (Value function);
- Функция полезности действия (Q-function);
- Q-обучение (Q-learning).



# Задача о многоруком бандите (*The multi-armed bandit problem*)

- Агенты с одним состоянием, т.е. состояние агента не меняется. У него фиксированный набор действий и возможность выбора из этого набора действий.
- Модель: агент в комнате с несколькими игровыми автоматами. У каждого автомата своё ожидание выигрыша.
- Нужно заработать побольше:  
Exploration vs. Exploitation  
(разведка против эксплуатации).
- Жадные и  $\epsilon$ -жадные стратегии  
(greedy &  $\epsilon$ -greedy)

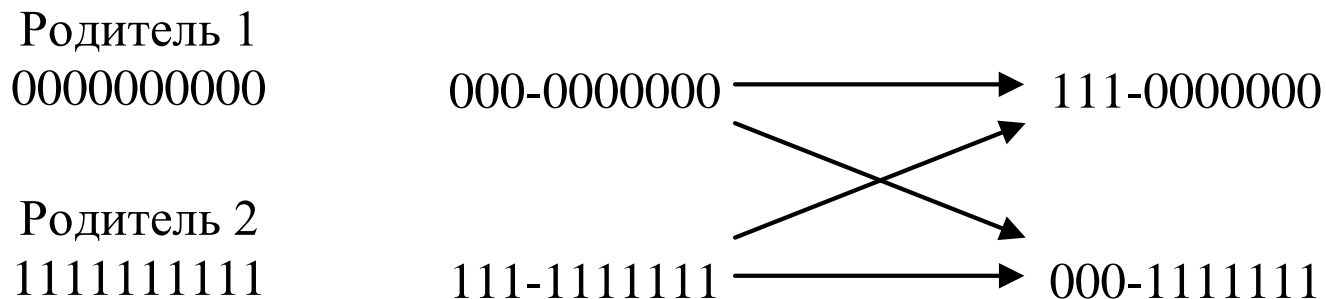


# Генетический алгоритм (*genetic algorithm*)

Эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в живой природе, таких как: скрещивание, наследование, мутации и отбор.

Область применения:

- поиск логических закономерностей - сегментация данных, поиск оптимального набора элементарных событий, выделение наиболее значимых значений признаков и их сочетаний;
- предобработка исходных данных в задаче классификации;
- поиск значений параметров аппроксимирующей функции для числовых последовательностей;
- обучение нейронной сети.



# Непрерывный генетический алгоритм

Генетический алгоритм для предобработки исходных данных в задаче классификации и поиска аппроксимирующего полинома для числовых последовательностей.

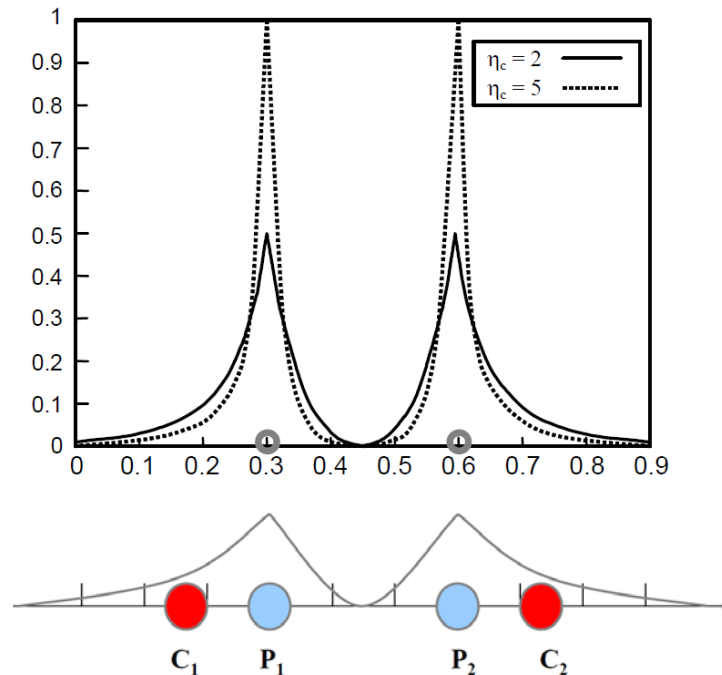
Целевая функция:

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

Оператор скрещивания - SBC (Simulated Binary Crossover) – кроссовер, который имитирует работу двоичного оператора скрещивания. Из двух векторов вещественных чисел  $\{a_0^1, a_1^1, a_2^1 \dots a_N^1\}$  и  $\{a_0^2, a_1^2, a_2^2 \dots a_N^2\}$  осуществляется формирование двух новых векторов.

$$a_i^{1(new)} = 0.5[(1 + \beta) \cdot a_i^1 + (1 - \beta) \cdot a_i^2]$$

$$a_i^{2(new)} = 0.5[(1 - \beta) \cdot a_i^1 + (1 + \beta) \cdot a_i^2]$$

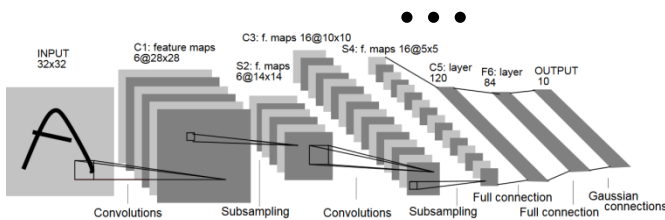
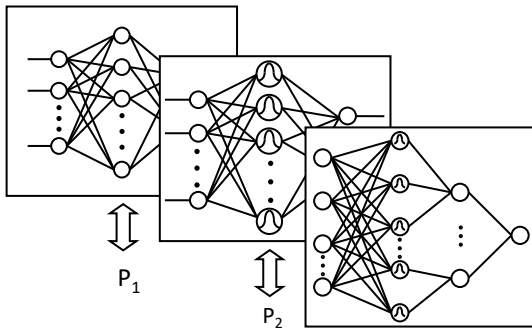


Важная особенность SBC– кроссовера заключается в том, что среднее значение функции приспособленности остается неизменным у родителей и их потомков.

$$\frac{a_i^{1(new)} + a_i^{2(new)}}{2} = \frac{a_i^1 + a_i^2}{2}$$

# Тема: «Нейронные сети»

Цель: выбор оптимальной топологии сети, значений параметров и структурных особенностей, которые бы наилучшим образом удовлетворяли решаемой задаче на имеющихся исходных данных.



- Многослойный персептрон
- RBF-сеть
- Гибридные нейросети
- Самоорганизующиеся карты Кохонена
- Сверточные сети
- Глубокое обучение
- и т.д.

Необходимость учитывать:

- Решаемая задача;
- Исходные данные (объем, структура и т.д.);
- Внутреннее представление информации нейросетью;
- Процесс обучения (обобщение vs переобучение);
- Интерпретируемость полученного результата.

# Инструменты Machine Learning

Инструменты машинного обучения используют на следующих этапах:

- сбор и подготовка данных;
- построение модели;
- интерпретация работы модели;
- обучение и развертывание приложений.

Для выполнения каждого из этих этапов могут применяться специализированные платформы. Они различаются по языку программирования (Python, R, Java, C, C++, Scala, CUDA), операционным системам (Linux, Mac OS, Windows) и тому, какие задачи можно решить с их помощью. Поддерживаются многими интегрированными средами разработки, в частности, R-Studio, R-Brain, Visual Studio, Eclipse, PyCharm, Spyder, IntelliJ IDEA, Jupyter Notebooks, Juno и др.

Сегодня на рынке представлено несколько десятков программных инструментов:

- TensorFlow;
- PyTorch ;
- Shogun;
- Keras.io;
- Rapid Miner;
- Google Cloud ML Engine;
- Amazon Machine Learning (AML);
- Accord.NET;
- Apache Mahout;
- Microsoft Azure ML;
- SberCloud ML Space.

# Перечень учебной литературы и дополнительных материалов, необходимых для освоения дисциплины

## Литература по дисциплине

- В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святков, Теория и практика машинного обучения : учебное пособие / Ульяновск : УлГТУ, 2017.
- Математические методы обучения по прецедентам (теория обучения машин) Курс лекций К.В.Воронцова, URL: <http://www.machinelearning.ru>.
- Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. – 2-е изд.: ДМК Пресс, 2018. – 652 с.
- СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА 2-е изд., испр. и доп. Учебное пособие для вузов / Бессмертный И. А. - 2022. - URL: <https://urait.ru/book/A7D41B16-CE66-4451-92E9-73529EA6C9F5>.
- Плас Дж. Вандер Python для сложных задач: наука о данных и машинное обучение / Плас Дж. Вандер ; пер. с англ. Пальти И. - СПб. : Питер, 2020. - 572 с. : рис., табл. - (Бестселлеры O'Reilly). - Библиогр. в конце глав. - ISBN 978-5-4461-0914-2.
- Саттон, Р. С. Обучение с подкреплением: введение : руководство / Р. С. Саттон, Э. Д. Барто ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2020. — 552 с. — ISBN 978-5-97060-097-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179453>
- Араки, М. Манга: Машинное обучение / М. Араки ; перевод с японского А. С. Слащевой ; Ватари Макана. — Москва : ДМК Пресс, 2020. — 214 с. — ISBN 978-5-97060-830-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179473>
- Шарден, Б. Крупномасштабное машинное обучение вместе с Python : учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти ; перевод с английского А. В. Логунова. — Москва : ДМК Пресс, 2018. — 358 с. — ISBN 978-5-97060-506-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/105836>

## Дополнительные материалы

- Основы искусственного интеллекта : учебное пособие / Е.В.Боровская, Н. А. Давыдова. 4-е изд.,электрон. М. : Лаборатория знаний, 2020. 130 с.
- Искусственный интеллект с примерами на Python. Джоши Пратик. Вильямс. 2019.
- Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем , 2-е издание. Жерон Орельен.Диалектика-Вильямс. 2020.
- Хенрик Бринк, Джозеф Ричардс, Марк Феверолф «Машинное обучение», Питер 2017.
- Как учится машина: Революция в области нейронных сетей и глубокого обучения. Ян Лекун. Альпина PRO. 2021.
- Грожаем глубокое обучение. Эндрю Траск. Питер. 2019.
- Обучение с подкреплением на PyTorch. Сборник рецептов. Юси Лю. ДМК Пресс. 2020.

# Методика оценки по рейтингу

**Промежуточная аттестация** по дисциплине проходит в форме дифференцированного зачета за курсовую работу и экзамена, контролирующего освоение ключевых, базовых положений дисциплины, составляющих основу остаточных знаний по ней.

Студент, выполнивший все предусмотренные учебным планом задания и сдавший все контрольные мероприятия, получает итоговую оценку по дисциплине за семестр в соответствии со шкалой:

Рейтинг	Оценка на экзамене, на дифф. зачете
85 – 100	отлично
71 – 84	хорошо
60 – 70	Удовлетворительно
0 – 59	неудовлетворительно

Оценивание дисциплины ведется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации студентов МГТУ им. Н.Э. Баумана.



**Спасибо за внимание**