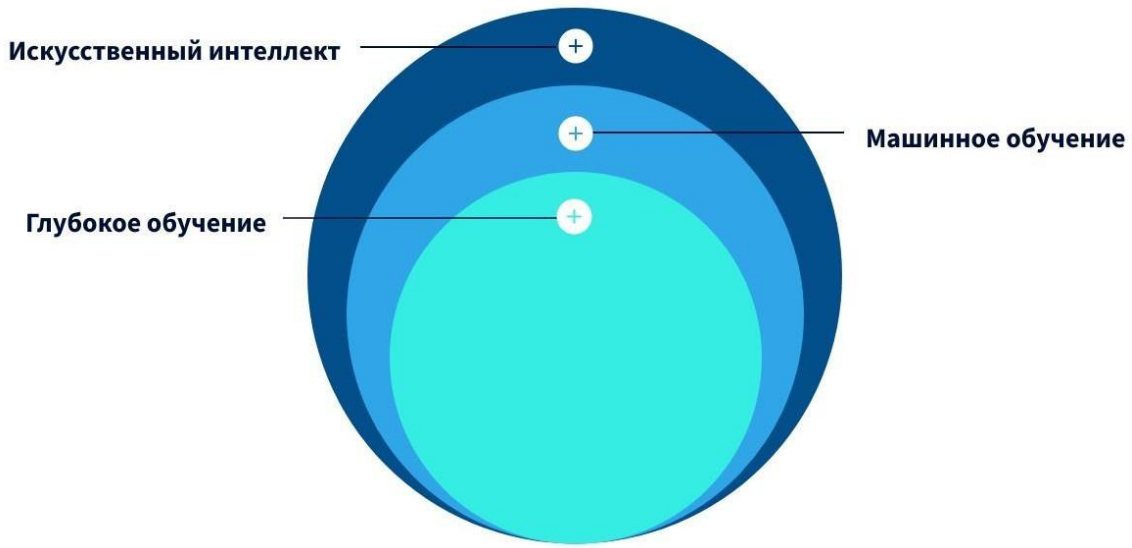
# Искусственный интеллект, машинное обучение и глубокое обучение

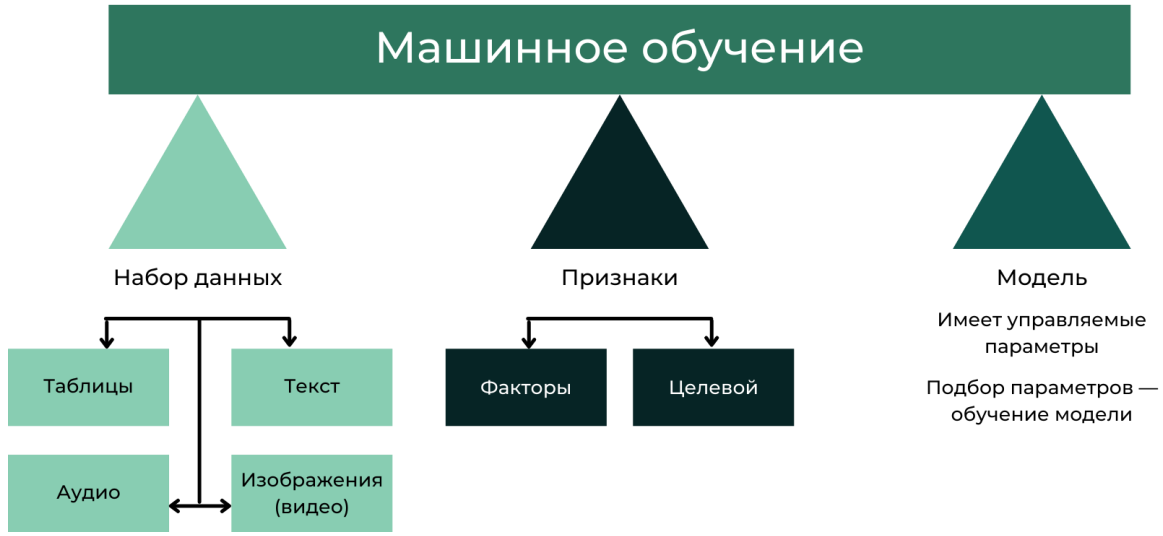
**Искусственный интеллект (Artificial Intelligence)** – это способность компьютерной системы имитировать когнитивные функции человека, такие как обучение и решение задач. ИИ позволяет компьютеру моделировать рассуждения людей для получения новых сведений и принятия решений (например, выдавать кредит человеку или нет).

**Машинное обучение (Machine Learning)** – это один из разделов науки об искусственном интеллекте. Машинное обучение заключается в построении моделей с помощью поиска закономерностей в данных и использовании их для того, чтобы спрогнозировать характеристики новых данных.

**Глубокое обучение (Deep Learning)** — подраздел машинного обучения. Раздел основан на изучении и применении в качестве инструмента для решения задач **искусственных нейронных сетей**. Данные алгоритмы основаны на имитации работы человеческого мозга.



## Основы машинного обучения:




**Набор данных** — это множество примеров (выборка), на котором происходит обучение модели. Это могут табличные данные, с которыми мы уже работали, текст, аудио, изображения (видео) и т. д.

**Признаки (features)** – это свойства, характеристики, которыми описываются наши объекты. Для недвижимости это могут быть площадь, этаж, район; для автомобиля — пробег, мощность двигателя, цвет и т. д.

Признак, который мы хотим предсказать, называется **целевым признаком (target feature)**. Иногда признаки, на основе которых мы хотим предсказать целевой, могут называться **факторами (factors)**. Например, хотим предсказать цену недвижимости: цена — наш целевой признак, остальные признаки (площадь, этаж, район) — факторы.

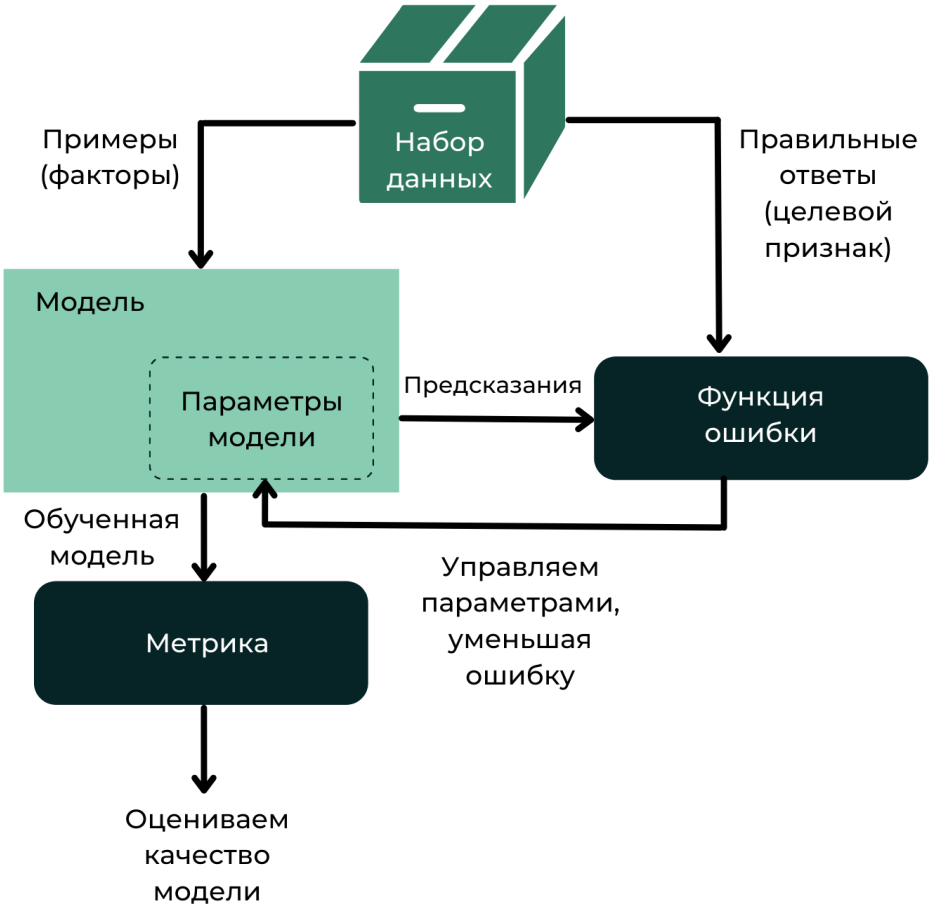
**Модель машинного обучения (ML-model)** – это некоторый математически формализованный метод (алгоритм) описания зависимости в данных. Как правило, модель имеет настраиваемые (регулируемые) параметры.

Управляя своими параметрами, модель подстраивается под зависимости в данных, чтобы описать эту зависимость и свести ошибку в предсказаниях к минимуму. Такой процесс называется **обучением модели (model learning)**.

## Основные схемы обучения:

- на основе минимизации ошибок (минимизация эмпирического риска);
- на основе «сходства» объектов;
- на основе вероятностных законов;
- на основе прироста информации.

## Схема обучения (минимизация эмпирического риска):




За управление параметрами отвечает некоторая **функция ошибки**, или **функция потерь (loss function)**, – это некоторая математическая функция, которая показывает различие между фактическими ответами и предсказаниями модели.

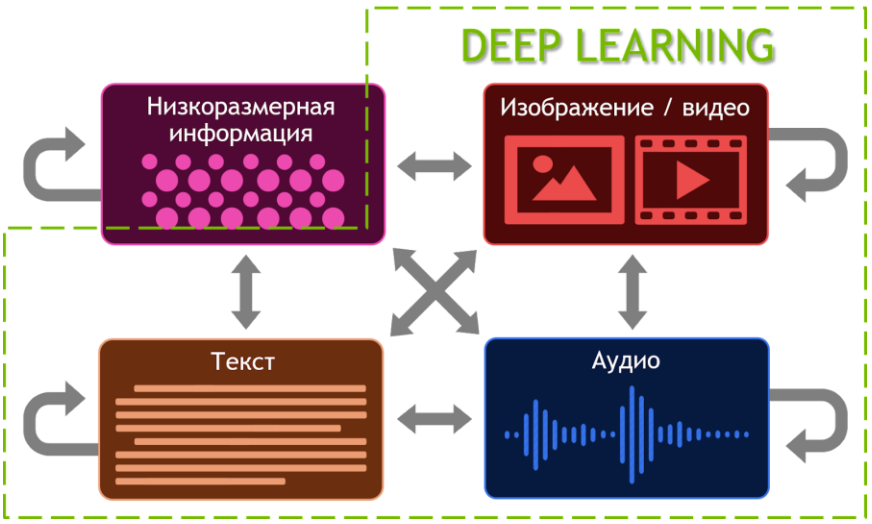
Мы пытаемся подобрать такие параметры модели, при которых функция ошибки нашего предсказания была бы наименьшей возможной на предоставленных данных. Такие параметры называются **оптимальными**.

Для оценки качества модели вводится еще одно понятие - **метрика**.

**Метрика (metric)** — это численное выражение качества модели (или её ошибки). Иногда метрика может совпадать с функцией потерь, но чаще всего они различны. Метрика, как правило, должна быть интерпретируемой и понятной — в этом её главное отличие от функции потерь.

**Метрика ≠ функция потерь**

## Области применения глубокого обучения:



## Карта мира машинного обучения:

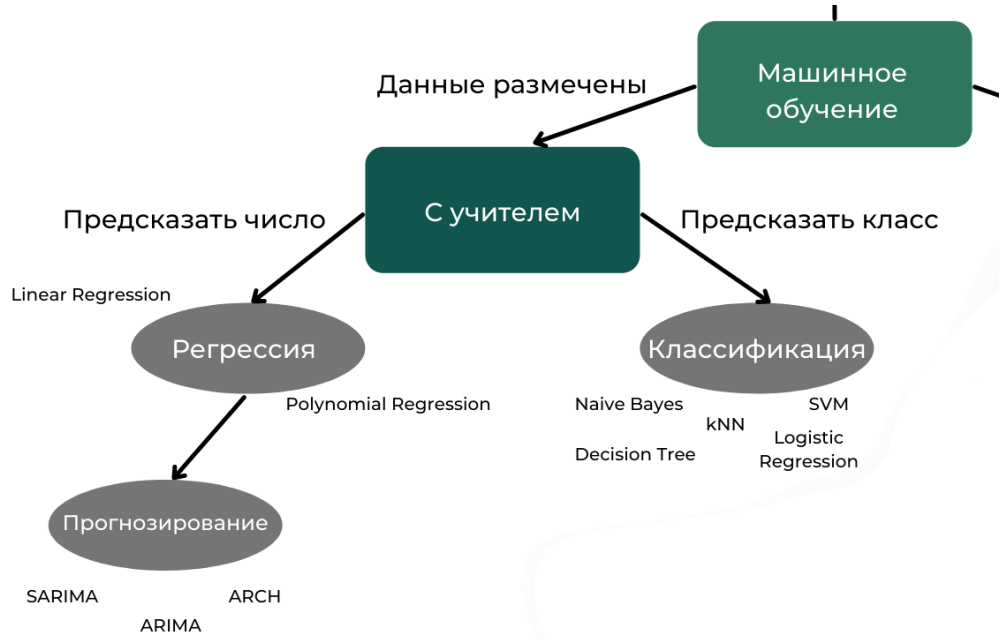

В зависимости от наличия разметки в данных и особенностей обучения выделяют следующие **виды машинного обучения**:

- **обучение с учителем (Supervised Learning)**;
- **обучение без учителя (Unsupervised Learning)**.

В отдельную категорию, не похожую на предыдущие, выделяют ещё один вид машинного обучения – **обучение с подкреплением (Reinforcement Learning)**.

При этом обучение с учителем и обучение без учителя содержат в себе отдельные типы задач машинного обучения, такие как **регрессия**, **классификация**, **кластеризация**, **понижение размерности** и **ассоциация**.

## Обучение с учителем

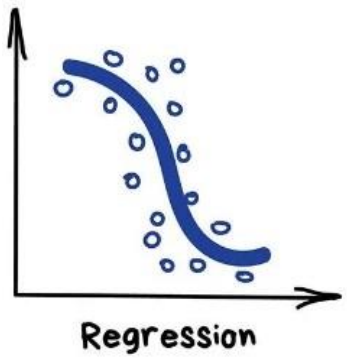



Обучение производится на размеченных данных.

> Данные, в которых содержится информация о целевом признаке, называются **размеченными**.

Вид обучения с учителем включается в себя два основных типа задач: **регрессия** — предсказание числа и **классификация** – предсказание категории объекта.

### Регрессия



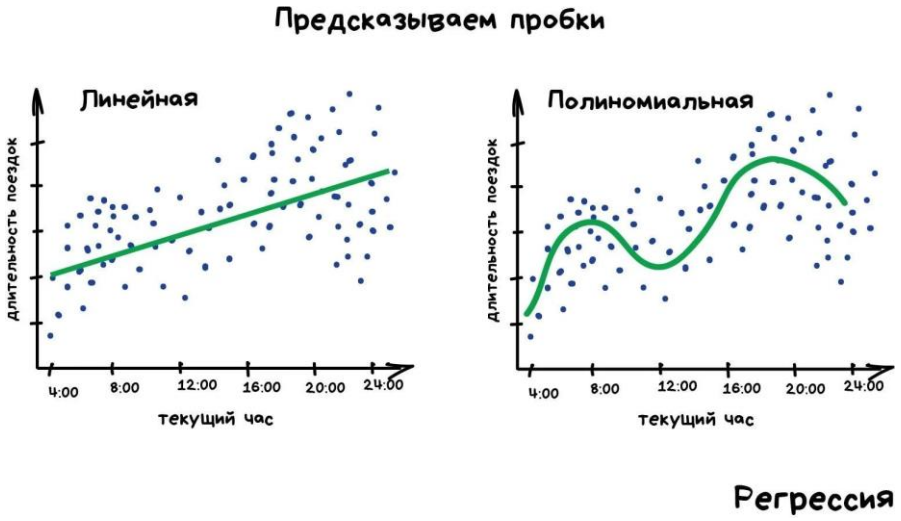
**Задача регрессии (regression)** – это задача, в которой мы пытаемся предсказать вещественное число на основе признаков в наборе данных. То есть задача сводится к предсказанию целевого признака, который является **числовым**.

**Примеры задач регрессии:**

- предсказываем цену квартиры,
- рейтинг вина при слепом тестировании,
- длительность поездки в такси в зависимости от времени суток,
- желаемую заработную плату соискателя.

**Цель обучения** – построить модель, которая бы отражала зависимость между признаками и целевой числовой переменной.

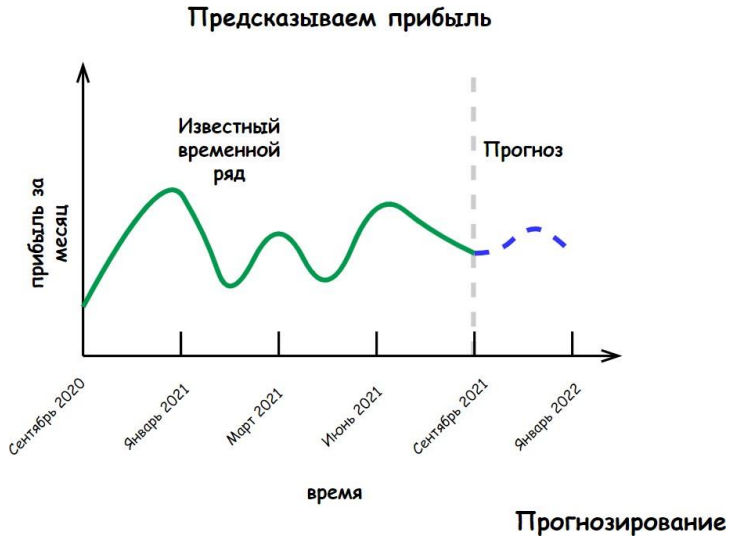
**Пример регрессии:**



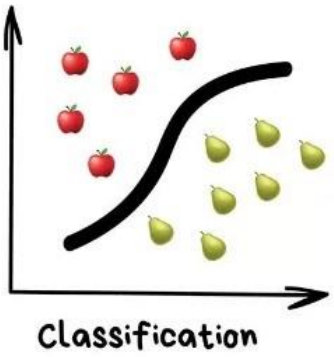

Нередко в качестве отдельного подвида задачи регрессии выделяют задачу **прогнозирования**.

**Прогнозирование (forecasting)** – это задача регрессии, в которой мы пытаемся предсказать будущее поведение временного ряда, то есть целевая переменная является **числовой** и **зависит от времени**. Причём каждому моменту времени соответствует одно конкретное значение.

**Пример прогнозирования:**




## Классификация



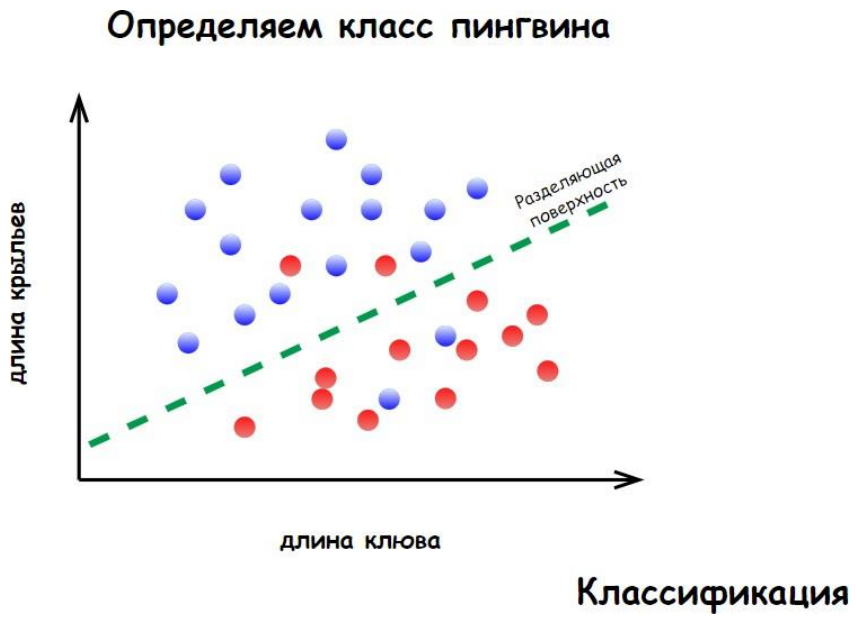

**Задача классификации (classification)** — задача, в которой мы пытаемся предсказать класс объекта на основе признаков в наборе данных. То есть задача сводится к предсказанию целевого признака, который является **категориальным**.

Чаще всего мы сталкиваемся с **бинарной** классификацией: целевой признак имеет две возможные категории (*да* — 1 или *нет* — 0). Например, мы можем предсказать, болен ли пациент раком, является ли изображение человеческим лицом, является ли письмо спамом и т. д.

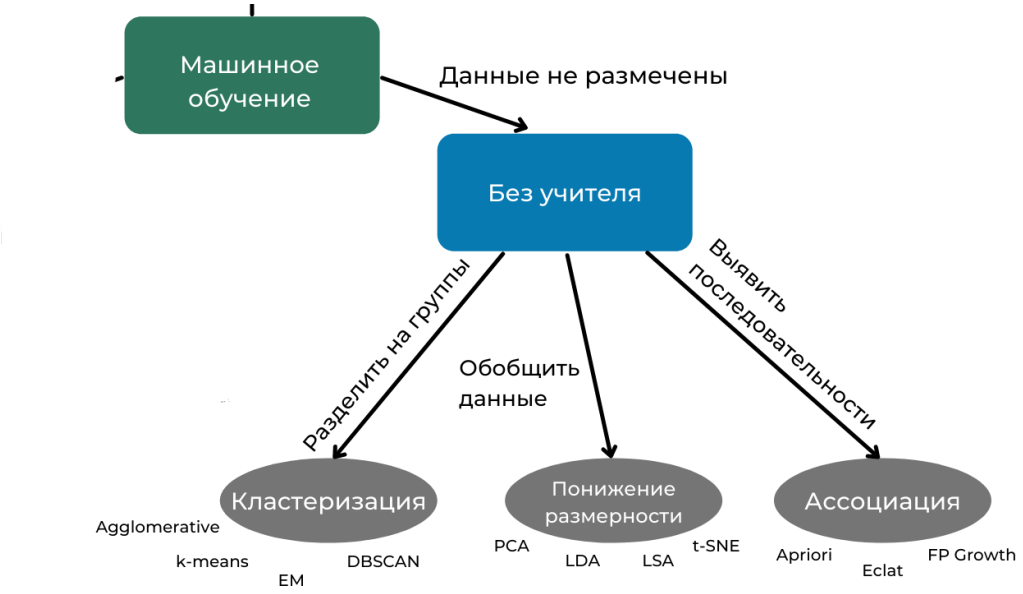
Когда классов, которые мы хотим предсказать, более двух, классификация называется **мультиклассовой (многоклассовой)**. Например, предсказание модели самолёта по радиолокационным снимкам, классификация животных на фотографиях, определение языка, на котором говорит пользователь, разделение писем на группы.

**Цель обучения** – построить модель, которая разделяет признаки на классы наилучшим образом. С математической точки зрения это значит построение разделяющей поверхности для классов в пространстве признаков.

**Пример классификации:**



## Обучение без учителя

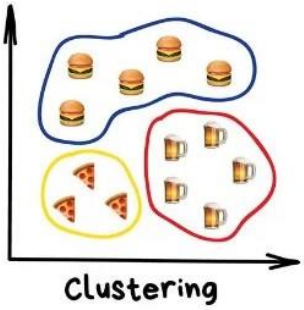



Обучение без учителя подразумевает, что у вас **нет правильных ответов**. То есть признак, который вы хотите предсказать, недоступен. Подход основан на том, что алгоритм самостоятельно выявляет зависимости в данных только на основе схожести объектов в данных между собой.

Обучение без учителя всё же чаще используют как метод анализа и предобработки данных. Данный вид машинного обучения разбивается на несколько самостоятельных **типов задач**:

- **кластеризация**,
- **понижение размерности**,
- **ассоциация**.

### Кластеризация



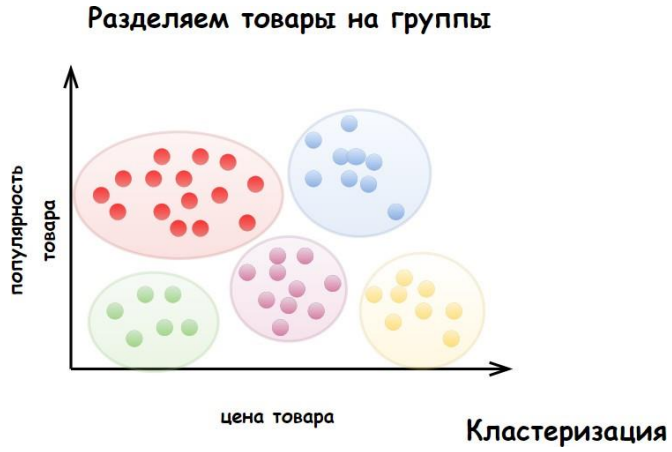
**Задача кластеризации (clustering)** – это задача, в которой мы разделяем данные на группы на основе признаков в данных.

**Примеры использования кластеризации:**

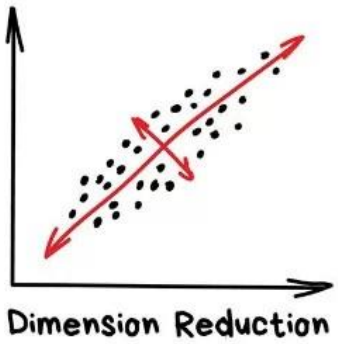
- сегментация рынка на категории,
- объединение близких точек на карте,
- разделение клиентов по уровню платёжеспособности,
- кластеризация студентов по их интересам или обучаемости,
- разметка новых данных.

**Цель обучения** – построить модель, которая наилучшим образом объединит «похожие» объекты в группы.

**Пример кластеризации:**




## Понижение размерности (обобщение)



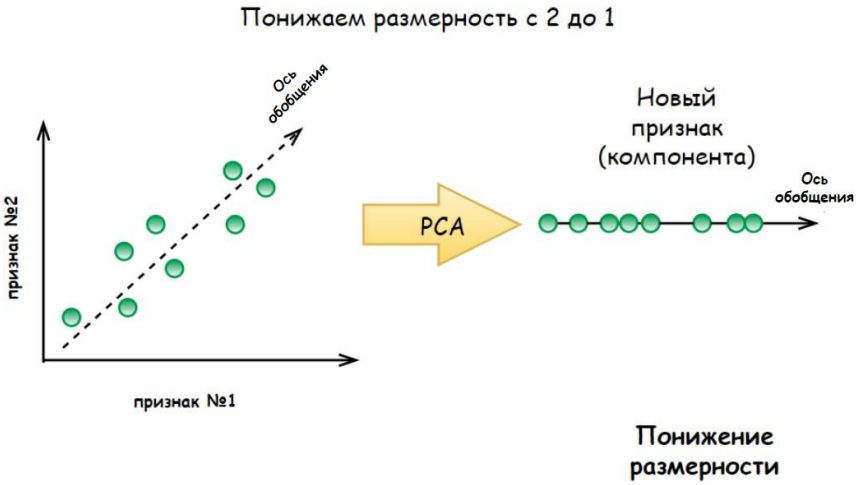
**Понижение размерности (dimensionality reduction)** – это задача, в которой мы пытаемся уменьшить количество признаков, характеризующих объект.

**Примеры использования методов понижения размерности:**

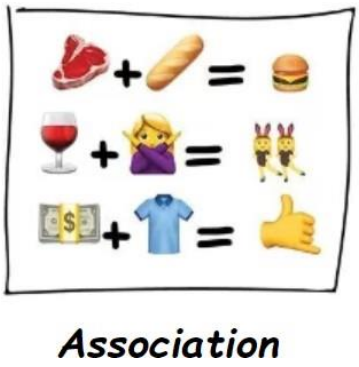
- визуализация,
- рекомендательные системы, определение тематик и поиск похожих между собой документов,
- анализ фейковых изображений.

**Цель обучения** – построить модель, которая переводит пространство признаков из размерности **n** в размерность **m** (m < n), при этом сохранив наибольшее количество информации.

**Пример понижения размерности:**




## Ассоциация

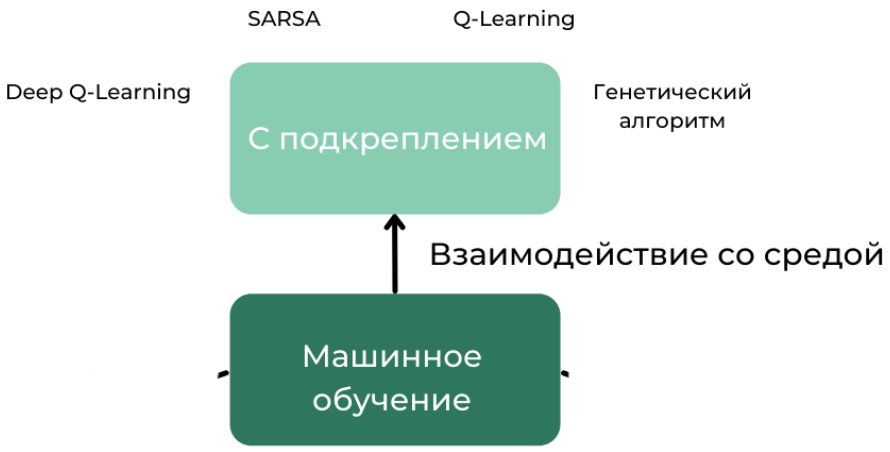


**Ассоциация (association)** – задача, в которой мы пытаемся найти правила и законы, по которым существует последовательность действий.

**Примеры использования ассоциации:**

- прогноз акций и распродаж,
- анализ товаров, покупаемых вместе,
- расстановка товаров на полках,
- анализ паттернов поведения на веб-сайтах.

## Обучение с подкреплением

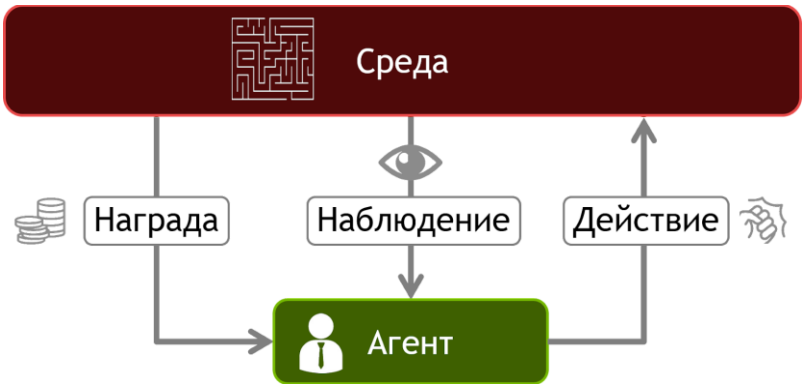



Обучение с подкреплением кардинально отличается от обучения с учителем и без него, поэтому его выделяют в отдельный вид обучения.

Это не задачи, связанные с анализом данных и предсказанием, а **задачи взаимодействия со средой и «выживания» в ней**.

Объект, который взаимодействует со средой (например, играет в игру), называется **агентом**.

Агент может от среды получать полные или частичные **наблюдения** о состоянии среды. Согласно своим наблюдениям, агент может выполнять **действия**. По мере совершения действий агент может получить **награду** от среды.




**Цель обучения** – не рассчитать все ходы, а построить оптимальную стратегию для взаимодействия со средой и максимизировать финальную награду.

**Примеры применения обучения с подкреплением:**

- боты в видеоиграх,
- роботы-пылесосы,
- беспилотные автомобили и летательные аппараты.

## Процесс разработки

### Методология Waterfall

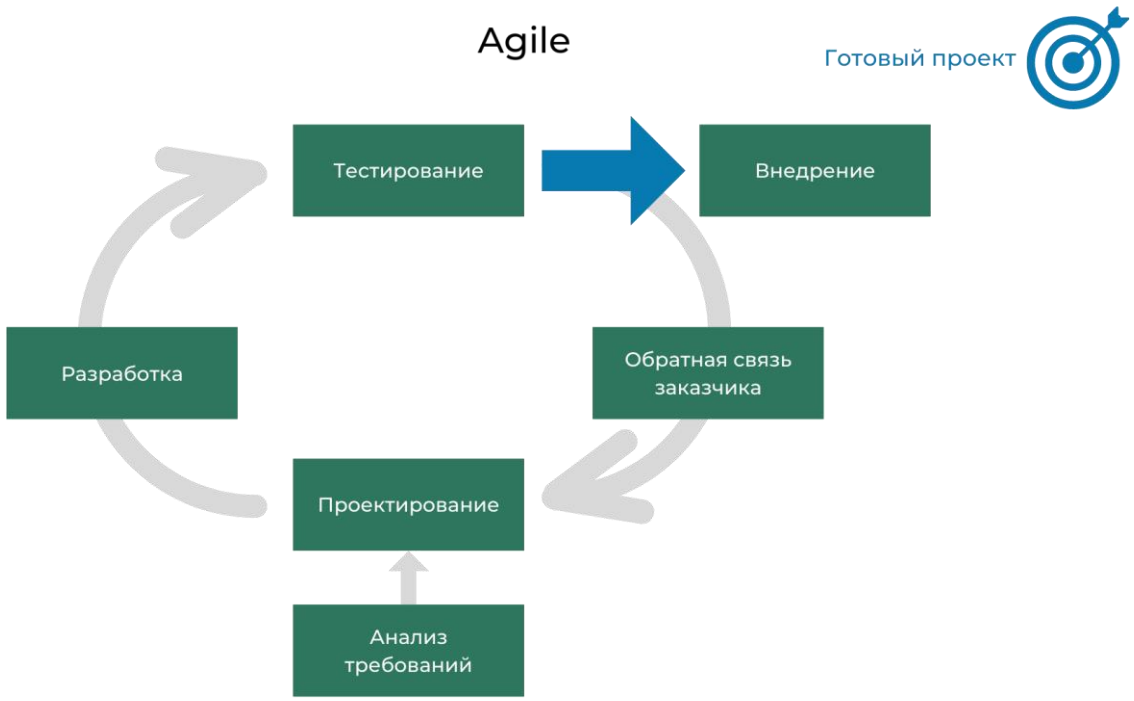
> **Водопадная методология (Waterfall model, «Водопад»)** – это модель процесса разработки ПО в виде потока последовательных фаз.

**Особенности методологии:**

- Разработка происходит строго последовательно этап за этапом. Переход на предыдущий этап не предусмотрен.
- Планирование ведётся на всю продолжительность проекта в самом его начале.
- Все действия максимально регламентированы и спланированы до мелочей. Установлены чёткие сроки окончания каждого из этапов.
- По окончании каждого из этапов происходит формальная сдача результатов именно этого этапа в виде большого числа документов.
- Результаты каждого из этапов тщательно проверяются на наличие ошибок.
- Готовый продукт передаётся заказчику только один раз, в конце проекта.

## Методология Agile

**Гибкая методология (Agile)** — это модель процесса разработки ПО с гибким возвратом к любому этапу: если тест спроектированной модели не дал нужного результата, разработчик может начать с самого начала.

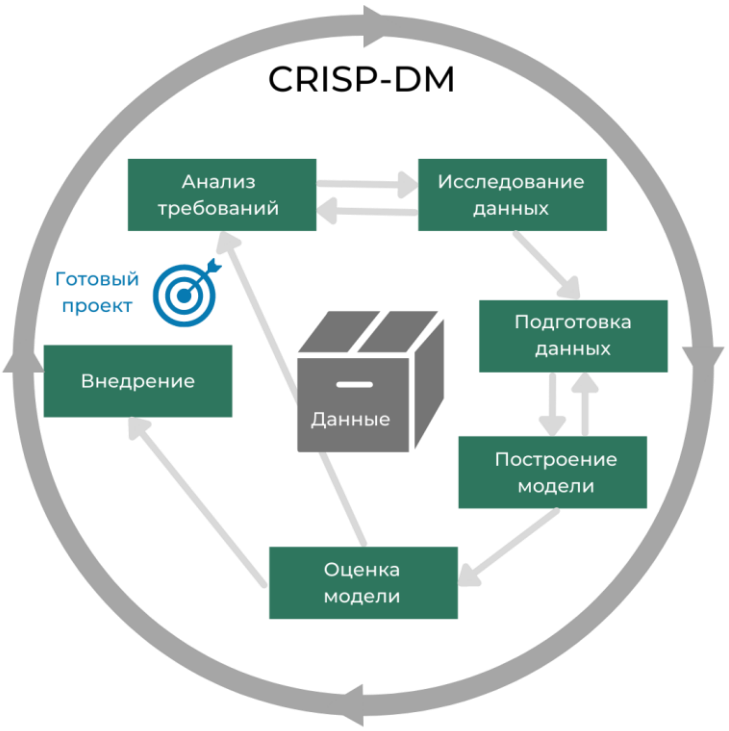


**Особенности методологии:**

- Разработка происходит по итерациям. В конце каждой итерации промежуточный результат демонстрируется заказчику. Заказчик даёт обратную связь (устраивает ли его эта часть функционала).
- Проект планируется только на один спринт. Длительность спринтов – от 1 до 4 недель.
- В случае, если у вас что-то не получилось, вы просто переходите на новую итерацию и теряете только время, потраченное на один спринт, а не на весь проект в целом.
- Не предусматривает множества формальных документов.
- Главный принцип – люди важнее процессов и инструментов.
- Заказчик видит продукт на протяжении всей разработки и может вносить коррективы.

## Методология CRISP-DM

**CRISP-DM (Cross-Industry Standard Process for Data Mining)** – это наиболее распространённая и проверенная методология по работе с проектами, завязанными на данных. Модель жизненного цикла исследования данных в методологии состоит из шести фаз, а стрелки обозначают наиболее важные и частые зависимости между фазами.



**Особенности методологии:**

- Методология разработана специалистами по работе с данными и учитывает особенности DS-проектов.
- Методология является обобщением методологии Agile для DS-задач.
- Последовательность этапов строго не определена, некоторые этапы можно менять местами. Возможна параллельность этапов. Предусмотрены возвраты на предыдущие этапы.
- Фиксирование ключевых моментов проекта.